UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A CORRECTIVE MOVEMENT-BASED APPROACH TO THE ONLINE ADAPTATION OF NEURAL DECODERS FOR PROSTHESIS CONTROL

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

EMMANUEL GOOSSAERT
Norman, Oklahoma
2010

A CORRECTIVE MOVEMENT-BASED APPROACH TO THE ONLINE ADAPTATION OF NEURAL DECODERS FOR PROSTHESIS CONTROL

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Dr. Andrew H. Fagg, Chair

Dr. Amy McGovern

Dr. Robert L. Rennaker II

Acknowledgments

I express my gratitude to Dr. Andrew H. Fagg for welcoming me in his laboratory, and for his support and patience during this work. I am also grateful to him for the knowledge he transmitted to me.

I want to thank Di Wang, Thomas Palmer, Joshua Southerland, Matthew Bodenhamer and Sandro Mussa-Ivaldi for their contributions.

I also think of Thomas Rousson, former student of the ISIMA who has been in the exchange program with the University of Oklahoma in 2006, and who has always been around to answer my questions regarding this program.

I am grateful to the University of Oklahoma for having considered me as a member of their community, and the ISIMA, that trusted me and permitted me to do this exchange program.

Finally I want to thank the NINDS for their grant, that has partly supported this work (NS048845).

Contents

Acknowledgments **iv**

Contents **v**

List Of Figures **vii**

Abstract **ix**

1 Introduction **1**

2 Background **3**
2.1 Brain-Machine Interfaces 3
2.2 Non-invasive approaches 3
2.3 Invasive approaches 5
2.4 Data Decoding 7
2.5 Development of Reaching in Humans 8
2.6 Adaptive Approaches to BMI 10

3 An Adaptive Decoder for Limb Control **16**
3.1 Control of a prosthetic arm 16
3.2 A Cursor Control Task 18
3.3 Automatically adapting the map 20
3.4 Task Configuration 26

4 Results **27**
4.1 Experimental protocol 27
4.2 Experimental Subjects 29
4.3 Data analysis 29
4.4 Experimental Results 31

5 Conclusion **36**

Bibliography **38**

Appendix A
Angle constraint function F_A 42

Appendix B
Individual performance curves for all six subjects 44

Appendix C
Experimental Maps 51

List Of Figures

2.1 This schematic represents a brain-machine interface process. During this process, brain activity is first recorded, and an estimate of the subject's intent is recovered by the decoder. This information is then used to control an effector, which can be a prosthetic arm 4

2.2 EEG helmet used to gather data from cerebral activity. Each circle is an electrode that can sense brain activity in a certain radius. This activity is transmitted through the wires connected on the electrodes to a recording device. *Picture courtesy of Brent Kirkwood* 6

2.3 Silicon microelectrodes that can be implanted directly into the brain *(picture courtesy of the SCI Institute at the University of Utah)* . . . 7

3.1 Relates the human to use of the prosthetic arm 16

3.2 Visual of the cursor control task, performed on a surface of 22 cm of side. The controllable cursor (blue dot) has a .3 cm diameter and the target to reach (red dot) a 1 cm diameter. The visual angles subtended by the surface, cursor and target are respectively 5.34 deg, .073 deg and .243 deg. 19

3.3 Relates the human to the cursor control task 20

3.4 Example of a correction. A sequence of two successive submovements, $\overrightarrow{AB}$ and $\overrightarrow{BC}$, is translated from hand motion, denoted as $\boldsymbol{H_t}$. $\overrightarrow{AC}$ is considered as the intended path for the first submovement $\overrightarrow{AB}$, which gives for each cursor position denoted as $\boldsymbol{x}$ a putative intended position along this path, denoted as $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$. 21

3.5 Variable update rate function 23

3.6 Corrective approach in action, for a sequence of two submovements of the hand in the cursor control task. 25

4.1 Subject performing the cursor control task 28

4.2 Reach of two successive targets. n indexes the division between submovements, t_n is the time the first target was reached, t_{n+1} is the time the next target was reached, x_n and x_{n+1} are the position of the targets, and d is the length of the cursor path between the reach of the two targets. 30

4.3 Naïve subjects #1 through #4 - Mean RTT, RPL and standard error (25-reach bins). Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 34
4.4 Naïve subjects #1 through #6 - Mean RTT, RPL and standard error (25-reach bins). Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 35

B.1 RTT and standard error of block #2 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 45
B.2 RTT and standard error of block #3 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 46
B.3 Cursor RPL and standard error of block #2 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 47
B.4 Cursor RPL and standard error of block #3 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 48
B.5 Hand RPL and standard error of block #2 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 49
B.6 Hand RPL and standard error of block #3 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively. 50

C.1 Natural map 51
C.2 Maps #1 through #6 52

Abstract

A subset of neurons in the primary motor cortex of primates has been shown to encode information about the ongoing movement of the limb. One can construct a model that translates the neural activity into a prediction of arm movement. If the model is accurate, predictions can be used to drive the motion of a robotic prosthesis that takes the place of the original limb. Classic approaches require explicit calibration phases to associate neural firing patterns to arm movements. However, due to changes in neuron behavior or in the signal extraction process (such as loss of neurons), the true behavior of the system will drift with respect to the model. While this model could be recalibrated on a daily basis by having the patient perform a specific task, such a process can be too obtrusive into the patient's daily activities. In this thesis, we propose an approach that uses the behavior of the patient to recalibrate the arm continuously.

While reaching to an object, a subject produces a sequence of submovements. Some of these submovements are corrections to the ones that precede them. These corrective movements can be considered as estimates of vectorial errors in the motor commands produced by the model, and used to drive the descent process of a quasi-gradient search. We propose a learning approach that continuously tunes the model to the subject using her own behavior. We demonstrate the utility of our approach using a non-invasive cursor control task performed by a set of human subjects. In this task, a human subject reaches targets with a cursor on a computer screen, using a sensor attached to his hand. Previously unknown models are presented to the subjects, simulating sudden drifts in the available neural signals. Our learning algorithm shows a significant advantage over the non-adaptive model by allowing the subjects to quickly learn to perform well with new models. This result has important implications for the possibility of continuously calibrating decoders and for achieving highly functional prostheses.

Chapter 1

Introduction

Neurons in the primary motor cortex encode information about limb movement. With invasive and non-invasive brain-machine interfaces, it is possible to sense and record the activity of individual or small groups of neurons. Using the recorded activity from invasive methods, it has been shown that both Cartesian hand position and velocity can be predicted using linear models (e.g., Paninski et al., 2004). Furthermore, it has been shown that joint torque can be predicted with comparable accuracy (Cabel et al., 2001; Fagg et al., 2009). Schwartz et al. (2006) and Serruya et al. (2003) have shown that if the model is robust enough to predict arm movement, then it can be used to control a robotic prosthesis.

In order for the prosthesis to act as a "drop-in" replacement for the arm, the model must accurately capture the relationship between cell activity and intended arm motion. In creating and sustaining such models, one faces several challenges. The first challenge is that of collecting the data necessary to construct the model. A typical approach is to simultaneously record neural data and either actively produced or passively observed arm motion. A model can then be determined by using a supervised learning approach that associates observed neural activity with arm motion (Novak et al., 2000; Taylor et al., 2002; Wahnoun et al., 2006). This process requires an explicit calibration period during which the subject has to perform specific arm movements while cell activity is simultaneously recorded. In other words, the intent of the subject is controlled through a cueing system.

Another challenge is that brain-machine interfaces are not stable over time. The true behavior of the prosthesis control system will drift with respect to the model, due to changes in neuron behavior or in signal extraction. Dickey et al. (2009) have shown that in the information gathered using chronically implanted electrodes, only

57% of the observed cells were firing consistently in the same direction over 7 days, whereas the others fire in unpredicted directions. Due to this instability, new models will need to be calibrated on a regular basis. While learning how to control the arm, a patient needs to recalibrate her prosthesis to new models at least once per week due to cell instability, which can be a significant intrusion into the patient's life routine.

Danziger et al. (2008) suggest a model that can be used on-line as the subject makes use of a (simulated) prosthesis. This model maps a set of control signals to a goal location for the prosthesis. The error in movement is assessed by comparing the known target location to position of the prosthesis at a given time after movement initiation. While such an approach can solve the problem of system drift, it has the drawback of using explicit knowledge of the intended goal of the movement. Systems in the field would likely not have information regarding the patient's true intent, making this approach unusable for continuous recalibration. A great improvement would be to assess movement error without explicit knowledge of the movement goal.

Berthier (1997) showed that when humans reach to an object, hand speed as a function of time exhibits a sequence of local maxima. This behavior suggests that a single reach is in fact a sequence of submovements, some of which may be corrections to the previous movement. Fagg et al. (1998) have suggested that it is possible to use these submovements as vectorial errors in a gradient descent search for improved reach control strategies. They have shown that inaccurate corrective reaches can lead to the development of reaching strategies.

In this thesis, I propose that BMI models can be learned on-line using movement errors estimated based on the sequence of corrections that are made by the patient. The developed approach starts by partitioning a movement into a set of submovements based on speed minima in the task space. Then the subset of submovements that corresponds to corrections is identified. The corrections are finally used as estimates of vectorial errors in task space in order to drive a quasi gradient descent learning approach. I demonstrate the viability of this approach using an abstract cursor control task in which the human subject uses arm position to specify the location of the cursor.

Chapter 2

Background

2.1 Brain-Machine Interfaces

Brain-machine interfaces (BMI), sometimes called brain-computer interfaces (BCI), are situated between neuroscience, psychology and computer science. As their name states, their purpose is to bind brains with machines. This implies that some amount of information is retrieved directly from the brain, and used to control a mechanical or electronic system. For instance, while it is possible to use information from the brain to control an artificial limb, one can also use information extracted from the brain to move a limb paralyzed from spinal cord injury (Hochberg et al., 2006).

Figure 2.1 illustrates the process of controlling a prosthetic arm using neural activity. The subject has the "intent," conscious or not, to perform a motor action. This action is represented by specific neural firing patterns in many areas of the brain, including the motor cortices (c.f. Schwartz et al., 2006). These patterns are extracted and decoded to estimate the intended action. Then, the decoded signal is used to control the effector, and feedback is given to the subject through visual information, proprioperception, or direct stimulation of neurons or nerves. Different technologies and approaches can be used for each of these stages.

2.2 Non-invasive approaches

Non-invasive techniques include electroencephalography (EEG), electrocorticography (ECoG), magnetoencephalography (MEG) and electromyography (EMG), and access

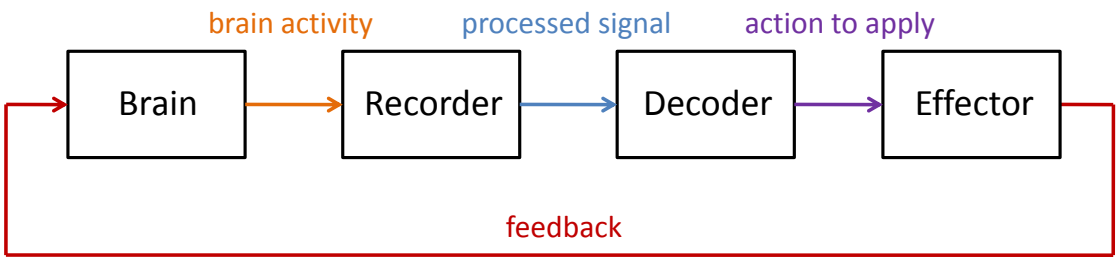


Figure 2.1: This schematic represents a brain-machine interface process. During this process, brain activity is first recorded, and an estimate of the subject's intent is recovered by the decoder. This information is then used to control an effector, which can be a prosthetic arm

information from outside the skull. EEG signals are recorded using an electrode network, placed on the subject's scalp like a helmet, as shown in Figure 2.2. These electrodes are sensitive to the average potential of a large number of nearby neurons. Because these electrodes don't have a direct access to the neurons, they can only detect general activity in large-scale areas of the brain. It is only possible to sense the activity of a population of neurons, and fine distinctions between neurons are averaged. Raw EEG data corresponds to changes of potential over time. EEG can also be expressed in terms of the frequencies present in the data. A Fourier transform is used to bring EEG from time domain into frequency domain. A feature vector describes the activity at an EEG electrode in terms of the signal power (e.g. the amplitude of the Fourier transform) at a range of frequency bins. The classic frequency band is 0.1-90 Hz for standard EGG system (Schwartz et al., 2006). Different frequencies correspond to different brain activity, such as normal use or active concentration, for instance.

As del R. Millán (2003) shows, this brain activity provides exploitable signals to control effectors in an approach that he calls *Adaptive Brain Interface*. In order to control an effector, a subject has to activate diverse areas of her cortex, such as fronto, central and parietal, by performing different types of mental tasks. These tasks include solving a basic mathematical operation, such as subtraction, performing word association, or imagining hand movement or 3D object rotation. The experiment consists of selecting one of three buttons on a computer screen, using mental states detected using an EEG system. This experiment allows the subjects to select letters of the alphabet in a 3-by-9 table by activating the buttons in different sequences. The EEG potentials are recorded with a sampling rate of 128 Hz on the scalp of the

subjects, at the eight fronto, central and parietal locations referred to as: F3, F4, C3, Cz, C4, P3, Pz and P4. These locations cover completely the top of the skull, and are separated from one another as shown in Figure 2.2.

During the experiment, the subjects are first asked to select three of the following mental tasks: relax, imagine left or right hand movement, imagine cube rotation, perform subtraction, and perform word association. Then, the ABI system uses a neural network to classify the extracted EEG potentials as on which mental task the subject is concentrating. For every subject, the three tasks showing the strongest response are selected and associated with the three buttons. In order to test the system, a subject chooses a button among the three available, and performs the mental task associated with that choice. If the button recognized based on the EEG signals matches the chosen one, the trial is considered to be successful, otherwise this is an error. After five days of training, the subjects were able to select a letter in 22 seconds on average, including recovery from error, with a recognition rate of 70%.

This ABI system, even though efficient compared to other EEG systems, still implies serious concerns regarding bandwidth and precision. Precision is a problem because the signal received at a single electrode on the scalp is influenced by the activity of a large number of neurons. Given the size of a single neuron's body (on the order of 100 micrometers in diameter) compared to the size of an electrode on the helmet (on the order of 1 centimeter in diameter, as shown in Figure 2.2), electrode potential is the result of the activity of a large number of cells. Therefore, it is impossible to differentiate fine details that are encoded at the single unit level (such as moving an arm to the right versus the left), which is why EEG produces only imprecise data. More precise solutions exist, and are presented in Section 2.3.

2.3 Invasive approaches

Invasive solutions imply that the equipment is placed as close as possible to the neurons, which means that electrodes are placed into the tissue or on the surface of the brain. Figure 2.3 shows the example of the Utah Electrode Array (UEA), created by the SCI Institute at the University of Utah. This array contains 100 micro-electrodes (in a 10x10 grid) that project out from a 4 mm^2 base, each electrode being separated from the others by 400 μm. The electrodes are made either of silicon,

Figure 2.2: EEG helmet used to gather data from cerebral activity. Each circle is an electrode that can sense brain activity in a certain radius. This activity is transmitted through the wires connected on the electrodes to a recording device. *Picture courtesy of Brent Kirkwood*

or of pure iridium insulated with parylene-C, as they very logically need to be made of biocompatible materials in order to be implanted into the brain. Connections to the electrodes are made either with individual insulated $25-\mu m$ wires or with a multilead polymide ribbon cable attached at the base of the array (Finn and LoPresti, 2003).

Each electrode senses the extracellular potential in a given area, providing localized high-fidelity information. Multiple neurons can be "heard" by a single electrode. Action potentials, or spikes, are short-duration depolarization events produced by individual neurons. Information is encoded by neurons in terms of the frequency of these events. Spikes are first detected from the signal extracted by the electrode. These spikes are then "sorted" by a form of waveform template matching into one of several categories that putatively correspond to individual neurons. Because it is impossible to put large arrays into the sculus, the arrays used only access regions at the surface of the brain. This is enough to give access to parts of the premotor cortex and primary motor cortex, which are specialized in the control of muscular

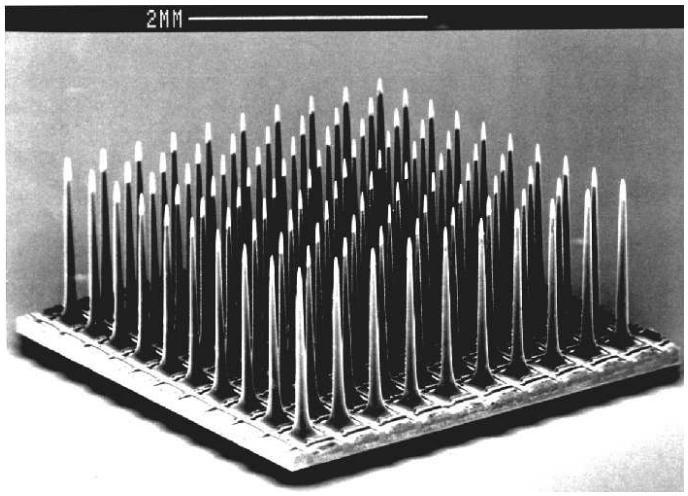


Figure 2.3: Silicon microelectrodes that can be implanted directly into the brain *(picture courtesy of the SCI Institute at the University of Utah)*

activity. Invasive methods allow access to cells that are involved in controlling the limbs that we are replacing with robotic prosthetic limbs. With non-invasive methods such as EEG, we have no choice but to use high-level brain activity to command the prosthetic limb. This approach requires modulation of activity in areas of the brain that have nothing to do with the intact limb.

Whether invasive or non-invasive approaches have to be chosen in the case of controlling a limb is source of disagreement in the community. It is hard to estimate with precision how these two classes of solutions will develop and which will be the best in the future. However, at this point of time we can argue that the best information is achieved from invasive approaches. Nevertheless, raw neural data cannot be used directly to pilot a prosthetic arm. Data has to be processed first, which is the topic of Section 2.4.

2.4 Data Decoding

Neural activity can predict limb movements using a model, whose job is to transform a description of the recent history of neural activity into a prediction of arm movement. More specifically, the linear function of the model is over the average spike rate of individual neurons exhibited during the entire movement. In order to address

arbitrary movements made over the workspace (including curved movements), a common technique is to describe the recent history of cell activity in terms of the average spike rate in a sequence of time bins. Hatsopoulos et al. (2004) advocate that this history should extend as far as one second prior to the prediction. The average spike rate (or equivalently, spike count) over a set of bins and a set of cells can be described using a feature vector. The predicted quantity is then some function defined over the resulting feature space, and is used to control the arm.

Georgopoulos et al. (1986) showed that the neural activity in the motor system, that includes the primary motor cortex and the premotor cortex, correlates arm movements. Using a linear model, Georgopoulos et al. (1986) showed that arm position and direction of arm movement can be predicted based on the activity of a population of neurons in the arm area of the primary motor cortex. In addition, both Cartesian hand position and velocity can be predicted using linear models (e.g. Paninski et al., 2004). Joint torque can be predicted with similar accuracy, still using linear models (Fagg et al., 2009). When the predictions can be made in real time, they can also be used to control an effector, like a prosthetic arm. On-line applications are possible, as neural data have served to control robotic prosthetic arms (Novak et al., 2000; Taylor et al., 2002; Wahnoun et al., 2006).

2.5 Development of Reaching in Humans

An open problem is how humans compute motor signals to produce arm trajectories. A feed-forward control solution is not possible due to the uncertainties in the actuation and sensing systems. Consequently, it is necessary to employ a degree of feedback control in the process. This implies integrating the results of recent motor commands in deciding the current motor output. However, feedback control is limited due to the significant delays in the biological feedback loops. As a result, the biological control system must employ a mixture of both predictive and feedback control (Berthier et al., 2005).

Von Hofsten (1979) and Berthier (1997) suggested that in reaching, some movements are corrections to previous movements. Two theories can explain the corrections that occur during hand reaching trials (Berthier and Robin, 1998). The first one speculates that the brain is pre-planning the whole movement towards the object,

aborting in case of error, and then re-planning. The second theory argues that a first submovement that brings the hand some point short of the target is planned ahead, and that additional submovements are computed to convey the hand closer to the target when the subject realizes that her current movement will not reach the object.

In an experiment with seven-month old toddlers, Berthier and Robin (1998) show that infants are capable of correcting the trajectory of a reach after the movement has been initiated. For this experiment, seven-month old infants were presented with a toy. The infant naturally tries to grasp the toy as soon as it is presented. As the infant reaches for the toy, the experimenter shifts the position of the toy, either to the left or to the right, forcing the infant to apply a correction to her movement in order to grasp the toy. Berthier and Robin (1998) examined the behavior that results from the change in reach target location. During the experiments, recorded hand path showed changes in direction towards the direction that the toy was shifted.

Based on a strict application of the first model, what the infant should do in case of movement error is stop her on-going movement, pause, re-plan a trajectory in accordance with the new position of the toy, and execute the new movement. Due to this pause and re-planning activity, the speed profiles of the infants during shift trials should show an interruption at some point. However, the speed profiles of the infants do not present any apparent interruption. This contradicts directly the re-planning theory. Berthier and Robin (1998) suggests that the second model is the one that is used by the infants. This would imply that infants use a sequence of *corrective movements* to reach for an object.

Following this idea, Fagg et al. (1998) demonstrated that it is possible to use corrective movements to update previous movements, but also that this corrective information can be used as feedback to the control system itself and alter its behavior for the future. In horizontal planes, adult reaching shows regularities, such as straight lines trajectories.

Fagg et al. (1998) used a simulated planar arm, with a model to command the elbow and shoulder flexion. A continuous signal controlled the magnitude of the movement for the relevant joints. The corrective movements provided a vectorial error signal that was then be used to adjust the joint movement magnitudes in the model. Fagg et al. (1998) proposed two approaches to use the corrective movements.

The first algorithm is a supervised learning method. In this approach, the arm is first placed in an initial position, and is given the position of the target to reach. A learning module moves the arm, reaching for the target. If at the end of the movement, the position of the arm is far from the target, a teaching module generates corrective movements that bring the arm closer to the target. These corrective movements are also used to update the model parameters.

2.6 Adaptive Approaches to BMI

Taylor et al. (2002) have proposed a 3D cursor control task with monkeys using invasive BMIs and a model called the "coadaptive" movement prediction algorithm (CMPA). The model is coadaptive mainly because both the model and the knowledge of the subjects are changing at the same time. This approach did not require physical limb movements or knowledge of cell tuning properties. The goal of the task was to control a spherical cursor to reach spherical targets. Experiments were performed with two monkeys, and lasted respectively four and six weeks. During these experiments, the monkey was not able to see the movements of its arm, and only had visual feedback from the cursor and target positions. Targets appeared successively at the eight corner locations of a cube in space, in a random order. At the beginning of each day, the monkey controlled the cursor using its hand position (hand-controlled task). At the end of each day, the monkey used a model based on recorded brain activity to control the cursor, and with his real arm restrained (brain-controlled task). By comparing trajectories computed off-line based on cortical activity recorded during hand-controlled experiments, and trajectories computed on-line during brain-controlled experiments, Taylor et al. (2002) demonstrated that accurate visual feedback of the cursor allows the subjects to perform reaches more accurately.

The model they used was calibrated using fixed tuning properties from the hand-controlled experiments of the same day. This means that the model was determined once every morning, and that it was not adapted to the cells' change in firing behavior that occurred during the day. As neuron tuning properties are known to change dramatically in short periods of time, they developed a coadaptive approach that refines the estimates of neuron tuning properties used by the model, by observing

neural activity as the monkey performs cursor movements. Cell-tuning properties from the hand-controlled and brain-controlled tasks were estimated and compared for each cell or group of cells. Using these estimates, Taylor et al. (2002) showed that cell tuning properties change in terms of preferred direction from hand-controlled to brain-controlled, due to the subject's learning of the task. Moreover, the cell directional tuning properties and accuracy of movement in the brain-controlled task showed improvement compared to the hand-controlled task. This means that the coadaptive method can be used to train immobilized patients to perform 3D-cursor reach, even though it would not work out of the box.

Taylor et al. (2002) demonstrated that visual feedback coupled with their coadaptive approach allowed the subjects to increase their performance in the case of the brain-controlled task.

In another 3D-cursor-control invasive task with monkeys, Wahnoun et al. (2006) use an interesting method to compute the initial calibration. Instead of using random or fixed parameters, a visual following task provides the information necessary to the calibration. The observed neurons were from M1 and PMd, in the cortex. The firing rates were determined using 90 ms sliding windows at 30 ms increments, zero meaned and then normalized by their maximum zero meaned value. The firing rate for a specific cell at given time is modeled as:

$$Fr(u,t) \quad = \quad A + \boldsymbol{PDc(u,t)D(t)^T}, \tag{2.1}$$

where A is the intercept, u an index to the individual units, t an index to submovement sampling (one point every 30 ms), $\boldsymbol{PDc(u,t)}$ the "preferred" direction vector for unit u at t, and $\boldsymbol{D(t)}$ the "ideal" current direction vector at t correcting the cursor to target location (the direction that would bring the cursor to the target). The preferred direction $\boldsymbol{PDc(u,t)}$ was then estimated by a linear regression between the firing rates and the ideal current direction $\boldsymbol{D(t)}$. Then, Wahnoun et al. (2006) used a model developed by Georgopoulos et al. (1986), in which the population vector $\boldsymbol{PV(t)}$ is computed every 30 ms. "Population vector" refers to a particular type of model in which each neuron is considered as having a preferred direction. Preferred directions of the observed neurons are weighted by their instantaneous firing rates and summed:

$$\boldsymbol{PV(t)} = \sum_{\forall u} \boldsymbol{PDc(u,t)Fr(un,t)^T}. \tag{2.2}$$

The initial model parameters were determined using a visual task. During this visual task, the monkey looks at recorded movements of a cursor for 12 to 15 seconds. At the end of this visual task, about 400 bins of data were collected (12 seconds with sliding bins at 90 ms every 30 ms). The position of the screen cursor was then determined using the population vector (c.f. Georgopoulos et al., 1986). The task was then switched to a direct control of the cursor, using the newly computed model. During the first five minutes of the monkey performing the cursor-control task, $\boldsymbol{PDc}$ values were estimated for every 30 second intervals. The model between hand and cursor was then refined through an on-line update process. At the end of the five minutes, a final preferred direction, $\boldsymbol{PDf}$, was computed for every observed cell.

Within the set of these observed neurons, not all neurons produced a valuable contribution for adequate control of the effector. Indeed, Wahnoun et al. (2006) demonstrated that about a third of the neurons were turning to random or undesired directions. As a consequence, these neurons did not contribute to the direction of the action being addressed by the population of neurons studied. Worse, these neurons were contradicting the general direction given by the other neurons, and therefore added noise and error to the process. In order to solve this problem, Wahnoun et al. (2006) developed a method called "individual removal error." This technique allowed them to remove from the neuron ensemble controlling the cursor movement the neurons that were not producing useful contributions. They experimented again with the cursor-control task, but this time using only the neurons producing useful contribution. Wahnoun et al. (2006) showed that the subjects performed the task better than with the full ensemble of neurons. For example, reach time decreased of 200 ms on average, which corresponds to an improvement of 10%.

2.6.1 Adaptive decoding in human experiments

Danziger et al. (2008) proposed an on-line approach for tuning models and demonstrated its viability in a human experiment. The subjects were asked to wear a CyberGlove™ on their right hand. This glove senses up to 19 different degrees of

movement freedom in the hand, including finger flexion, palm arch, and thumb and finger adduction/abduction. Using a linear model, the glove inputs were transformed into an angle vector:

$$\boldsymbol{\theta} = \boldsymbol{W^T H}, \tag{2.3}$$

where $\boldsymbol{W}$ is a 2 by 19 transformation matrix, $\boldsymbol{H}$ is the glove input vector, and $\boldsymbol{\theta}$ is the output angle vector. The state of the angle vector determines the joint angles of a simulated, planar, two degree-of-freedom arm. The endpoint of the arm determines the location of a cursor:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos(\theta_1) & cos(\theta_1 + \theta_2) \\ sin(\theta_1) & sin(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} \tag{2.4}$$

where A is the transformation matrix, $[x\ y]^T$ the location of the end of the effector in the planar space, and $[L_1\ L_2]^T$ is a constant parameter vector that includes the link lengths (Danziger et al., 2008).

The subject receives visual feedback on a screen as to the locations of the end effector and the reach target. The targets appear on the computer screen one after the other, at four different fixed locations and in a random order. The subjects used different hand configurations to bring the cursor to the target. The subjects were given unlimited time to plan each movement. They were also required to keep the cursor on top of the target for one second in order to successfully complete the reach and switch to the next target. After 800 ms, the color of the cursor was changed, but the subjects were allowed to complete their reach. This 800 ms time limit was chosen such that the subjects could not perform the reach within the time limit on most of the trials. That way, there was a difference between the position of the cursor and the position of the target. This difference was taken as the movement error.

Two methods of model adaptation were tested: a least mean squares gradient descent algorithm (LMS) and a Moore-Penrose pseudo-inverse algorithm (MPP). Subjects were divided into three groups: LMS, MPP, and control (no learning method used). Subjects were trained for three days, 11 epochs per day, each epoch consisting

of 24 reaches. Errors in movement were computed based on the differences between the cursor and target positions during the different epoch. The parameters of the model, stored in the transformation matrix $\boldsymbol{W}$, were updated *off-line* based on these errors at the end of each epoch.

The LMS solution tries to minimize the square of the performance error norm by updating the parameters of the model. The update is applied at the end of each epoch, based on the movements performed during the epoch, which are indexed by t. The parameter update is:

$$w_{i,j}(N+1) \;=\; w_{i,j}(N) - \alpha \sum_t e_i(t) h_j(t), \tag{2.5}$$

where N is the iteration index, α is the step size of the update, $h_j(t)$ is the state of the jth glove parameter at 800 ms into the reach, and $e_i(t)$ is the error in joint i for movement t.

The MPP approach uses the Moore-Penrose pseudo-inverse transformation to find the minimum norm solution for a set of recent reaches. The $\boldsymbol{A}$ matrix is updated such that:

$$\boldsymbol{A} \;=\; \boldsymbol{\Theta}\boldsymbol{H}^{+}, \tag{2.6}$$

where $\boldsymbol{H}$ is a matrix of h-vectors each taken 800 ms after movement onset, $\boldsymbol{H}^{+}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{H}$ and $\boldsymbol{\Theta}$ is the matrix of targets in joint angle space.

The results show that the MPP group did not show any improvement over eight epochs. The MPP group showed a 50% increase in error, whereas the LMS and control groups both showed a 50% reduction in error after eight epochs. In addition, the LMS and control groups both showed a 70% reduction in error at the end of the experiment (after 24 epochs), and both converged to the same level of final performance. However, the LMS group reached asymptotic performance faster than the control group. The LMS case asymptoted approximately 12 epochs before the control case (after 13 epochs). This is the primary benefit demonstrated by this group to the experiment.

Regarding the feasibility of the developed approach in the real use of a prosthetic arm, the main issue here is the knowledge of goal. Knowledge of the target position is used to compute the error that serves to update the model. However, this is information that would not necessarily be available under conditions in which the subject was choosing the target locations autonomously.

Chapter 3

An Adaptive Decoder for Limb Control

3.1 Control of a prosthetic arm

The process of controlling a prosthetic arm using the activity of a set of neurons is depicted by Figure 3.1. As we can see in this figure, the brain receives the positions of the prosthetic arm and of the object to reach to through visual (and possibly proprioperceptive) feedback. The activity of the neurons in MI encodes arm movement information. Using a model, this neural activity is processed to recover the "intended" arm movement and to produce the associated motor signals that are necessary to move the prosthetic arm.

Several successful approaches have been proposed for adaptive brain-machine interfaces that allow a model to be tuned to better support the use of the BMI by the subject. However, some of these approaches are incremental in nature in that the model is not immediately updated as new performance information is available.

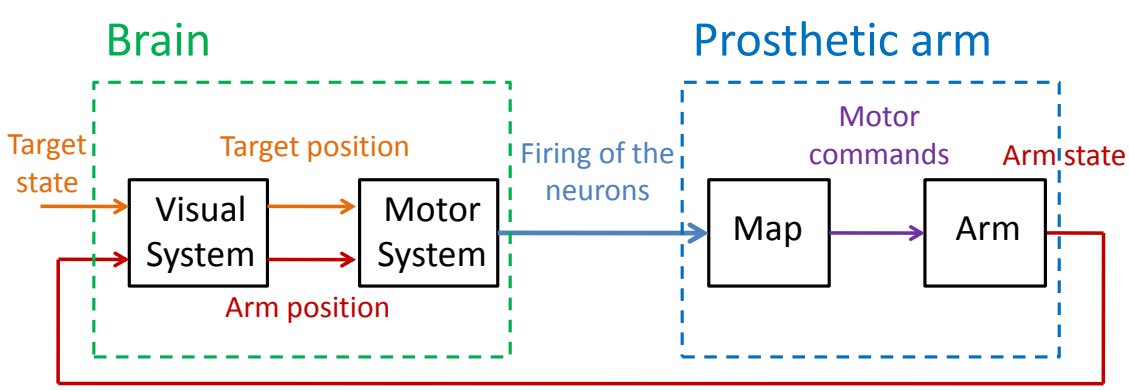


Figure 3.1: Relates the human to use of the prosthetic arm

Furthermore, all of the approaches assume explicit knowledge of the target of the reach being executed.

We want to design an algorithm that would help subjects to adapt faster to changing situations. This algorithm should work without any knowledge of the target position. We also want this approach to be on-line, so that any explicit calibration phase could be avoided.

In such an approach, accurate control of the arm depends in part on the quality of the model that translates the neuron firing patterns into motor commands for the prosthesis. With the classic approaches, the model is determined using a calibration period before the arm is first used (Wahnoun et al., 2006). During such a calibration process, the patient is asked to perform specific actions during a certain period of time. The issue is that with time, the true behavior of the system will drift with respect to the model. These drifts can be due to a variety of factors, including changes in the behavior of neurons (e.g., due to synaptic plasticity or cell death) and in the extraction of the neural signals (e.g., drift of the electrode array relative to the neurons). The patient would have to go through a new calibration process every time a significant drift occurs. Dickey et al. (2009) have studied the stability of single electrodes, and have demonstrated that 57% of the electrodes provided stable signals through a week. Therefore, a patient would have to recalibrate her device at least weekly.

Due to this commitment for frequent calibration, the classic approaches have the drawback of creeping into the patient's life routine, as it requires her to regularly stop her activity and to go through a new calibration process if she wants to take fully advantage of her prosthetic arm. It would be a major improvement if such a recalibration process could be avoided. In this work, our goal is to use the behavior produced by the patient while she uses the arm in order to automatically drive the recalibration. This would allow recalibration to be performed continuously and would avoid the need for the patient to explicitly go through a calibration process.

I propose an approach to the recalibration process that uses information extracted from the patient's reaching behavior to refine the model parameters as the patient is using the arm. In order to implement this approach, several key problems have to be solved. First, a complete arm movement must be decomposed into a sequence of submovements. Next, we need to determine which of these submovements are

corrective. The corrective information conveyed by these submovements have then to be transformed into error vectors for the gradient-descent search. Finally, these error vectors must be used to update the model, in a way that does not interfere with the ongoing performance of reaching tasks. We will illustrate these issues in the context of a cursor control task.

3.2 A Cursor Control Task

As our goal is to develop the details of an adaptive learning algorithm, we employ a non-invasive task in this work. This task is an abstraction for a prosthetic arm being used to reach objects successively. In this task the subject controls a cursor on the screen using arm movements. Thus, instead of having the brain control the prosthetic arm, the position of a subject's hand determines the position of the cursor on a computer screen. We express the transformation between hand and cursor positions as follows:

$$\boldsymbol{x} = F_{\boldsymbol{W}}(\boldsymbol{H}), \tag{3.1}$$

where $\boldsymbol{W}$ is the set of map parameters, $\boldsymbol{H}$ is the hand position, and $\boldsymbol{x}$ is the cursor position. Thus, we can transform $\boldsymbol{H}$ into $\boldsymbol{x}$ using the map function $F_{\boldsymbol{W}}$, a function of $\boldsymbol{W}$. For the instant, we employ a linear transformation from hand to task coordinates:

$$\boldsymbol{x_t} = \boldsymbol{W^T H_t}. \tag{3.2}$$

Figure 3.2 shows what the human subject sees on the computer screen during the experiment. The goal of the task is to reach the target with the cursor and then slow the cursor below a specified speed threshold. Once the target has been reached, it disappears, and another target appears, forcing the subject to move the cursor again. The targets appear at uniformly random positions on the screen. Figure 3.3 illustrates the components involved in the cursor control task. As we can see on this figure, cursor and target states are received by the visual system, which will transmit

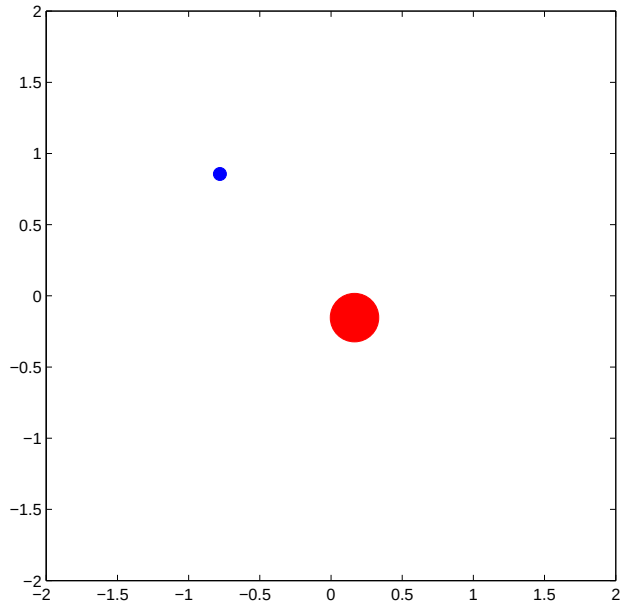


Figure 3.2: Visual of the cursor control task, performed on a surface of 22 *cm* of side. The controllable cursor (blue dot) has a .3 *cm* diameter and the target to reach (red dot) a 1 *cm* diameter. The visual angles subtended by the surface, cursor and target are respectively 5.34 deg, .073 deg and .243 deg.

the information to the arm control system in order to move the hand. This position of the hand is then transformed by the map into the position of the cursor.

A particular configuration for the map parameters is the one in which the cursor tracks the hand as the hand moves in the same plane as the screen. We call this configuration the "natural map." This map is very intuitive, as a subject facing the screen just has to move her hand in the direction she wants the cursor to move on the screen. For example, moving the hand left makes the cursor move left and moving the hand up moves the cursor up. A representation of this map is given in Appendix C. However, arbitrary relationships are possible. For example, moving the hand towards the left may make the cursor move downwards. This allows us to experimentally simulate the case in which a new population of neurons is suddenly the only means of controlling the prosthesis. We therefore examine how the combined adaptive systems of patient and learning algorithm cooperate to learn to effectively perform the task. In this case, the task is to learn the relationship between hand and cursor motion in order to complete a sequence of reaches.

When facing new parameters for the first time, it is hard for a subject to reach the targets on the screen, because the relationship between hand sensor and screen cursor

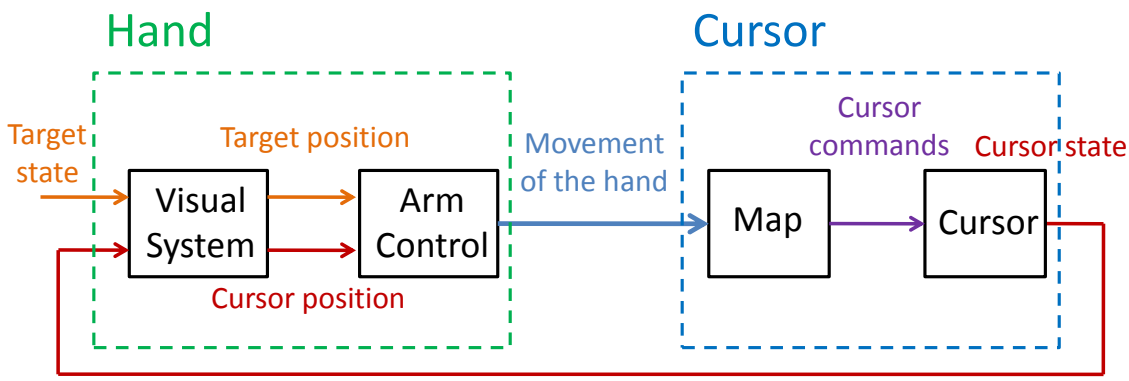


Figure 3.3: Relates the human to the cursor control task

is unknown to her. The subject typically produces many corrective submovements during the course of these reaches. We want to use these corrective submovements to drive the learning approach. As a first step, we need to detect the submovements in a complete movement of the hand.

In experiments with monkeys, Novak et al. (2000); Fichbach et al. (2005, 2007) have used jerk and snap zero crossings to decompose movements into a series of submovements. Here, we follow von Hofsten (1991) by using speed minima in the task (cursor) space. Each cursor speed minimum defines the boundary between two submovements. Then, we need to distinguish those submovements that are corrective from those that represent *changes in task*. For instance, suppose that a prosthetic-armed patient wants to reach to an object. After the object has been reached, the subsequent transport of this object should not be considered as a corrective movement, but as a change in task. Because corrective movements are often smaller than the movements that are being corrected (Berthier and Robin, 1998), we choose to define a corrective submovement as one that is shorter in both duration and distance than the submovement that precedes it. All other submovements are considered as changes in task.

3.3 Automatically adapting the map

Figure 3.4 illustrates this corrective process. In this figure, the trajectory of the hand sensor in the space, defined by the sequence of hand positions H_t, is translated into a cursor trajectory using the map. This cursor trajectory is divided into two

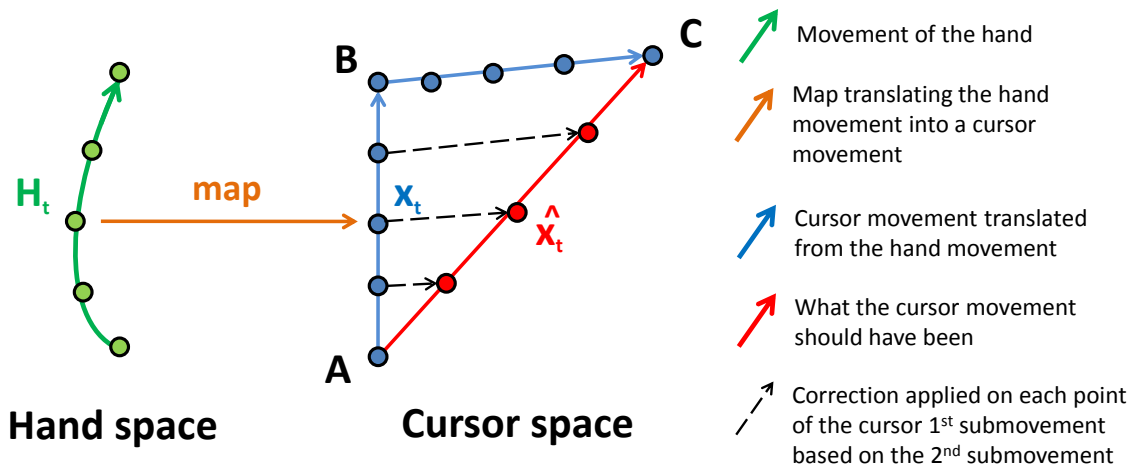


Figure 3.4: Example of a correction. A sequence of two successive submovements, $\overrightarrow{AB}$ and $\overrightarrow{BC}$, is translated from hand motion, denoted as $\boldsymbol{H_t}$. $\overrightarrow{AC}$ is considered as the intended path for the first submovement $\overrightarrow{AB}$, which gives for each cursor position denoted as $\boldsymbol{x}$ a putative intended position along this path, denoted as $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$.

submovements $\overrightarrow{AB}$ and $\overrightarrow{BC}$. As $\overrightarrow{BC}$ is smaller in both length and duration than $\overrightarrow{AB}$, it is a corrective movement to $\overrightarrow{AB}$. Therefore, we assume that the "intended" path of this sequence of two submovements is $\overrightarrow{AC}$. Our assumption is that every point, $\boldsymbol{x_t}$, that comprises the first submovement $\overrightarrow{AB}$, should be brought to the corresponding position on $\overrightarrow{AC}$. This defines a sequence of error vectors $(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t})$, shown in dashed black on Figure 3.4. These vectors are then used to update the set of parameters $\boldsymbol{W}$ of the linear model presented in Equation 3.2. We assume that this will make the task easier for the subject.

Changes to the parameters are applied with a gradient descent search, using the following cost function:

$$\boldsymbol{E} = \frac{1}{2}\sum_{t=A}^{B}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t})^T(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t}). \tag{3.3}$$

The gradient of this cost function in parameter space is:

$$\frac{\boldsymbol{\delta E}}{\boldsymbol{\delta W}} = \frac{1}{2}\sum_{t=A}^{B}\frac{\boldsymbol{\delta}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t})^T}{\boldsymbol{\delta W}}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t}) + (\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t})^T\frac{\boldsymbol{\delta}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x_t})}{\boldsymbol{\delta W}}, \tag{3.4}$$

and as $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$ is assumed to be constant, we have $\frac{\delta \hat{\boldsymbol{x}}_{\boldsymbol{t}}}{\delta \boldsymbol{W}} = 0$, which gives us a gradient of:

$$\begin{aligned}
\frac{\delta \boldsymbol{E}}{\delta \boldsymbol{W}} &= -\frac{1}{2}\sum_{t=A}^{B} \frac{\delta \boldsymbol{x}_{\boldsymbol{t}}^{T}}{\delta \boldsymbol{W}}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}}) + (\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}})^{T}\frac{\delta \boldsymbol{x}_{\boldsymbol{t}}}{\delta \boldsymbol{W}}, \\
&= -\frac{1}{2}\sum_{t=A}^{B} 2\frac{\delta \boldsymbol{x}_{\boldsymbol{t}}^{T}}{\delta \boldsymbol{W}}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}}), \\
&= -\sum_{t=A}^{B} \frac{\delta(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{H}_{\boldsymbol{t}})^{T}}{\delta \boldsymbol{W}}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}}), \\
&= -\sum_{t=A}^{B} \frac{\delta(\boldsymbol{H}_{\boldsymbol{t}}{}^{\boldsymbol{T}}\boldsymbol{W})}{\delta \boldsymbol{W}}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}}), \\
&= -\sum_{t=A}^{B} \boldsymbol{H}_{\boldsymbol{t}}{}^{\boldsymbol{T}}(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}}). \qquad (3.5)
\end{aligned}$$

The error term $(\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}})$ is estimated as a vector with same direction as $\overrightarrow{BC}$, scaled based on how close $\boldsymbol{x}_{\boldsymbol{t}}$ is to the end of the submovement. This scaling is done using a scaling factor denoted as γ_t:

$$\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}} \approx \gamma_t(\boldsymbol{x}_{\boldsymbol{C}} - \boldsymbol{x}_{\boldsymbol{B}}), \qquad (3.6)$$

$$\gamma_t = \frac{(\boldsymbol{x}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{A}})^{T}(\boldsymbol{x}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{A}})}{(\boldsymbol{x}_{\boldsymbol{B}} - \boldsymbol{x}_{\boldsymbol{A}})^{T}(\boldsymbol{x}_{\boldsymbol{B}} - \boldsymbol{x}_{\boldsymbol{A}})}. \qquad (3.7)$$

The gradient becomes:

$$\frac{\delta \boldsymbol{E}}{\delta \boldsymbol{W}} \approx -\sum_{t=A}^{B} \boldsymbol{H}_{\boldsymbol{t}}{}^{\boldsymbol{T}}\gamma_t(\boldsymbol{x}_{\boldsymbol{C}} - \boldsymbol{x}_{\boldsymbol{B}}). \qquad (3.8)$$

With the update rate denoted as α, we get the parameter update:

$$\begin{aligned}
\boldsymbol{W} &\leftarrow \boldsymbol{W} - \alpha\left(\frac{\delta \boldsymbol{E}}{\delta \boldsymbol{W}}\right)^{T}, \\
&= \boldsymbol{W} + \alpha(\boldsymbol{x}_{\boldsymbol{C}} - \boldsymbol{x}_{\boldsymbol{B}})^{T}\sum_{t=A}^{B}\gamma_t\boldsymbol{H}_{\boldsymbol{t}}. \qquad (3.9)
\end{aligned}$$

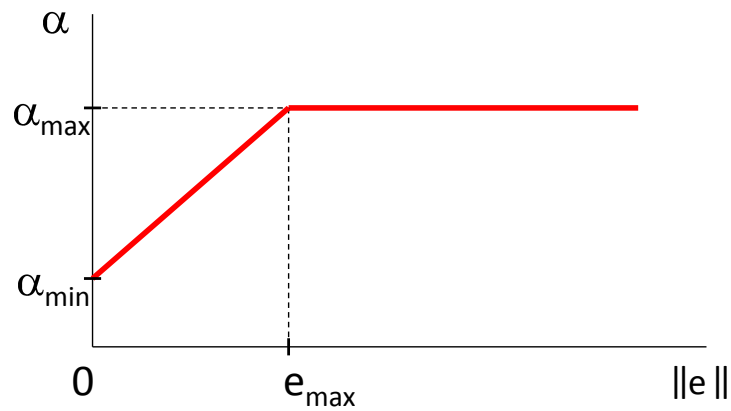


Figure 3.5: Variable update rate function

When coming closer to the target, the subjects attempt to complete the reach as fast as they can by jittering the cursor around the target. This creates a lot of small submovements, that do not represent real corrective movements, but just a strategy generally employed by subjects to perform the task. Some of these submovements will be considered as corrective by our algorithm when they should not. We want to suppress this noise, and for that we need to minimize the role of small corrective movements. In order to deemphasize these small corrective movements in the learning process, we modulate the update rate α as a function of the corrective movement magnitude:

$$\alpha = \begin{cases} \alpha_{max} & \text{if } ||\boldsymbol{e}|| > e_{max}, \\ \alpha_{min} + \frac{\alpha_{max} - \alpha_{min}}{e_{max}} ||\boldsymbol{e}|| & \text{otherwise,} \end{cases}$$

where $\boldsymbol{e} = (\boldsymbol{x_C} - \boldsymbol{x_B})$ is the error, and with α_{min} and α_{max} respectively the minimum and maximum update rate values, and e_{max} the error magnitude threshold. This function is shown in Figure 3.5.

In practice, direct updates of the map parameters based on a corrective movement can result in a "jumping" of the cursor every time the map is updated. This jumping effect decorrelates the hand and cursor movements, making it difficult for the subject

to know what the relationship is between her actions and the movement of the cursor. This lack of *knowledge of results* can dramatically impede the subject's ability to learn how to perform the task. In order to avoid such jumps, updates to the map are buffered and gradually introduced as the subject performs the task. The parameter update is stored in a buffer:

$$\boldsymbol{B} \leftarrow \boldsymbol{B} - \alpha \left(\frac{\boldsymbol{\delta E}}{\boldsymbol{\delta W}} \right)^T, \tag{3.10}$$

At every time step, a limited quantity of the buffer, the magnitude of which is denoted as ϵ, is shifted to the model:

$$\epsilon = \begin{cases} 1 & \text{if } ||\boldsymbol{B}|| \leq \mu \ , \\ \frac{\mu}{||\boldsymbol{B}||} & \text{otherwise,} \end{cases}$$

where:

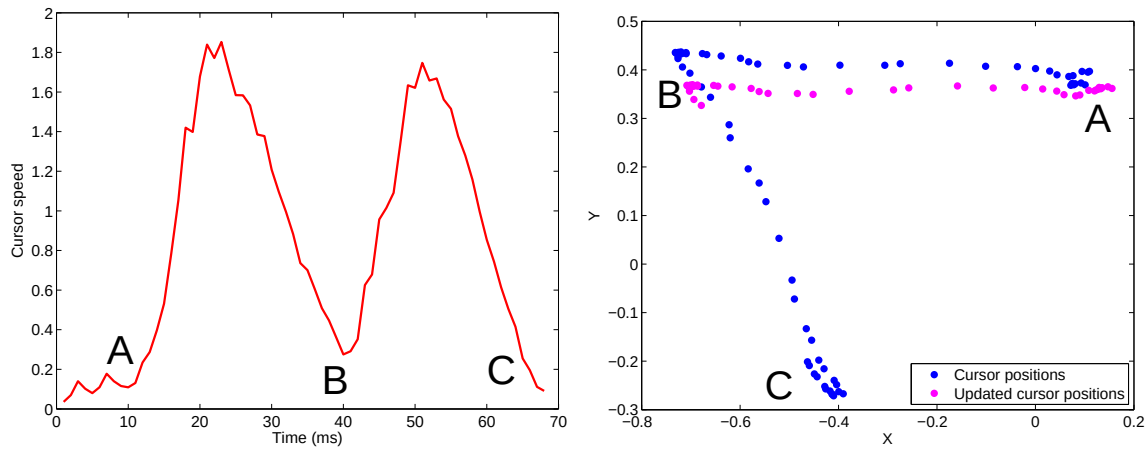
$$||\boldsymbol{B}|| = \max_{i,j} b_{i,j},$$

and μ is the shift threshold. Finally, the map and the buffer are updated at each time step:

$$\boldsymbol{W} \leftarrow F_a(\boldsymbol{W} + \epsilon \boldsymbol{B}), \tag{3.11}$$

$$\boldsymbol{B} \leftarrow (1 - \epsilon)\boldsymbol{B}, \tag{3.12}$$

where F_a is a function that controls scale and that ensures the column vectors of $\boldsymbol{W}$ are constrained to have at least an angle of $a_{min} = 70°$ between one another, and at most $180° - a_{min}$. If the angle is too small, it is opened to a_{min}, and if the angle is too big, it is reduced to $180° - a_{min}$. Changes are made such that the plane defined by the two vectors stays the same. This constraint is applied in order to avoid a side effect of the update process that happens when $\boldsymbol{W}$ loses full rank. When full rank is lost, the column vectors of $\boldsymbol{W}$ are no longer linearly independent, which translates in the task space into the loss of control over one of the two degrees of freedom of the cursor. The details of this function are given in Appendix A.



(a) Cursor speed

(b) Corrected movement

Figure 3.6: Corrective approach in action, for a sequence of two submovements of the hand in the cursor control task.

Figure 3.6(a) shows the cursor speed profile for an example movement of the hand. Local minima in the speed profile, indicated by the labels A, B and C, are used to determine the separation between the submovements. Two submovements are detected here, $\overrightarrow{AB}$ and $\overrightarrow{BC}$. Looking at the actual cursor path, the blue dots in Figure 3.6(b), we clearly see two submovements. As we can see in Figure 3.6(b), the second submovement, $\overrightarrow{BC}$, is shorter in both length and duration than the first submovement $\overrightarrow{AB}$, and therefore satisfies the corrective movement criteria for the first submovement. In Figure 3.6(b), the purple dots represent the cursor path that would have resulted from the same hand trajectory with the updated map. This purple cursor path, with update, is closer to the sum of the two paths than the first submovement, $\overrightarrow{AB}$.

The map is updated using the corrective movement as an estimate of the error. This changes how the hand sensor position will be translated into a screen cursor position, in such a way that future similar hand movements will bring the cursor closer to the target. But a major advantage of our approach over what has been done for decoder adaptation is that no knowledge of the real "intent" in movement is used to estimate the error. Instead, we employ the sequence of movements that is produced by the subject.

3.4 Task Configuration

As the task uses different map configurations over successive repetitions, we have to make sure that repetitions of the task do not interfere with one-another. We present the natural map, introduced in Section 3.2, between presentations of unintuitive maps, in order to "wash out" the effects of learning one novel map before a new one is attempted. Using this same initial configuration with all of the subjects, we can better compare the results from different subjects.

As the goal of this experiment is to study the learning capabilities of the combination of human and algorithm, the difficulty level has to be selected with care so that the learning process can be examined. Moreover, we want the subjects to learn to perform the different configurations in a timely fashion, completing the tasks within approximately 15 minutes. Configurations that are too easy will not show clear learning effects, and configurations that are too hard will require too much time for the subjects to learn. Therefore, we choose task difficulty levels that enable subjects to show learning effects in the matter of minutes.

A solution to adjusting task difficulty is to change the movement precision by reducing the size of the target on the screen. We selected a target size so that learning performance will asymptote well within the 15 minutes allotted to each trial. If target size is selected to be too small, visual fatigue can increase dramatically, as smaller targets require a lot more of concentration from the subject.

The initial map parameters also play an important role in the performance of the subjects. A way to make sure configurations are hard to learn is to switch at least one of the axes of the map. We do this by selecting particular parameters for the model. Thus, when translating hand movement to cursor movement, at least one of the directions of movement in sensor space is not the same as the direction of movement in cursor space. We show this set in Appendix C.

Finally, in order to study the benefits of the learning algorithm, we employ a control condition in which the learning algorithm is not active. Under this condition, the only changes in subject behavior are due to learning by the subject. We call the condition in which the subject is learning by herself "constant," and the condition where the subject is learning with the help of our learning algorithm "adaptive."

Chapter 4

Results

4.1 Experimental protocol

A Patriot Polhemus™ magnetic sensing system is used to track hand motion. It has been attached to a plastic tube, which is grasped by the subject using a palmar grasp, as shown on Figure 4.1(a). This setup has been chosen to enforce the subject's movement strategy. With a palmar grasp and with the sensor located near the palm, the subject is forced to move her arm to control the cursor, which limits the possible use of wrist movements. The subject performs the task standing in front of the screen, two meters away from the screen as presented by Figure 4.1(b).

The cursor control task is an abstraction for the use of a prosthetic arm. Thus, the presentation of a new map stands in for the occurrence of a drift in the inputs of the model. When facing a new map, a subject has no idea of the relationship between the hand sensor and the screen cursor, and she must learn this relationship. We are interested in studying how the performance of a subject changes with time when she is exposed to a new map. Furthermore, as our goal is to show that the approach we propose allows humans to adapt faster to new maps, we compare the subject's performance alone with her performance when coupled with the adaptive learning algorithm.

In order to compare the two conditions, we present a subject with multiple novel maps, one for each trial. An experimental session consists of six trials of 15 minutes each, operated on six different maps. During each trial, the subject has to reach the targets on the screen that appear.

(a) Palmar grasp of a tube on which a magnetic sensor has been fixed to track hand movements

(b) Task from the subject's perspective

Figure 4.1: Subject performing the cursor control task

This approach opens the way for "meta-learning" effects, in which the subject learns general skills over time that enable her to perform better on maps that are presented later in the training process. Having multiple maps allows us to address the issue of meta-learning. We refer to the maps with numerical identifiers from 1 to 6. Counterbalancing the map ordering deals with variation in difficulty of individual maps. For instance, some subjects received the map order: 1, 2, 3, 4, 6, while others received: 2, 1, 4, 3, 6, 5.

In an effort to minimize the transfer of specific knowledge from one trial to the next, every trial starts with a warm-up period of 50 reaches on the natural map, which has been discussed in Section 3.4. As this configuration is easy to get used to and is easy to perform for all subjects, it tends to "wash away" the specifics of the hand-to-cursor mapping from the previous trial. This allows us to more consistently compare one trial within and across subjects. After the warm-up period, the map is switched from the natural map to a new map with which the subject has no experience.

For the analysis, trials are grouped by blocks of two trials, such that trials #1 and #2 form block #1, trials #3 and #4 form block #2, and trials #5 and #6 form block #3. Every block consists of an adaptive trial followed by a constant trial. This condition ordering gives the benefit to the constant condition with respect to meta-learning effects, as the constant trials are always performed with more experience in the task.

In addition, because all the trials are performed in a single session, there is a possible fatigue issue. The duration of 15 minutes has been chosen as a compromise, as it enables the subjects to perform enough reaches to allow statistical analysis, and it is short enough for the subject not to develop too much fatigue. In addition, there are short rest periods every 50 reaches and more substantial ones between trials. As the adaptive condition is presented before the constant condition in every block, if there is fatigue, then it would give an advantage to the adaptive condition over the constant condition.

4.2 Experimental Subjects

Six subjects gave informed consent to participate in this study. The experiment has been approved by the University of Oklahoma Internal Review Board (IRB #12569). The subjects were recruited on campus, by face-to-face solicitations in the laboratory and by an email sent to the Department of Computer Science. Based on previous exploratory experiments, a clear factor affecting performance in the task is the degree of experience with video games in which hand-to-video movements must be constantly re-mapped. Therefore, the subjects selected were all older than 25. This guarantees that even though they likely have some experience with video games, they do not play them on a daily basis at the moment of the experiment.

4.3 Data analysis

The data gathered during the different trials are analyzed using the following performance metrics: *relative time to target* (RTT) and *relative path length* (RPL).

The relative time to target represents the time required by a human subject to reach from one target to the next, divided by the Euclidean distance between the two

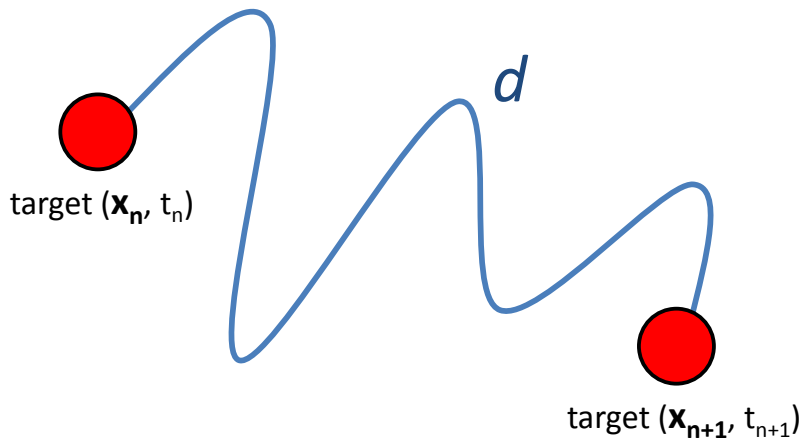


Figure 4.2: Reach of two successive targets. n indexes the division between submovements, t_n is the time the first target was reached, t_{n+1} is the time the next target was reached, x_n and x_{n+1} are the position of the targets, and d is the length of the cursor path between the reach of the two targets.

targets. Figure 4.2 outlines the elements of the task that are used to compute the metric values. As Equation 4.1 shows, RTT is the inverse of average speed:

$$RTT = \frac{t_{n+1} - t_n}{||\boldsymbol{x_{n+1}} - \boldsymbol{x_n}||}. \tag{4.1}$$

The cursor relative path length is the distance traveled by the cursor between two target reaches, divided by the Euclidean distance between these two targets. Equation 4.2 presents cursor RPL:

$$RPL = \frac{d}{||\boldsymbol{x_{n+1}} - \boldsymbol{x_n}||}. \tag{4.2}$$

With near perfect performance, the RPL should be close to 1, since the subjects would move the cursor in approximately straight lines. It is also possible to compute *hand RPL*, by using the hand path between two target reaches instead of cursor path.

For these three metrics, RTT, cursor RPL and hand RPL, values decreasing over time show that subjects are improving in their performance of the task. As subjects adapt to new maps, one part of the improvement in performance is due to the subject learning the map, and the other part is due to the approach we propose.

In all the displays of subjects' performance presented below, statistical significance between the two conditions is indicated by vertical lines. Green vertical lines indicate an advantage of adaptive over constant. In addition, these vertical lines have been marked with "+" and "x" symbols to indicate that the statistical tests passed with p-values of $p < .05$ and $p < .01$ respectively. The statistical test is a left-tailed two-sample t-test. All p-values are Bonferroni-adjusted to account for multiple comparisons performed over a trial (Jensen and Cohen, 2000).

4.4 Experimental Results

The set of subjects who were tested on our approach are six naïve subjects. These subjects have no prior experience in the task. Subjects #1 and #3 received the map order: 1, 2, 3, 4, 5, 6, subjects #2 and #4 received: 2, 1, 4, 3, 6, 5, subject #5 received a rotation: 5, 6, 1, 2, 3, 4, and subject #6 received another rotation: 6, 5, 2, 1, 4, 3. We first analyze the data of subjects #1 through #4 only, as two subjects, one for each rotation, are missing in order to counterbalance properly the rotated subjects #5 and #6.

The data from block #1 are thrown away, as discussed in Section 4.1. Every curve in Figure 4.3 is an average of the performance of four subjects, all subjects performing on the same four maps. For each subject, two of these maps were performed with the constant condition, and the two others with the adaptive condition.

Figure 4.3(a) and 4.3(b) show for 25-reach bins mean RTT and standard error, for the blocks #2 and #3, respectively. Regarding fatigue effects, we can see here that this is not an issue. Indeed, looking at blocks #2 and #3, the adaptive and constant conditions asymptote both around the same mean RTT and around the same reaches (approximately 150 reaches for block #2 and 250 reaches for block #3).

Also, the presence of asymptotic performance in the last bins of every trial suggests that the subjects have largely adapted to the map, and managed therefore to perform consistently on the new maps that we present to them.

In Figure 4.3(b), the adaptive curve seems flat compared to the constant curve, which exhibits a nice asymptotic learning shape. Though the mean RTT appears flat for the adaptive case, the first 10 reaches do show non-optimal behavior. In addition, and as stated in Section 4.1, the condition ordering always benefits to the constant

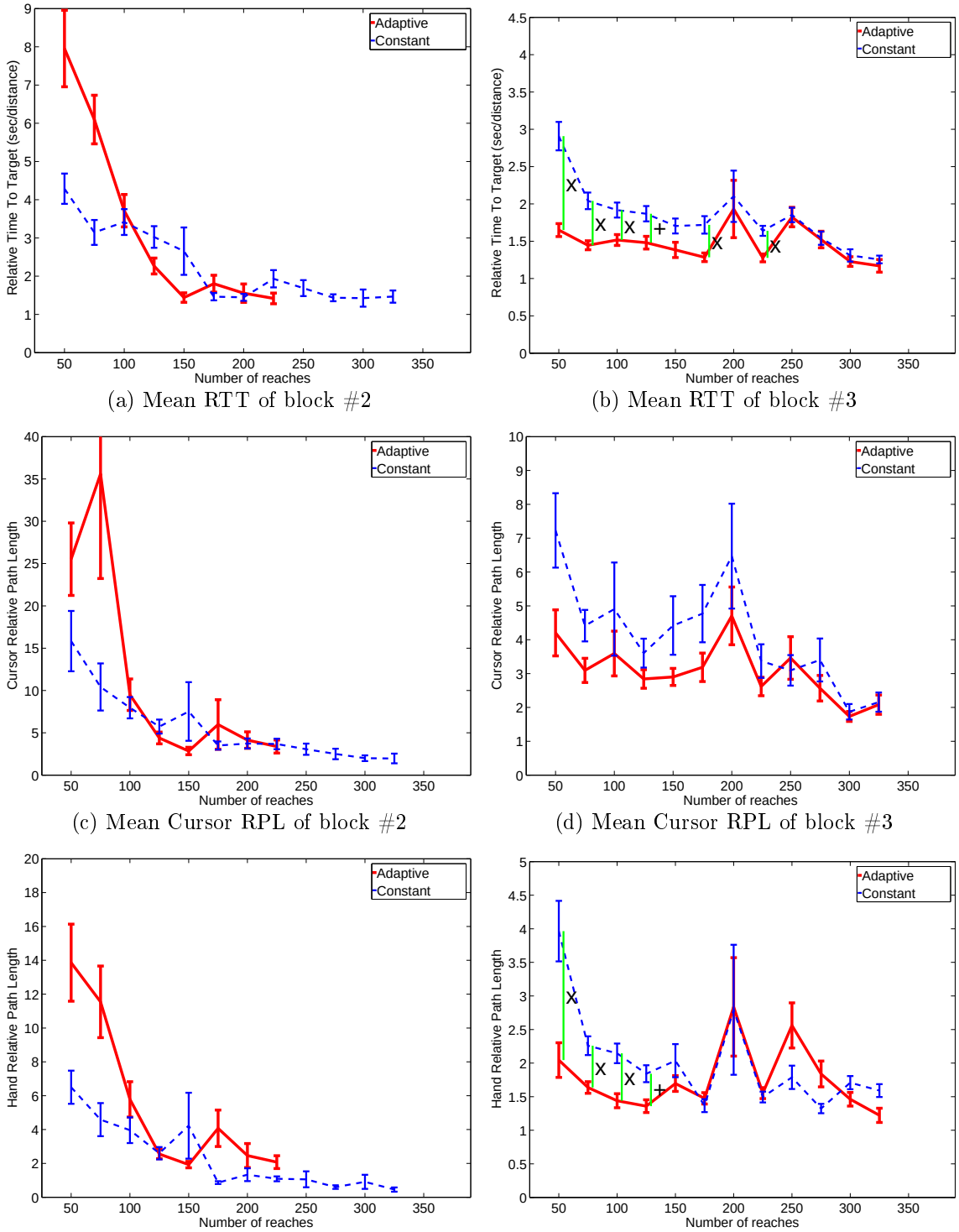
condition, as it is always presented after an adaptive condition within all blocks. Reaches #50 through #175 show a significant advantage of adaptive at $p < 0.01$, except around reach #125, that passes at $p < 0.05$. Therefore here the adaptive condition allows the subjects to perform better, even though the ordering of adaptive and constant conditions gives the adaptive condition a handicap.

An interesting feature can be observed by comparing the first reaches of blocks #2 and #3, for both RTT and RPL. Looking at RTT for block #2 in Figure 4.3(a), the subjects perform worse in the adaptive condition than in the constant condition. However, in block #3 (Figure 4.3(b)), we see that the adaptive condition allows the subjects to perform better than the constant condition, from the beginning until the end of the trial. The difference between mean RTT values in the adaptive and constant conditions for blocks #2 and #3 are largely due to meta-learning effects. Note also that each of the block #3 traces exhibit modes around reach #200. These are in fact due to two outliers and do not represent a general trend. A similar pattern is demonstrated in the cursor and hand RPL metrics, in Figures 4.3(c) and 4.3(e) for block #2, and Figures 4.3(d) and 4.3(f) for block #3. In particular, hand RPL correlates to RTT with a significant advantage of the constant condition over the adaptive condition for reaches around #50 and #75 with $p < .05$, and a significant advantage of the adaptive condition over the constant condition from reaches #50 through #125, with $p < .05$ and $p < .01$.

Finally, Figure 4.4 shows the performance of all six subjects. The analysis of these curves is not developed further, as two rotated subjects are missing to complete the counter-balancing. Nevertheless, the shapes of the curves of subjects #1 through #4 is quite similar when adding the data of subject #5 and #6. The difference between the adaptive and constant conditions is even bigger than for subjects #1 through #4 only. Also, individual performance curves for all subjects are presented in Appendix B.

As a conclusion of our analysis of block #3, and based on mean RTT and RPL values, we show a clear advantage of the adaptive condition over the constant condition. Furthermore, we argue that the advantage of the constant condition over the adaptive in block #2 should be seen only as a transitory phenomenon due to the learning of the task. Block #3 is more indicative of what we would expect in the

long term, which is why only the data from block #3 should be considered for further analysis.



(a) Mean RTT of block #2

(b) Mean RTT of block #3

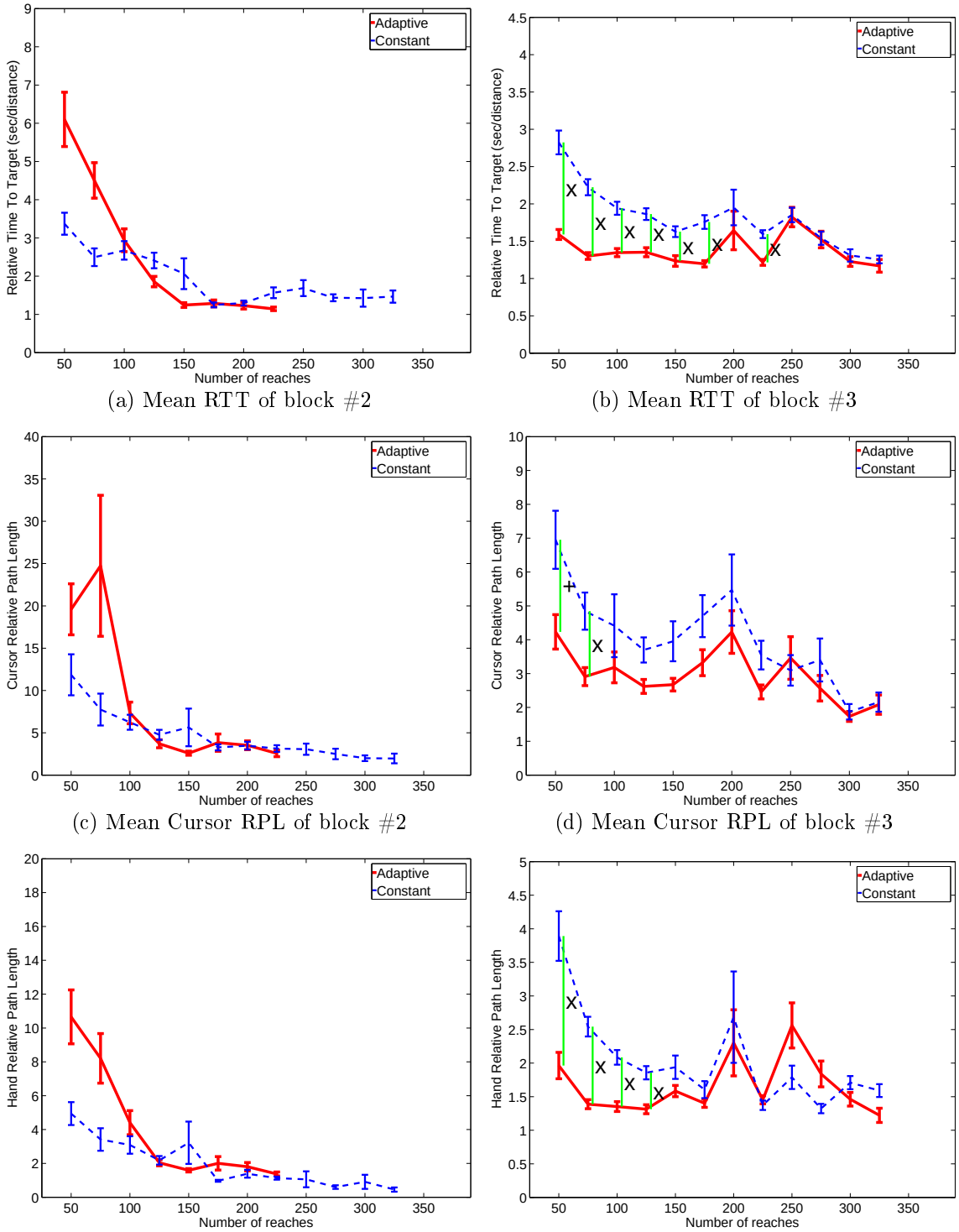
(c) Mean Cursor RPL of block #2

(d) Mean Cursor RPL of block #3

(e) Mean Hand RPL of block #2

(f) Mean Hand RPL of block #3

Figure 4.3: Naïve subjects #1 through #4 - Mean RTT, RPL and standard error (25-reach bins). Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.



(a) Mean RTT of block #2

(b) Mean RTT of block #3

(c) Mean Cursor RPL of block #2

(d) Mean Cursor RPL of block #3

(e) Mean Hand RPL of block #2

(f) Mean Hand RPL of block #3

Figure 4.4: Naïve subjects #1 through #6 - Mean RTT, RPL and standard error (25-reach bins). Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.

Chapter 5

Conclusion

In this study, we proposed a novel method for the continuous calibration of prosthesis command systems. This learning approach make use of errors in movements in order to update the map as the patient uses the robotic arm. These errors were used to drive the descent of a gradient-descent search, for a map that better interprets the motion "commands" produced by the subject.

We developed a non-invasive cursor control task as an abstraction for the control of a real prosthetic arm, in order to show the benefit of the algorithm we proposed. This task asks human subjects to control a screen cursor using a sensor attached to their hands. The learning algorithm alters the relationship between the hand position and the cursor. Due to meta-learning effects, only the data from block #3 (trials #5 and #6 of six trials total) has been considered to be relevant. Asymptotic performance was the same for both adaptive and constant decoders, which showed an overall adaptation of the subjects to the task. However, the adaptive condition showed a significantly improved performance in early trials of block #3 compared to the constant condition. Thus, as the subject performed the task, the learning algorithm made the subjects' learning process faster by adapting the maps to individual subjects.

As the task consisted of reaching to many targets, it induced a certain degree of visual and muscle fatigue. Fatigue effects would give an advantage to early trials compared to the ones that follow them. As our experimental protocol starts blocks with adaptive conditions, this also means that the adaptive condition could have an advantage over the constant one. Nevertheless, the asymptotic behavior has been the same for adaptive and constant condition in the same blocks and across different blocks, which means that fatigue is not a problem.

Finally, an interesting question is to know whether or not this algorithm will be usable in a fielded scenario. To begin with, the algorithm does not need any knowledge of the patient's intent. Instead, it uses estimates of the error in movement, computed using the behavior developed by the patient in the task. A major prerequisite for our approach is that the subject must be able to successfully perform movements within a reasonable period of time. If the patient cannot perform movements because of the difficulty of the task, there will be no meaningful corrections to serve as input for the algorithm to update the map appropriately. Arbitrary updates would be made to the map, and the map would not converge to a configuration helping the patient. A solution to this issue is to use traditional methods for initial calibration, and then to switch to automatic calibration thereafter.

An open problem is whether the approach can be applied to situations in which the input feature space approaches that which is typical in modern BMI systems (on the order of 1,000 inputs). As the error is computed in task space (arm control, three degrees of freedom), and not command space (motor command, on the order of 1,000 degrees of freedom), numerical issues are possibly avoided, and the error can be used to drive the calibration process in a meaningful way. Therefore, due to its design, we hope that the algorithm we propose will be usable for higher degree of freedom systems.

The use of a set of neurons to control a prosthetic arm is a truly deep and interesting problem. The outcomes of the research in that domain could help amputees to have a normal life thanks to robotic prosthetic limbs.

Bibliography

N.E. Berthier. Analysis of reaching for stationary and moving objects in the human infant. In J. Donahoe and V.P Dorsal, editors, *Neural Network Models of Complex Behavior - Biobehavioral Foundations*, pages 283–301. North Holland: Amsterdam, 1997.

N.E. Berthier and D.J. Robin. Mid-reach corrections to shifting targets by seven-month-old infants. *Journal of Motor Behavior*, 30:290–300, 1998.

N.E. Berthier, S.P. Singh, A.G. Barto, and J.C. Houk. Distributed representation of limb motor programs in arrays of adjustable pattern generators. *Journal of Cognitive Neuroscience*, 5(1):56–78, 1993.

N.E. Berthier, R.K. Clifton, V. Gullipalli, D.D. McCall, and D.J. Robin. Visual information and object size in the control of reaching. *Journal of Motor Behavior*, 28(3):187–197, 1996.

N.E. Berthier, M.T. Rosenstein, and A.G. Barto. Approximate optimal control as a model for motor learning. *Psychological Review*, 112(2):329–346, 2005.

B. Biguer, M. Jeannerod, and C. Prablanc. The coordination of eye, head, and arm movements during reaching at a single visual target. *Experimental Brain Research*, 46(2):301–304, 1982.

D.W. Cabel, P. Cisek, and S.H. Scott. Neural activity in primary motor cortex related to mechanical loads applied to the shoulder and elbow during a postural task. 86 (4):2102–2110, 2001.

R.L. Carrico and N.E. Berthier. Vision and precision reaching in 15-month-old infants. *Infant Behavior and Development*, 31(1):62–70, 2008.

U. Castiello, K. Bennett, and H. Chambers. Reach to grasp: the response to a simultaneous perturbation of object position and size. *Experimental Brain Research*, 120(1):31–40, 1998.

R.K. Clifton, D.W. Muir, D.H. Ashmead, and M.G. Clarkson. Is visually guided reaching in early infancy a myth? *Child Development*, 64(4):1099–1110, 1993.

P.R. Cohen. *Empirical methods for artificial intelligence*. MIT press, 1995.

Z. Danziger, A. Fishbach, and F.A. Mussa-Ivaldi. Adapting human-machine interfaces to user performance. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, volume 1, pages 4486–4490, 2008.

J. del R. Millán. Adaptive brain interfaces. *Communications of the ACM*, 46(3): 74–80, 2003.

M. Desmurget, C. Prablanc, M. Arzi, Y. Rossetti, Y. Paulignan, and C. Urquizar. Integrated control of hand transport and orientation during prehension movements. *Experimental Brain Research*, 110(2):265–278, 1996.

A.S. Dickey, A. Suminski, Y. Amit, and N.G. Hatsopoulos. Single-unit stability using chronically implanted multielectrode arrays. *Journal of Neurophysiology*, 102(2): 1331–1339, June 2009.

A.H. Fagg, J.C. Houk, and A.G. Barto. Learning to reach via corrective movements. In *Proceedings of the Tenth Yale Workshop on Adaptive and Learning Systems*, pages 179–185, 1998.

A.H. Fagg, N.G. Hatsopoulos, G. Ojakangas, and L.E. Miller. Kinetic trajectory decoding using motor cortical ensembles. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(5):487–496, 2009.

A. Fichbach, L.E. Miller, S.A. Roy, C Bastianen, and James C. Houk. Kinematic properties of on-line error corrections in the monkey. *Experimental Brain Research*, 164(4):442–457, 2005.

A. Fichbach, L.E. Miller, S.A. Roy, C. Bastianen, and J.C. Houk. Deciding when and how to correct a movement: discrete submovements as a decision making process. *Experimental Brain Research*, 177(1):45–63, 2007.

W.E. Finn and P.G. LoPresti. *Handbook of neuroprosthetic methods*. CRC, 2003.

T. Flash and E. Henis. Arm trajectory modifications during reaching towards visual targets. *Journal of Cognitive Neuroscience*, 3(3):220–230, 1991.

A.P. Georgopoulos, R. Caminiti, and J.F. Kalaska. Static spatial effects in motor cortex and area 5: quantitative relations in a two-dimensional space. *Experimental Brain Research*, 54(3):446–454, 1984.

A.P. Georgopoulos, A.B. Schwartz, and R.E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

P. Haggard and A. Wing. Coordinated responses following mechanical perturbation of the arm during prehension. *Experimental Brain Research*, 102(3):483–494, 1995.

N. Hatsopoulos and P. Donoghue. The Science of Neural Interface Systems. *Annual Review of Neuroscience*, 32:249–266, 2009.

N. Hatsopoulos, J. Joshi, and J.G. O'Leary. Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles. *Journal of neurophysiology*, 92(2):1165, 2004.

L.R. Hochberg, M.D. Serruya, G.M. Friehs, J.A Mukand, M. Saleh, A.H. Caplan, A. Branner, D. Chen, R.D. Penn, and J.P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, July 2006.

M. Jeannerod. Intersegmental coordination during reaching at natural visual objects. In *Attention and Performance IX: Proceedings of the Ninth International Symposium on Attention and Performance, Jesus College, Cambridge, England, July 13-18, 1980*, pages 153–169. Lawrence Erlbaum, 1981.

M. Jeannerod. The timing of natural prehension movements. *Journal of Motor Behavior*, 16(3):235–254, 1984.

D.D. Jensen and P.R Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309–338, 2000.

D.J. McFarland and J.R. Wolpaw. Brain-computer interface operation of robotic and prosthetic devices. *Computer*, 41(10):52–56, October 2008.

T.E. Milner. A model for the generation of movements requiring endpoint precision. *Neuroscience*, 49(2):487–496, July 1992.

K. Munhall and A. Löfqvist. Gestural aggregation in speech. *The Journal of the Acoustical Society of America*, 84(S1):S82, 1988.

K.E. Novak, L.E. Miller, and J.C. Houk. Kinematic properties of rapid hand movements in a knob turning task. *Experimental Brain Research*, 132(4):419–433, 2000.

L. Paninski, M.R. Fellows, N.G. Hatsopoulos, and J.P. Donoghue. Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *Journal of neurophysiology*, 91(1):515–532, 2004.

J.H. Piater, P.R. Cohen, X. Zhang, and M. Atighetchi. A randomized ANOVA procedure for comparing performance curves. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, pages 430–438, 1998.

C. Prablanc and O. Martin. Automatic control during hand reaching at undetected two-dimensional target displacements. *Journal of Neurophysiology*, 67(2):455–469, 1992.

D. Rotermund and U.A. Ernst K.R. Pawelzik. Towards on-line adaptation of neuroprostheses with neuronal evaluation signals. *Biological Cybernetics*, 95(3):243–257, 2006.

A.B. Schwartz, X.T. Cui, D.J. Weber, and D.W. Moran. Brain-controlled interfaces: Movement restoration with neural prosthetics. *Neuron*, 52(1):205–220, 2006.

M. Serruya, N. Hatsopoulos, M. Fellows, L. Paninski, and J. Donoghue. Robustness of neuroprosthetic decoding algorithms. *Biological Cybernetics*, 88(3):219–228, 2003.

M. Taira, J. Boline, N. Smyrnis, A.P. Georgopoulos, and J. Ashe. On the relations between single cell activity in the motor cortex and the direction and magnitude of three-dimensional static isometric force. *Experimental Brain Research*, 109(3): 367–376, 1996.

D.M. Taylor, S.I.H. Tillery, and A.B. Schwartz. Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574):1829–1832, 2002.

D. Tkach, J. Reimer, and N.G. Hatsopoulos. Observation-based learning for brain–machine interfaces. *Current Opinion in Neurobiology*, 18(6):589–594, 2008.

C. von Hofsten. Development of visually directed reaching: The approach phase. *Journal of Human Movement Studies*, 5:160–178, 1979.

C. von Hofsten. Structuring of early reaching movements: a longitudinal study. *Journal of Motor Behavior*, 23(4):280–292, 1991.

C. von Hofsten and A. Forsström. Visually directed reaching of children with motor impairments. *Developmental Medicine and Child Neurology*, 24(5):653–661, 1982.

R. Wahnoun, J. He, and S.I.H. Tillery. Selection and parameterization of cortical neurons for neuroprosthetic control. *Journal of Neural Engineering*, 3(2):162–171, 2006.

W. Wu, MJ Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, and JP Donoghue. Neural decoding of cursor motion using a Kalman filter. *Advances in neural information processing systems*, 15:133–140, 2003.

Appendix A

Angle constraint function F_A

The F_A function constraint the angle between the $\boldsymbol{W}$ row vectors, $\boldsymbol{W}$ being the set of parameters for the model. This constraint is applied in order to avoid a side effect of the update process that happens when $\boldsymbol{W}$ loses full rank. When full rank is lost, the row vectors of $\boldsymbol{W}$ are no longer linearly independent, which translates in the task space into the loss of control over one of the two degrees of freedom of the cursor. In order to avoid these drawbacks, the vectors of $\boldsymbol{W}$ are modified when it is necessary, such that the angle between them is set to a particular minimum value. We denote as $\boldsymbol{W}$ the input of this function, and $\boldsymbol{W_{new}}$ the output. We denote as $\boldsymbol{A}$ and $\boldsymbol{B}$ the two row vectors of the 2-by-3 model parameter matrix, $\boldsymbol{W}$, such that $\boldsymbol{W} = [\boldsymbol{A},\ \boldsymbol{B}]$. We start by using the Al-Kashi theorem (also known as cosinus theorem) to compute the current angle between $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\theta_{current} = arccos\left(\frac{\boldsymbol{A}\boldsymbol{B}^T}{\sqrt{\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{B}^T}}\right). \tag{A.1}$$

Then, we compute the objective angle, that is to say the angle we want to see between $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\theta_{objective} = \begin{cases} \theta_{min} & \text{if } \theta_{current} < \theta_{min}, \\ \pi/2 - \theta_{min} & \text{if } \theta_{current} > 180 - \theta_{min}, \\ nil & \text{otherwise.} \end{cases} \tag{A.2}$$

If $\theta_{objective} = nil$, $\boldsymbol{A}$ and $\boldsymbol{B}$ do not need to be adjusted. Otherwise, $\boldsymbol{A}$ and $\boldsymbol{B}$ are to stay in the same plane and to be adjusted by half the angle that we need to correct:

$$\theta = \frac{\theta_{current} + \theta_{objective}}{2}, \tag{A.3}$$

and we can now compute the new $\boldsymbol{A}$ and $\boldsymbol{B}$ vectors. $\boldsymbol{A_{new}}$ is computed as follows:

$$\boldsymbol{A_{new}} = \alpha_{\boldsymbol{A}}\boldsymbol{A} + \beta_{\boldsymbol{A}}\boldsymbol{B}, \tag{A.4}$$

where:

$$\alpha_{\boldsymbol{A}} = \sqrt{\frac{\boldsymbol{AA}^T - \frac{\boldsymbol{BB}^T\boldsymbol{AA}^T cos(\theta)^2}{\boldsymbol{BB}^T}}{\boldsymbol{AA}^T - \frac{(\boldsymbol{AB}^T)^2}{\boldsymbol{BB}^T}}}, \tag{A.5}$$

$$\beta_{\boldsymbol{A}} = \frac{\sqrt{\boldsymbol{BB}^T}\sqrt{\boldsymbol{AA}^T}cos(\theta) - \alpha_{\boldsymbol{A}}\boldsymbol{AB}^T}{\boldsymbol{BB}^T}. \tag{A.6}$$

$\boldsymbol{B_{new}}$ is computed with a similar method:

$$\boldsymbol{B_{new}} = \alpha_{\boldsymbol{B}}\boldsymbol{A} + \beta_{\boldsymbol{B}}\boldsymbol{B}, \tag{A.7}$$

where:

$$\beta_{\boldsymbol{B}} = \sqrt{\frac{\boldsymbol{BB}^T - \frac{\boldsymbol{AA}^T\boldsymbol{BB}^T cos(\theta)^2}{\boldsymbol{AA}^T}}{\boldsymbol{BB}^T - \frac{(\boldsymbol{BA}^T)^2}{\boldsymbol{AA}^T}}}, \tag{A.8}$$

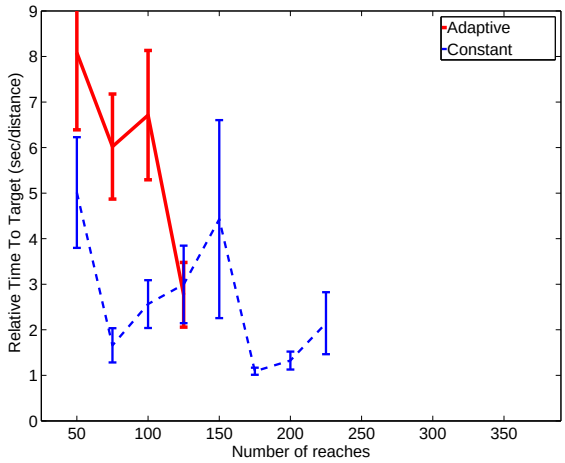
$$\alpha_{\boldsymbol{B}} = \frac{\sqrt{\boldsymbol{AA}^T}\sqrt{\boldsymbol{BB}^T}cos(\theta) - \beta_{\boldsymbol{B}}\boldsymbol{BA}^T}{\boldsymbol{AA}^T}. \tag{A.9}$$

Finally, $\boldsymbol{A_{new}}$ and $\boldsymbol{B_{new}}$ are paired to form the output of F_A, $\boldsymbol{W_{new}} = [\boldsymbol{A_{new}},\ \boldsymbol{B_{new}}]$.

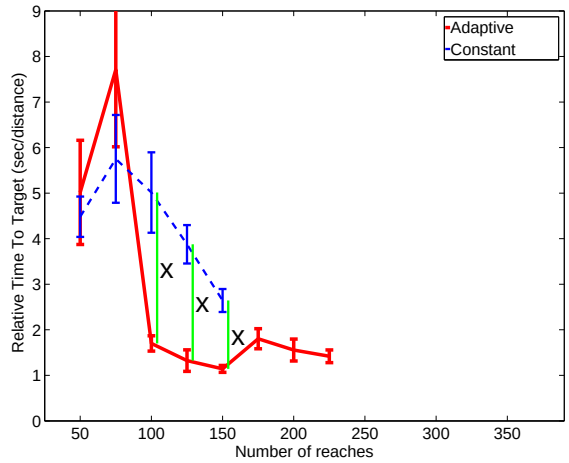
Appendix B

Individual performance curves for all six subjects

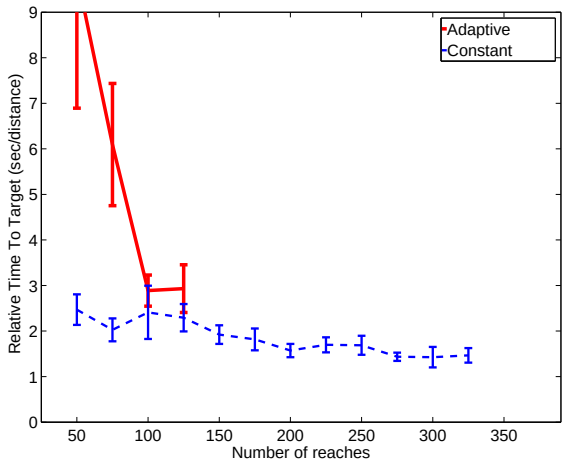
Each page of this appendix shows the performance curves of all six subjects for each block. The caption of each figure at the bottom of each page gives the observed block and metric. Blocks are either #2 or #3, and metrics are RTT, cursor RPL or hand RPL. The caption of each subfigure states which subject is observed and on which maps. For instance, the caption of Figure B.1 tells us that the curves represent RTT and standard error during block #2. Moreover, the caption of Figure B.1(a) tells us that this subfigure represents the performance of subject #1, and that she was performing the task with map #3 during her adaptive trial (A), and with map #4 during her constant trial (C).



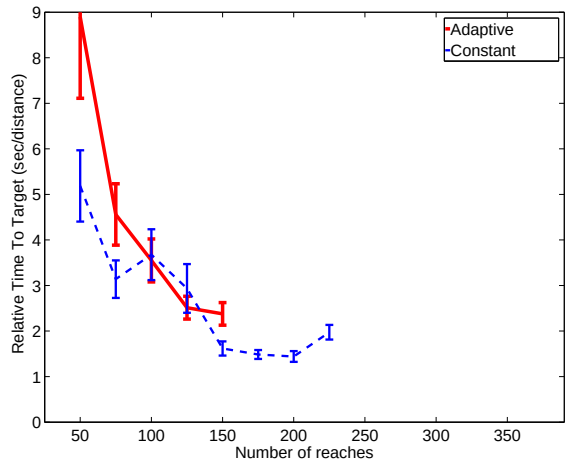
(a) Subject #1 - A: map #3, C: map #4



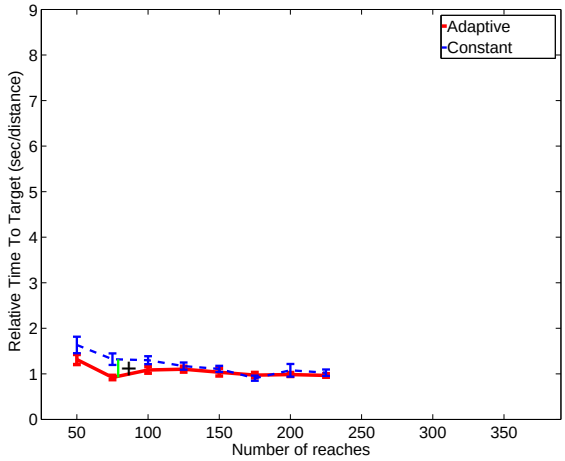
(b) Subject #2 - A: map #4, C: map #3



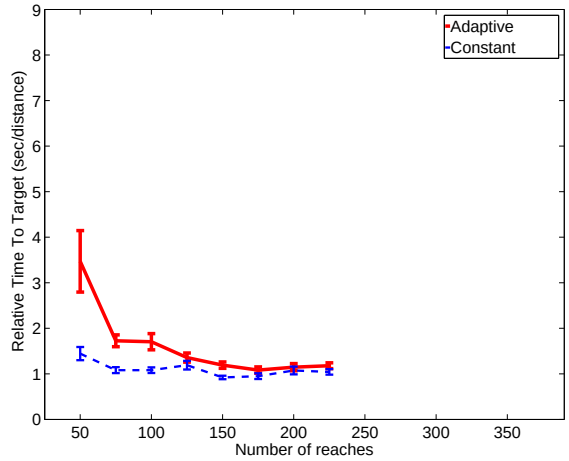
(c) Subject #3 - A: map #3, C: map #4



(d) Subject #4 - A: map #4, C: map #3

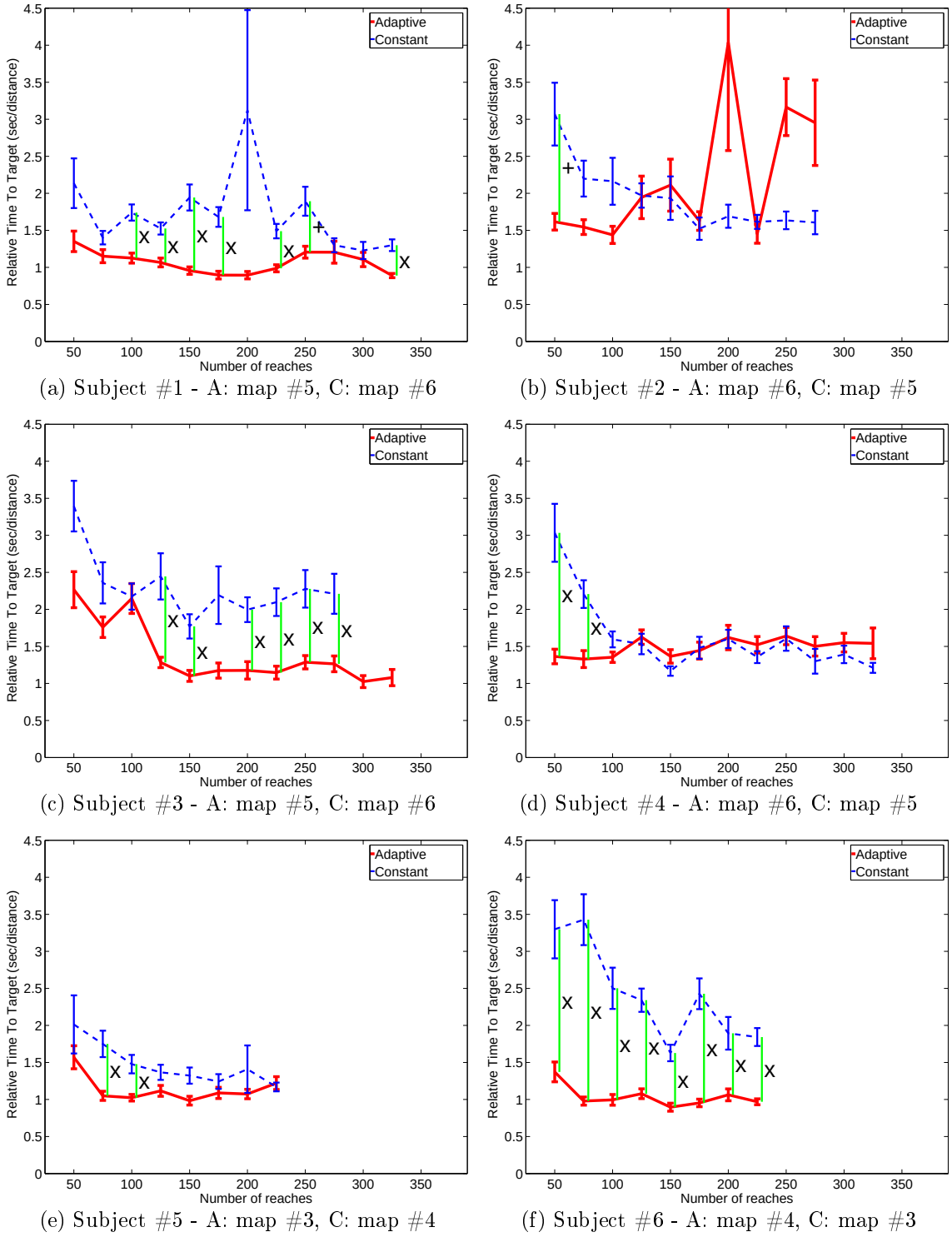


(e) Subject #5 - A: map #1, C: map #2



(f) Subject #6 - A: map #2, C: map #1

Figure B.1: RTT and standard error of block #2 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.



(a) Subject #1 - A: map #5, C: map #6

(b) Subject #2 - A: map #6, C: map #5

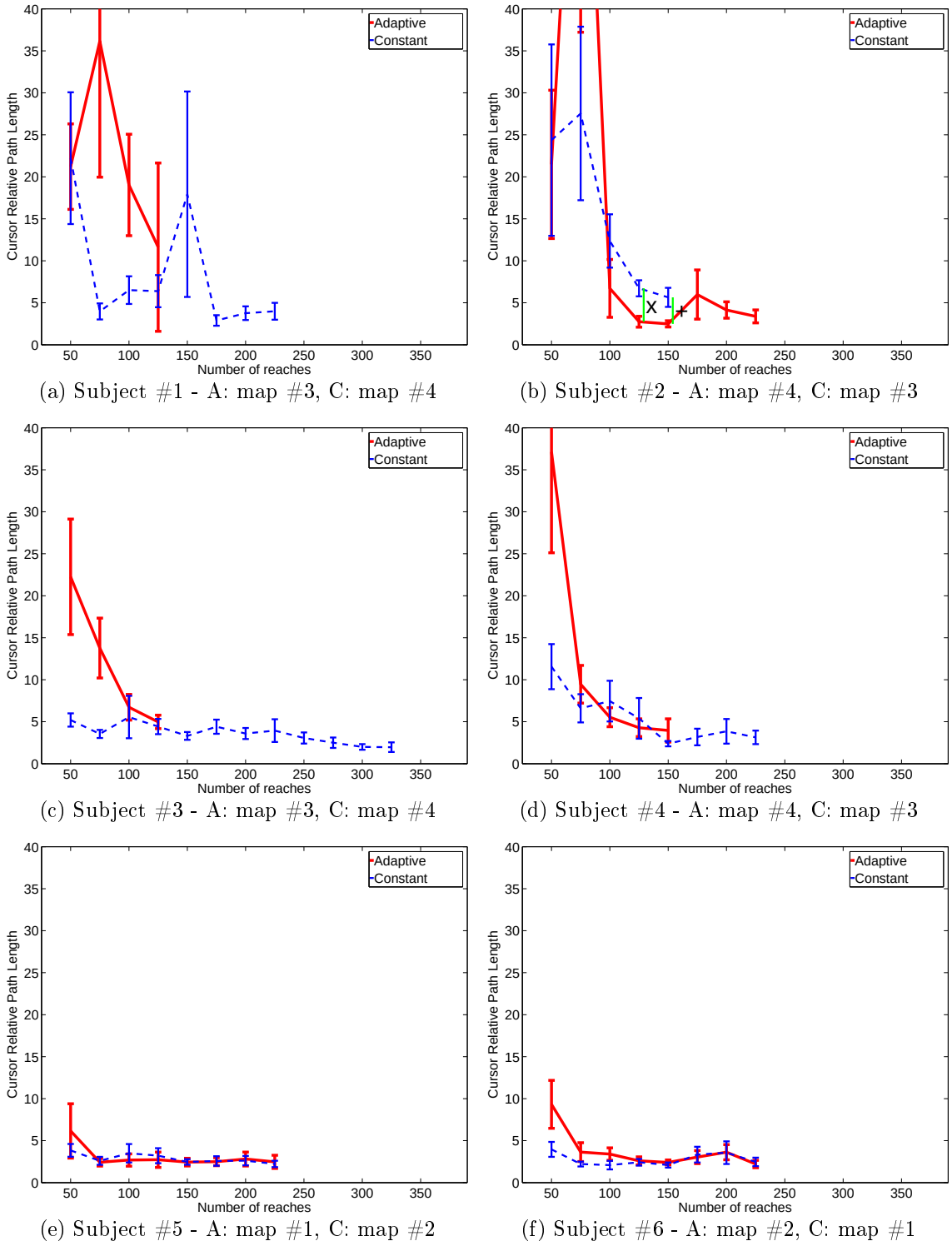
(c) Subject #3 - A: map #5, C: map #6

(d) Subject #4 - A: map #6, C: map #5

(e) Subject #5 - A: map #3, C: map #4

(f) Subject #6 - A: map #4, C: map #3

Figure B.2: RTT and standard error of block #3 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.



(a) Subject #1 - A: map #3, C: map #4

(b) Subject #2 - A: map #4, C: map #3

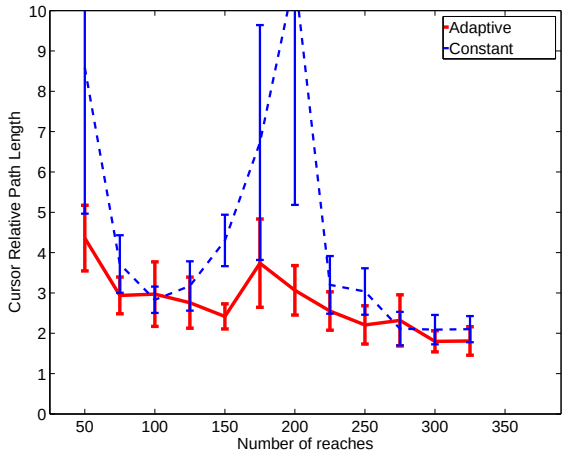
(c) Subject #3 - A: map #3, C: map #4

(d) Subject #4 - A: map #4, C: map #3

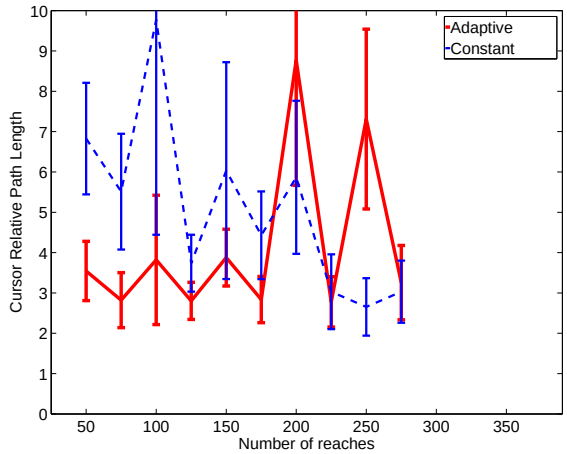
(e) Subject #5 - A: map #1, C: map #2

(f) Subject #6 - A: map #2, C: map #1

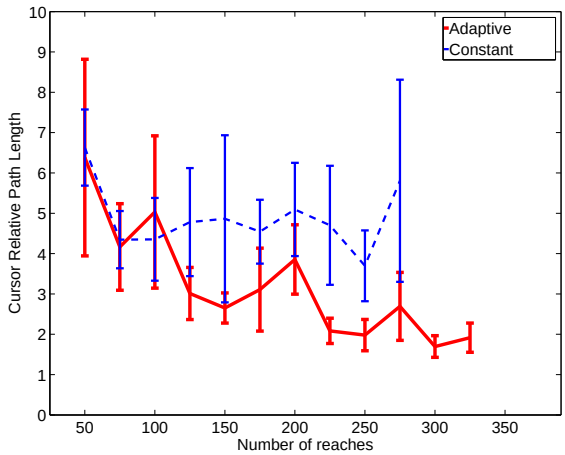
Figure B.3: Cursor RPL and standard error of block #2 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.



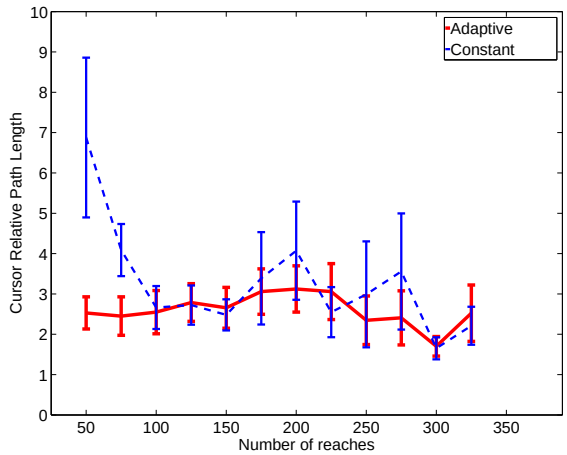
(a) Subject #1 - A: map #5, C: map #6



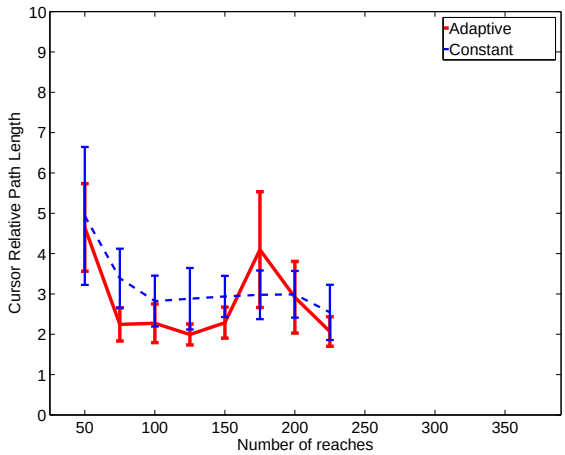
(b) Subject #2 - A: map #6, C: map #5



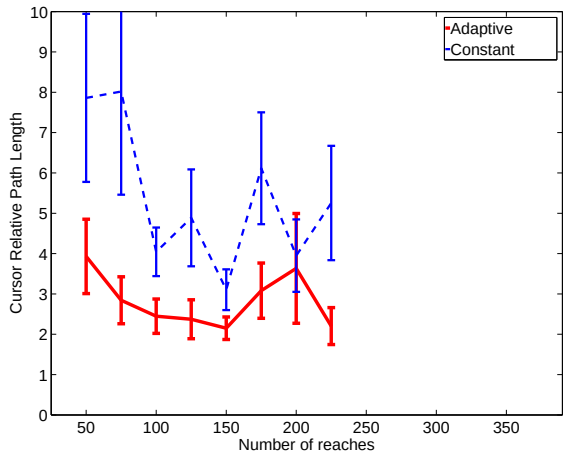
(c) Subject #3 - A: map #5, C: map #6



(d) Subject #4 - A: map #6, C: map #5

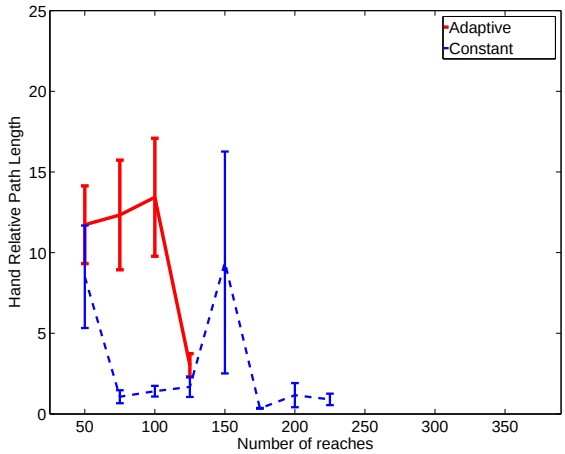


(e) Subject #5 - A: map #3, C: map #4

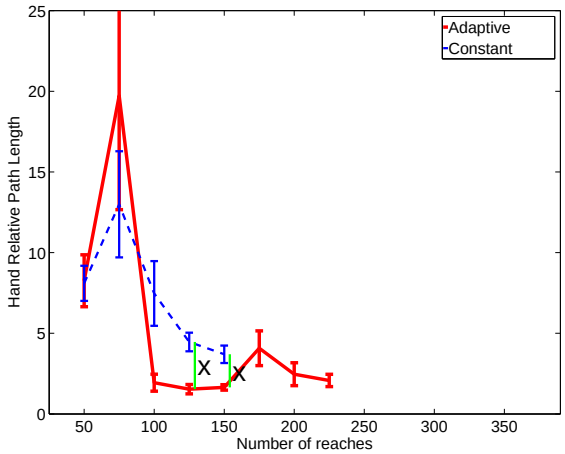


(f) Subject #6 - A: map #4, C: map #3

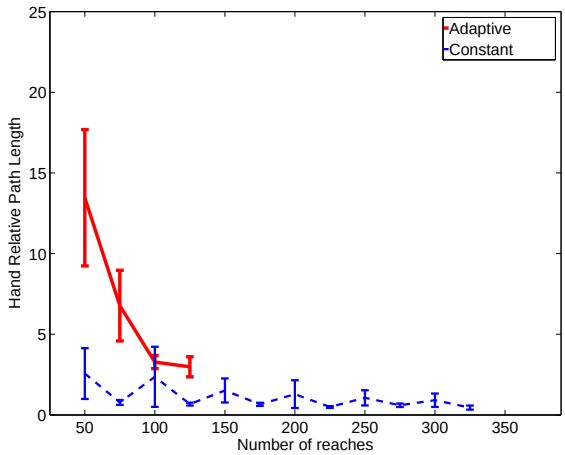
Figure B.4: Cursor RPL and standard error of block #3 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.



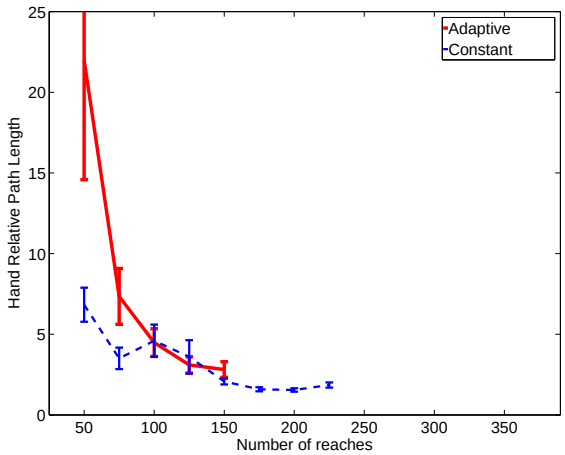
(a) Subject #1 - A: map #3, C: map #4



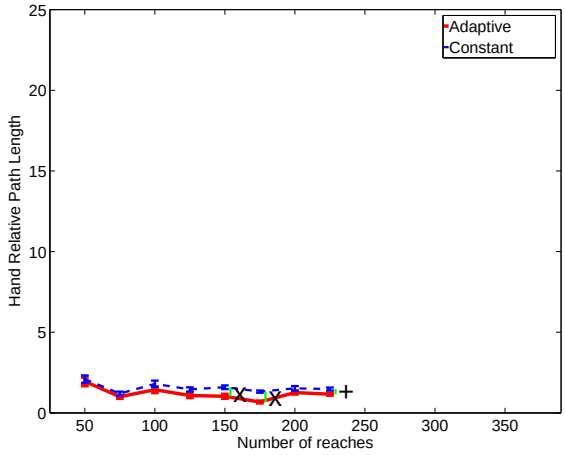
(b) Subject #2 - A: map #4, C: map #3



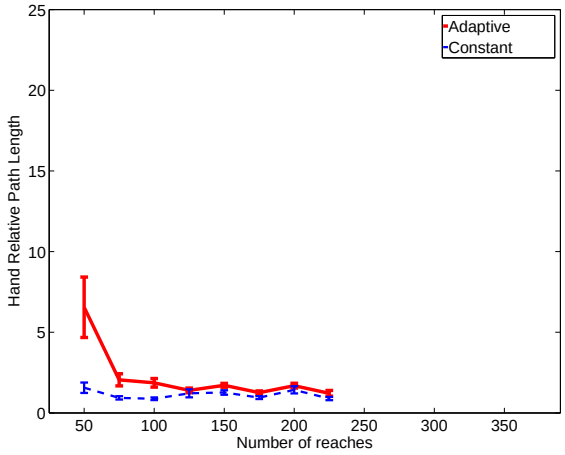
(c) Subject #3 - A: map #3, C: map #4



(d) Subject #4 - A: map #4, C: map #3

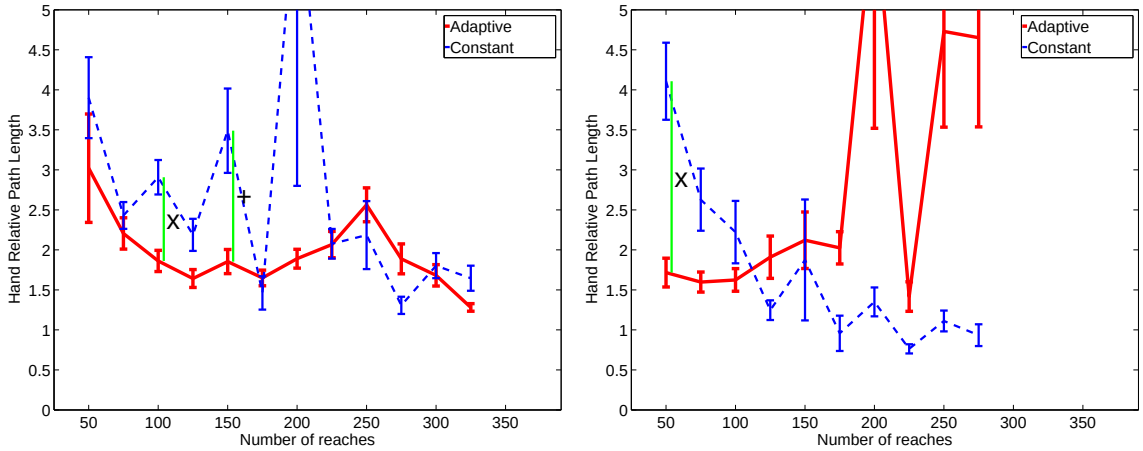


(e) Subject #5 - A: map #1, C: map #2



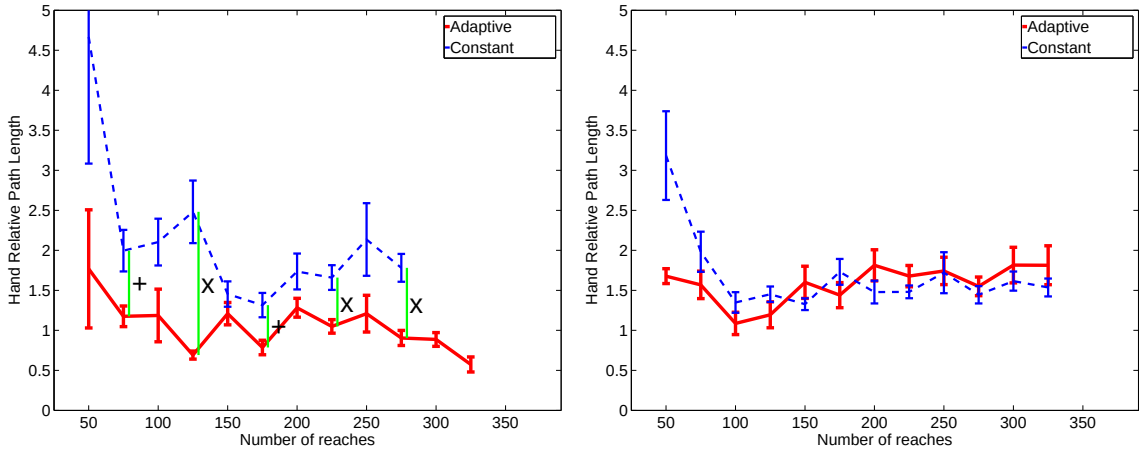
(f) Subject #6 - A: map #2, C: map #1

Figure B.5: Hand RPL and standard error of block #2 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.



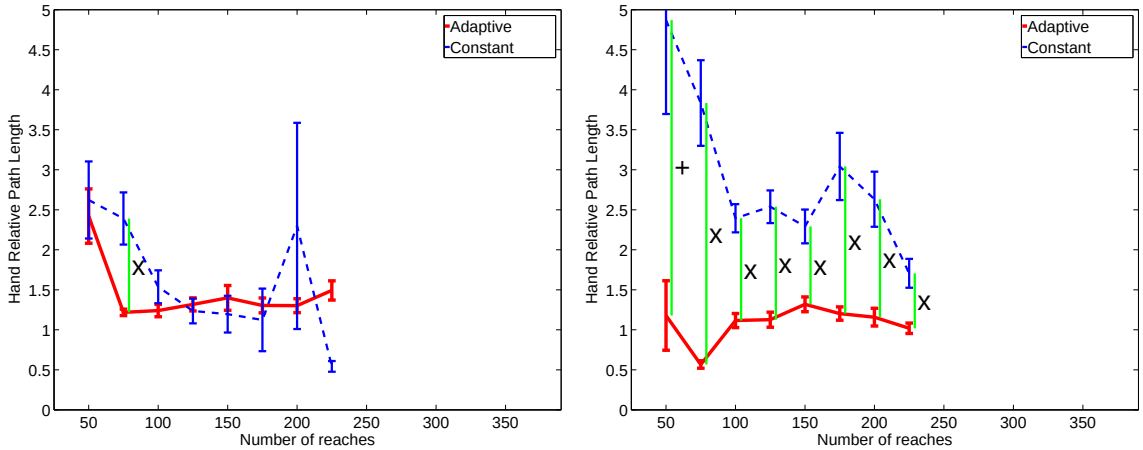
(a) Subject #1 - A: map #5, C: map #6

(b) Subject #2 - A: map #6, C: map #5



(c) Subject #3 - A: map #5, C: map #6

(d) Subject #4 - A: map #6, C: map #5



(e) Subject #5 - A: map #3, C: map #4

(f) Subject #6 - A: map #4, C: map #3

Figure B.6: Hand RPL and standard error of block #3 for all six subjects. Vertical lines with the "+" and "x" symbols indicate statistical advantages of adaptive over constant with p-values of $p < .05$ and $p < .01$ respectively.

Appendix C

Experimental Maps

The six maps used during the experiment have been projected in space as planes, in order to give a spatial representation of the different model parameters. As the studied configurations are just alterations of the natural map shown in Figure C.1 (c.f. Section 3.2), markers are placed on the planes to show how the maps have changed. The blue lines represents where the bottom of the natural map used to be, and the blue square represents where the bottom-left corner of the natural map used to be. Moreover, the red circle shows where the corner of the plane that is the closest to the viewer. This gives a hint as to how is the plane oriented.

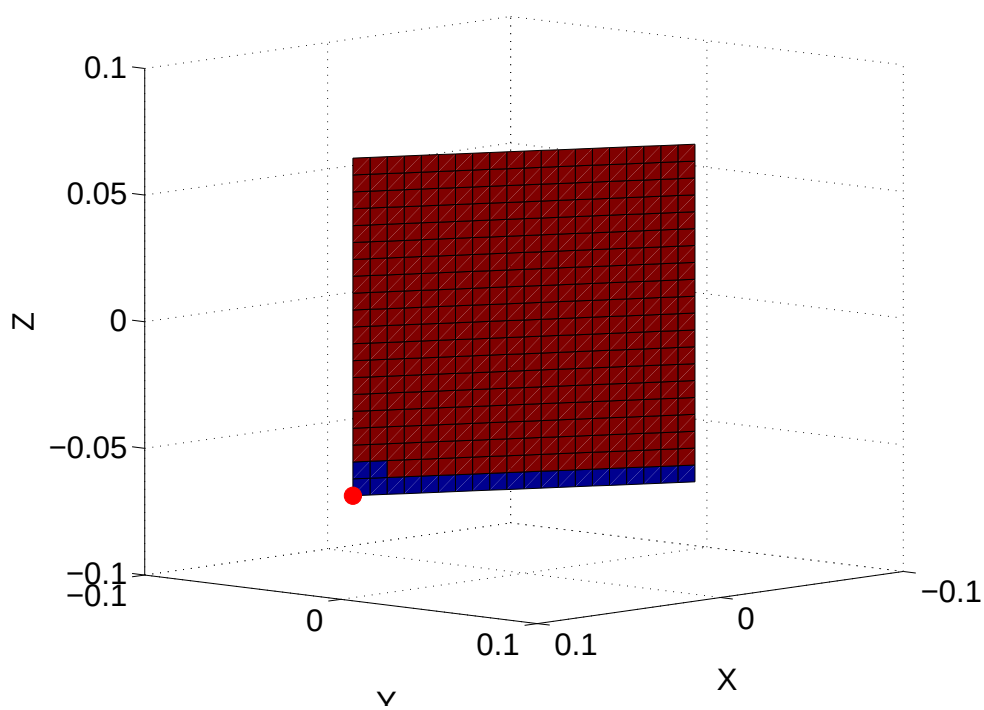
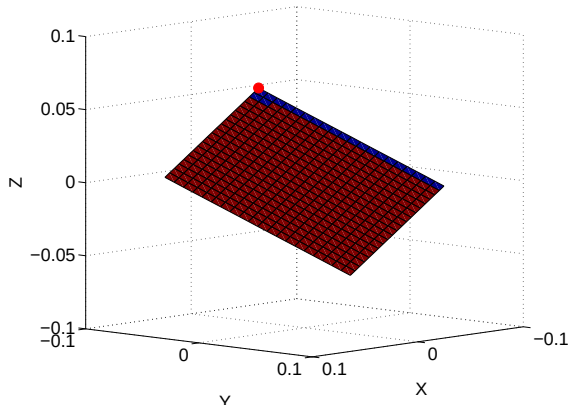
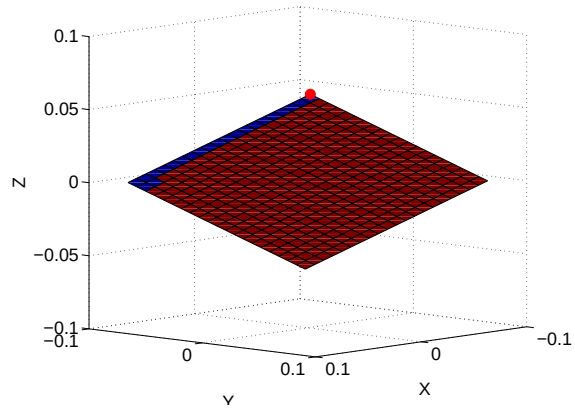


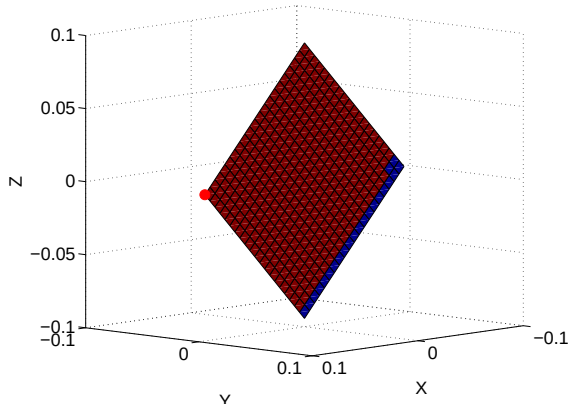
Figure C.1: Natural map



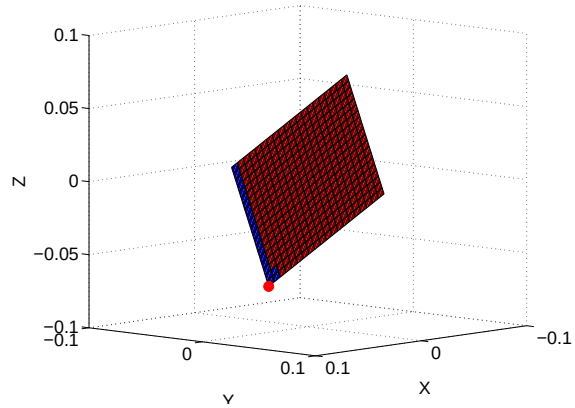
(a) Map #1



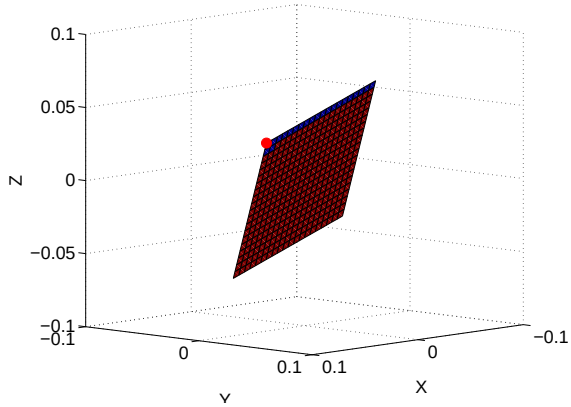
(b) Map #2



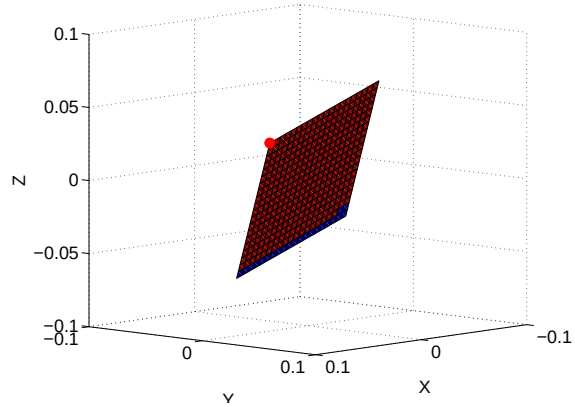
(c) Map #3



(d) Map #4



(e) Map #5



(f) Map #6

Figure C.2: Maps #1 through #6