Strategies for Control of Neuroprostheses through Brain-Machine Interfaces

by

Hyun K Kim

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mechanical Engineering at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
August 2005

© 2005 Massachusetts Institute of Technology All Rights reserved

Author

Department of Mechanical Engineering August 22, 2005

Certified by

Dr. Mandayam A. Srinivasan
Senior Research Scientist, Mechanical Engineering Thesis Supervisor

Accepted by

Prof. Lallit Anand
Professor of Mechanical Engineering Chair, Department Committee for Graduate Students
DISCLAIMER OF QUALITY

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available. If you are dissatisfied with this product and find it unusable, please contact Document Services as soon as possible.

Thank you.

The images contained in this document are of the best quality available.
Strategies for Control of Neuroprostheses through Brain-Machine Interfaces

by

Hyun K Kim

Submitted to the Department of Mechanical Engineering on August 21, 2005 in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mechanical Engineering

ABSTRACT

The concept of brain controlled machines sparks our imagination with many exciting possibilities. One potential application is in neuroprostheses for paralyzed patients or amputees. The quality of life of those who have extremely limited motor abilities can potentially be improved if we have a means of inferring their motor intent from neural signals and commanding a robotic device that can be controlled to perform as a smart prosthesis. In our recent demonstration of such Brain Machine Interfaces (BMIs) monkeys were able to control a robot arm in 3-D motion directly, due to advances in accessing, recording, and decoding electrical activity of populations of single neurons in the brain, together with algorithms for driving robotic devices with the decoded neural signals in real time. However, such demonstrations of BMI thus far have been limited to simple position control of graphical cursors or robots in free space with non-human primates. There still remain many challenges in reducing this technology to practice in a neuroprosthesis for humans.

The research in this thesis introduces strategies for optimizing the information extracted from the recorded neural signals, so that a practically viable and ultimately useful neuroprosthesis can be achieved. A framework for incorporating robot sensors and reflex like behavior has been introduced in the form of Continuous Shared Control. The strategy provides means for more steady and natural movement by compensating for the natural reflexes that are absent in direct brain control. The Muscle Activation Method, an alternative decoding algorithm for extracting motor parameters from the neural activity, has been presented. The method allows the prosthesis to be controlled under impedance control, which is similar to how our natural limbs are controlled. Using this method, the prosthesis can perform a much wider range in of tasks in partially known and unknown environments. Finally preparations have been made for clinical trials with humans, which would signify a major step in reaching the ultimate goal of human brain operated machines.
Acknowledgements

I would first like to thank my advisor, Dr. Mandayam Srinivasan for providing me with the opportunity to work on this exciting project. Srini granted me the intellectual freedom to find a research topic that truly excited, yet his guidance was invaluable when I required it the most.

Special thanks to my thesis committee members, Professor Neville Hogan, Professor Emilio Bizzi and Dr. James Biggs for their constructive criticism and encouragement. The opportunity to tap in to their wealth of knowledge has improved the quality of my work immensely. I would like to thank James especially for his role as a supervisor and lab mate. I relied on his input daily, in all aspects of my work, and I was greatly inspired by his enthusiasm for science and engineering.

I would like to express my gratitude to those at Nicolelis Lab at Duke University; Professor Miguel Nicolelis, Dr. Jose Carmena, Dr. Misha Lebedev, Tim Hanson, Erin Phelps and Joseph O'Doherty. They opened their doors to their lab and allowed me the opportunity to pretend to be a neuroscientist for a year.

My lab mates at the Touch Lab provided a fun working environment for the last five years. I would especially like to thank those who I have had the privilege of working with on various projects, Dr. David Schloerb, Dr. Jung Kim, Dr. Manivannan Muniyandi, Dr. Joono Cheong and Minseung Ahn.

My friends; room mates, soccer guys, drinking buddies and mentors, thank you for making the past nine years in Cambridge the most enjoyable years of my life.

Finally, I would like to thank my parents and my sister for their constant support and prayers. Without them, I would truly be lost.
Contents

Introduction .......................................................................................................................................... 21
  1.1 Previous work ............................................................................................................................ 22
  1.2 BMI system description ............................................................................................................ 24
  1.3 Goal of the thesis ...................................................................................................................... 26

Continuous Shared Control ............................................................................................................. 29
  2.1 Introduction ................................................................................................................................ 29
  2.1.1 Challenges of Brain Machine Interface .................................................................................. 29
  2.1.2 Continuous shared control ...................................................................................................... 30
  2.1.3 Design of the shared control system ...................................................................................... 31
  2.2 Methods .................................................................................................................................... 33
  2.2.1 Hardware – Proximity sensors co-located with the gripper ......................................................... 33
  2.2.2 Software – Shared control with sensor readings driving simulated reflexes ......................... 36
  2.2.3 Engineering tests .................................................................................................................... 41
  2.2.4 Grasping tasks ....................................................................................................................... 42
  2.3 Results ...................................................................................................................................... 45
  2.3.1 Engineering tests .................................................................................................................... 45
2.3.2 Grasping task ................................................................................................................................. 46
2.4 Discussion .................................................................................................................................................. 48
2.5 Conclusion .................................................................................................................................................. 52

3 Impedance Control in BMI ............................................................................................................................. 55
3.1 Introduction .................................................................................................................................................. 55
3.2 Methods ...................................................................................................................................................... 59
  3.2.1 Musculoskeletal arm model ..................................................................................................................... 59
  3.2.2 Prediction Algorithm .............................................................................................................................. 70
  3.2.3 Online System for Impedance Control .................................................................................................. 74
  3.2.4 Electrophysiology ................................................................................................................................... 76
  3.2.5 Behavioral tasks ..................................................................................................................................... 80
3.3 Results .......................................................................................................................................................... 83
  3.3.1 Prediction of Motion parameters .......................................................................................................... 83
  3.3.2 Prediction during force field tasks ......................................................................................................... 98
  3.3.3 Muscle Activation .................................................................................................................................. 105
  3.3.4 Kinematic Neurons ................................................................................................................................ 112
3.4 Discussion .................................................................................................................................................... 115
3.5 Conclusions .................................................................................................................................................. 118

4 Human Trials ................................................................................................................................................... 121
4.1 Introduction .................................................................................................................................................. 121
4.2 Preliminary trials ......................................................................................................................................... 122
4.3 Online human BMI system ........................................................................................................................ 124
4.4 EMG teleoperation ....................................................................................................................................... 128
  4.4.1 Previous work .......................................................................................................................................... 129
  4.4.2 System hardware and software .............................................................................................................. 129
  4.4.3 Teleoperation demonstration ............................................................................................................... 137
4.5 Conclusion .................................................................................................................................................... 142

5 Conclusion and Future Studies ......................................................................................................................... 143
List of Figures

Figure 1-1 System diagram for BMI setup. ................................................................. 25

Figure 2-1 The general problem of shared control in Brain Machine Interfaces. The goal is to share control between commands arising from the user’s brain (B) and sensors (S) on the slave device in a way that improves task performance. Both types of signals must be transformed (e.g., by $TB$, $TS$) into a “normal” command (e.g., a set of joint angles $q$) on which a control system ($H$) may operate. .................................................................................. 32

Figure 2-2 Trials were conducted with a pneumatic gripper mounted on a back-driveable robot. Three triangulating infrared proximity sensors are mounted at the gripper base. The centroid of the sweet spot of the gripper is marked (yellow “o”), along with portions of the gripper that contacted a target object when it was grasped in the sweet spot (highlighted purple). The sweet spot was the location that results in the most stable grasp of the target object. The lines of sight of the sensors are shown (dashes aligned with $\hat{u}_{\text{sens}}$). Range readings ($r_1...r_3$) from a simulated surface illustrate terms in Equations 2 and 6. Cartoon springs displaced from rest (dashes up to $r_0$) illustrate how the sensor control laws embodied in equations 2 and 6 repelled the gripper from surfaces in front of flanking sensors ($r_1$ and $r_2$) and attracted it to surfaces in front of the central sensor ($r_2$). ........................................................................................................................................... 34

Figure 2-3 Infrared proximity sensor calibration data. Error bars signify standard deviation............................................................................................................................... 36

Figure 2-4 Artificial potential field introduced by the sensor reflexes. Without shared control, the robot would move the gripper sweet spot (yellow “o”) along the brain-commanded trajectory. With shared control, the combination of sensors and target causes a potential field that produces an augmented trajectory. As the gripper progresses, sensor 1 scans the target object, creating the nearest repulsive potential (a), which urges the
gripper “downhill” in the direction $-\hat{u}_{\text{sensors}}$. With adequate gain, this effect prevents collisions between the gripper and target object. When sensor 2 scans the target object, the central attractive region (b) is produced, urging the gripper “downhill” toward the object, in the direction $+\hat{u}_{\text{sensors}}$. Sensor 3 produces the most distant repulsive field (c).

Figure 2-5 Block diagram of the control system for the proposed continuous shared control paradigm. Inputs are the Cartesian coordinates extracted from the brain signals and the range information from the sensors.

Figure 2-6 Workspace reproduced offline with a robot (SensAble Technologies, Phantom 3.0), a target object, infrared proximity sensors and a pneumatic gripper. The dimensions of the workspace and target object are in proportion to those presented to the monkey on a computer screen during online trials. The dashed blue line shows a sample of a trajectory towards a target from a starting position marked by “o”.

Figure 2-7 Robot behavior augmented with continuous shared control. The original trajectories are commanded by a human user. (a) The robot is able to hover above a planar surface and avoid collision with obstacles. (b) A raw noisy trajectory attempting to locate the handle of a coffee mug (marked “M”) and the corresponding augmented trajectory (c). In (b) the robot collides with the surface of the mug and fails to locate the exact position of the handle. With sensor-based reflexes turned on, (c), the robot avoids collision, successfully hovers over the curved surface and locates the handle successfully.

Figure 2-8 Percentage success rate for the reproduced BMI trials.

Figure 2-9 A sample trajectory of the robot gripper with various levels of sensor-based control added. Coordinates of the target object (black circle) are indicated, as well as the initial coordinates of the gripper sweet spot “o”, and the positions of the infrared proximity sensors (3 vertical lines).

Figure 2-10 A segment of the robot trajectory. This segment of the trajectory was chosen because the users original intention was to move away from the target and shows one
possible drawback of shared control. Coordinates of the target object (black circle) are
indicated, as well as the initial coordinates of the gripper sweet spot (o), and the positions
of the infrared proximity sensors (3 vertical lines). As the weight of the sensor-based
reflexes is increased the trajectory becomes increasingly different from that derived
purely from brain signals (upper left trajectory, 0% CSC).

Figure 3-1 Muscle geometry for the musculoskeletal arm model. The following
abbreviations are used for the muscle group; pectoralis major - PMJ, biceps short head -
BSH, biceps long head - BLH, deltoids - DEL, triceps long head - TRIO and triceps
lateral head - TRIA. The letters (ai - di) mark the coordinates of the origin and insertion
points for each muscle group. The coordinate values are shown in Table 1. A is an
enlarged view of the shoulder joint and B is an enlarged view of the elbow joint.

Figure 3-2 Block diagram for muscle force generation model. Plot (a) shows the passive
force and plot (b) shows the active force. The total force is the sum of the passive and
active muscle forces is shown in plot (c).

Figure 3-3 Normalized force-displacement property for muscle tissue as shown in Zajac
(1989). The curve is for full activation, a=1.

Figure 3-4 (a) Force-velocity properties for muscle tissue as shown in Zajac (1989) and
(b) the corresponding curve given by equation 3.

Figure 3-5 Estimated operating ranges of the elbow flexors and extensors plotted by
Murray et al. (Murray, Buchanan et al. 2000). Estimated fascicle excursions were
normalized by optical fascicle length and superimposed on a normalized force-length
curve. The dotted lines bounded by red circles show the same operating ranges for the
musculoskeletal model developed for this study.

Figure 3-6 Block diagram of the prediction algorithm. The algorithm takes neuronal
activity as input. The ANN outputs estimated muscle activation as an intermediate
variable. The muscle activations drive the computational musculoskeletal model.
Figure 3-7 Schematic of proposed online implementation of impedance control (an enlarged version of the musculoskeletal model can be seen in Figure 3-6). ................................. 76

Figure 3-8 (a) Location of the four implants in the first monkey (PMd, M1, M1, and S1) shown with the landmarks used to locate the desired areas. (b) Array of electrodes being inserted through a craniotomy. ................................................................................................................................. 77

Figure 3-9 The micro-wire arrays are fixed into position using adhesives, and grounded using metal screws fixed to the cranium. Dental cement is used to cover up the head stage. ........................................................................................................................................... 78

Figure 3-10 Screenshot of sort client 2 weeks after electrodes were implanted in the first monkey .......................................................................................................................... 79

Figure 3-11 The KINARM exoskeleton used for the primate behavior tasks. The arm troughs constrain the monkey to planar movement and the motors are able to apply external torques during the tasks. The tasks is displayed visually to the monkey on a reflective surface in front of the animal. ................................................................................................. 80

Figure 3-12 The center out task. The task involved holding the cursor in the red center target for 2 seconds and then moving to the green outer target and then holding for another 2 seconds. The gray dot on the cursor disappears when the cursor intersects the target. ........................................................................................................................................... 81

Figure 3-13 The monkey seated in the KINARM, performing the task ................................................. 83

Figure 3-14 (a) Correlation coefficient for Cartesian position prediction for a linear Wiener filter during 12 sessions. (b) Correlation coefficient for predicting muscle torques at the elbow and shoulder. The plot compares the linear Wiener filter with the Muscle Activation Method (MAM). ................................................................................................................................. 84

Figure 3-15 Box plot for the prediction correlation coefficient of shoulder torque from the two prediction algorithms. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are the dashed lines extending from each end of the boxes to show the extent of the rest of the data. The maximum length of the whiskers is 1.5
times the interquartile range. Outliers are data with values beyond the maximum whisker length and are shown as red crosses if they exist................................. 85

Figure 3-16 Box plot for the prediction correlation coefficient of elbow torque from the two prediction algorithms. ........................................................................................................ 86

Figure 3-17 Box plot for the prediction correlation coefficient of shoulder torque from MAM and position (x) from a linear filter. The red + is an outlier (outside 1.5 times the interquartile range)............................................................................................................ 87

Figure 3-18 (a)SER (signal to error ratio) plotted for muscle torques at the shoulder. (b) SER plotted for muscle torques at the elbow........................................................................... 88

Figure 3-19 Box plot for the prediction SER of (a) shoulder torque and (b) elbow torque from MAM and linear filter. The red + are outliers (outside 1.5 times the interquartile range). ............................................................................................................................... 90

Figure 3-20 1 minute of sample predicted trajectory from MAM and corresponding observed trajectory for muscle torque at the shoulder (T_s) and elbow (T_e) for five sessions ((a)-(e)). Each plot is accompanied by a cross plot of the predicted and observed values. ........................................................................................................................................... 97

Figure 3-21 Observed muscle torques at the shoulder for a training (viscous force field) and test set (null force field) for the two sessions ........................................................................................................ 99

Figure 3-22 Predicted trajectory from a linear filter for shoulder torque given a viscous force field training set and a null field test set................................................................. 100

Figure 3-23 Predicted trajectory from a linear filter for shoulder torque given a viscous force field training set and a viscous force field test set....................................................... 101

Figure 3-24 Predicted trajectory from the Muscle Activation Method (MAM) algorithm for shoulder torques given a viscous force field training set and a null field test set. .... 102
Figure 3-25 Predicted trajectory from the Muscle Activation Method (MAM) algorithm for torques at the shoulder (Figure 3-24) on an enlarged scale with their cross plots. 104

Figure 3-26 Prediction of shoulder torques during a viscous force field, using joint angle, velocity and acceleration as input instead of neural data. 105

Figure 3-27 (a)-(c) 3 × 100 second samples of normalized surface EMG measurements plotted with the intermediate variable from the Muscle Activation Method (MAM) algorithm. Each figure contains two plots, one from the biceps and the other from the triceps. 107

Figure 3-28(a)-(c) 3 × 100 second samples of normalized surface EMG measurements plotted with the intermediate variable from the Muscle Activation Method (MAM) algorithm. Each figure contains two plots, one from the biceps and the other from the triceps. (Low pass filtered at 0.5Hz). (d) Cross plots of MAM activation and EMG measurements for biceps and triceps. The data points are from the complete 300 second sample (5 minutes) plotted in (a)-(c). 111

Figure 3-29 Cross plots of triceps and biceps activity from EMG recordings and the MAM algorithm. 112

Figure 3-30 Prediction of position (x) using a linear filter with a viscous force field training set and null force field test set. 113

Figure 3-31 Directional tuning properties of neurons during curl force field tasks. 114

Figure 4-1 Screen shot of graphical display of the task. The patient controlled the black cursor to track a randomly time-varying target by applying variable gripping force to a pressure measurement device in the contralateral hand. 122

Figure 4-2 Squeeze ball for measuring the patients’ grip force. 123

Figure 4-3 Prediction of gripping force from recorded neuronal activity by linear and nonlinear decoding algorithms, A Wiener filter, B ANN, C LMS, D Kalman filter. E, Increasing predictive quality with longer training time. F, increasing accuracy of
prediction with larger neuronal ensembles. The plots are from results shown in Patil, Carmena et al. 2004. ................................................................. 124

Figure 4-4 Human BMI system ................................................................. 125

Figure 4-5 A back-drivable gripper with three infrared proximity sensors mounted for Continuous Shared Control...................................................... 126

Figure 4-6 Pneumatic force feedback pressure cuff (cuff is worn around the wrist in this picture). ................................................................. 127

Figure 4-7 (a) Surface EMG electrodes and amplifier. (b) Electrode placements...... 131

Figure 4-8 Two link manipulandum for recording position during planar movements. 132

Figure 4-9 (a) Phantom Desktop (Sensible Technologies, Woburn, MA), (b) Nano-17 force transducer (ATI Industrial Automation, Apex, NC)............................ 136

Figure 4-10 System overview of EMG teleoperation system (enlarged version of the musculoskeletal model can be seen in Figure 3-6)............................. 137

Figure 4-11 The teleoperation task. A sharpened pencil was attached to the end of the robot. The task involved drawing straight lines in the target boxes of the framed paper without piercing the surface................................................................. 138

Figure 4-12 Predicted shoulder torque (a) and elbow torque (b) from EMG recordings and the Muscle Activation Method. The predicted and observed trajectories are and their corresponding plots are shown......................................................... 140

Figure 4-13 Resulting drawings from teleoperation trials. First row is without impedance control and the second row is with impedance control. The sequence of each row shows trial progression. No piercing occurred in the final two trials of impedance control .... 142

Figure 4-14 Mean length of pierced paper for the teleoperation trials with position and impedance control. The error bars show the standard deviation for the 5 trials............. 142
List of Tables

Table 1 Dimensions for the musculoskeletal arm model muscle geometry shown in Figure 3-1 ........................................................................................................................................ 62

Table 2 Linear regression for PCSA (cm$^2$) against total body weight (kg) for macaca mulatta ........................................................................................................................................ 65

Table 3 Linear regression for segment mass and moment of inertia against total body weight (kg) for macaca mulatta ........................................................................................................................................ 69

Table 4 ANOVA table for the prediction correlation coefficient of shoulder torque from the two prediction algorithms ........................................................................................................................................ 85

Table 5 ANOVA table for the prediction of elbow torque from the two prediction algorithms ........................................................................................................................................ 86

Table 6 ANOVA table for the prediction correlation coefficient of shoulder torque from MAM and position (x) from a linear filter ........................................................................................................................................ 87

Table 7 ANOVA table for the prediction SER of (a) shoulder torque and (b) elbow torque from MAM and linear filter ........................................................................................................................................ 89
1 Introduction

Operators capable of sending brain signals to control robots, spaceships and other mechanical devices have been a popular subject of science fiction in the past. Combining the versatility and amazing computational power of the human brain with the "super-human" capabilities of machines is a fascinating prospect. There have been numerous examples of Hollywood movies showcasing super heroes and super villains controlling robotic body parts or mechanical sidekicks as deftly and effortlessly as they were control their own bodies.

With recent technological advances in measuring and decoding the electrical activity of large populations of single cortical neurons, "neurobots" have come closer to being a reality. The first demonstration of a Brain-Machine Interface (BMI) where a robot was controlled using cortical neural signals was presented in 2000 (Wessberg, Stambaugh et al. 2000). Many similar demonstrations have followed since. In one of the most recent studies, a monkey was able to control a 3-dof robot to perform reach and grasp tasks under "brain-control" through hundreds of neuronal signals recorded from multiple chronic implants (Carmena, Lebedev et al. 2003).

Science fiction and Hollywood aside, BMIs have promising potential as neuroprotheses for paralyzed individuals. Traumatic spinal cord injuries inflict motor deficits to more than 200,000 patients in the US. About half of these patients are quadriplegic, meaning they do not have use of their arms or legs. Also, there are more than 5,000,000 stroke survivors and 400,000 amputees. These patients depend on continuous assistance to accomplish even the simplest motor tasks. It goes without saying that these deficits are devastating to the livelihood of those inflicted. Although, some cellular and pharmacological therapies exist, many of these disorders and lesions remain irreparable. Restoring motor skills through robot manipulators controlled by brain signals
would have a profound effect on the quality of life of those individuals suffering from paralysis.

This thesis introduces strategies for improving the control of such neuroprostheses, which are still at an early stage of development, so that eventually a practical and useful prosthesis can be achieved.

1.1 Previous work

Schmidt (1980) originally proposed direct interfaces between spared cortical or subcortical motor centers and artificial actuators could be employed to bypass spinal cord injuries. The purpose would be to restore basic motor functions in patients suffering from severe body paralysis. This novel idea captured the imagination of many neuroscientists and engineers. However, technology constraints in recording of cortical neurons and real-time computation have hampered the development of BMIs until recently.

Earlier studies in BMI used scalp electroencephalograms (EEGs) for their non-invasive nature and easy application. Wolpaw and colleagues developed an EEG-based system which allowed human users to move a cursor in 1-D through modulating brain rhythms (1991). In Birbaumer et al. (1999), patients learned to regulate their slow cortical potentials through EEGs to control a cursor on a spelling aid. However, EEG signals have shown limitations in their applications beyond simple cursor control. Scalp EEG recordings reflect the common electrical activity of millions of neurons in widespread areas of the cortex, and subsequently lack both the resolution and the response time required for dexterous control of robot arms in real-time.

Chapin et al. (1999) first demonstrated BMI with intracranial recording in rodents, by simultaneously recording from arrays of electrodes chronically implanted in the primary motor (M1) cortex and ventrolateral (VL) thalamus of the rats. The rats were trained to control a robot in 1-D motion to deliver water. Wessberg et al. (2000) was the first study to show primates controlling robots in 1-D and 3-D motion through neural signals from chronic implants. The recordings came from 96 electrodes in arrays implanted in multiple cortical areas of monkeys and the results suggested that motor
planning and execution is done in a distributed manner across the cortex. The predictions of movement were obtained by linear filters and neural networks constructed from previous movement recordings. Serruya et al. (2002) used a similar method, but were able to use smaller number neurons (7-30) in M1 to control the 2-D motion of a cursor off-line. Taylor et al. (2002) demonstrated real-time BMI, with a monkey controlling a cursor in 3-D motion while given visual feedback. The study showed that real-time visual feedback allowed the subjects to change the tuning properties of the cells during brain-control, resulting in good performance even with a small number of neurons. Carmena et al. (2003) included a robot in to the loop through visual feedback of the robot trajectory to the subject. The study showed that recording from large neuronal ensembles gives higher accuracy during BMI and also the ability to extract more motor parameters. Also, the functional reorganization in multiple cortical areas suggested that BMI/robot dynamics were incorporated into the motor and sensory representation. Taylor et al. (2003) showed a similar demonstration of 3-D robot movement through BMI.

All of the above BMI demonstrations employed the strategy of continuous control of endpoint trajectory. Recently, Andersen and colleagues have introduced a new approach, where higher level signals related to goal of movement are decoded from the cortical areas and the detailed trajectories for the movement are left to the robot (Buneo, Jarvis et al. 2003; Shenoy, Meeker et al. 2003; Musallam, Corneil et al. 2004). Information about position, type and size of intended target could be extracted from implants in the parietal reach region (PRR) of the posterior parietal cortex (PPC) and the dorsal premotor cortex (PMd). Such tactics have obvious advantages. The resultant trajectories would be accurate and robust and there would be less strain on the user. Such a system is similar to supervisory control used in teleoperation (Sheridan 1992), and would be suitable for use in a structured environment where tasks are well defined and repetitive. However, the lack of direct control of the trajectories would result in problems in flexibility and responsiveness of the BMI. The system would only be able perform actions that are predefined as motion primitives. For instance, with a system containing primitives for a reach and grasp tasks only, the user would find it difficult to pick up and twist open a bottle. It would be difficult to define an exhaustive set of motion primitives required for the BMI to perform a wide range of every day tasks.
Pesaran et al. (2002) proposed the use local field potentials (LFP) for BMI instead of the neuronal spikes used in previous studies. They found that LFP activity in parietal cortex discriminated between preferred and anti-preferred directions with similar accuracies to the spike rate and predicted the time of planned movement with better accuracy than the spike rate. This suggests that LFP may be a good alternative for BMI systems that use the supervisory control framework suggested by Andersen’s group. However, it is not clear that LFP signals contain enough information to control a manipulator through continuous control of trajectory. A BMI demonstration with LFP is yet to been done, and there is marginal advantage in terms of invasiveness of the implants since they still require intracranial recordings.

There is still considerable debate over what is the best system for BMI. Whether single unit, multunit or field potentials are the best type of brain signals for the application is still being debated. Research groups that focus on single unit activity diverge on the assessment of whether large (hundreds to thousands) (Wessberg, Stambaugh et al. 2000; Carmena, Lebedev et al. 2003) or a substantially smaller (eight to thirty) (Serruya, Hatsopoulos et al. 2002; Taylor, Tillery et al. 2002) or number of single unity neuron activity is sufficient for efficient BMI. Also, whether a goal-based supervisory approach or a continuous control of trajectory approach is better for BMI is not clear. However, it appears that recording large population of single unit activity from intracranial implants in multiple cortical areas and extracting a continuous trajectory (Wessberg, Stambaugh et al. 2000; Carmena, Lebedev et al. 2003) has been the most successful approach to date, in terms of robustness, flexibility and performance of the system.

1.2 BMI system description

The work that will be described in the thesis will use the Wesseberg et al. (2000) and Carmena et al. (2003) BMI system as the test bed (Figure 1-1). This section will describe the most recent version of the setup, as a brief introduction to the system.
Monkeys (Macaca mulatta) were implanted with multiple arrays containing 16-64 microwires each, with separation of 300 μm between adjacent microwires. Implanted areas included the dorsal premotor cortex (PMd), supplementary motor area (SMA), and the primary motor cortex (M1) in both hemispheres. Stereotaxic coordinates, published microstimulation maps and intra-operative neural mapping were used to locate the desired cortical areas. The microwires were 50 μm diameter tungsten electrodes manufactured at the Nicolelis Lab (Duke University, NC).

A 512-multichannel acquisition processor (Plexon Inc., Dallas, TX) was used to simultaneously collect and record all the neuronal data. The server connected to the multichannel acquisition processor (MAP) sorted the signals from the channels to pick out single and multi-unit spikes from the surrounding noise in real-time. The spike counts were binned in 100 ms bins for processing. To extract the relevant motor parameters (hand position, velocity and grip force) from the binned neural data, various mathematical models were used including linear filters and nonlinear artificial neural networks. The models were established from the preceding ten minutes of movement.

Figure 1-1 System diagram for BMI setup.
data, which was fitted to the corresponding neuronal activity. The predictions of various mathematical models proved to yield comparable correlation coefficients offline. For the online demonstration a multidimensional linear regression, or Wiener filter, was chosen for its simplicity and fast convergence.

The predicted end-point position and velocity information, along with the grip force was sent to a 6-dof robot arm equipped with a 1-dof gripper (The Arm, Exact Dynamics, Didam, The Netherlands). The 2-D trajectories were played back on the robot and the resulting end-point position was visually fed back to the monkey as a graphical cursor on a monitor. The task presented to the monkey was to move the cursor into a target and increase the grip force until the size of the cursor matched the target. The monkey was trained to initially perform the task using a joystick and then switch to brain control half way through the session. After several months of training, the monkey was able to perform the task successfully for hour-long sessions under brain control. Eventually, the monkey was able to learn to not move the joystick at all during brain control (Carmena, Lebedev et al. 2003).

1.3 Goal of the thesis

Future applications of BMI may range from the fantastic (fighter jet pilots using BMI helmets) to the life altering (neuroprotheses). There still remain many obstacles to achieving an effective BMI for such real-life applications. There are health and safety concerns with surgically implanting the multiple electrodes for intracranial recordings. Also, there needs to be significant development in high performance micro-electrode array design, biocompatibility of the chronic brain implants, microelectronics (for example, miniaturization of hardware for multichannel neural signal condition and telemetry), power management and real-time computational modeling (Nicolelis 2003).

This thesis addresses some of the main bottlenecks preventing the application of current BMI technology towards a practical motor neuroprosthesis. Many of the engineering requirements listed above have readily available solutions that simply require application. However, one of the main challenges with current BMI, that needs
addressing, is the lack of certainty in decoding the neural signals. This uncertainty leads to generation of noisy trajectories and unsteady manipulation that hinders practical use. Also, useful information contained in the cortical recordings is lost due to incorrect decoding. These are the main reasons for BMI demonstrations being thus far restricted to movement of graphical cursors and free motion of a robot through space without any real world tasks being achieved. Current understanding of the neural circuitry is limited and thus, it is impossible to decode exactly what each neuron firing signifies. Also, considering that only a minute sample of the total population is being recorded, an entirely accurate reproduction of the intended trajectory may not be attainable. Given these limitations, this thesis introduces strategies for obtaining the most out of the neural signals, so that a practically viable and ultimately useful neuroprosthesis can be achieved.

In Chapter 2, a Continuous Shared Control (CSC) paradigm is introduced where low-level robot intelligence is implemented to augment the unsteady trajectories from the neural signal. Using a robot to reach and grasp objects in unstructured environments can be a difficult tele-manipulation task, even for an able-bodied individual. Controlling the slave directly with neural signals instead of a hand-master adds further challenges due to added unsteadiness. In CSC, robot sensors would add reflex-like reactions to facilitate dexterous manipulation.

Chapter 3 addresses the issue of maximizing the information content extracted from the neural spike recordings. Currently, the signal is treated as a pure motion source where only trajectories are extracted for the manipulator. This approach neglects information about force and dynamics of the movement, that are also present in the cortical neurons. A model-based method is developed for extracting all motion parameters in a consistent and co-dependent manner. The parameters will include impedance information, which are critical to the control of movement and interaction with objects. This strategy allows the application of impedance control to the manipulator such that it can perform a wider range of everyday tasks.

In Chapter 4, the practical issue of bringing BMI technology to human use is examined. Preparations made for a clinical trial is described, and impedance control is demonstrated through EMG teleoperation with human subjects.
Finally, conclusions of the thesis and suggestions for future studies are made in Chapter 5.
2 Continuous Shared Control

2.1 Introduction

2.1.1 Challenges of Brain Machine Interface

Uncertainty in determining the command signal presents one of the main difficulties in agile control of a BMI robot. One probable source of this uncertainty is the sparse sampling of the population of cortical neurons. Current microwire arrays allow for recording of approximately $10^2$ neurons, an order of magnitude better than was possible until very recently, but still a miniscule fraction of the approximately $10^{11}$ neurons in the brain. Placing the microwire arrays in brain areas known to be related to motor control obviously improves the performance, but the undersampling problem seems likely to remain an aspect of BMI systems for some time. This undersampling is partially offset by redundancy in motor information in the neuron firings. However, this redundancy does not provide assurance that all the relevant information is present in the neurons that are being recorded. Even in arm motor and premotor areas, it appears that only a relatively small portion of cortical neurons are involved with direct encoding of movement features. Studies show large portions of the neurons are correlated to more abstract characteristics such as potential targets, position of obstacles and orientation of spatial attention (Crammond and Kalaska 1994; Shen and Alexander 1997; Lebedev and Wise 2001).

Due to limited understanding of the information represented by neuronal discharges and the sparse sampling, current models can not reconstruct exactly the intended motion from the neural activity recorded. Current microelectrode arrays gather enough data to predict 2D or 3D hand coordinates with an average correlation coefficient ($R^2$) of about 0.5-0.8 (Wessberg, Stambaugh et al. 2000; Carmena, Lebedev et al. 2003). If position is predicted, the decoded output resembles the monkey’s arm trajectory with noise added.
The unsteadiness of the output trajectory makes dexterous manipulation very difficult. Although a low pass filter may be used to reject some of the noise, such a filter results in a delayed response and loss of the ability to make quick movements.

The frequency of the decoded command signals can also present problems. In algorithms used in current BMI systems, cortical activity is commonly binned and processed at 10-20 samples/second, for optimal prediction. This works adequately for positioning a robot limb in free space, or moving a cursor on a computer monitor. However, this is a very low update frequency for commanding a slave robot interacting with real objects. Purposeful human movements range from approximately 0.01 Hz (posture) to about 10 Hz (physiologic tremor) (Jones 1997). To command a robot limb over this range of frequencies requires updates of at least 20 samples/second to meet the Nyquist criterion, or 200 samples/second in order to have 10 samples per half-cycle. Most robot controllers in fact update at above 1 kHz. The neural recordings can be binned more frequently than the typical 10-20 cycles/second. However, doing so may affect the accuracy of the prediction, may increase the noise in the predicted signals, and may require more computational power than current processors can provide.

Dexterous manipulation is further complicated by the lack of tactile feedback. Numerous teleoperation studies have shown that force feedback can improve performance dramatically (Hannaford, Wood et al. 1991; Massimino and Sheridan 1994). Unless the interface stimulates somatosensory cortex by passing small currents through electrodes, or the skin with tactile stimulation, the robot and gripper are insensate to the BMI user. Key problems of an insensate gripper are damage and delicacy. A user who does not feel pain when a slave gripper collides roughly with an object may damage the gripper mechanism or sensors by accident (humans) or indifference (monkeys). Furthermore, delicate grasping of fragile objects usually requires sensing of grip force (Meek, Jacobsen et al. 1989) and incipient slip (Burstedt, Edin et al. 1997).

2.1.2 Continuous shared control

Despite the difficulties just outlined, the BMI user needs dexterous manipulation of the robot in unstructured environments. To meet this challenge, a new control paradigm
is proposed where the control is shared between the command signals from the user’s brain and reactive signals from local sensors. This control paradigm can be described as one of continuous shared control. While the command signal from the user retains continuous control of the end position, the robot also reacts to local sensors detecting proximity, collisions or other relevant information.

The idea of control shared between sensors and high-level command signals has a long history. Salisbury (1988) originally proposed a control strategy for a robotic hand, where the human can intervene in the autonomous task executed by the robot and, conversely, the robot can augment direct commands issued by the operator. Hayati and Venkataraman (1989) presented a strategy detailing a framework for task level sharing of motion trajectories between the sensors and user. Since then, shared control has been used for numerous applications where disability of the user makes the command signal unreliable, where the environment is unpredictable and where the task is too difficult for direct teleoperation, or a combination of these factors. Examples of applications can be seen in navigation of wheelchairs and walkers (Levine, Bell et al. 1999; Wasson, Gunderson et al. 2001), trajectory control of satellites (Oda, Inaba et al. 1999), and, most widely, for dexterous manipulation in teleoperation (Hannaford, Wood et al. 1991; Backes 1992; Brunner, Arbter et al. 1994; Michelman and Allen 1994; Turki and Coiffet 1995; Li, Cox et al. 1996; Ambrose, Aldridge et al. 2000; Griffin 2003). Although the idea of shared control is not new, the present study is, to the authors’ knowledge, the first application of shared control to Brain Machine Interfaces.

2.1.3 Design of the shared control system

The phrase “continuous shared control” will be used to distinguish the particular form of control implemented in this study from the range of possible BMI control schemes. First, the control is continuous because the interaction is immediate and does not have the “wait and see” characteristics of a planner-based approach or the switching characteristic of a traded-control (Hayati and Venkataraman 1989). The control is shared because it always reflects input of both brain and sensor, again distinguished from traded
control where control switches discreetly from direct operator control to the autonomy of the robot depending on task and situation (Figure 2-1).

![Figure 2-1](image)

Figure 2-1 The general problem of shared control in Brain Machine Interfaces. The goal is to share control between commands arising from the user’s brain (B) and sensors (S) on the slave device in a way that improves task performance. Both types of signals must be transformed (e.g., by $T_B$, $T_S$) into a “normal” command (e.g., a set of joint angles $q$) on which a control system ($H$) may operate.

As in prior studies, the continuous shared control system makes the robot’s reaction to the environment partially dependent on a mapping from a sensor set to an associated action-based set of rules, in a purely reactive fashion (versus planner-based). This is similar to what might be found in the zeroth layer of the subsumption architecture presented by Brooks (1986). Such reflexive actions have been previously incorporated in grasp execution of Turki and Coiffet’s robot hand (Turki and Coiffet 1995), the Utah/MIT dexterous hand (Michelman and Allen 1994) and NASA’s Robonaut humanoids hand (Ambrose, Aldridge et al. 2000). In these cases, dexterous and delicate grasping of objects, that would otherwise be impossible, is achieved through force sensing and visual servoing.

Although reflexive reactions have been used previously to stabilize grasping with general-purpose robots, it is worth noting the special importance of sensor-based reactions to BMIs. Sensor-based reactions added to a BMI system do more than merely improve the behavior of the robot. Rather, they can be viewed as a component necessary to replace the function of missing spinal and cerebellar reflexes that would normally facilitate manipulation. Muscle activity due to reflexes of the cerebellum and spinal cord
or due to passive mechanical properties of the muscles are not reflected in the cortical electrodes of BMI at all or not until much after the activity has taken place. Therefore, for the system to approach normal human limb performance, it should be necessary to substitute for these reflexes in hardware and software. Accordingly, it is reasonable to choose control laws for sensor-based reactions that directly mimic or at least functionally replace biological motor reflexes and/or muscle mechanical properties.

2.2 Methods

Continuous shared control was implemented in a robotic manipulator with proximity sensors in order to replace grasping functionality normally provided by two reflex reactions in biological organisms. The first reaction was a collision avoidance reflex roughly analogous to "pain withdrawal" (Levinsson, Garwicz et al. 1999). This reflex withdrew the robot when either of the sensors just outside the outer edges of the gripper detected the presence of an obstacle, or impending inappropriate contact with the target. The second reflex replaced the functionality of the "infant palmar grasp reflex" (Zafeiriou, Tsikoulas et al. 1995). When the sensor viewing the space within the span of the gripper detected an object, it attracted the robot towards it and biased the gripper toward closing. Engineering tests and a grasping task experiment, which used previously recorded brain-commanded trajectories, were then performed with the system.

2.2.1 Hardware – Proximity sensors co-located with the gripper

Previous studies on shared control in teleoperation have shown that local sensors are most useful in obstacle avoidance and accurate grasping. Accordingly, the implementation of the continuous shared control paradigm presented in this thesis used proximity sensors co-located with a robot gripper in order to provide obstacle avoidance and stabilized grasping. Figure 2-2 shows the three triangulating infrared proximity sensors (Sharp GP2D120 Infrared Ranger, Acroname Inc., Boulder CO) attached to the base of a pneumatic gripper (Festo, HGR-10-A) mounted on a robot (PHANTOM 3.0, Sensable Technologies, Woburn MA).
Figure 2-2 Trials were conducted with a pneumatic gripper mounted on a back-driveable robot. Three triangulating infrared proximity sensors are mounted at the gripper base. The centroid of the sweet spot of the gripper is marked (yellow “o”), along with portions of the gripper that contacted a target object when it was grasped in the sweet spot (highlighted purple). The sweet spot was the location that results in the most stable grasp of the target object. The lines of sight of the sensors are shown (dashes aligned with $r_s$). Range readings ($r_1...r_3$) from a simulated surface illustrate terms in Equations 2 and 6. Cartoon springs displaced from rest (dashes up to $r_0$) illustrate how the sensor control laws embodied in equations 2 and 6 repelled the gripper from surfaces in front of flanking sensors ($r_1$ and $r_0$) and attracted it to surfaces in front of the central sensor ($r_2$).

The design of the location and action of the sensors in this study was deliberately biomimetic. Sensors were chosen that could report range of an object near the gripper much like the antennae of an ant or the vibrissae of a rat report range of an object near the animal’s mouth. A variety of sensors mounted at a variety of positions could have reported the same information. Ultrasound, scanned laser, stereovision, and a co-located monocular camera system (Matsikis, Schmitt et al. 1999) were all considered. Co-located 1-dof optical range sensors were chosen for several reasons.
Co-location was chosen because it seemed anatomically correct. Many animals that grasp with the mouth have vibrissae or antennae situated near it. Even animals with prehensile limbs, such as primates, rely highly on co-located tactile feedback from the finger pads for dexterous grasping. Co-location of the sensor also minimized potential problems of an obstruction lying between the sensor and target.

Optical sensing was chosen for cost-effectiveness, robustness, and lightness. Optical range sensors inexpensive enough to replace (~$10 each) were selected so that the present investigation might have some relevance to real real-world prosthetics, which must be made with parts inexpensive enough to replace when broken. Also, the lack of moving parts made the sensors robust. Since they didn’t require physical contact with objects, they were less prone to damage than tactile sensor arrays. Furthermore, at 3.5g each, they were extremely light, reducing power consumption in the motors of the backdrivable robot.

A final advantage of the optical range sensors selected for this study was their low computational expense compared to more complicated machine vision systems. Machine vision and visual servoing are well-established fields where the basic paradigms have largely been defined (Horn 1986; Hutchinson, Hager et al. 1996; Kragic and Christensen 2003). In a typical vision-based robotic grasping system stereo cameras are used to determine the position and the orientation of the object. In many cases a 3-D model of the object is generated to visually track the target object. The same 3-D model is used for model-based simulations to plan the grasp and movement trajectories. During the execution, the visual system also monitors task execution. The system usually requires substantial computational power for processing the information and a “wait-see” approach during the planning stage. In the present BMI application, low computational overhead was required in order to maintain the relatively high servo rate (1000 Hz) required by the backdrivable robot (PHANTOM 3.0). The three infrared range sensors required far less data processing than a scan of every point in a 3D scene would have. The simplicity of the computation was in tune with that of a biological reflex loop. In reflex actions, the responses are immediate and uncomplicated and vibrissae and antennae serve as simple range sensors much like the infrared sensors used.
The voltage outputs of the three infrared sensors were measured at various distances for calibration. The average output is plotted in Figure 2-3. To recover continuous range data, the plot was divided into three segments and least squares polynomial fits were done for each segment. The overall function was not monotonic and so a continuity constraint had to be added to obtain the exact range.

\[
V = -1.3914d^2 + 4.1807d + 0.00893
\]

\[
V = 0.0057d^3 - 0.0909d^2 + 0.1839d + 2.8693
\]

\[
V = 0.0017d^2 - 0.0962d + 2.0971
\]

Figure 2-3 Infrared proximity sensor calibration data. Error bars signify standard deviation.

2.2.2 Software – Shared control with sensor readings driving simulated reflexes

A reasonable, general-purpose paradigm for Continuous Shared Control can be introduced in the form of the artificial field potentials described by Khatib (1986). In this paradigm the robot moves in a field of forces. In the BMI Continuous Shared Control version of this concept, the brain-derived field $U_{brain}$ has an attractive pole at the Cartesian coordinates estimated to be the desired gripper location. When one or more sensors collocated with the gripper detects the proximity of an object, it causes a sensor-
based field potential, $U_{\text{sensors}}$, which may have local attractive or repulsive regions depending on the control laws of the individual sensors. The field potential for the gripper moving under shared control is the sum these two fields (Equation 1).

$$U_{\text{shared}} = U_{\text{sensors}} + U_{\text{brain}}$$ (1)

The only constraint on $U_{\text{sensors}}$ is that the system be asymptotically stable. Thus, the design space for $U_{\text{sensors}}$ is large and depends on the desired reflex response for the task and environment. For the implementation shown in this paper, a simple and effective quadratic potential field function was chosen. It treated range information from the sensors ($r_1, r_2, r_3$) as displacements of linear springs (Figure 2-3) from rest ($r_0$).

$$U_{\text{sensors}} = \frac{1}{2} K_s \left[ \left( \Delta r_1^{+} \right)^2 - (\Delta r_2^+)^2 + (\Delta r_3^+)^2 \right]$$ (2)

where:

$\Delta r^+ = \max[\Delta r, 0]$

$\Delta r_i = r_0 - r_i$

$r_i$ = calibrated range reading from left flanking sensor (mm)

$r_2$ = calibrated range reading from central sensor (mm)

$r_3$ = calibrated range reading from right flanking sensor (mm)

$r_0$ = range corresponding to rest length of virtual springs (mm)

$K_s$ = spring constant

The negative sign of $\Delta r_2$ produced a linear spring-like attraction towards objects within the span of the gripper, in the line-of-sight of sensor $r_2$. This attraction facilitated task performance by advancing the gripper toward graspable objects. Conversely, the positive signs of $\Delta r_1$ and $\Delta r_3$ gave the gripper a linear spring-like repulsion away from objects just outside the gripper span, in the lines-of-sight of the flanking left and right sensors $r_1$ and $r_3$. This repulsion facilitated task performance by withdrawing the gripper from objects with which the outer gripper surface was about to collide. The resulting potential field for $U_{\text{sensors}}$ with a target object is shown in Figure 2-4.
Figure 2-4 Artificial potential field introduced by the sensor reflexes. Without shared control, the robot would move the gripper sweet spot (yellow “o”) along the brain-commanded trajectory. With shared control, the combination of sensors and target causes a potential field that produces an augmented trajectory. As the gripper progresses, sensor 1 scans the target object, creating the nearest repulsive potential (a), which urges the gripper “downhill” in the direction -  **u**<sub>sensors</sub>. With adequate gain, this effect prevents collisions between the gripper and target object. When sensor 2 scans the target object, the central attractive region (b) is produced, urging the gripper “downhill” toward the object, in the direction +  **u**<sub>sensors</sub>. Sensor 3 produces the most distant repulsive field (c).

The field potential **U**<sub>brain</sub> for control of the brain-derived trajectory was straightforward. In online trials, binned neural signals were processed through a Wiener filter, as detailed elsewhere (Appendix A), in order to compute a brain-derived estimate of the correct gripper coordinates x<sub>brain</sub>. In the offline experiments on continuous shared control reported here, the inverse kinematics of the robot were solved for the corresponding set of joint angles, q<sub>brain</sub>. The choice of proportional-derivative control in joint angle space gave the brain-derived potential field **U**<sub>brain</sub> the following form in joint-angle space:
\[ U_{\text{brain}} = \frac{1}{2} K_p (q - q_{\text{brain}})^2 + \frac{1}{2} K_d (\dot{q} - \dot{q}_{\text{brain}})^2 \]  \hspace{1cm} (3)

where:

- \( U_{\text{brain}} \) = field potential derived from brain signals
- \( q \) = joint angle encoder readings of the robot
- \( q_{\text{brain}} \) = brain derived estimate of the correct joint angles
- \( \dot{q} = \frac{[q(t) - q(t - \Delta t)]}{\Delta t} \)
- \( \dot{q}_{\text{brain}} = \frac{(q_{\text{brain}}(t) - q_{\text{brain}}(t - \Delta t))}{\Delta t} \)
- \( K_p \) = Gain of proportional control (diagonal matrix)
- \( K_d \) = Gain of derivative control (diagonal matrix)

Solving Lagrange's equations for the energy functions resulted in the following implementation of the control (Equations 4-8). The motor torques of the robot, decoupled from dynamic terms, were a weighted sum of commands coming from two sources: 1) sensors on the slave robot 2) the user's brain signals.

\[ \tau_{\text{shared}} = \tau_{\text{sensors}} + \tau_{\text{brain}} \]  \hspace{1cm} (4)

The first set of torques in Equation 4, \( \tau_{\text{sensors}} \), were those derived from sensor readings. When no object was in the line of sight of the sensors, these torques were zero. However, when objects were present, the sensor-based potential field \( U_{\text{sensors}} \) became sloped, and the local gradient in the sensor-derived potential field (Equation 2) was equivalent to a virtual spring force \( F_{\text{sensors}} \) acting on the gripper:

\[ F_{\text{sensors}} = \nabla[U_{\text{sensors}}] \]  \hspace{1cm} (5)

This virtual force had a magnitude, \( F_{\text{sensors}} \), equal to the sum of three spring forces with potential energy defined previously in Equation 2. For any given set of sensor readings \( \{r_1, r_2, r_3\} \), the virtual force acting on the gripper was:
This virtual force acted in a direction aligned with the sensors lines-of-sight ($\dot{\mathbf{u}}_{\text{sensors}}$), and produced torques about the robot joints ($\mathbf{\tau}_{\text{sensors}}$) determined by the transpose of the robot Jacobian ($J^T$):

$$\mathbf{\tau}_{\text{sensors}} = \mathbf{F}_{\text{sensors}} J^T \dot{\mathbf{u}}_{\text{sensors}}$$  \hspace{1cm} (7)$$

The second torque term in Equation 4, ($\mathbf{\tau}_{\text{brain}}$), was proportional-derivative control that urged the robot joints toward the brain-derived pose ($\mathbf{q}_{\text{brain}}$). Since the potential field $U_{\text{brain}}$ associated with this command was expressed in joint coordinates (Equation 3), the form of the potential gradient was simple. Using variables previously defined in Equation 3, it had the form:

$$\mathbf{\tau}_{\text{brain}} = K_p (\mathbf{q} - \mathbf{q}_{\text{brain}}) + K_d (\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{brain}})$$  \hspace{1cm} (8)$$

Finally, nonlinear control components were added to $\mathbf{\tau}_{\text{shared}}$ to compensate for the dynamic and gravitational forces of the manipulator.

$$\mathbf{\tau} = \mathbf{\tau}_{\text{shared}} + \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q})$$  \hspace{1cm} (9)$$

The three dynamic terms include inertial torques, $\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r$, Coriolis and centrifugal torques, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r$, and gravitational torques, $\mathbf{g}(\mathbf{q})$, where $\ddot{\mathbf{q}}_r$ and $\dot{\mathbf{q}}_r$ are the referenced joint accelerations and velocities respectively (Slotine and Li 1991). The variables $\ddot{\mathbf{q}}_r$ and $\dot{\mathbf{q}}_r$ are given by Equation (10) and converge to $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$. Figure 2-5 shows the overall control system in block diagram form.

$$\ddot{\mathbf{q}}_r = \ddot{\mathbf{q}}_{\text{brain}} - \lambda (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{\text{brain}})$$

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_{\text{brain}} - \lambda (\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{brain}})$$  \hspace{1cm} (10)$$
2.2.3 Engineering tests

A preliminary evaluation of the system was performed by observing the effect of shared control on the robot's trajectory near obstacles. In the first test, the robot was commanded to move the gripper in a straight line over a horizontal surface such that, without shared control, it would collide with a rectangular obstacle resting on the surface (see Figure 2-7a). In the second test, a master control was used to manually generate the commanded trajectory. In this test, the task was to approach a coffee mug laying on its side (which served as the obstacle), move up and over the mug parallel to its curved side without touching it, and then come to rest with the gripper fingers on either side of the
handle. The gripper collided with the cup and did not come to rest properly on the handle when the commanded trajectory selected for the test was generated (see Figure 2-7b). Note that while the rest of the system was the same as in the grasping task, a different gripper was used than the one shown in Figure 2-3. The gripper used in the engineering tests was smaller and it had permanently-open fixed-fingers. Also, the robot used in the engineering tests was a small workspace Phantom 1.5 (Sensible Technologies, Woburn, MA) as opposed to the large workspace Phantom 3.0 that was used in the grasping task.

2.2.4 Grasping tasks

To evaluate the usefulness of shared control for BMI, the system was next tested on previously recorded brain-commanded trajectories to reach and grasp real objects. The neural data used were recorded from multiple cortical areas (M1, PMd, SMA and S1) of the monkey (Nicolelis, Dimitrov et al. 2003), performing a reaching and grasping task with graphical feedback of targets and robot end position. In this original study, a robot (MANUS, Exact Dynamics, Netherlands) was included in the loop, but the monkeys did not view it directly. Instead the monkeys viewed a computer monitor that displayed the location of the target object as a colored circle, and the location of the robot hand as another circle of different color. The monkey was trained to move the cursor to the target, and to modulate its grip force (indicated by cursor size) to achieve a target level of force (indicated as a ring around the target). The trial was considered successful if the monkey held the center of the cursor within the target for more than 150ms, with the appropriate grasping force. The monkey could modulate the force prior to moving to the target. To make the slave robot’s task analogous to the graphical task for which the monkeys were trained, a foam block was permanently mounted in the robot’s gripper. This made the monkeys’ task in online trials one of moving the slave robot to the target location while squeezing with the target force. There was no object placed at the target location and, hence, the difficulties of inadvertently knocking the object away, or closing the gripper prematurely were not part of the online task. Also, the robot used for the original neural control trials was not equipped with continuous shared control. The details of the online task have been reported previously (Carmena, Lebedev et al. 2003).

42
In the online trials the monkey's hand position and grip force were recorded, as well as the activity of 183 neurons. Ten minutes of measured position and force data were used to train a linear model that approximated the mapping from neuronal activity to hand position and grip force. Once the model was trained, the cursor position and size of the cursor displayed for the monkey were determined solely by neuronal activity.

In the current study, the previously recorded data of Carmena et al. (2003) were used in offline experiments on reaching and grasping of real objects. The predicted end point positions and grip forces, along with the position of the target objects were reproduced in a real workspace with a robot capable of continuous shared control (Figure 2-6). Relative sizes of the target object, workspace, gripper, and robot movement were the same as in the graphical game presented to the monkey when the neural data were recorded in the study of Carmena et al. In contrast to the online experiments, however, the current experiments included real-world complexities such as prematurely closing the gripper and knocking the target object away. Also, to allow for the fact that the robot used in the offline tests did not have an articulated wrist, target locations were rotated so that the target object was always directly in front of the robot at the start of each trial.
The scale of the real workspace was chosen so that the real object was successfully situated within the gripper’s span at times when the graphical cursor had been successfully inside the graphical target. The origin of the real workspace was chosen so that the gripper’s sweet spot (shown in Figure 2-3) centered on the real object at times when the graphical cursor had been centered on the graphical target. A total of 83 trials of the brain-controlled reach and grasp task were examined. For these trials, all cursor movement was controlled solely by neural signals without pole movement. In the online graphical game of Carmena et al., all 83 trials were counted as successes.

In the offline setup, a trial was counted as successful if the gripper successfully grasped the object without knocking it over. The percentage of successful trials was recorded with continuous shared control, with the contribution of the sensor commands weighted at 0%, 7%, 15%, 30% and 48% of the weight of brain-derived commands. The percentage weight was determined by comparing the stiffness of the sensor response (Figure 2-5, $K_s$) with stiffness of proportional controller (Figure 2-5, $K_p$). To facilitate
comparison, the Jacobian of the robot was used to map both stiffnesses into Cartesian space at the target location, where they could be expressed as (N/m) in the axial direction (Figure 2-3, $\mathbf{u}_{\text{sen}}$). In results comparing sensor-based weighting to brain-based weighting, the unitless ratio of these stiffnesses (N/m) / (N/m) is reported. The stiffness of the controller ($K_p$) was initially tuned prior to the trials to give optimal position control (small overshoot and rise time) and subsequently remained constant throughout the trials. The stiffness of the sensor response ($K_s$) was changed during the trials to produce the reported weights for the sensor commands.

In the prior study of Carmena et al. (2003) the size of the graphical cursor was determined solely by brain signals. In the present study, the grasping force of the gripper was determined solely by brain signals when the sensor-based commands were weighted 0%. However, in the continuous shared control mode, when sensor-based commands were weighted 7-48%, the proximity of the target cued the gripper activation.

### 2.3 Results

#### 2.3.1 Engineering tests

In engineering tests, these two reflexes were found to modify the commanded trajectory in several useful ways. Withdrawal commanded by the exterior sensors caused the gripper to avoid collisions of the outer gripper surfaces with objects in the scene. When the gripper approached large objects from the side, one of the outer two repulsive sensors detected the object first. This caused reflex withdrawal that tended to lift the gripper above it. Such reactions occurred when the exterior of the gripper was about to collide with the edge of a block (Figure 2-7a). When the gripper was situated above an object large enough to be detected by all three sensors, the central attractive sensor achieved an equilibrium with outer repulsive sensors. This led the gripper to hover conveniently over both flat surfaces, such as the top of a block (Figure 2-7a) and gently curved surfaces such as the exterior of a coffee mug (Figure 2-7c), despite a command trajectory that would have caused collisions (Figure 2-7b). However, objects or protuberances small enough to grasp, such as a coffee mug handle (Figure 2-7b and c),
tended to selectively activate the central sensor, attracting the gripper and biasing it toward closure.

![Diagram](image)

Figure 2-7 Robot behavior augmented with continuous shared control. The original trajectories are commanded by a human user. (a) The robot is able to hover above a planar surface and avoid collision with obstacles. (b) A raw noisy trajectory attempting to locate the handle of a coffee mug (marked “M”) and the corresponding augmented trajectory (c). In (b) the robot collides with the surface of the mug and fails to locate the exact position of the handle. With sensor-based reflexes turned on, (c), the robot avoids collision, successfully hovers over the curved surface and locates the handle successfully.

2.3.2 Grasping task

The number of successful trials initially increased with increased weighting of the sensor-derived commands. With continuous shared control off (0% weight), 22 of the 83 trials were successful. Increasing the weight of sensor-derived commands to 7%, 15%, 30% of the brain-derived signals increased success to 47, 51 and 48 trials respectively (Fig. 8). With the gain of sensor commands turned up to 48%, the number of successful trials dropped dramatically back down to 12, because the strong attraction of the gripper tended to knock the target object over. The number of successful trials that resulted in the object being grasped in the sweet spot of the gripper was also plotted (Figure 2-8), showing a success rate of 6%, 27%, 35%, 46% and 7% (5, 22, 29, 39 and 6 trials) with
increasing weight of sensor command. Note that the sweet spot was the best location to
hold the target object in order to maintain grip and orientation.

![Graph showing success rate for BMI trials](image)

**Figure 2-8** Percentage success rate for the reproduced BMI trials.

Figure 2-9 shows a sample of trial trajectories for 0%, 15%, 30% and 48% shared
control. 7% shared control plots are omitted, but can be approximately interpolated from
the 0% and 15% plots.
Figure 2-9 A sample trajectory of the robot gripper with various levels of sensor-based control added. Coordinates of the target object (black circle) are indicated, as well as the initial coordinates of the gripper sweet spot “o”, and the positions of the infrared proximity sensors (3 vertical lines).

2.4 Discussion

As reported in Carmena et al. (2003), monkeys were able to use brain signals to successfully squeeze an object with the required force at the right location during online trials. The object was attached to the gripper and shown to the monkey as a virtual object on the computer screen. In the real world task reported here, the gripper needed to grasp an object positioned in the target location instead. This introduced difficulties such as knocking the target object away or closing the gripper prematurely. Without continuous shared control this resulted in successful grasps only 27% of the time, which emphasizes how difficult it is to achieve dexterous manipulation through a BMI once the complexities of real world telerobotic grasping are introduced.
The grasping success rate was increased dramatically (Figure 2-8) by the addition of sensors and Continuous Shared Control. Adding an optimal level of sensor-based reflex action increased the number of successful grasps by 2.3-fold. It also increased the number of grasps that secured the target object in the gripper “sweet-spot” by 7.7-fold.

The rise and fall in task performance with progressively increasing weighting of the sensor-based reflexes was a satisfyingly biomimetic result. Just as the human body can have too little reflex activity (hyporeflexive flaccidity) or too much (hyperreflexive tremor), so could the robot have too little or too much reflex activity produced by its sensors. The best task performance occurred when brain-based commands were weighted 70-85%, and sensor based reflexes were weighted 15-30% (Figure 2-8). It will be interesting to see whether this weighting produces optimal results in online trials as well.

The success rate of the trials may not be the only factor in choosing the ideal control weight for the sensors. As the weight of sensor-based commands is increased, the sensor-based components of the trajectories may eventually obscure the user’s intention. As shown, (in Figure 2-10), the monkey’s intended trajectory becomes less and less clear, as the attraction to the target becomes predominant. This may not be a problem in this simple reaching and grasping task, especially since there is only one target and the user can still withdraw from the target by pulling away in order to overcome the gripper’s attraction to a graspable object. However, for more complicated tasks in more cluttered environments excessively high sensor weights might cause the user to lose control of the robot.
Figure 2-10 A segment of the robot trajectory. This segment of the trajectory was chosen because the user's original intention was to move away from the target and shows one possible drawback of shared control. Coordinates of the target object (black circle) are indicated, as well as the initial coordinates of the gripper sweet spot (o), and the positions of the infrared proximity sensors (3 vertical lines). As the weight of the sensor-based reflexes is increased the trajectory becomes increasingly different from that derived purely from brain signals (upper left trajectory, 0% CSC).

The question of how much autonomy is ideal in BMI systems is an interesting one. Similar questions have been raised in telerobotic systems in the past. In teleoperation, total control can be given to the operator in a direct bilateral control system (Lawrence 1993; Yokokohji and Yoshikawa 1994). At the other extreme, the robot can run autonomously in a hierarchical supervisory control framework where the operator only provides higher-level commands (Sheridan 1992). Also, there are various degrees of shared control that represent the middle ground, examples of which were given in the introduction. The level of autonomy given to the robot in teleoperation generally depends on the task, the environment, and the time delays involved in the system. Usually, if the tasks are well defined, performed in a known environment, involve a small number of
motion primitives, and/or involve long time delays, more autonomy is given to the robot. If the task is performed in an unpredictable environment where the movement of the robot must be immediately responsive to the user’s input and decisions have to be made instantaneously, the operator is given more direct control of the robot. Similar criteria would apply to use of shared control in BMIs.

Yet, there is an added dimension to the application of shared control to BMI. During brain control, the cortex has a limited communication bandwidth for commanding the robot, determined by the number of neurons that are being recorded. Carmena et al., (2003) showed that there is plastic cortical reorganization when the monkeys switch from operation with the hand to brain control. Primates appeared to optimize neural activity in order to control the robot/cursor through the limited communication channel provided by the recorded neuronal signal. Thus, it seems reasonable to ask what level of robot autonomy would provide for optimal cortical reorganization and robot control. Would a little robot autonomy lessen the computational load on the user’s neuronal network, thus facilitating robot control? Might too much autonomy lead to poor correlations of the robot trajectory with the user’s natural arm movements, hindering cortical reorganization and robot control? The question of how the cortex responds to different levels of robot autonomy is an interesting direction for future studies. Such a study is currently ongoing using the system described in this chapter. Graded levels of robot autonomy might also facilitate training users in BMI control. For example, the training of a paralyzed human user, or a non-human primate, could start with pure supervisory control. As cortical control of the robot is mastered, one might slowly decrease the level of robot autonomy, allowing the user’s cortical plasticity to gradually subsume low-level flexible control of the manipulator.

A potential drawback of shared control is that flexibility of the system may be lost. It is worth noting that the type of partial autonomy given to the robot in this study would not have made any of the points in a workspace unreachable to a human or non-human primate user. Because control was shared, every point in the space remained reachable to a user capable of modulating motor commands well enough to offset the contribution of the sensors. In principle, objects could still be “almost but not quite” touched, if that were the user’s goal. Likewise, a user could have pushed around an object with the outer
edges of the gripper by adjusting his or her motor command to offset the sensor-based withdrawal reflex.

Also, it is important to acknowledge one of the main shortcomings of the results described in this paper. As described previously, continuous shared control has been demonstrated here using pre-recorded neural data. The offline nature of this demonstration omits the important effects of visual feedback to the brain. Given feedback, the brain would be able to adapt and learn the new dynamics introduced by shared control, potentially producing different results from those reported here. Despite this limitation, the results show unambiguously that continuous shared control improved task success significantly, even without online adaptation and learning of the brain. It seems reasonable to suppose that online visual feedback to the monkey of the resulting robot trajectories might lead to further enhancements in performance.

The noise level in the brain-commanded trajectory has not been reduced in these demonstrations. However, noise reduction can now be done by low-pass filtering the predicted trajectories from neural signals. Previously, it was stated that low-pass filtering of the trajectories was not ideal, because the system would become less responsive. However, with the Continuous Shared Control strategy, responsiveness can be added to the system through the sensor-based reflex reactions. The low-pass filtered brain commands can suggest the general direction of the movement and allow the sensors to respond according when the manipulator comes closer to the target object. Applying a low-pass filter to brain-commanded trajectories is also in tune with our neuromuscular system, because muscle recruitment dynamics act as a low pass filter between cortical commands and motion output.

2.5 Conclusion

The design space of BMI shared control systems appears to be large, and rich with interesting possibilities. For example, the shared control paradigm provides a convenient framework for incorporating the software equivalent of a central pattern generator into a walking BMI. A more pedestrian improvement to shared control BMIs would be to put the weight of the sensor-derived commands under the user’s instantaneous brain control.
Activity of cortical neurons is well known to reflect focused attention (Colby and Goldberg 1999; Reynolds, Chelazzi et al. 1999; Kastner and Ungerleider 2000; Lebedev and Wise 2001). Therefore, monitoring user’s spatial attention and analyzing the corresponding neuronal patterns may provide the key to adjusting sensor-derived commands on the fly. It also looks feasible to make the type of reflex response that the sensors generate (attraction, repulsion) subject to the user’s continuous control. Furthermore, it would be interesting to replace the practical, linear, attractive/repulsive reflex actions used in this study with time-varying reflexes designed to mimic spinal reflexes as closely as possible. This would facilitate quantitative comparisons between the optimal weighting of robot reflexes and the optimal weighting of spinal reflexes compared to the user’s central motor drive. In future applications the type of control, ranging from purely supervisory to purely reflexive, might be determined post-surgery, based on the type of information that the subsample of recorded neurons best represents. If neurons representing higher-level information such as task goals and target objects were mainly acquired, the robot could operate close to supervisory control, whereas if neurons carrying lower-level trajectory information were acquired, the cortex could be given more direct control.

The intention has been to introduce a general purpose Continuous Shared Control paradigm for brain-machine interfaces. The specific problem of grasping in unstructured environments, and the specific implementation of the gripper, range sensors, and reflex actions were intended primarily to illustrate the approach. It seems clear the present study has only scratched the surface of what sensor based actions and increased autonomy could add to the capabilities of BMIs. Regardless of the specific implementation, the paradigm of continuous shared control can provide a framework for adding more complex and interesting BMI components in the future.
3 Impedance Control in BMI

3.1 Introduction

There have been numerous demonstrations of non-human primates controlling robots or graphical cursors in real-time through signals collected from cortical areas. These demonstrations can be divided largely into two categories: either continuous control of position (Wessberg, Stambaugh et al. 2000; Serruya, Hatsopoulos et al. 2002; Taylor, Tillery et al. 2002; Carmena, Lebedev et al. 2003; Taylor, Tillery et al. 2003), or discrete control of more abstract information such as intended targets, intended actions, and onset of movements (Shenoy, Meeker et al. 2003; Andersen, Burdick et al. 2004; Musallam, Corneil et al. 2004).

In BMI systems that use the continuous control of position strategy, much of the work thus far has treated the primate as a pure motion source. This makes intuitive sense considering that numerous studies show that much cortical activity is tuned broadly to higher-level features of hand movement such as position (Georgopoulos, Kalaska et al. 1982) and velocity (Schwartz 1994; Moran and Schwartz 1999). Using this approach, neural recordings have been used to predict hand trajectory with reasonable accuracy (Wessberg, Stambaugh et al. 2000; Serruya, Hatsopoulos et al. 2002; Taylor, Tillery et al. 2002; Carmena, Lebedev et al. 2003; Taylor, Tillery et al. 2003). This strategy has proved to be an appropriate first pass at BMI, confirming the potential for decoding movement features from cortical areas in order to drive neuroprostheses for paralyzed patients.

In this chapter, we discuss the next logical step -- impedance control through a brain machine interface.
A motion source has infinite mechanical impedance. It is the complement of a force source, which has zero mechanical impedance. These ideal sources have behavior independent of the load they drive. However, the mammalian neuromuscular system is neither an ideal force source nor an ideal motion source, and its mechanical impedance is far from either of the two extremes. In fact, the neuromuscular system is highly adaptive and can modulate mechanical impedance. Numerous studies show that this modulation is essential for versatile interaction with the environment (Hogan 1985; Burdet, Osu et al. 2001; Rancourt and Hogan 2001; Hogan 2002; Franklin, Burdet et al. 2003). For example, when writing neatly on tax forms or soldering circuits, we increase the stiffness of our hand movements for precise position control. When catching an egg or handling delicate objects, we decrease the stiffness.

In prior work on BMIs this stiffness information was lost, along with other dynamic parameters such as intended inertia and viscosity. The decoding algorithms did not have mathematical forms that could capture these aspects of motor performance. The main goal of the new algorithm that will be introduced in this chapter is to decode mechanical impedance (intended inertia, viscosity and stiffness) from neural activity, so that impedance control (Hogan 1985) can be implemented on a slave robot, and the user’s intended impedance can be displayed to the environment. If successful, this approach would give the neuroprosthesis user the sort of prosthesis control necessary to perform the wide range of tasks that interaction with objects demands, whether that task is filling out a tax form or catching an egg.

There are two general approaches to the problem of decoding impedance. The approach one takes depends on a deep-rooted argument in neuroscience about how motion is encoded in the cortex. Does the cortex only specify high level movement features and leave the conversion of this relatively abstract representation into muscle commands for neurons downstream, or does the motor cortex, along with other cortical areas, calculate the inverse dynamics and kinematics of the limb in order to send muscle activity information directly?

In support of the “abstract motion” view, Georgopoulous et al. (1982) originally observed that M1 neurons were broadly tuned to hand movement direction, noting that each neuron had a preferred direction for which it exhibited maximal firing rate.
Although this finding ran counter to Evarts (1968) original observation that M1 firings are correlated to forces, subsequent work has shown good correlation between hand space movement and neural activity (Georgopoulos, Caminiti et al. 1983; Schwartz 1994; Moran and Schwartz 1999). Thus, many have accepted the idea of population vectors as a working model of the motor cortex, and hold the view that the motor cortex controls the higher level features of hand movement, rather than lower level details related to joints and muscles.

Yet, there is persuasive evidence supporting the opposing view as well. For example, Kalaska et al. (1989), and Sergio and Kalaska (1998) observed that the same cells that show directional tuning for movement also encode force on an object during movement in an isometric task. Also, several studies have found systematic differences in neural activity depending on joint configuration for identical movement directions (Caminiti, Johnson et al. 1991; Scott and Kalaska 1995; Kakei, Hoffman et al. 1999). These studies suggest that the M1 area does not purely encode motion in hand space. In recent publications, Todorov (2000; 2003) has argued that the motor cortex activity is most highly correlated with muscle activity. His hypothesis is based on studies that show correlation between muscle activation (magnitude and onset time) and M1 firings (Humphrey and Reed 1983; Scott and Kalaska 1995; Scott 1997; Scott, Gribble et al. 2001). In support of his argument, he has linearly mapped motor cortex activity to muscle activities, and used these activities to drive a simplified musculoskeletal model. The approach reproduced many of the observations of Georgopolous et al. (1982) and Schwartz (1994) among others.

Which model better describes the function of the motor cortex is still not certain. The most likely answer is that it is a mixture of both. In fact, studies by Kakei et al. (1999) and Li et al. (2001) show that large percentages of cortical neurons correlate well with muscle activities and equally large percentages correlate well with direction. Returning to the question at hand, it seems likely that impedance can therefore be decoded using either of the two models.

In the first “high level abstraction” approach, one would proceed from the assumption that neural activity in the cortex correlates well with higher-order impedance information, such as the end effector stiffness matrix, just as it correlates with higher-
order movement features like hand position. One would suppose that, since humans can make conscious decisions about desired stiffness, it should be possible to do a “direct extraction” of this information from cortical areas. To implement this approach, impedance would be measured during a primate’s movement, and fit directly to neural activity, just as position and velocity have been fit to neural activity in previous BMIs.

In the second “muscle activity” approach, one would proceed from the assumption that the cortex contains an inverse model of limb dynamics and kinematics, and that therefore neurons with output well correlated with muscle activity are present. Since muscles act like damped springs driving limb inertias, signals related to muscle activity already contain information about the intended limb impedance. To implement this approach, the endpoint impedance would be extracted by decoding the outputs of neurons correlated with muscle activity, and inputting these estimates of muscle activation into a computational musculoskeletal model that would predict the state of the limb subjected to those muscle activities.

Although the first method - direct extraction of impedance parameters - might be straightforward to understand, practical concerns limit the probable success of this approach in BMIs. Direct extraction requires measurements of limb impedance in order to train any prediction algorithm. In principle, this could be done. Impedance parameters of the primate limb have been estimated by applying small force perturbations under static conditions (Mussa-Ivaldi, Hogan et al. 1985) and during movement (Gomi and Kawato 1996). In practice, however, impedance is an inconvenient parameter to measure. It varies with arm posture, the degree of muscular co-contraction, and the direction of perturbation. Thus, to implement the direct method would require an onerous set of measurements – a sampling of all possible arm configurations, perturbed in all possible directions, during all levels of muscular co-contraction. Gathering this data would be tedious for a human subject, particularly if it had to be performed before every BMI trial. Unfortunately, it would also be a difficult topic to investigate in an animal model, since training a monkey to apply different levels of co-contraction is likely to be an unwieldy task in itself.

The more practical option is to take the second approach and to predict impedances by establishing a musculoskeletal model. This approach, which we call the Muscle
Activation Method (MAM), has several advantages. In the Muscle Activation Method, a fitting algorithm takes the neural data as input, and outputs an intermediate variable. The intermediate variable is analogous to muscle activation. This "muscle activation" is input to the musculoskeletal arm model, which, in turn, outputs all parameters related to motion - both kinematic and dynamic.

The first, key advantage of the Muscle Activation Method is that all motion parameters are properly inter-correlated and mutually consistent, whereas in previous BMIs, movement parameters (e.g., force and position) have been treated as independent, potentially leading to conflicting predictions about what the prosthesis should do. The second, key advantage of the Muscle Activation Method is that the need for data collection is greatly reduced. Given a training set that spans the entire "muscle activation space" of the test set, the algorithm can be trained satisfactorily on unimpeded limb trajectories, without requiring difficult and time consuming measurements of limb impedances.

The primary focus of this chapter is to introduce and characterize the Muscle Activation Method, a musculoskeletal model-based extraction algorithm for impedance control. To demonstrate a working algorithm, neural data from a behaving monkey was collected and used to test the algorithm in two ways. The first test determined whether the extraction algorithm, which includes a musculoskeletal arm model, can predict movement parameters to a level of accuracy that is comparable to that shown in previous BMI studies. The second test investigated how well the Muscle Activation Method predicts the outcome of dynamic interactions with the environment, something that a traditional trajectory prediction algorithm can not do.

3.2 Methods

3.2.1 Musculoskeletal arm model

A realistic model of the monkey’s arm was needed in order to extract motion parameters from the cortical signals. The model was developed by combining published biomechanical data with published models of muscle force generation.
3.2.1.1 Muscle geometry

The muscle geometry - namely the insertion point, origin point, and line of action of each muscle group – provides two pieces of information required for computing joint torques. First, the muscle geometry gives the transformation from joint space to muscle length space. Muscle lengths and velocities permit calculation of muscle forces. Second, the muscle geometry gives the transformation from joint space to moment arm space. The moment arms permit calculation of the torques that each muscle force causes about each joint.

For simplicity, the arm model was constrained to planar movement with two degrees of freedom, one at the shoulder and the other at the elbow. The six major muscle groups used in Gribble et al.’s (1998) planar arm model were included: a single-joint flexor and extensor at the shoulder (pectoralis and deltoid) and at the elbow (biceps long head and triceps lateral head) and also two double-joint muscles spanning both the shoulder and elbow (biceps short head and triceps long head). The origin and insertion points of each muscle group were taken from data measured by Wood et al. (1989), who used a 3-D digitizer to measure arm anatomy of a human male cadaver. In our 2-D model, Wood et al.’s 3-D data was projected on to axial plane of arm movement (Figure 3-1). All dimensions were scaled linearly for the arm lengths of the monkey used in the experiments (upper arm = 11 cm, forearm = 20 cm to the center of the hand).
Muscle origins mounted to main body (Fixed)

- Muscle origins/insertions mounted to upper-arm (Moving)
- Muscle insertions mounted to lower-arm (Moving)
Figure 3-1 Muscle geometry for the musculoskeletal arm model. The following abbreviations are used for the muscle group; pectoralis major –PMJ, biceps short head – BSH, biceps long head – BLH, deltoids – DEL, triceps long head – TRIO and triceps lateral head – TRIA. The letters (a_i - d_i) mark the coordinates of the origin and insertion points for each muscle group. The coordinate values are shown in Table 1. A is an enlarged view of the shoulder joint and B is an enlarged view of the elbow joint.

The following abbreviations identify the muscle groups in Figure 3-1; pectoralis major – PMJ, biceps short head – BSH, biceps long head – BLH, deltoids – DEL, triceps long head – TRIO and triceps lateral head – TRIA. Origin points for the pectoralis, deltoid, bicep short head and triceps long head are given as x-y coordinates in the world coordinate frame, fixed with respect to the torso. The coordinates of the muscle origins for the biceps long head and triceps lateral head, as well as the insertion points for the pectoralis and deltoid remain fixed to the upper-arm, and are expressed as radii and angles from the center or rotation of the upper limb. The insertion point coordinates for the biceps long head, biceps short head, triceps long head and the triceps lateral head are given as radii and angles from the center or rotation of the lower limb. The dimensions are given in Table 1.

<table>
<thead>
<tr>
<th>Muscle Group</th>
<th>Origin</th>
<th>Insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a_i</td>
<td>b_i</td>
</tr>
<tr>
<td>1 PMJ</td>
<td>-0.0477 m</td>
<td>0.0278 m</td>
</tr>
<tr>
<td>2 BSH</td>
<td>-0.0080 m</td>
<td>0.0126 m</td>
</tr>
<tr>
<td>3 BLH</td>
<td>0.0089 m</td>
<td>126.1°</td>
</tr>
<tr>
<td>4 DEL</td>
<td>-0.0051 m</td>
<td>-0.0060 m</td>
</tr>
<tr>
<td>5 TRIO</td>
<td>0.0058 m</td>
<td>-0.0093 m</td>
</tr>
<tr>
<td>6 TRIA</td>
<td>0.0496</td>
<td>8.69°</td>
</tr>
</tbody>
</table>

Table 1 Dimensions for the musculoskeletal arm model muscle geometry shown in Figure 3-1.

Tendons of the three extensors (DEL, TRIA, TRIO) wrap around bones at both the shoulder and elbow. These segments are modeled as pulleys of radius 1.43 cm and 0.711 cm at the shoulder and elbow respectively. The instantaneous moment arms of the
extensors remain constant and correspond to the radius of the pulley, as is modeled in Gribble et al. 's (1998). The moment arms of the flexors (PMJ, BLH, BSH) were calculated from muscle kinematics. The direction in which a given muscle pulled on the skeleton was approximated as a line between the muscle origin and insertion. The length and its rate of change of the muscles could also be calculated with the same assumptions used for the moment arms. Graham and Scott (2003) have measured moment arms for the same planar movement in monkeys (Macaca mulatta) and have reported polynomial coefficients for moment arm regressions to joint angles. These regressions could have been used instead of our geometrical model.

### 3.2.1.2 Muscle Force Model

A Hill-type model (Zajac 1989) was used for muscle force generation, with the muscle forces divided into passive and active components. The two components were given analytical expressions by approximating the shape of the dimensionless force-length curves shown by Zajac (1989). The region where active muscle force is generated is nominally between 0.5\(l_o\) and 1.5\(l_o\), where \(l_o\) is the optimal fascicle length at which maximum force is generated (Edman 1966; Gordon, Huxley et al. 1966; Close 1972; Gans 1982). The force-length property of a less than fully activated muscle tissue can be considered to be a scaled version of the fully activated muscle (Hatze 1977; Winters and Stark 1985). Therefore, as an approximation in this model, the active force was assumed to be quadratic in length and linear in activity, with the maximum force, \(F_{\text{max}}\), occurring at the optimal fascicle length, \(l_o\) of each muscle group. The muscle activation, \(\alpha\), scales the quadratic function. This is given by equation 1 and shown in Figure 3-2a.

\[
F_a = \alpha F_{\text{max}} \left[ 1 - 4 \left( \frac{l - l_o}{l_o} \right)^2 \right] 
\]

where,

\([u]^+ = \max[u,0]\).
The passive force component was given by the equation,

\[ F_p = \left[ \frac{F_{\text{max}}}{\exp(K_{sh}) - 1} \right] \exp\left( \frac{K_{sh}(l - l_o)}{0.5l_o} \right) - 1 ]^{1/2} \]

for \( l \geq l_o \) and \( F_p = 0 \) otherwise. The shape of the exponential was determined by the variable \( K_{sh} \). We set \( K_{sh} = 3 \). The resulting passive force-length curve is shown in Figure 3-2b.

\[ F = \begin{cases} \frac{F_{\text{max}}}{\exp(K_{sh}) - 1} \exp\left( \frac{K_{sh}(l - l_o)}{0.5l_o} \right) - 1 \end{cases} \]

\[ \text{length} \]

\[ \text{activation} \]
(pectoralis – 7.9 cm, biceps short head – 6.6 cm, biceps long head – 5.4 cm, deltoids – 2.7 cm, triceps long head – 4.3 cm and triceps lateral head – 3.8 cm). $F_{\text{max}}$ was assumed to be proportional to the physiological cross sectional area (PCSA) of the muscles (Gribble, Ostry et al. 1998; Cheng and Scott 2000). The PCSA for the muscles were found by the linear regression of *macaca mulatta* PCSA against total body weight, provided by Cheng and Scott (Cheng and Scott 2000)(Table 2).

<table>
<thead>
<tr>
<th>Muscle</th>
<th>m (slope)</th>
<th>b (y-intercept)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMJ</td>
<td>0.61</td>
<td>2.73</td>
</tr>
<tr>
<td>BSH</td>
<td>0.17</td>
<td>1.14</td>
</tr>
<tr>
<td>BLH</td>
<td>0.44</td>
<td>1.14</td>
</tr>
<tr>
<td>DEL</td>
<td>0.54</td>
<td>0.83</td>
</tr>
<tr>
<td>TRIO</td>
<td>1.10</td>
<td>1.73</td>
</tr>
<tr>
<td>TRIA</td>
<td>0.68</td>
<td>2.34</td>
</tr>
</tbody>
</table>

Table 2 Linear regression for PCSA (cm$^2$) against total body weight (kg) for *macaca mulatta*.

A scale factor of 22.5N/cm$^2$ (Lieber and Friden 2000) from PCSA to $F_{\text{max}}$ was used. The resulting $F_{\text{max}}$ values for the monkey used in this study (5.23 kg) were; pectoralis – 133 N, biceps short head – 45.7 N, biceps long head – 77.4 N, deltoids – 82.2 N, triceps long head – 168 N and triceps lateral head – 133 N.

The force of each muscle was the sum of active and passive components as shown in Figure 3-2c. The resulting shape of the force-displacement-activation curve appears to be a satisfactory approximation of the normalized curve presented by Zajac (Zajac 1989) which characterizes empirical measurements. Figure 3-3 shows the curve as presented in his paper. Comparing the two plots (Figure 3-3 and Figure 3-2c), the quadratic approximation of the active force-length relation appears to be greatest source of inaccuracies in the model. The gradient of the curve is steeper at the extremes for the quadratic function than that shown in Figure 3-3. Also, the empirical data is not as symmetrical as the quadratic model. However, in the range of muscle lengths used for
this study (shown later in Figure 3-5), the maximum error in the gradient is less than 10% and thus the model serves as a satisfactory approximation.

![Graph showing force-displacement property for muscle tissue](image)

Figure 3-3 Normalized force-displacement property for muscle tissue as shown in Zajac (1989). The curve is for full activation, $\alpha=1$.

The velocity dependence of the total muscle force was calculated as:

$$f_{\text{total}} = [a_1 + a_2 \tan(a_3 + a_4 \dot{l})]a_5 + f_p, \quad (3)$$

where $\dot{l}$ is the rate of change of the muscle length and with $a_1 = 0.80$, $a_2 = 0.50$, $a_3 = 0.43$ and $a_4 = 58 \text{ s/m}$. This has the same form as the sigmoidal function used in Gribble et al. (1998), and is consistent with Zajac’s (1989) force-velocity curve (Figure 3-4).
The visco-elasticity of the tendons was not modeled separately. Rather, for simplicity the tendon length ratios were assumed constant, and the muscle fiber lengths were calculated as a constant percentage of the total origin-to-insertion length. The percentage of muscle fiber length, $l$, per total origin-to-insertion length was assumed 81% for biceps short head, 79% for biceps long head, 83% triceps and 95% for the pectoralis and deltoids. Muscle fiber length ratios were estimated from anatomical drawings (Martini and Timmons 1997). The resulting muscle fiber lengths from these approximations were compared to experimental measurements of muscle fiber lengths (Murray, Buchanan et al. 2000) and are shown in Figure 3-5. The plots are for elbow flexions in the range of 20° to 120°. The muscle operating ranges shown in the figure agree quite well. The range of bicep muscle length from Murray et al. (2000) and the musculoskeletal model are both below the optimal fascicle lengths. This suggests that biceps do not go into passive tension within this range of elbow flexion. The triceps do extend slightly past the optimal fascicle length and into the passive tension zone. Given that range of motion during the behavioral tasks (described later in section 3.2.5) is actually smaller than that shown in Figure 3-5, one can expect the motion to be in the region where the active forces are dominant over passive forces.
Figure 3-5 Estimated operating ranges of the elbow flexors and extensors plotted by Murray et al. (Murray, Buchanan et al. 2000). Estimated fascicle excursions were normalized by optical fascicle length and superimposed on a normalized force-length curve. The dotted lines bounded by red circles show the same operating ranges for the musculoskeletal model developed for this study.

The muscle recruitment and force development dynamics could have been modeled with a second (Gribble, Ostry et al. 1998) or first order (Zajac 1989) low pass filter. However, these phenomena were omitted from the musculoskeletal model for simplicity, and instead were intentionally left for processing by the artificial neural network through the delayed time bins, as discussed later.
3.2.1.3 Dynamic Model of the Arm

The dynamics of the arm during planar movement were modeled as a two-link manipulator with frictionless rotational joints (Figure 3-1). The equation of motion was:

$$\tau_\theta + \tau_\text{external} = \tau_\text{muscles}$$

$$\begin{bmatrix} I_{11} & I_{12} \\ I_{12} & I_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_s \\ \ddot{\theta}_e \end{bmatrix} + \begin{bmatrix} -h\dot{\theta}_e - h\dot{\theta}_e - h\dot{\theta}_e \\ h\dot{\theta}_e \end{bmatrix} + \tau_\text{external} = \tau_\text{muscles}$$

$$I_{11} = m_1 l_1^2 + I_1 + m_2 (l_1^2 + l_2^2 + 2l_1 l_2 \cos \theta_e) + I_2$$
$$I_{22} = m_2 l_2^2 + I_2$$
$$I_{12} = m_2 l_1 l_2 \cos \theta_e + m_2 l_2^2 + I_2$$
$$h = m_2 l_1 l_2 \sin \theta_e$$

where $\tau_\theta$ are the inertial torques, $C(\dot{\theta}, \theta)\dot{\theta}$ are the coriolis and centripetal torques, $\tau_\text{muscles}$ are the torques generated by the muscles and $\tau_\text{external}$ are the external torques applied by the environment. The inertial parameters were obtained from Cheng and Scott’s linear regressions on monkey data (Table 3)(Cheng and Scott 2000).

<table>
<thead>
<tr>
<th>Segment</th>
<th>Upper arm</th>
<th>Forearm + Hand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass (g)</td>
<td>m (slope)</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>b (intercept)</td>
<td>0.17</td>
</tr>
<tr>
<td>$I_c$ (g cm$^2$)</td>
<td>m</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table 3 Linear regression for segment mass and moment of inertia against total body weight (kg) for macaca mulatta.
3.2.2 Prediction Algorithm

A variety of mathematical approximators could have been placed between the neural data and the musculoskeletal arm model in order match arm movements predicted from neural data to those observed in the limb. Both a linear filter and an ANN were considered. Because the inputs to the musculoskeletal model were muscle activations that should not have negative values, linear filters – which can give negative values - were rejected. Since a nonlinear function was required to generate the muscle activations, an ANN with non-negative sigmoidal transfer functions was chosen (Figure 3-6).
Neuronal activity (136-207 cells with 9 time lag bins)

$E = \sum \frac{1}{2} (d_i - y_i)^2$

6 x muscle activations

Arm dynamics
- Force to torque transformation
- Angle to muscle length transform

Force generation model x 6 muscles

Computational musculoskeletal arm model
The standard back propagation algorithm for adjusting the weights of the neural net had to be modified to account for the arm model between the ANN and the output. This problem was solved by treating the final layer of the ANN as a hidden layer and the arm model as an output layer. With this approach, changes in the weights of the ANN could be calculated in the usual way (Haykin 1994), and gradient descent could be used to minimize the cost function. The algorithm for updating a weight in a hidden layer (l-th layer) is given by:

$$
\Delta w_{y} = \eta \delta_{i}^{l} x_{j}^{l-1} \\
\Delta b_{i} = \eta \delta_{i}^{l} \\
\delta_{i}^{l} = \frac{\partial E}{\partial x_{i}^{l}} f'(u_{i}^{l}),
$$

where $\eta$ is the learning rate, $x_{j}^{l-1}$ is the activity of j-th neuron in the l-1-th layer, E is the cost function defined by half the sum of the squared errors of the output,

$$
E = \frac{1}{2} (y_{k} - d_{k})^{2},
$$

$f(u)$ is the sigmoidal transfer function of each node in the artificial neural net,

$$
f(u) = \frac{1}{1 + e^{-u}}
$$

and the activity of each node depends on the weighted sum of the inputs $x_{j}$ and the node bias $b_{i}$:

$$
u_{i}^{l} = \sum_{j=1}^{n} w_{ij}^{l} x_{j}^{l-1} + b_{i}^{l}.
$$
The value of $\delta_i$ for the final layer in the ANN in our combined algorithm is given by,

$$
\delta_i^l = \frac{\partial E}{\partial x_i^l} f'(u_i^l)
$$

$$
= \sum_k \left( \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial x_i^l} \right) f'(u_i^l)
$$

$$
= \sum_k \left( y_k - d_k \right) \frac{\partial y_k}{\partial x_i^l} f'(u_i^l),
$$

where, $y_i$ is one of the $n$ selected outputs from the musculoskeletal arm model and $d_i$ is the corresponding "target" value from the training set. The gradient $\frac{\partial y_k}{\partial x_i^l}$ (the partial differential of the $k$th output variable with respect to $i$th muscle activation) can be found analytically or numerically. We found it numerically by applying a small change in muscle activation and observing the resulting change in output (Wang and Buchanan 2002) as shown in Equation 10. Over many iterations (2000-30000), the algorithm converged to minimize the cost function, given the right choice of learning rate and initial conditions.

$$
\frac{\partial y_k}{\partial x_i^l} = \frac{\partial y_i}{\partial x_i^l}. 
$$

At this point it is worth noting some elegant features of this approach. First, provided that the training set spans the entire muscle activation ranges present in the test set, the algorithm should predict motion parameters in the test set to a reasonable degree of accuracy. No explicit measurements of limb impedance are required. Second, the muscle activations in the hidden layer do not have to be measured. If the movement in the training set spans the muscle activation space, the hidden layer should, in theory, converge to the actual muscle activations with accuracy depending on the accuracy of the arm model. Correlation of the trained hidden layer with indicators of muscle activity,
such as surface EMG, then serves as an independent validation of the correctness of the modeling approach.

In this study, a training set of 10 minutes of neural signals and movement data was used to train the network to predict the subsequent 1 minute of movement. The neural activity was processed for the model by grouping the neural spikes in time bins of a reasonable size (100 ms). The input to the ANN was the current bin of neural activities from all recorded cells, plus a train of prior bins that provided enough history for the network to approximate the effects of transmission delay, muscle recruitment dynamics, and lingering intention signals. For this purpose, nine prior 100 ms bins were used. The bin size and number of prior bins were determined by optimizations previously conducted with a linear filter for best correlation coefficient (Wessberg, Stambaugh et al. 2000; Carmena, Lebedev et al. 2003). The parameters make sense biologically, since voluntary movement of the upper limb has a bandwidth of less than 5Hz (Neilson 1972) and conduction and muscle recruitment delays usually range from 25 ms-500 ms.

In principle, any motion variable downstream from muscle activations could have been predicted using this musculoskeletal model approach, and the output of the ANN optimized with respect to error in that variable. For this study, the torques generated by the muscles at the two joints (shoulder and elbow) were selected. Muscle torques were selected because they are a compact set of variables (just two in this case) that, together with limb inertias, determined all motion parameters (accelerations, velocities and positions). In minimizing the model error, errors in the two torques were weighted equally and summed. The estimated torques in the monkey’s joints were calculated by inputting the measured joint angles, angular velocities and angular accelerations to the dynamics equation (Equation 4). Velocities and accelerations were computed from the joint encoder values using finite-difference approximations.

3.2.3 Online System for Impedance Control

To apply impedance control to an online manipulator, the desired trajectory along with the intended inertia, viscosity and stiffness has to be sent to the robot. Impedance information can be extracted from the musculoskeletal model through a series of Jacobian
and Hessian transformations. Given the intended impedances \((K_d, B_d, M_d)\) and trajectory \((X_d, V_d)\), the control torques for the robot can be found by,

\[
I(0)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau + J^T F_{\text{ext}}
\]

\[
F_{\text{ext}} = K_d (X_d - \dot{X}) + B_d (V_d - \dot{V}) - M_d \frac{dV}{dt}
\]

\[
\tau = J^T W^{-1} M_d^{-1} \left[ K_d (X_d - X) + B_d (V_d - V) \right] + J^T W^{-1} \left[ J^T C \ddot{\theta} - J^T \dot{\theta} \right] + J^T \left( I - W^{-1} M_d^{-1} \right) F_{\text{ext}}
\]

where \(I(0)\ddot{\theta}\) and \(C(\theta, \dot{\theta})\dot{\theta}\) are the inertial and coriolis torques of the robot and \(g(\theta)\) is the gravitational torque (Hogan 1985).

However a simpler approach is to simulate the interaction within the musculoskeletal model. A force sensor can be attached to the manipulator to measure the external forces. Then the simulated external forces can be applied to the dynamic equation (Equation 4) of the online musculoskeletal model. The resulting trajectories can be then sent to the robot. With this approach, the robot can now be controlled under pure motion control given by,

\[
\tau = K_p (0 - \theta_d) + K_v (\dot{\theta} - \dot{\theta}_d) + I(0)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta),
\]

where \(K_p\) and \(K_v\) are gain constants for the PD control determined by the performance criteria of the robot for position control.
Figure 3-7 Schematic of proposed online implementation of impedance control (an enlarged version of the musculoskeletal model can be seen in Figure 3-6).

### 3.2.4 Electrophysiology

An adult female monkey (*Macaca mulatta*) was used in this study. All procedures conformed to the National Research Council's *Guide for the Care and Use of Laboratory Animals* (1996) and were approved by the Duke University Animal Care and Use Committee. The monkey was implanted with four arrays containing 32 microwires each (with 1mm separation between each wire) in frontal and parietal cortical areas. The microwire electrodes were 15 microns in diameter and were made of tungsten with gold plated tips.
The implants were in the motor cortex (M1 – 64 electrodes), sensory motor cortex (S1 – 32 electrodes) and the premotor cortex (PMd – 32 electrodes) of the left hemisphere. For each implant, small craniotomies were made using coordinates from a stereotaxic device and anatomical maps. The dura mater and arachnoid layer were removed and landmarks such as the central sulcus and arcuate sulcus were used to locate the desired area for the implants. The electrodes penetrated the pia mater and were inserted until a good population (more than 2 cells in all channels) of neuronal activity was recorded. The neuronal activity in the channels was observed online during the implants. On average, the penetration depth was about 1.5 mm from the surface of the pia, which corresponds to the depth of layer 5 of the motor cortex. Once a reasonable signal was achieved, the implants were fixed with various adhesives and covered with composites used for dental filling (Figure 3-9).
The signals from the electrodes (128 channels) were pre-amplified (×3000) and sent to a 512-multichannel acquisition processor (Plexon Inc., Dallas, TX). The processor performed simultaneous A/D conversions of all channels at 40 kHz at 12-bit resolution. The signal was digitally filtered with an 80 Hz 2-pole low-cut filter and an 8 kHz 6-pole high-cut filter. Sort client software was used in conjunction with digital signal processing hardware to sort signals of the individual neuron from each other and the noise, in real-time. The neuronal signals were initially processed by only capturing the characteristic triggered spikes of neuronal activity by manually adjusting a threshold. The sorting of the individual neuron signals in each channel was done by identifying clusters in a 2 dimensional principal component space. The neuronal signals were large and formed well separated clusters in most of the channels, making the sorting relatively easy. The sort client was able to sort up to 4 neurons per channel online and store the time stamps of each cell. Figure 3-10 shows a screen shot of the sort client, 2 weeks after the electrodes
were implanted in the monkey. The windows in the right show the triggered signals in all 128 channels. The first four columns are in the premotor cortex (PMd), the next eight are motor cortex (M1) and last four columns are sensory motor cortex (S1). The window in the top left corner shows an enlarged shot of one of the channels. Four distinct signals can be seen in this channel, with each signal coming from a different neuron in the vicinity of the electrode and having its own characteristic waveform. The principal component space clusters and the waveform of each neuron are shown in a row just below the large window. The activity time stamp trains of each neuron are also shown at the bottom. The selected channel shows an example of a good electrode, where the signals are much larger than the noise and well separated. Fortunately, there were many such channels for this set of implants.

Over thirteen sessions, the number of active cells simultaneously recorded ranged from 201 to 391. For the musculoskeletal model, the neuronal activity of only the 136-207 cells in the motor cortex implants in the first monkey were used. It was assumed that
the M1 neurons were most likely to be highly correlated with muscle activation (Todorov 2000; Todorov 2003) and thus best suited for the Muscle Activation Method predictions.

3.2.5 Behavioral tasks

The monkey was trained to perform center-out tasks to 12 targets (7-15cm from the center), while seated in a KINARM exoskeleton (BKIN Technologies). The KINARM constrains the monkey’s arm in planar motion, and has motors that are able to apply external torques at the two joints. The cursor and target were displayed on a reflective surface in front of the monkey and above the monkey’s hand. The cursor was superimposed on top of the monkey’s hand and updated to follow the hand’s motion.

![Image of KINARM exoskeleton](image)

Figure 3-11 The KINARM exoskeleton used for the primate behavior tasks. The arm troughs constrain the monkey to planar movement and the motors are able to apply external torques during the tasks. The tasks is displayed visually to the monkey on a reflective surface in front of the animal.

The center out task involved holding the cursor within the red center target for 2 seconds. Subsequently, one of the 12 green outer targets would appear. The monkey would have to move the cursor to the target and hold for 2 seconds to obtain a juice
reward. Once the juice was rewarded, the center target would reappear and the task would be repeated (Figure 3-12).

**Figure 3-12** The center out task. The task involved holding the cursor in the red center target for 2 seconds and then moving to the green outer target and then holding for another 2 seconds. The gray dot on the cursor disappears when the cursor inters the target.

Four different force conditions were planned for the tasks.

Task 1: Null force task – No force is applied.

Task 2: Viscous force task – A force field is applied in proportion to velocity in hand space (Cartesian space) and opposite to the direction of motion.

\[
\begin{bmatrix}
F_x \\
F_y
\end{bmatrix} =
\begin{bmatrix}
-B & 0 \\
0 & -B
\end{bmatrix}
\begin{bmatrix}
v_x \\
v_y
\end{bmatrix}
\]  

(13)

Task 3: Curl force task – A force field is applied proportional to the velocity in hand space and perpendicular to direction of motion.

\[
\begin{bmatrix}
F_x \\
F_y
\end{bmatrix} =
\begin{bmatrix}
0 & C \\
C & 0
\end{bmatrix}
\begin{bmatrix}
v_x \\
v_y
\end{bmatrix}
\]  

(14)

Task 4: Divergent force task – A force field is applied perpendicular and linearly divergent from a desired path. (The force field acts like a negative spring pulling the arm away from the desired path.)
Viscous force fields have been applied extensively in many behavioral studies. Gribble and Scott (Gribble and Scott 2002) used the KINARM to produce viscous forces at different joints to show that internal models in M1 overlap. Shadmehr and Mussa-Ivaldi (Shadmehr and Mussa-Ivaldi 1994) used a similar device with viscous forces on human subjects to show that adjusting internal models were not constructed like look-up tables. Curl force fields have been previously used by Li et al. (Li, Padoa-Schioppa et al. 2001) to show the presence of kinematic, dynamic and memory neurons. The two key differences between curl and viscous force fields are that in curl force fields the direction of the force is not along the direction of movement and that after effects are easily observable. Curl force fields are conservative; meaning the direction of force is always perpendicular to the direction of movement. Divergent force fields were used by Burdet et al. (Burdet, Osu et al. 2001) to investigate the human’s ability to optimize arm stiffnesses. The purpose of the divergent force fields of different magnitudes proposed for Task 4, was to induce the monkey to change arm stiffnesses. In doing so, it was hoped that direct representation of stiffness in the cortex could be observed.

Overall, the purpose of Task 2-4 was to apply forces that were coupled to the movement, where arm impedances would determine the resultant motion. Previous BMI tasks were purely kinematics tasks or tasks with uncoupled forces (grip forces), and thus it was possible to use a pure motion source or pure force source approach.

The monkey was able to learn to perform Task 1 and Task 2 well. The animal was also trained on Task 3. The quality of the neuronal recordings during Task 3 sessions was much reduced due to loss of cell signals. Furthermore, the neuronal recordings did not last long enough for the monkey to be trained on Task 4. It is common that neuronal recordings from chronic implants last between 1-2 months. Dying cells, relative movement of electrodes within the tissue and the formation of new tissue around the electrodes are usually responsible for the deterioration of the signal. The results presented in this study will primarily be from 13 sessions where Task 1 and Task 2 were preformed.
3.3 Results

Neural data from the monkey performing the task was used to test the Muscle Activation Method. As proposed in the introduction, two tests were conducted. The first determined whether Muscle Activation Method predicts motion parameters as well as other, more common, extraction algorithms. The second determined how well the Muscle Activation Method can predict motion parameters that current methods cannot.

3.3.1 Prediction of Motion parameters

Before proceeding further, we first determined whether the neuronal activities were related to motion at all. This was done with a simple linear filter (Appendix A), which prior studies have shown to be a quick and accurate way of predicting kinematic parameters for online BMI (Wessberg, Stambaugh et al. 2000; Schwartz, Taylor et al. 2001; Serruya, Hatsopoulos et al. 2002; Carmena, Lebedev et al. 2003). In the most
recent of these studies (Carmena, Lebedev et al. 2003), a linear Wiener filter was used to give a correlation coefficient for predicting position of \( r = 0.55-0.85 \). This is a meager correlation for engineering purposes; however it is accepted as a significant correlation in neuroscience given the uncertainty in decoding in neuronal signals described in previous chapters. Figure 3-14(a) shows the correlation coefficient for hand position prediction for the monkey in this study, using the same linear filter as in Carmena et al. (Carmena, Lebedev et al. 2003). The results are comparable. Thus, it was reasonable to assume that the implants in the monkey recorded neural data that was sufficiently correlated with arm movements. It was therefore sensible to proceed with fitting the musculoskeletal arm model.

Figure 3-14 (a) Correlation coefficient for Cartesian position prediction for a linear Wiener filter during 12 sessions. (b) Correlation coefficient for predicting muscle torques at the elbow and shoulder. The plot compares the linear Wiener filter with the Muscle Activation Method (MAM).

Turning now to the first hypothesis presented in the introduction, the results show that the predictions of the Muscle Activation Method have correlation coefficients equivalent to those of a linear filter. Figure 3-14(b) shows the prediction correlation coefficients for the monkey’s muscle torques (shoulder and elbow) across 13 sessions. Coefficients of the Muscle Activation Method and a traditional linear Wiener filter are both plotted.

\[ r^2 \] is also widely used as the correlation coefficient and is the squared product of the coefficient \( r \) which is used in this thesis.
Not only were the predictions of the Muscle Activation Method comparable to those of the linear filter, they were actually significantly better for most sessions. This was further verified through statistical tests. An analysis of variance (ANOVA) was conducted on the correlation coefficients from the 13 sessions. When comparing the mean of shoulder torque predictions for the two methods, the ANOVA test showed that the $p$-value for the null hypothesis that the means of the groups are equal is 0.0005% (Table 4). The box plots (Figure 3-15) show that the lower quartile of the predictions from the Muscle Activation Method is higher than the higher quartile of predictions from the linear filter. Therefore, it is safe to conclude that MAM performed better than the linear filter for predicting the shoulder torques.

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>$F$</th>
<th>Prob&gt;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>0.58425</td>
<td>1</td>
<td>0.58425</td>
<td>34</td>
<td>5.16752e-006</td>
</tr>
<tr>
<td>Error</td>
<td>0.41238</td>
<td>24</td>
<td>0.01718</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>0.99662</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4 ANOVA table for the prediction correlation coefficient of shoulder torque from the two prediction algorithms.

Figure 3-15 Box plot for the prediction correlation coefficient of shoulder torque from the two prediction algorithms. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are the dashed lines extending from each end of the boxes to show the extent of the rest of the data. The maximum length of the whiskers is 1.5 times the interquartile range. Outliers are data with values beyond the maximum whisker length and are shown as red crosses if they exist.
Similar statistics can be shown for the elbow torque predictions. The $p$-value for the null hypothesis that the means of the MAM and linear filter predictions for elbow torque are equal is 0.2% (Table 5). The box plots (Figure 3-16) show that the median of the predictions from the Muscle Activation Method is higher than the higher quartile of the predictions from the linear filter. Therefore, again, it is safe to conclude that MAM performed better than the linear filter for predicting the elbow torques.

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Prob&gt;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>0.24669</td>
<td>1</td>
<td>0.24669</td>
<td>12.02</td>
<td>0.002</td>
</tr>
<tr>
<td>Error</td>
<td>0.49248</td>
<td>24</td>
<td>0.02052</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>0.73917</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 ANOVA table for the prediction of elbow torque from the two prediction algorithms.

![Box plot for the prediction correlation coefficient of elbow torque from the two prediction algorithms.](image)

Figure 3-16 Box plot for the prediction correlation coefficient of elbow torque from the two prediction algorithms.

To complete the statistical analysis, the predictions of shoulder torque from the Muscle Activation Model are compared to the prediction of position from a linear filter. The prediction of a kinematic feature through a linear filter can be considered as a good representation of previous prediction levels. Therefore, the fact that the difference in the means of the two prediction levels is statistically insignificant ($p=30.4\%$) shows that the
Muscle Activation Method can predict to levels that are comparable to traditional algorithms.

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Prob&gt;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>0.02211</td>
<td>1</td>
<td>0.02211</td>
<td>1.1</td>
<td>0.3043</td>
</tr>
<tr>
<td>Error</td>
<td>0.48166</td>
<td>24</td>
<td>0.02007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>0.50377</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6 ANOVA table for the prediction correlation coefficient of shoulder torque from MAM and position (x) from a linear filter.

Figure 3-17 Box plot for the prediction correlation coefficient of shoulder torque from MAM and position (x) from a linear filter. The red + is an outlier (outside 1.5 times the interquartile range).

The correlation coefficient is used as the standard metric for measuring the "goodness" of prediction in all BMI studies conducted thus far. It is also used as the standard metric in this study. The correlation coefficient is given by,

\[
r = \frac{\sum XY - \frac{\sum X \cdot \sum Y}{N}}{\sqrt{\sum X^2 - \frac{(\sum X)^2}{N}} \sqrt{\sum Y^2 - \frac{(\sum Y)^2}{N}}}.\tag{15}
\]
It is essentially the covariance of the two data sets divided by the standard deviations of each. As a result, the correlation coefficient gives the quality of a least square fitting of the two data sets using a linear correlation. A correlation coefficient of 1 signifies a perfect linear fit between the two data sets, and a coefficient of 0 means there is no linear relationship at all (unless one data set contains all zeros).

In this study an additional statistic, which may give a more intuitive feel for the quality of prediction, is shown as an alternative to the correlation coefficient. The signal to error ratio (SER), defined as power of desired signal divided by the power of the error signal, is plotted in Figure 3-18(a) and (b).

\[
SER = \frac{\sum X^2}{\sum (X - Y)^2},
\]

(16)

![Figure 3-18 (a)SER (signal to error ratio) plotted for muscle torques at the shoulder. (b) SER plotted for muscle torques at the elbow.](image)

The SER from MAM is again comparable or better for all sessions for both elbow and shoulder torque. Again, the ANOVA tests substantiate this fact (Table 7).
Table 7 ANOVA table for the prediction SER of (a) shoulder torque and (b) elbow torque from MAM and linear filter.
Returning to the correlation coefficient, the coefficient is only effective when the correlation between the predictions and the observed values is linear. In a case where the two data sets are perfectly correlated in a nonlinear manner, the correlation coefficient is not a good represent of their correlation. A cross plot of the predicted values versus the actual observed values is a good check to see if the relation is in fact linear. Also, the correlation coefficients are insensitive to scale and offsets. Thus, a plot of the actual trajectories should be used in conjunction with the correlation coefficients and cross plots as a measure of the quality of prediction.

Figure 3-20 show typical torque predictions of the Muscle Activation Method for one minute of test data, together with actual joint torques estimated from the arm trajectory. The corresponding cross plots are provided below each trajectory.
Figure 3-20 1 minute of sample predicted trajectory from MAM and corresponding observed trajectory for muscle torque at the shoulder ($T_s$) and elbow ($T_e$) for five sessions ((a)-(e)). Each plot is accompanied by a cross plot of the predicted and observed values.

The plots show predictions from five sessions with descending levels of accuracy. Overall, the Muscle Activation Method predicts the torque profile well, even during high frequency movements. The MAM especially predicts segments with no motion fairly well, whereas traditional linear filters have proven to be notoriously noisy when predicting still segments (Chapter 2). The cross plots show that a linear correlation is a fairly good model for the prediction correlation. There is some evidence from the cross plots that there is saturation at the higher amplitude torques where the predictions are not able to reach the peaks, especially for lower correlation coefficient sessions. This may be due to the saturation of the sigmoidal function (Equation 7) used to produce the muscle activations from the ANN in the Muscle Activation Method. In fact, some cross plots roughly match the shape of the sigmoidal function. This suggests that the MAM may perform better with a linear half-wave rectification function in the neural net, instead of the sigmoidal function. However, this was not verified for this study.

The cross plots do not show evidence of any consistent phase difference between the predicted and observed trajectories. This suggests that an appropriate number of time lag bins have been chosen for the inputs so that the ANN can properly model the transmission and muscle recruitment delays for the eventual predicted output. Also, the
scale appears to be correct in most sessions, and none exhibit a significant offset in the predictions that is observable to the eye.

3.3.2 Prediction during force field tasks

Turning now to the second hypothesis presented in the introduction, these results (Figure 3-21 to Figure 3-25) determine whether one of the main advantages of the Muscle Activation Method is in fact achieved. Specifically, they address the question of whether the Muscle Activation Method successfully predicts how the monkey limb interacts with a novel environment.

Viscous force fields (Task 2) were applied during two sessions to see the effects on the prediction of the linear filter and the Muscle Activation Method. For the Muscle Activation Method, the training set had to span all of the muscle activation space present in the test set. Therefore the algorithms were trained on viscous force conditions to predict a null force test set, rather than the other way around. The training set contained a wide range of torques, whereas the test set contained only smaller torques, with amplitudes about 5 times smaller than that of the test set (Figure 3-21).
Figure 3-21 Observed muscle torques at the shoulder for a training (viscous force field) and test set (null force field) for the two sessions.

With the linear filter, the predicted torque amplitudes from the neural activity were significantly larger than the actual observed torques. Figure 3-22 illustrates this mismatch. The correlation coefficients for the predicted shoulder torques for the two sessions were essentially zero ($r = 0.037$ and 0.002). Thus, given a training set with viscous force, the linear filter could not accurately predict torque under a different force field condition. It appears that the optimized linear filter was not able to extract the correct correspondence between the neural activities and the muscle activities that determine the dynamics of the movements.
Figure 3-22 Predicted trajectory from a linear filter for shoulder torque given a viscous force field training set and a null field test set.

Figure 3-23 confirms that this is due to the linear filter’s inability to perform under a novel force field rather than its inability to perform in a force field in general. The figure shows that the linear filter is able to predict at a reasonable level if the test set is in a viscous force field, even for the similarly smaller torque amplitudes (due to smaller movements, rather than a null force field).
Figure 3-23 Predicted trajectory from a linear filter for shoulder torque given a viscous force field training set and a viscous force field test set.

The Muscle Activation Method, on the other hand, performs better in the new environment, as hoped. Figure 3-24 shows the prediction of shoulder torque using MAM for the same sessions plotted on the same scale. The correlation coefficients for predicting the null force field torques with the Muscle Activation Method were 0.572 and 0.549 for the two sessions. This is significantly higher than prediction correlation coefficients achieved using the linear filter and the difference in accuracy is apparent when comparing the plots in Figure 3-22 and Figure 3-24. The Muscle Activation Method successfully predicted the reduced torque amplitudes of the null force field condition from the neuronal activity from a training set containing much higher amplitude torques resulting from a viscous force field. This suggests that the Muscle Activation Method was able to extract information at the level of muscle activations rather than trying to make direct correlations to the torques. The relationship between the neural activity and the muscle activation should remain constant, regardless of force field conditions.
Figure 3.24 Predicted trajectory from the Muscle Activation Method (MAM) algorithm for shoulder torques given a viscous force field training set and a null field test set.

Figure 3.25 shows the same plots as Figure 3.24 on an enlarged scale with their corresponding cross plots.
It is interesting that the linear filter made such poor predictions of joint torque during the null-force trials, whereas the combined action of the nonlinear ANN, the muscle force model, and the muscle geometry performed well. This indicates that there are nonlinearities in the relationship between cortical neural activity and joint torque that are too large to neglect. These may include the nonlinear transformation from joint angles to muscle lengths, the transformation from forces to torques, and the nonlinearities in the generation of muscle force. Some of these maybe approximated using a linear model. For instance the relationship between the angles and muscle lengths or angles and moment arms could be approximated linearly. However, there are functions and transformations that a linear filter will not be able to approximate effectively. For example, the muscle forces are a function of both muscle length and activation. However, these two variables do not affect muscle force in an additive manner. A more realistic linear approximation would be a bilinear model, which can not be established in a simple linear filter. Another example is the transformation of muscle forces into torques. The muscle forces have to be multiplied by the moment arm which in turn varies nonlinearly with joint angles. A linear filter is not able to multiply two independent variables and thus this transformation can not be adequately approximated. Although, multiple linear filters or multiple layers in an ANN may be able to represent such multiplications, the Muscle Activation Method
contains these nonlinear relationships explicitly and thus would provide a more accurate approximation.

It is also interesting to note the large amplitudes of the incorrect predictions of the linear filter during the null-force trials. They suggest that the linear filter correlated neural activity with higher order kinematic features such as joint angles and velocities that are loosely related to the joint torques. These kinematic features did not change under the null-force condition, and thus the linear filter might have erroneously predicted higher amplitude outputs than those in the test set. If this hypothesis is correct, one can assume that the noisy prediction during the rested motion in session 2 of Figure 3-22 is due to the non-kinematic neurons that are not able to fit to anything meaningful.

In fact, using joint angle, velocity and acceleration as input, a linear filter was able to predict muscle torques to a very high level of accuracy \( r = 0.9650 \), given the same viscous force field conditions for the training and test set (Figure 3-26). A linear filter was also able to predict joint angle \( r = 0.6649 \), velocity \( r = 0.6559 \) and acceleration \( r = 0.5428 \) fairly well given neural input.

![Figure 3-26 Prediction of shoulder torques during a viscous force field, using joint angle, velocity and acceleration as input instead of neural data.](image)

**3.3.3 Muscle Activation**

The above results imply that the Muscle Activation Method converged to a solution that was consistent at the level of muscle activations and therefore was able to make the correct predictions. Figure 3-27(a)-(c) provides further evidence for this. Gold disc electrodes (Grass Instrument Co., West Warwick, RI) were used to measure surface...
EMGs at the biceps (BLH) and triceps (TRIA) of the monkey during a session with the viscous force field task. The EMG signals were amplified, high-pass filtered, rectified, smoothed by convolution with an 80 millisecond triangular kernel, and normalized. The processed EMG signals are plotted on Figure 3-27 along with the intermediate variable from the hidden layer of the trained Muscle Activation Method.
Figure 3-27 (a)-(c) 3 × 100 second samples of normalized surface EMG measurements plotted with the intermediate variable from the Muscle Activation Method (MAM) algorithm. Each figure contains two plots, one from the biceps and the other from the triceps.
The correlation coefficient for the biceps EMG and MAM activation were 0.4821, 0.5003 and 0.3759 for the three plots shown in Figure 3-27. The corresponding triceps correlation coefficients were 0.5067, 0.3934 and 0.4211.

An improvement on the correlation could be seen by taking the enveloped profile of the signals through low-pass filtering (at 0.5 Hz with a 5-th order butterworth filter) both the MAM activation and measured EMG signals (Figure 3-28). This spectrum would correspond to signals from slow intentional movements. The corresponding correlation coefficients for the low pass filtered signals were 0.6241, 0.7390 and 0.6441 for the biceps, and 0.7209, 0.7399 and 0.6805 for the triceps.
(c) Biceps

(d) Triceps

EMG Activity vs. MAM Activity
Figure 3-28(a)-(c) 3 × 100 second samples of normalized surface EMG measurements plotted with the intermediate variable from the Muscle Activation Method (MAM) algorithm. Each figure contains two plots, one from the biceps and the other from the triceps. (Low pass filtered at 0.5Hz). (d) Cross plots of MAM activation and EMG measurements for biceps and triceps. The data points are from the complete 300 second sample (5 minutes) plotted in (a)-(c).

It was previously postulated that, given a full training set, the hidden layer would converge to muscle activation values through the back-propagation of error. As hypothesized, the activation times and amplitudes of the measured EMGs and the muscle activation variables resulting from the algorithm, shown in Figure 3-27, are roughly in agreement. When comparing the two, one must consider the fact that surface EMGs are not direct measures of activation inputs to the entire muscle group. Rather, surface EMG is a noisy signal that correlates primarily with the activity of motor units near the electrode (De Luca 1997). Given the approximate nature and intrinsic noise of the EMG signals, the likeness of the two signals shown in the plots is remarkable. Since the ANN was not trained on these muscle activities, the correspondence of these two signals provides an independent validation of the correctness of the method. The plots provide strong evidence that the hidden layer has in fact converged to muscle activations, as desired. Also, comparing the triceps and biceps activations, it appears that the co-contractions of the muscles are extracted by the Muscle Activation Method. This can also be seen clearly in the cross plots of biceps and triceps activity shown in Figure 3-29. Co-contraction of the algorithm activity seems to be less pronounced than in the EMG recordings (especially in session (a)), yet it is clearly present. The co-contraction of the muscles determines the stiffness of the arm, and thus this provides further evidence that the correct impedances are being predicted.
Figure 3-29 Cross plots of triceps and biceps activity from EMG recordings and the MAM algorithm.

3.3.4 Kinematic Neurons

Data in the present study are also useful for addressing the question of “kinematic” versus “dynamic” neurons. We can report that some of the neurons maintained high
correlation coefficients for predicting position with a linear filter under different force conditions. These neurons are most likely the “kinematic neurons” (Li, Padoa-Schioppa et al. 2001) that encode higher order parameters of position and velocity. Figure 3-30 shows prediction of position in a null force field using a linear filter trained on a viscous force field task. The presence of kinematic neurons results in a correlation coefficient for prediction of position \( r = 0.6935 \) that is much higher than that of prediction of torque under new force conditions \( r = 0.0037 \), shown previously. However, the trajectory does appear noisier than the usual position trajectories predicted with a linear filter. This is possibly due to the inclusion of neurons that do not encode kinematics information directly and thus were predicting incorrect information under the different force condition (similar to the noise in session 2 of Figure 3-22). If only the kinematic neurons are located and exclusively used, a better prediction of position may be possible using the linear filter.

![Figure 3-30 Prediction of position (x) using a linear filter with a viscous force field training set and null force field test set.](image)

To discriminate the kinematic neurons from the “dynamic neurons” (Li, Padoa-Schioppa et al. 2001), the directional tuning properties of the cells were investigated during curl force field tasks (Task 3). Dynamic neurons are those correlated to more abstract motor features such as muscle activation, forces and dynamics of the movement. Kinematic neurons would not vary in tuning properties under different force fields. The directional tuning of dynamic neurons, on the other hand, would vary significantly in a curl force field, because the force is perpendicular to the direction of motion.
A. Top 50 neurons for predicting hand position through linear filter
B. Bottom 50 neurons for predicting hand position through linear filter

**Figure 3-31** Directional tuning properties of neurons during curl force field tasks.

Figure 3-31 shows the directional tuning properties of the neurons during curl force field tasks. The neurons were ranked by their correlation coefficient for predicting position in a curl force field using a linear filter trained on null force field tasks. The top 50 neurons were considered to be kinematic neurons and the bottom 50 neurons were considered to be dynamic neurons and others. The directional tuning is shown by representing the activity of each neuron as a color plot across the direction of movement of the monkey’s cursor (-180° to 180°). The plots in the first column show the neurons directional tuning during a null force field and the second column shows the same for a curl force field. The purpose of the plot was to show that the kinematic neurons do not change in directional tuning properties for different force fields, where as dynamic neurons do so depending on the direction of the force field. The plots show this to some degree. Unfortunately, the quality of the cells was fast deteriorating by the time the monkey was able to perform curl force field tasks and thus the plots are not as convincing as hoped. The directional tuning plots during viscous force fields would not show the
dramatic change in tuning properties for dynamic neurons because the direction of the force field is in the same direction as the movement.

It is clear from the fact that kinematic features can be predicted regardless of force field conditions, that kinematic neurons do exist in the recordings. By locating and recording from kinematic neurons alone, a BMI could be operated under pure motion control for all force field conditions. However, these neurons would not contain impedance information and thus versatile interactions with the environment would not be possible.

3.4 Discussion

Measurements of the prediction accuracy across multiple sessions show that the Muscle Activation Method performs just as well as the linear model for free movement. In fact, in most sessions, the Muscle Activation Method predicted the muscle torques at the joints significantly better than the linear model. This may be because joint torques are lower level variables that may not often be represented directly in the cortical areas.

Higher order kinematic variables such as position and velocity have been predicted with great success using linear filters, because a large number of neurons correlate directly to these variables. However, to the author’s knowledge, there has been no evidence to show that cortical neurons directly encode muscle torques at the joint and this may account for the slightly lower correlation coefficients. On the other hand, there is abundant evidence that cortical neurons are directly linked to muscle activities (Humphrey and Reed 1983; Scott and Kalaska 1995; Scott 1997; Scott, Gribble et al. 2001). A linear filter may also extract muscle activation successfully if fitted to measurements of the activation. However, even when muscle activation is related to neuronal activity, the linear filter can not model the nonlinearities of muscle force generation and nonlinearities in the transformations to moments that are needed to predict muscle torques accurately, as discussed previously.

By selecting muscle torques as the output variable for comparison, one of the main strengths of the Muscle Activation Method has been highlighted. A linear filter, or any other non-model based extraction algorithm, is most effective only when the predicted
motor parameters are directly and strongly represented in the cortical areas. However, using the Muscle Activation Method, any motor parameter that is downstream from the muscle activities in the musculoskeletal system can be predicted. These include torques, forces, impedances, positions, and velocities, among others. Furthermore, the parameters are extracted so that they are all mutually consistent, whereas non-model based algorithms can only extract the parameters independently which may result in an inconsistent set of predicted outputs.

The plots of muscle activation versus EMG measurements in Figure 3-27 and Figure 3-28 provide good evidence for the Muscle Activation Methods ability to predict a consistent set of motor parameters. Although the Muscle Activation Method algorithm was trained to minimize torque errors at the shoulder and elbow, it was also able to converge to a solution that could predict muscle activations to a reasonable degree of accuracy. This also reflects favorably on the accuracy of the musculoskeletal model. It implies that the approximations made in transformations from muscle activation to joint torques were fairly accurate.

The new Muscle Activation Method has also demonstrated the ability to predict interactions with force fields. More notably, it predicted the interactions of the limb with a new force field condition that was not present in the training set. Again this can be attributed to extraction at the muscle activation level which contains co-contraction information and also the models of the force-displacement and force-velocity properties of the muscles.

The fact that the Muscle Activation Method can use a relatively small training set to generalize for a wide range of interactions will be important in future BMI applications, when prosthetic limbs interact with objects. A BMI can not be trained on all possible force interactions with the environment. Given a limited training set, the BMI must generalize to successfully handle new interactions, with the goal of making the artificial appendage always respond like a natural limb. No current extraction algorithm, be it a linear filter, an artificial neural network, population vectors, or a recurrent neural network, is able to generalize for all possible interactions given a limited training set. However, the only requirement on the training set for the Muscle Activation Method is that it covers the entire muscle activation ranges. This is a relatively lax requirement and
is easily achievable. Training and prediction at the muscle activation level results in a comprehensive algorithm that is not restricted by the environmental forces in the training set and is able to predict a wide range of interactions during the test set.

The ability of the Muscle Activation Method to generalize for all possible interactions is equivalent to its ability to predict impedances. The effective stiffness, viscosity and inertia information is always present in the model and can be displayed to the environment via the prosthesis to achieve the intended interaction. The algorithm is further strengthened by the fact that direct measurements of impedances, which are difficult to acquire, are not required. Although not included in this study, a behavioral task with the divergent force fields (Burdet, Osu et al. 2001) where the monkey would have had to control its arm stiffness to complete task, would have shown the advantages of predicting impedances more conspicuously.

It has to be pointed out that the musculoskeletal model in the Muscle Activation Method is not entirely accurate. Vast simplifications have been made. Spinal reflexes were entirely omitted. Also, monkey data was not available for several parameters and human data was used instead. In fact, a large portion of the error in predictions may be attributed to inaccuracies in the model. Continued effort can be made to establish a more accurate model. However, it is also hoped that in an online system, the plasticity in the cortex will adjust to inaccuracies in the model. In other words, it is hoped that the user will be able to learn to control an arm with different dynamics. It is apparent in our everyday activities that the human brain is adept at learning new dynamics, for example of the arm holding a hammer, so it is not overly optimistic to expect that users will successfully learn the dynamics of a musculoskeletal arm model that imperfectly matches the dynamics of their original limb.

An effective alternate strategy for the Muscle Activation Method may be to use "muscle synergies" (Tresch, Saltiel et al. 1999) for the intermediate output of the ANN, instead of individual muscle activations. Individual muscle groups are not usually activated independently. Furthermore, studies have shown that a small set of combinations of muscle activation patterns may be the building blocks for constructing motor behaviors in the CNS (Tresch, Saltiel et al. 1999; Saltiel, Wyler-Duda et al. 2001; d'Avella, Saltiel et al. 2003; d'Avella and Bizzi 2005). Thus, using a set of muscle
synergies as intermediate variables could be a more “natural” way of controlling the musculoskeletal model for cortical neurons. This strategy could be implemented by identifying a set of muscle synergies that are the primitives for the planar arm motion, and adding an intermediate layer of these synergies to the Muscle Activation Method.

An online demonstration of BMI with the Muscle Activation Method was not conducted. However, offline analysis has shown enough evidence that a working online system would be achievable. Previous studies have shown that, given visual feedback of an online BMI system, the primate is actually able to perform better (Carmena, Lebedev et al. 2003) than the offline predictions suggest (Wessberg, Stambaugh et al. 2000).

3.5 Conclusions

The main goal of this chapter has been to introduce a new approach to extracting continuous motion parameters for BMI. The Muscle Activation Method was built from empirical measurements and muscle models found in the literature. It has been tested with behavioral tasks on a monkey with cortical implants. It has passed two tests that establish the significance and broad utility of the approach. The tests showed that the Muscle Activation Method predicts motor parameters as well as a linear filter, and that it successfully predicts interactions with a novel environment, something that prior extraction algorithms can not do. Furthermore, the Muscle Activation Method can extract a complete set of motor variables that are consistently inter-correlated within the dynamics of the system. With the consistent set of variables, impedance control can be implemented in a BMI system in order to provide more versatile and useful control of neuroprostheses.

The main contribution made in this chapter can be summarized as follows. We have successfully demonstrated a working first pass at a scheme for connecting cortical events with world dynamics. Traditional BMI systems have ignored a significant portion of the information present in the cortical neurons, by only extracting kinematic parameters. This study has shown that it is possible to extract muscle activations from cortical signals, as suggested by (Todorov 2003) and (Scott 1997), in order to extract impedances. In fact, extracting at the level of muscle activations has proven to be the key for interacting with novel environments. Also, constraining the extraction algorithm with the biomechanics of
the arm provided a means for extracting a full set of consistent variables. There is room for improvement and future studies may concentrate on a more accurate muscle force generation model or development of an online system. However, two facts have been established that have laid the foundations for future work; the necessity of extraction at the muscle activation level and the need for musculoskeletal model based constraints on that extraction algorithm.
4 Human Trials

4.1 Introduction

Until recently, Brain-machine interface systems have been demonstrated exclusively using recordings from nonhuman primates (Wessberg, Stambaugh et al. 2000; Pesaran, Pezaris et al. 2002; Serruya, Hatsopoulos et al. 2002; Taylor, Tillery et al. 2002; Carmena, Lebedev et al. 2003; Taylor, Tillery et al. 2003; Musallam, Corneil et al. 2004). The only human demonstration of BMI has been from single unit neuronal recordings from human cerebral cortex which were used to drive graphical cursors (Kennedy and Bakay 1998; Kennedy, Bakay et al. 2000). Although, the single unit recording demonstrations show the feasibility of human BMI systems, nonhuman primate studies have suggested that multi-neuronal recordings (at least 50-100 neurons) are needed for predicting movement adequate for neuroprosthetic application.

For current BMI technology to eventually be used in neuroprostheses for patients with neurological injury, systems with multi-electrode implants need to be tested in human trials. Such tests are approached cautiously due to health and safety concerns with surgically implanting the multiple electrodes for intracranial recordings. However, clinical trials with humans would signify a major step towards achieving the ultimate goal of human brain operated machines. Thus, recently, many groups have concentrated their efforts on applying their BMI technology to human trials. Our current plans for a comprehensive human BMI system with a robotic manipulator have been stalled due to pending approval from the FDA.

In this chapter, our ongoing efforts at applying BMI technology to human trials will be described.
4.2 Preliminary trials

Preliminary trials with multi-electrode arrays were performed in an intraoperative setting on patients with tremor disorders or Parkinson’s disease during deep brain stimulator (Medtronics, Minneapolis, MN) placements in the subthalamic nucleus and thalamic motor areas (ventralis oralis posterior (VOP) and ventralis intermediate nucleus VIN). These subcortical areas, in addition to cortical motor regions, have been shown to be involved in the planning and execution of motor tasks (Lenz, Kwan et al. 1990; Abosch, Hutchinson et al. 2002). The study was approved and monitored by the Duke University Institutional Review Board.

The trials were conducted on 11 awake patients and lasted up to 10 minutes during deep brain stimulator surgery. The task involved the patient modulating grip force to match a target force presented on a graphical display (Figure 4-1), and holding that level of force for a short period (0.5-1.0s). To monitor the patients grip force, a pressure sensing squeeze ball was developed, by inserting a miniature pressure sensor (Entran Devices, Inc., Fairfield, NJ) inside a sealed rubber bulb (Figure 4-2). Typically, the patients could perform the task for 5 minutes continuously before tiring, allowing up to 50 task repetitions for a single session. The intraoperative neuronal recordings along with the grip pressure recordings were stored for offline analysis.

![Figure 4-1 Screen shot of graphical display of the task. The patient controlled the black cursor to track a randomly time-varying target by applying variable gripping force to a pressure measurement device in the contralateral hand.](image)

122
Analysis showed that, overall, 61% of subthalamic nucleus neurons and 81% of VOP/VIM neurons modulated with gripping force. Ensemble of these neurons were able to predict the gripping force in 30-second test periods with moderate accuracy ($r^2 = 0.26 \pm 0.04$), using various linear and nonlinear models (Figure 4-3).
Figure 4-3 Prediction of gripping force from recorded neuronal activity by linear and nonlinear decoding algorithms. A Wiener filter, B ANN, C LMS, D Kalman filter. E, Increasing predictive quality with longer training time. F, increasing accuracy of prediction with larger neuronal ensembles. The plots are from results shown in Patil, Carmena et al. 2004.

Although, the task was a simple one degree of freedom task and online feedback of predictions were never provided to the patients, the findings serve as a feasibility study for future human BMI.

4.3 Online human BMI system

An upgraded version of the Carmena et al. (Carmena, Lebedev et al. 2003) monkey BMI system introduced in Chapter 1 has been developed for upcoming clinical trials with quadriplegic patients. The hardware devices, including a four-processor server computer, a Plexon multichannel acquisition processor (MAP) and displays are conveniently rack mounted in a portable frame to permit both intraoperative recording and long-term
recording in a patient room (Figure 4-4). The system is currently capable of recording 128 spike channels and 128 local field potentials (LFP). Extracellular spikes are sorted in hardware by the MAP, and their timestamps and waveforms are transferred to the software, which stores and bins them. From there, the software coordinates all activities essential for a BMI, including fitting and running multiple adaptive models, interfacing to a robotic manipulators (Phantom 3.0, Sensable Technologies, Woburn, MA), displaying visual feedback to the patient and therapist, and providing information about the quality of model fit and prediction. The software implements a very general and expandable set of fast multiparametric plotting routines that allow realtime multiscale visualization of estimated firing probability for any neuron against any recorded or generated continuous variable; this allows the user to instantly assess the information content of neuronal firing. Plotting is also used to gauge model (including Wiener, nLMS, recursive least squares filters, PCA, and a musculoskeletal model of the arm) fitting and predicting.

![Figure 4-4 Human BMI system.](image)

The system incorporates technology developed in the previous two chapters. A DC motor cable-drive gripper, mounted with infrared proximity sensors was developed to provide Continuous Shared Control to the robot for reaching and grasping tasks (Figure 4-5). This second generation gripper provides position control and is more responsive
than the pneumatic gripper used in Chapter 2. Also, the Muscle Activation Method algorithm has been incorporated into the software to gauge its prediction levels and possibly provide an online demonstration with visual feedback to the user. A pneumatic force feedback device has been developed for feedback of contact forces through a pressure cuff. A bellow was attached to a 1-dof motion platform (SmartAxis, Adept Technology Inc., Livermore, CA) to modulate the air pressure in a rubber pressure cuff. A pressure sensor (Entran Devices, Inc., Fairfield, NJ) was inserted into the cuff to feedback pressure information to implement closed loop control of the pressure. The cuff provides one degree of freedom force information and suitable for use in the clinical setting because it does not produce the electrical noise levels of traditional motor driven devices.

Figure 4-5 A back-drivable gripper with three infrared proximity sensors mounted for Continuous Shared Control.
The primary difference between the clinical trials and the previous nonhuman primate trials is that there is no measurement of kinematic data for the models to be trained on. The patients will have various degrees of paralysis and would not be able to produce normal movements with their natural limbs. Therefore, an idealized trajectory will be produced for the training set of these trials. Studies have shown that normal human reaching movements resemble minimum jerk trajectories towards the target (Flash 1987). The linear and nonlinear models would fit the neuronal recordings to the minimum jerk trajectories and then used the models to predict brain-controlled movements.

One advantage of the human trials is that better communication with the subject is possible. The subject is likely to require a much shorter training period to perform the task than the monkeys. Also, feedback to and from the subject about task performance would be possible. It is hoped that this would result in a more successful BMI trial.

Currently the system is ready, volunteers have been recruited and we are waiting approval from the FDA for use of long term implants (1-4 weeks) in the patients.
4.4 EMG teleoperation

Although neural signals from human subjects could not yet be used for the online system, other biological signals could, when obtained by less invasive methods. Surface EMG signals can be readily measured and contain low level motor related information. An EMG controlled manipulator could potentially have wide applications. Controlling manipulators through a human-machine interface is essentially a teleoperation task and is not restricted to neuroprostheses. Teleoperation with robots is widely used for manipulation in hazardous environments, remote operations under water or in outer space, and operations requiring enhanced movement or force control such as laparoscopic surgery or manipulation of heavy objects. In many of these applications where interaction and manipulation of different objects are involved, impedance control can enhance performance dramatically. However, measuring the intended impedance on the master is not easy. Usually, force perturbations are required to know the user’s impedance. In teleoperation, this can be done through force feedback. However, even in a bi-lateral system with force feedback, delays in the system may hinder performance by failing to reflect the instantaneous impedance of the master. Traditional methods of commanding impedances in unilateral system (without force feedback) have been to measure grip forces at the master or to pre-program impedances from known information about the environment.

Measuring muscle activities through EMGs can provide a more intuitive way of commanding impedances. It has been shown in the previous chapter that the Muscle Activation Method algorithm can be used to extract impedance information from the cortical recordings, given kinematic parameters of the training set. The same can be done for the surface EMG recordings, which should in fact be more directly related to the muscle activation levels which are input in to the musculoskeletal model.
4.4.1 Previous work

There has been a large body of work involving the use of EMG signals to control robotic devices. The first applications for EMG controlled manipulators were in upper extremity prostheses. The Boston Arm (Jearad, Williams et al. 1974) and the Utah Arm (Jacobsen, Knutti et al. 1982) used EMG measurements from remnant muscles to drive the prosthesis through velocity control. More recently, (Abul-Haj and Hogan 1990) used remnant muscle activity to set impedance of the Cybernetic Elbow Prosthesis through a simple bi-linear model.

In most of the studies since then, robots and prosthetics have been driven under supervisory control (Sheridan 1992) through spectral analysis and pattern recognition of EMG signals (Hudgins, Parker et al. 1993; Zardoshti-Kermani, Wheeler et al. 1995; Farry, Walker et al. 1996; Nishikawa, Yu et al. 1999). Continuous position control through EMGs has previously been limited to virtual arm models (Koike and Kawato 1995; Wang and Buchanan 2002). Recently, Crawford et al. (Crawford, Miller et al. 2005) have demonstrated continuous position control of a robot from EMG signals through a linear support vector machine algorithm.

In the work most closely resembling our system, Ito et al. (Ito, Hori et al. 2002) have proposed a continuous position control strategy that includes impedance control of a simulated manipulator. The EMG signals are used to modulate the stiffness and viscosity at the joints through a simple bilinear model. The model assumes the sum of the flexor and extensor activities linearly modulate the impedance at the joint. An accurate model of muscle force generation or musculoskeletal geometry was not included.

This study marks the first time a detailed musculoskeletal model has been used to drive a physical manipulator under impedance control from EMG recordings.

4.4.2 System hardware and software

The prediction algorithm presented in Chapter 3 was used with two modifications. The musculoskeletal model parameters needed to be updated for the human subject instead of the monkey. PCSA (physiological cross-sectional area) and inertial parameters were taken from Gribble et al. (Gribble, Ostry et al. 1998). Optimal fascicle lengths were
taken from Murray et al. (Murray, Buchanan et al. 2000). Muscle geometry was scaled back to the original human data given by Wood et al. (Wood, Meek et al. 1989).

The other modification was that the input signals were sampled EMG signals from the arm muscles rather than binned activity from the neuronal recordings. The size of the input signals were much smaller (6 channels \(\times\) 9 time lags), so two more hidden layers could be used in the ANN without causing the training period to be unacceptably long (no more than 5,000 iteration required). It was hoped that the multilayered ANN would be able to model the muscle activation dynamics and the cross talk between the channels to output individual muscle activations to the musculoskeletal model.

Surface EMG measurements were taken using single differential electrodes (Delsys Inc., Boston, MA) (Figure 4-7(a)). The electrodes were placed on the same six muscle groups modeled in the musculoskeletal model presented in Chapter 3; single-joint flexor and extensor at the shoulder (pectoralis and deltoid) and at the elbow (biceps long head and triceps lateral head) and also two double-joint muscles spanning both the shoulder and elbow (biceps short head and triceps long head). The exact location for the electrode placements, shown in Figure 4-7(b), were determined by anatomical drawings of the musculoskeletal system (Martini and Timmons 1997) and prior studies using measurements from the same muscle groups (Koike and Kawato 1995; Yoshida, Dome et al. 2002).
The EMG signals were amplified ($\times 1000$) and sampled at 1 kHz. The sampled signal was then band-pass filtered at 30-300 Hz, which brackets the significant portion of the myoelectric-signal spectrum (Abul-Haj and Hogan 1990). The resulting signal was then rectified and low-pass filtered at 5 Hz with a 5'th order butterworth filter. It was suggested that the low-pass filtered EMG signals reflect the firing rate of $\alpha$ motor neurons (Koike and Kawato 1995). The low-pass of 5 Hz also corresponds to the frequency range of voluntary movement (Neilson 1972).

A light weight two link manipulandum with ball bearing joints was built to restrict and support the arm for planar motion (Figure 4-8). Two Hall effect rotary sensors were used to measure angular position from the manipulandum for kinematic data.
During the training session, EMG and kinematic data were recorded during 10 minutes of random motion. The subject periodically stiffened and relaxed the arm throughout the session, so that the training set would significantly span the entire muscle activation space for planar motion. The Muscle Activation Method algorithm was trained on this data as described in Chapter 3.

The resulting model was then used to demonstrate impedance control with a robot (Phantom 3.0, Sensable Technologies, Woburn, MA). The traditional algorithm for impedance control has the form,

\[
I(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau + J^T F_{ext}
\]

\[
F_{ext} = K_d(X_d - X) + B_d(V_d - V) - M_d \frac{dV}{dt}
\]

\[
\tau = J^T W^{-1} M_d^{-1} \left[ K_d(X_d - X) + B_d(X_d - X) \right] + J^T W^{-1} (J^T C \dot{\theta} - J \ddot{\theta}) + J^T (1 - W^{-1} M_d^{-1}) F_{ext}
\]

where \( I(\theta) \) and \( C(\theta, \dot{\theta}) \) are the inertial and coriolis torques of the robot and \( g(\theta) \) is the gravitational torque. A simplification was made for this demonstration. Static conditions were assumed, and thus the viscosity and inertia terms \((B_d \text{ and } M_d)\) were ignored. This was a reasonable assumption for tasks involving small, slow movements. The elegant
solution shown in Equation 1 does not require inverse kinematics. However, it does require external force measurements and the inverse of $M_d$. The inverse kinematics of the 3-dof robot has been previously established. Thus, it was not necessary to use Equation 1. A simpler implementation,

$$\tau = K_d (\theta - \theta_d) + I(\dot{\theta}) + C(\theta, \dot{\theta}) + g(\theta),$$

(2)

could be used instead. (The inertial components of the robot in Equation 2 could also be ignored for this demonstration yet was included because it was already available from previous robot control algorithms).

To find the desired $K_d$ in robot joint space the stiffness of each muscle needs to be found and an appropriate transformation is required. The stiffness values for the muscles were found by differentiating the analytical force expressions for muscle force (Equation 1 and Equation 2 in Chapter 3) with respect to muscle length,

$$k = \frac{dF}{dl} = \frac{8\alpha F_{max}}{l_o} \left(1 - \frac{l}{l_o}\right)^2 + \frac{K_{sh} F_{max}}{0.5l_o \exp(K_{sh}) - 1} \exp\left(\frac{K_{sh} (l - l_o)}{0.5l_o}\right).$$

(3)

Therefore, the $K_{\text{muscle}}$ matrix in muscle space would be,

$$K_{\text{muscle}} = \begin{bmatrix} k_{PMJ} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{BLH} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{BSH} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{DEL} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{TRIO} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{TRA} \end{bmatrix}.$$  

(4)

The transformation for stiffness in muscle space to robot joint space is given by,

$$K_d = J^T K_{\text{muscle}} J + \frac{\partial}{\partial \theta} (J^T F_{\text{ext}})$$

(5)
where,

\[
J = J_{\text{arm}} J_{\text{cart}}^{-1} J_{\text{robot}}
\]  

(6)

and

\[
J_{\text{arm}} = \begin{bmatrix}
\frac{dl_{PMJ}}{d\theta_j} & 0 \\
\frac{dl_{BLH}}{d\theta_j} & \frac{dl_{BLH}}{d\theta_e} \\
\frac{dl_{DML}}{d\theta_j} & 0 \\
\frac{dl_{TRIO}}{d\theta_j} & \frac{dl_{TRIO}}{d\theta_e} \\
0 & \frac{dl_{TRIA}}{d\theta_e}
\end{bmatrix}
\]  

(7)

\[
J_{\text{cart}} = \begin{bmatrix}
-L_1 \sin \theta_x - L_2 \sin(\theta_x + \theta_e) & -L_2 \sin(\theta_x + \theta_e) \\
L_1 \cos \theta_x + L_2 \cos(\theta_x + \theta_e) & L_2 \cos(\theta_x + \theta_e)
\end{bmatrix}
\]  

(8)

\[
J_{\text{robot}} = \begin{bmatrix}
(l_1 \cos \theta_2 + l_2 \sin \theta_1) \cos \theta_1 & -l_1 \sin \theta_1 \sin \theta_2 & l_2 \sin \theta_1 \cos \theta_3 \\
0 & l_1 \cos \theta_2 & l_2 \sin \theta_3
\end{bmatrix}
\]  

(9)

\(J_{\text{arm}}\) is the Jacobian for arm joint space to muscle length space, \(J_{\text{cart}}\) is the Jacobian for arm joint space to Cartesian space and \(J_{\text{robot}}\) is the Jacobian for robot joint space to Cartesian space. \(\theta_x\) and \(\theta_e\) are the arm joint angles, \(\theta_1, \theta_2, \text{and} \theta_3\) are the robot joint angles, \(L_1\) and \(L_2\) are the upper arm and lower arm lengths and \(l_1\) and \(l_2\) are the robot link lengths.

For stiffness control of the robot, the EMG measurements were inserted into the trained ANN from the Muscle Activation Method algorithm. The ANN outputs muscle
activation outputs were then input into Equation 3 and $K_d$ was found from the transformation in Equation 5.

Although impedance information could be extremely useful, accurate position control is of primary importance in most teleoperation tasks. Motion trajectories can be commanded from the musculoskeletal model of the Muscle Activation Method. However, there would be a significant loss in positional accuracy due to prediction error ($\text{SER} \approx 2.0-6.0$, (Figure 3-18)) and drift due to error minimization in torques rather than position. The more sensible approach would be to use a position tracking device, which are readily available and adequately accurate, and use the Muscle Activation Method algorithm only to command stiffnesses. To avoid mismatches in the two, position information from tracking device was also used for determining the joint angles and muscle lengths for Equation 3, Equation 7 and Equation 8. In this case, the motion dynamics of the musculoskeletal model would only be used for the training of the algorithm. When predicting the stiffness of the algorithm, kinematic data from the position tracking device would be used. For this study, a Phantom Desktop device (Sensible Technologies, Woburn, MA) (Figure 4-9(a)) was attached to the end of the manipulandum as the motion tracker. The encoders were less noisy than the Hall effect rotary sensors and the device could potentially provide force feedback if desired. Also, a Nano-17 six-axis force/torque transducer (Figure 4-9(b)) was attached to the end effector of the robot to measure $F_{ext}$ for Equation 5.
To summarize, the Muscle Activation Method was used to generate an ANN model that is capable of predicting muscle activation from surface EMG recordings. This model was used in teleoperation system to generate muscle stiffness information from EMG recordings online. The muscle stiffnesses were transformed to robot joint stiffness and displayed to the environment. Figure 4-10 shows this overall system.
Training Phase

![EMG recordings](image1)

Motion

Back propagation of error

\[ E = \sum \frac{1}{2} (d_i - y_i)^2 \]

Musculoskeletal arm model

Teleoperation Phase

![EMG recordings](image2)

Motion tracking device

Motion

Motion

SEMG recordings

Trained ANN

Muscle force model

Force sensor

Stiffness

Robot

4.4.3 Teleoperation demonstration

The main advantage of this approach to impedance control is that the command of impedances is natural and that the directional information of impedances is not lost. The entire stiffness ellipse for the endpoint can be extracted, which means the user has the ability to modulate stiffness in different directions for the robot, as they are able to do with their natural arm. A task which demonstrates the ability to naturally modulate stiffness and stiffness direction is shown.

A sharpened pencil was mounted onto the force sensor which was in turn mounted on to the end of the robot. Four target rectangles were marked on a sheet of paper, which was framed and mounted in front of the robot. The task was to make four straight lines on the target boxes without piercing the sheet of paper. The task required the subject to increase stiffness in the direction parallel to the paper surface for tight position control.
but lower the stiffness in the direction normal to the surface so that the pencil does not pierce through (Figure 4-11).

Figure 4-11 The teleoperation task. A sharpened pencil was attached to the end of the robot. The task involved drawing straight lines in the target boxes of the framed paper without piercing the surface.

Ten minutes of surface EMG recordings and kinematic recordings were used to train the Muscle Activation Method. The subsequent one minute of data was also recorded to verify that the trained model could predict trajectories with a reasonable degree of accuracy. Figure 4-12 shows the predicted trajectories and their cross plots. The prediction correlation coefficients were 0.4659 and 0.4036 for shoulder torque and elbow torque, respectively. The moderate correlation coefficients suggest that the models have fit to a reasonable degree and could be used for the teleoperation.
The subject attempted the drawing task 5 times each with two modes. The first mode was without impedance control, for which the stiffness was preset at high values optimized for effective position control. The second mode was with surface EMG commanded impedance control. The two modes were alternated for each trial. The subject was not able to complete the drawing without piercing the paper in any of the 5 trials without impedance control. In the impedance control mode, the subject initially had to learn to modulate stiffness in elbow and shoulder for drawing. This was unusual because drawing usually involves modulation wrist stiffness. However, by the fourth trial, the subject was able to successfully perform the trial. The drawings from all 10 trials are shown in Figure 4-13. The average pierced target lengths for the trials are plotted in Figure 4-14.
Position control

Impedance control
Figure 4-13 Resulting drawings from teleoperation trials. First row is without impedance control and the second row is with impedance control. The sequence of each row shows trial progression. No piercing occurred in the final two trials of impedance control.

![Graph showing平均穿过的纸长度 for the teleoperation trials with position and impedance control. The error bars show the standard deviation for the 5 trials.]

4.5 Conclusion

This chapter described the preparations made for the upcoming clinical trials. The possibility of demonstrating a human operated, brain-machine interface driven manipulator is very exciting. Much of the technology is duplicating that used for non-human primate trials, yet the demonstration would have much significance in that it is a major and crucial step towards realizing a working neuroprosthesis for the disabled.

We have also shown a successful online demonstration of the advantages of impedance control through the Muscle Activation Method, with EMG signals. The demonstration showed the possibilities of using other bio-signals as the source for human-machine interactions. Signals from cortical and subcortical neurons, cerebellum, spinal chord and muscles could be used separately or in conjunction for an effective neuroprosthesis in the future. The level of information content would vary depending on the site of the recordings. The Muscle Activation Method has been shown to work successfully with both higher level (cortical neurons) and lower level recordings (EMGs).
5 Conclusion and Future Studies

Current technologies in brain-machine interfaces offer an exciting look into the future. Potential applications could be life altering for those suffering from loss of limbs and paralysis. Also, many “super-human” tasks could be imagined by combining the accuracy and strength of machines with the versatility and computational power of the human neural circuitry.

However, demonstrations of BMI to date have been limited to showcasing proof of concept and feasibility, rather than the performance of useful tasks. This thesis provides strategies for bridging the gap between simple demos of monkeys moving cursors and robots in free space and that of humans performing useful real world tasks.

A Continuous Shared Control framework has been introduced where sensor-based reflexes could be used to augment the brain commanded trajectories. This strategy was successfully demonstrated to improve the reaching and grasping of real objects with a robot manipulator. The strategy could potentially serve as the framework for generating motions such as walking, where higher level cortical cells are more likely concerned with abstract features such as direction and speed of movement, and the detailed trajectories are left for lower level circuitry.

The prediction of motor parameters was improved through the Muscle Activation Method. The method provided the ability to predict a complete and coherent set of parameters that contained all information required for determining the dynamic interaction with the environment. The new algorithm was tested on neuronal data and was found to predict motion parameters with as much accuracy as a traditional extraction algorithm. Furthermore, it successfully predicted limb interactions with novel force fields, which is a new and significant capability lacking in other algorithms. The results showed clear benefits of extracting at the level of muscle activations and the need to
constrain the algorithms with a biomechanical model that can produce a consistent set of variables.

In terms of application to neuroprostheses, the Muscle Activation Method provides the framework for impedance control. Impedance control of the manipulator would allow versatile interaction with objects. Versatility would be one of the keys to the usefulness of the prosthesis. A successful online demonstration of the algorithm was performed using EMG measurements from a human subject.

Preparations have been made for clinical trials of BMI using previous technology and those described in this thesis. The trials would signify a major milestone in the pursuit of a working neuroprosthesis, because it would show the viability of a human operated system for the first time.

Work on all the systems described in this thesis is ongoing and are subjects for further studies. The Continuous Shared Control system is currently being used for nonhuman primate trials to see how the cortex responds to different levels of robot autonomy. The study would attempt to find the right mixture of brain control and robot autonomy to optimize both task performance and versatility for an effective BMI. Continuous Shared Control will also be used in the clinical trials to facilitate the initial training and also to maximize performance in reaching and grasping tasks.

The Muscle Activation Method algorithm would be tested online in both human and nonhuman primate trials. Efforts would be made on improving the accuracy of the musculoskeletal model prior to the trials. Sensitivity studies could be conducted on the model to fine tune the parameters and also to determine where the largest source of error might be. Potential sources would be in the muscle force generation approximations and in the exclusion of the reflex loop. Muscle synergies would also be investigated as a possible alternative to muscle activations as the input to the musculoskeletal model.
Bibliography


Appendix A

The Wiener filter: Motor parameters such as help position, velocities, joint torques and others were modeled as weight linear combination of neuronal activity using a multidimensional linear regression. The basic form is given by,

\[ y(t) = b + \sum_{u=-m}^{n} a(u)x(t-u) + \varepsilon(t). \]

\( x(t-u) \) is an input vector of neuronal firing rates at time \( t \) and timelag \( u \), \( y(t) \) is a vector of the output (motor parameter) at time \( t \), \( a(u) \) is a vector of weights at timelag \( u \), \( b \) is a vector of y-intercepts, and \( \varepsilon(t) \) are the residual errors. The lags in the summation can in general be negative (in the past) or positive (in the future) with respect to present time \( t \). In this study, only lags into the past were considered.

The above equation can be represented in the matrix form as,

\[ Y = XA, \]

where each row in each matrix is a unit of time and each column is data vector. Note that matrix \( X \) would contain lagged date and thus has a column for each lag multiplied by the number of neurons. The y-intercept is handled by attaching a column of ones at the end of matrix \( X \). \( A \) is then solved by,

\[ A = inv(X^TX)X^TY. \]