Novel Algorithms
for Better Decoding of Neural Signals
for Intracortical Brain Computer Interfaces

Mark L. Homer
B.S. and M.S. in Mechanical Engineering
Massachusetts Institute of Technology 1995

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Division of Biology and Medicine
and School of Engineering, Biomedical Engineering
Brown University

Providence, Rhode Island
May, 2014

This dissertation by Mark Louis Homer is accepted in its present form by the Division of Biology and Medicine and the School of Engineering at Brown University as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Dr. Leigh R. Hochberg, Advisor — Date

Recommended to the Graduate Council

Dr. Michael J. Black, Thesis Committee Member — Date

Dr. Eric M. Darling, Thesis Committee Member — Date

Dr. John P. Donoghue, Thesis Committee Member — Date

Dr. Matthew T. Harrison, Thesis Committee Member — Date

Dr. Arto V. Nurmikko, Thesis Committee Member — Date

Dr. Klaus R. Müller, Reader — Date

Approved by the Graduate Council

Dr. Peter M. Weber, Dean of the Graduate School — Date

Curriculum Vitae

Mark Louis Homer

marklhomer@gmail.com

www.marklhomer.com

508.269.1011

245 Boxwood Lane

Bridgewater, MA 02324

Education

2009-present

Ph.D. Candidate, Biomedical Eng., Brown University, Providence, RI, USA

Thesis: Novel algorithms for better decoding of neural signals for intracortical brain computer interfaces

1995

B.S and M.S., Mech. Eng., Massachusetts Institute of Technology, Boston, MA, USA

Thesis: A parametric model for reducing the manufacturing cost of the traveling wave tube

Professional Experience

2006-2009

Technical Director and Project Leader, C.S. Draper Laboratory, Cambridge, MA, USA

2005-2006

Systems Engineering Group Manager, BAE Systems, Burlington, MA, USA

2000-2005

Sr. Member of Technical Staff, C.S. Draper Laboratory, Cambridge, MA, USA

1998-2000

Process & Controls Engineer, Biogen, Cambridge, MA, USA

1996-1998

Operations Analyst, Biogen, Cambridge, MA, USA

1995-1996

Decision Support Analyst, Alphatech, Burlington, MA, USA

Publications

Homer ML, Perge JA, Black MJ, Harrison MT, Cash SS, Hochberg LR, "Adaptive Offset Correction for Intracortical Brain Computer Interfaces," IEEE Transactions on Neural Systems and Rehabilitation Engineering. (epub Nov. 2013)

Homer ML, Harrison MT, Black MJ, Perge JA, Cash SS, Friehs G, Hochberg LR, "Mixing Decoded Cursor Velocity and Position from an Offline Kalman Filter Improves Cursor Control in People with Tetraplegia," Proceedings of 6th International IEEE EMBS Conference on Neural Engineering, 2013.

Homer ML, Nurmikko AV, Donoghue JP, Hochberg LR, "Sensors and Decoding for Intracortical Brain Computer Interfaces," Annual Review of Biomedical Engineering, vol. 15, 2013.

Perge JA, Homer ML, Malik WQ, Friehs G, Donoghue JP, Hochberg LR, "Intra-day Signal Instabilities Affect Decoding Performance in an Intracortical Neural Interface System,"

Journal of Neural Engineering, vol. 10, no. 3, 2013.

Homer ML, Irvine JM, Wendelken S, A Model-Based Approach to Human Identification Using ECG, Proceedings of SPIE, Optics & Photonics in Global Homeland Security V and Biometric Technology for Human Identification VI, 2009.

Homer ML, Lim S, Lopez J, Corbett M, "Machine-Aided Design of an Air Launched Missile Defense System," Proceedings of AIAA Missile Sciences Conference, 2008.

Forest F, Kessler L, Homer ML, "Design of a Human-Interactive Autonomous Flight Manager (AFM) for Crewed Lunar Landing," Proceedings of Infotech@Aerospace, 2007.

Pettit R, Homer ML, "An Autonomous Threat Evasion Response Algorithm for Unmanned Air Vehicles During Low Altitude Flight," Proceedings of AIAA 1st Intelligent Systems Technical Conference, 2004.

Homer ML, "A Quantitative Approach for Determining the Level and Sophistication of Automation in Unmanned Vehicles," Proceedings of the AUVSI, 2003.

Homer ML, "Handling Event Driven Messaging in Distributed Flight Critical Systems," Proceedings of the Digital Avionics Systems Conference, 2002.

Bertsekas DP, Homer ML, Logan DA, Patek SD, Sandell NR, "Missile Defense and Interceptor Allocation by Neuro- Dynamic Programming," IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol. 30 Issue: 1, 2000.

<u>Presentations</u>

Homer ML, Harrison MT, Perge JA, , Simeral JD, Hochberg LR, "Decoding of Target Prox-

imity Improves Estimation of Attempted Movement in People with Tetraplegia," Society for Neuroscience Annual Meeting, 2013.

Homer ML, Perge JA, Harrison MT, Black MJ, Hochberg LR, "A Method for Determining Neural Signal Contributions to Kalman Filter Based Decoding in Intracortical Brain Computer Interfaces," BMES Annual Meeting, 2012.

Homer ML, Perge JA, Harrison MT, Black MJ, Hochberg LR, "Characterizing Neural Signal Nonstationarities During Operation of an Intracortical Brain Computer Interface by People with Tetraplegia," Society for Neuroscience Annual Meeting, 2012.

Homer ML, Perge JA, Harrison MT, Black MJ, Hochberg LR, "Detecting Neural Signal Nonstationarities in Intracortical Brain Computer Interfaces Using a Model Selection Method," Sixth International Workshop on Statistical Analysis of Neuronal Data, 2012.

Homer ML, Perge JA, Hochberg LR, "Mitigating Nonstationarities During Neural Cursor Control with Noise Offset Correction," Society for Neuroscience Annual Meeting, 2011.

Parry J, Homer ML, Openshaw G, "Autonomy: How Much is Right for Future Commercial Applications?," UUVS, 2003.

Awards

2009, Levi Fellowship

2008, Top paper in Draper Technology Digest

2007, Selected to serve on innovation committee at Draper Laboratory

2004, Best paper in session at AIAA 1st Intelligent Systems Technical Conference

2004, Team recognized as top Lockheed Martin contractor

Skills

Management: personnel, software projects, supply chains, production equipment design & delivery

Algorithms: machine learning, artificial intelligence, data mining, signal processing, operations research, optimization, bioprocess control systems, vehicle guidance/trajectory/mission planning

Software: C/C++, JAVA, MATLAB, industrial control systems (PLCs & SCADAs), embedded C, HP microcontrollers

Design: intracortical brain computer interfaces, physiology of the stress response, ECG, Mars landers, rotorcraft, micro-satellites, antimissile missiles, bioreactors, supply chains

Teaching/Training

2012

Wheeler School, Middle and high school invited speaker on biomedical engineering

2011

Tissue Engineering, Teaching Assistant

2005

Mark M. Hickie, M.S., Masters Thesis Supervisor, Massachusetts Institute of Technology

Thesis: Behavioral Representation of Military Tactics for Single-Vehicle Autonomous Rotorcraft via Statecharts

2004

Ryan L. Pettit, M.S., Masters Thesis Supervisor, Massachusetts Institute of Technology

Thesis: Low Altitude Threat Evasive Trajectory Generation for Autonomous Aerial Vehicles

Selected Industry Activities

2008-2009

Principle Investigator exploring ways for people to guide unsupervised machine learning classifiers.

2008-2009

Lead developer of pattern classification software that discerns an individuals arousal state from physiological sensors. Project sponsored by the Department of Homeland Security.

2007-2008

Co-researcher on computer network intrusion detection software.

2007-2007

Technical Director of a study that explored design alternatives for the Missile Defense Agencys air launched, hit-to-kill interceptor.

2007-2007

Designed an activity scheduling routine for an autonomous micro-satellite sponsored by the

Air Force Research Laboratory.

2006-2007

Formulated a trajectory planner for NASAs Autonomous Landing and Hazard Avoidance Technology (ALHAT) program.

2006-2007

Developed a rapid path planner for an unmanned undersea vehicle for the Navy.

2005-2005

Organized and led multi-day long training seminar for Boeing developers on the systems integration of battle management software for aerial drones.

2005-2006

Personnel manager of five engineering staff members, directly responsible for hiring decisions, their work schedules, career guidance, performance reviews, and yearly incentive packages.

2005-2006

Oversaw system engineering activities during the development of battle management software for an unmanned air vehicle under the J-UCAS program sponsored by DARPA.

2003-2005 Led team to prototype command & control software for an autonomous helicopter. Effort sponsored by DARPA & the Army.

2002-2004

Evaluated the mission effectiveness of different autonomous helicopter software architectures.

2000-2002

Developed a prototype distributed avionics architecture including an embedded operating system and voting logic to achieve fault tolerance.

1998-2000

Managed the specification/ construction of pharmaceutical production equipment. Worked with contractors and production personnel to meet cost, time, and quality targets.

1998-2000

Developed software for the companys manufacturing systems. Worked with closed-loop control techniques, data acquisition, PLCs, and related hardware.

1996-1998

Formulated a strategy for senior management to minimize the company's risk of a product recall. Results implemented by CEO and his operating team.

1996-1998

Ran the companys supply chain schedule using a custom developed software tool.
Coordinated manufacturing and quality activities of Avonex and four other pharmaceutical products.

1995-1996

Constructed software using genetic algorithms and dynamic programming that positions and schedules missile defense and radar resources.

1995-1996

Used linear programming for crisis management of supply chains.

Funding Awards

2008-2009

Led internal research and development grant by Draper Laboratory for enabling a person to guide an unsupervised machine learning algorithm.

2005-2006

Co-led $4M proposal selected by the Army for autonomous mission planning and routing of drone aircraft.

2004-2005

On proposal team that won subcontract by Lockheed Martin for threat evasion software to fly on an autonomous helicopter.

2004-2005

Led internal research and development grant by Draper Laboratory for a general software architecture for automating unmanned vehicles.

2002-2003

Led internal research and development grant by Draper Laboratory for choosing an unmanned vehicles level of autonomy.

Other Interests

Practicing 5th degree black belt in Kenpo Karate & Taiho-Jitsu; Sensei, teaching children and adults

Sailing, skiing, hiking, meditation

Acknowledgements

Working towards a Ph.D. has a lot in common with an apprenticeship. An advisor leads, guides, manages, and sets a good example professionally. Training under Dr. Leigh Hochberg has been a wonderful experience and his efforts are much appreciated. His excellence and grace pushed me to improve the quality of my work. At the same time, he taught me the power of kindness even in the face of long-term, persistent struggle.

My thesis advisory committee was instrumental in insuring the research was interdisciplinary and strived to achieve results of long lasting academic benefit. Inputs from Dr. Arto Nurmikko, Dr. Matthew Harrison, Dr. Michael Black, Dr. Eric Darling, and Dr. John Donoghue are all appreciated. I would like to thank Dr. Müller for his comments after reading the thesis; they further improved the dissertation's contents and broadened my thinking.

Many past and present members of the Hochberg Lab were involved in running the clinical research sessions, development of the overall BrainGate platform, providing data preparation software, and associated administrative activities: Dr. Beata Jarosiewicz, Dr. John Simeral, Dr. János Perge, Dan Bacher, Dr. Nicholas Masse, Dr. Tomislav Milekovic, Nick Schmansky, Sergey Stavisky, Jad Saab, Brittany Sorice, Kathryn Tringale, Erin Gallivan, Anish Sarma, Beth Travers, David Rosler, and Laurie Barefoot. Dr. Gerhard Friehs, Dr. Emad Eskandar, and Dr. Sydney Cash were instrumental in implanting the microelectrode arrays in people. This work also analyzed monkey data from the Donoghue Lab, which were collected by Naveen Rao as part of his Ph.D. research with involvement from Dr. Carlos Vargas-Irwin.

Productive research typically emerges from a fertile ground of ideas and intellectual stimulation. A grateful nod is thus in order to Dr. John Donoghue, Dr. Wasim Malik, Dr. Wilson Truccolo, Michael Rule, Fabien Wagner, Naubahar Agha, Dr. Blaise Yvert, Dr. Yun Park, Dr. Shaomin Zhang, Dr. Ernest Ho, Thomas Wiecki, Yu-Ting Dingle, Dr. Diane Hoffman-Kim, Travis May, Dr. Yi-Ning Wu, Jason Pacheco, Michael Hughes, Jeff Miller, Dr. David Brandman, Jessica Feldman, Brandon King for many great conversations over the years.

I have deep appreciation for the people who are referred to as S3 and T2, for their participation in the BrainGate clinical research trial.

Special thanks to Dr. János Perge, Dr. Matthew Harrison, and Dr. Michael Black for all the research and career guidance, everything from mathematical theorems to the roots of our evolutionary history to the "Tao" of academic life - cheers!

For my family, Amy, Alexander, Benjamin, Maxwell, Hang-Ling, Rosa, Judy, Roger, Michael, Julee, Johnny, Jenn, Charlotte, Rosalie, Louis, Margaret, Roberta, and Horace, your support has meant all the world to me. :-)

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 INTRACORTICAL BRAIN COMPUTER INTERFACES

This chapter is derived from a review paper we have already published [1]. Voluntary limb movement in primates, including able-bodied people, is achieved in part by the conveyance of high level motor commands from neurons in the motor cortex and other cortical areas to neurons within the spinal cord. In response to input from these so-called upper motor neurons, as well as inputs from other descending systems and cells local to the spinal cord, alpha (lower) motor neurons then activate muscle fibers generating motion. Injury or disease, such as brainstem stroke or loss of a limb, can disconnect the cortex from its effector target. Ongoing research is focused on building technological bridges between the motor cortex and external devices in order to restore motor function. When these systems work as designed, the intent to move again becomes translated into overt action.

Variously labeled as neural interfaces, neural prostheses, brain machine interfaces (BMI) or brain computer interfaces (BCI), this technology promises to be useful to a large number of people. In the United States alone, it is estimated that 1.6 million Americans live with limb loss [2] and 5.5 million people experience some form of paralysis [3] including 50,000 individuals who have complete tetraplegia (severe functional impairment or paralysis of both arms and legs) [4]. Commonly, people with tetraplegia rely on others to perform routine, everyday actions. Responding to the need for better assistive devices, BCI technology could

provide a control signal for any range of devices: a computer, robotic or prosthetic arm, or a powered wheelchair. For a true restorative function, people suffering from paralysis of an arm might, in the future, be able to recover control by re-animating the affected muscles via brain-controlled functional electrical stimulation (FES); others with limb loss could employ a cortically-controlled robotic prosthesis that emulates arm and hand actions.

This thesis work focuses on intracortical brain computer interfaces (iBCIs) where, for example, microscale probes having 96 fine tipped microlelectrodes in a compact 4 x 4 mm array are inserted into the cortex (reviews of scalp-based electroencephalography (EEG) and brain surface-based electrocorticography (ECoG) BCIs can be found in [5] and [6] respectively). The sensors are sensitive enough to pick up the discrete all-or-none output of single neurons, the action potential, commonly referred to as a spike, as well as the summed voltage fluctuations from small to large numbers of neurons, called field potentials. Each electrode provides spiking from up to a few neurons (typically 1 or 2), yielding the population's time evolving output pattern. These represent but a small sample of the entire set of neurons in this limited region, as spiking can only be detected by microelectrodes closely approximated to a neuron.

Spikes are desired because they are the information-rich signal nearly universally employed by neurons for communication in the brain. Beginning approximately 50 years ago, experiments by Evarts and others helped to reveal how individual cortical neurons, recorded one at a time, control voluntary movement, helping to lay the groundwork for current iBCI work, e.g., [7–12]. Multielectrode approaches helped to provide [13, 14] early examples of real-time prediction of an animal's wrist position from the decoded motor cortex output [15, 16], and enabled animals to perform simple tasks via an external device [17]. The first clinical realization of these technologies occurred in 1998, when a person with amyotrophic lateral sclerosis (ALS) paralysis controlled an on-off switch using two electrodes implanted in the motor cortex [18].

Based upon the many years of publicly funded basic science, research beginning in 2000 demonstrated that monkeys could use motor cortex spiking patterns to control a computer cursor in two or three dimensions [19–23]. Soon afterwards, in 2004, the first demonstrations of intracortically-directed 2D cursor movements and simple robotic control

were accomplished by people with tetraplegia using an iBCI [24]. This was followed by iBCI-enabled 3-D control of a prosthetic arm by non-human primates [25]. Most recently, multi-dimensional robotic arm control was successfully achieved for reaching and grasping actions by people with tetraplegia [26, 27]. The future promise of brain-controlled FES devices was further demonstrated by iBCI-driven muscle control in monkeys with temporary arm paralysis [28–30], further underscoring the potential highlighted in an initial demonstration by a human with paralysis using an iBCI to control a simulated, dynamic FES system and arm [31].

Modern iBCIs have several major components, illustrated in Figure 1. The user imagines or attempts a movement, resulting in motor cortical neuronal activity. An implanted multiprobe sensor of electrodes, commonly referred to as a microelectrode array (MEA), records neural activity in the form of extracellular voltage signals. Data acquisition (DAQ) hardware transports, amplifies, and digitally samples these signals. The data is processed on a dedicated, real time hardware and software platform, which hosts various decoding routines. In the typical real-time data processing stream, neural signals first are processed to extract informative features related to intended movement, after which estimation techniques translate those features into device commands. The device then carries out the desired action as well as provides feedback to the user.

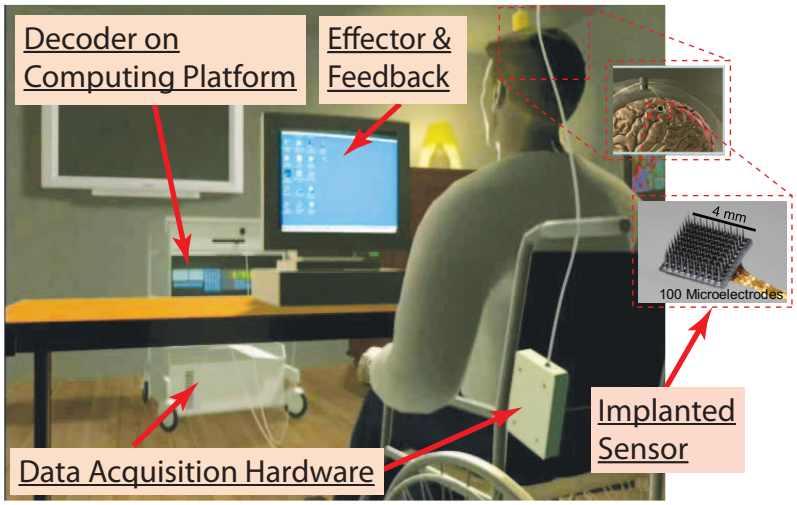

Figure 1.1: Illustration of an iBCI setup.

The core elements of an iBCI (Figure 1.1), from neural signal collection to actuation

of the effector, each require the iBCI neuroengineering team to make important decisions. Each of these elements (which could also be broken into several smaller elements) also presents opportunities for focused research and development. An engineering perspective is presented in this review, where the domain of decoding encompasses the combination of the feature extraction, estimation, and calibration elements [32]; this viewpoint is used to organize subsequent sections.

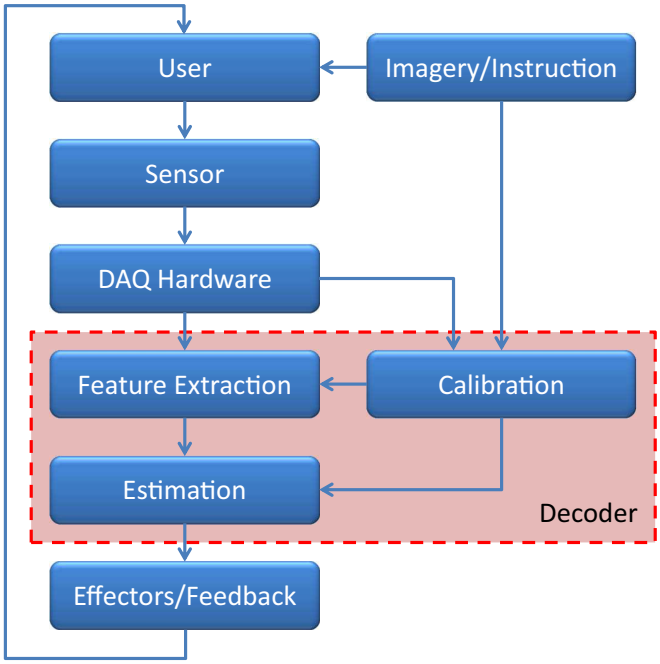

Figure 1.2: Block diagram of iBCI architecture. Arrows indicate the major flows of information, and the red-shaded box contains elements associated with decoding.

Prior to use, the BCI must be calibrated. Plainly stated, a map must be built between the observed neural activity associated with intended movement and the actual movement of the effector. First, the user (with, for example, tetraplegia due to stroke or ALS) engages in a series of instructed behavioral tasks where various real world actions, such as the preprogrammed motion of a computer cursor, are observed while imagined or attempted by the user (who is unable to make them). Next, with both the neural signals and the attempted (or, at least, instructed) movements being known quantities, calibration techniques tune the decoder. It is important to note that during this process only the decoder has been calibrated; no real training or learning has (yet) occurred from the user's standpoint. While reasonable to believe that users presented with a new and reasonably useful tool (an iBCI)

will become more proficient in its use over time [33], important questions regarding recorded signal stability as well as the potential for simultaneous co-adaptation [34] of both the decoder and the user yield the clear demonstration of such learning to future research.

A thorough review of the several active research areas highlighted in Figure 2 would extend far beyond this introduction (see [35] for several excellent chapters in this regard). Here we focus on the state-of-the-art in decoding algorithms. The emphasis is on results that have been or may become widely adopted by the iBCI research community.

1.2 NEURAL SIGNAL FEATURE EXTRACTION

1.2.1 Overview

Once the recorded signals have gone through data acquisition steps, the decoder first performs the step of neural signal feature extraction (Figure 1.3). During each operational cycle of the decoder (aka time step), a vector of values is computed from the multivariate time series sampled from the many voltage waveforms streaming in from the sensor. One classical approach involves visually inspecting extracellular voltage gradients, emanating from individual neurons and shaped by filter settings, e.g. [24]. The typically biphasic, ms-long voltage waveforms of each channel are visually inspected to manually set time and amplitude windows, converting them to discrete events in time, permitting detection of spikes and their assignment to a single neuron or a cluster of them. A unit's spikes are then time binned to compute the spike counts of each unit in each of the time bins, creating a point process time series for each channel.

1.2.2 Threshold Crossings

Sometimes, the shape and size of a unit's waveform can change significantly over time. A systematic analysis of instabilities found several cases where the combination of time-amplitude bins traditionally used in spike sorting failed to detect spikes due to drifts in their amplitude which can result from even small movements (on the order of a few tens of microns of the electrode recording site with respect to a particular neuron) [37,38]. A simpler and more robust approach may be to count the number of voltage threshold crossings within

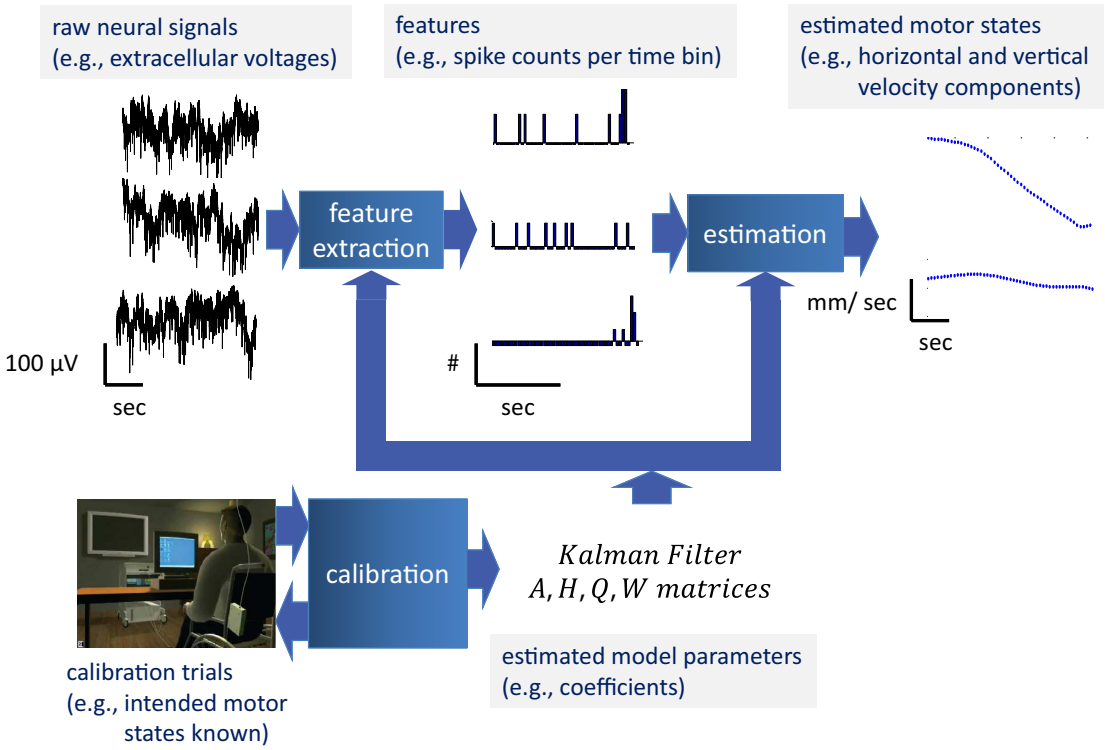

Figure 1.3: Illustration of a decoder's architecture and the flow of information passing through the various components. Matrices A, H, W, and Q represent model parameters in the Kalman filter. Specifically, A represents the motor state transition matrix, and W is the covariance matrix on the respective noise. Similarly, H captures how the neural features vary as a function of motor state (the observation matrix), and Q is the covariance matrix on the associated noise. See [36] for a detailed explanation of the Kalman filter.

a given channel and time bin [39]. Termed threshold crossings, the technique enabled no appreciable drop in decoding performance despite significant reductions in spike amplitudes [37]. Use of threshold crossings is gaining adoption with their use a key element in recent pilot clinical studies [26], but success may be sensitive to signal to noise ratio, electrode properties, the size and shape of neurons and other unknown factors.

1.2.3 Multiunit Activity

An alternative or addition to detecting spikes analyzes signal fluctuations available in the high frequency band, e.g., between 300 - 6,000 Hz. The multiunit activity (MUA) can also be used for iBCI control after passing through signal processing such as a nonlinear filter [40]. More recently, in offline decoding, non-linear filtering of MUA was reported to yield better results than either spike detection and sorting or threshold crossings [41]. 3D reach and grasp in monkeys was decoded offline by extracting power information in the 200-400 Hz frequency bands (which extends into the MUA range) [42]. In preliminary closed-loop, head-to-head comparisons by a person with tetraplegia operating an iBCI, MUA performance was found comparable to spike counts [43]. Perhaps most attractive is the evidence given of MUA's non-stationary statistics being less pronounced than other high frequency signal options [41]. Again, caution is needed because these advantages may be modified by amplifier or electrode properties.

1.2.4 Local Field Potentials

Other frequency bands within the range of signals generated by neural activity also contain information about movement intent. Unlike spikes, the primary sources for a given local field potential (LFP) signal measured in the motor cortex remains unclear. Although it has been postulated that LFPs are the result of synaptic currents, many factors such as spike afterhyperpolarizations can play a role [44]. Prominent oscillations, one example of which is the commonly referred to beta rhythm, have been linked to attention [45] and found to noticeably decrease in amplitude between movement preparation and onset [12]. In the last decade, reach direction was decoded from LFPs in monkeys in offline studies [46–48]. The features extracted varied, for example, a Gaussian kernel smoother [46] or a

multitaper spectral analysis [48]. The frequency bands studied may contain different types of information about intended movement; for example, the average amplitude within the 63-200 Hz band increased at movement onset relative to baseline, whereas it was observed to decrease between 16-42 Hz [47]. Three dimensional reach and grasp in monkeys can be decoded, offline, from intracortically-recorded low, midrange and higher frequency bands, with the quality approaching, but not surpassing, that of spikes [49]. Figure 1.4 provides an example from a person with tetraplegia with a sensor placed in the arm area of motor cortex. In the presented spectrogram, there is a decrease in amplitude during intended arm movement in the 12-40 Hz range, with both increases and decreases in power seen at other frequency bands throughout the intended movement.

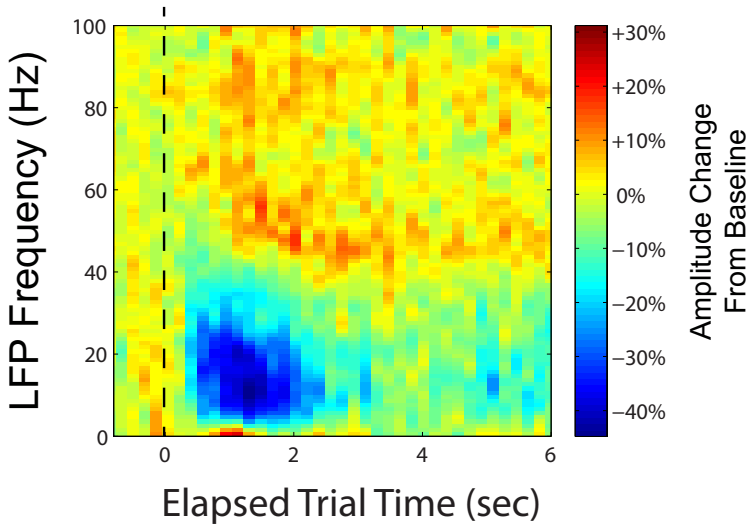

Figure 1.4: Representative spectrogram of the local field potential in one channel, averaged over 70 repeated trials. Time 0 denotes the instructed onset of movement (dashed line). Colors indicate amplitude, normalized by dividing by average baseline activity prior to movement onset (blue represents greatest reduction in activity; red represents greatest increase in activity). The data were recorded from a person with tetraplegia intending to move her own hand during repeated cursor control calibration trials. Note the appearance of at least three differing bands (approximately 0-3, 3-40, and 40-100 Hz). (Printed with permission from [38])

Using two arrays and frequency bands, 0.3-4 Hz and 48-200 Hz, classification of eight target directions was achieved to an accuracy comparable with spikes [50]. Other reports decoded natural arm reaches to predict arm endpoint motion and grip aperture [49, 51]. Despite these advances, comparisons between decoding with LFPs vs. spikes do not point

to LFPs as a clear winner [46, 49]. The real value then of the signal is thought to lie in providing more robust features, particularly over the course of time. For example, the ability to detect a spike might be susceptible to array micromotions whereas the aggregate nature of LFPs may mitigate signal changes due to tiny array movements because of their volume conduction properties. This advantage might explain one study's report of channels that had lost their spikes still retaining most of their predictive power in the LFP features [52].

An alternative use for LFPs lies in identifying discrete states linked to movement. In electrocortical studies, a person's visual attention (versus eye movement) can be classified above chance levels into one of eight directions [53]. In the posterior parietal cortex (PPC), 0-10 and 20-40 Hz bands have revealed the onset of motion as compared to the preparation phase [54]. The accomplishment is an especially significant step because the decoding was done online where the monkey used the iBCI to perform the task. Thus, LFPs could play a part in a hierarchal decoding strategy where identification of the discrete motor state dictates how to approach estimating the continuous variables.

Also, the precise features extracted from these broadband neural signals can make a substantive difference in their decoding utility. Instead of LFP amplitude, recent work is starting to reveal the role of LFP phase. Investigations into beta oscillations, for example, have found a statistical dependence between the oscillation's phase and target reach direction [55]. These observations are supported by the revealed coherence between beta oscillations and EMG recordings [56]. Spiking activity has also been linked to the phase of beta oscillation, where directional information of the former was a function of the latter [57]. These new scientific insights suggest that extracting phase information from LFPs might improve iBCI decoding by considering the influence of LFP waveform phase on modulations in spiking activity.

1.3 DECODER CALIBRATION

1.3.1 Overview

In order to customize the decoder to the user, initial filter calibration exercises engage the iBCI user. In animals, the subject is trained to perform the behavioral task, using

overt arm actions. Meanwhile, recordings of the brain activity are collected [17, 19–21]. In contrast, people with paralysis are asked to attempt arm and hand motions that would, for example, generate planar mouse trajectories and clicks matching the computer controlled cursor on the screen [24, 58]. In these calibration sessions, the recorded neural signals and any information processed from them are not presented to the user and thus are called open-loop. Similar approaches can be used for initial training for the control of multi-dimensional reach and grasp movements of a robotic or prosthetic arm [26].

The recorded neural signals and the corresponding movement information are then fed into calibration routines which set parameters within the decoder. For example, when a person with paralysis used an iBCI to control a computer cursor's motion, linear regression processed the calibration data to arrive at the needed matrices for the linear filter [24]. In the case of a Bayesian classifier, once the neural signal samples are separated into their respective classes, class conditioned distributions can be fit from the data, e.g., calculating means for independent Poisson distributions [23]. The calibration procedures usually extend to processing the features themselves. For instance, from the set of spiking units collected, only those well modulating with reach direction as shown through ANOVA were selected for inclusion into the decoder [22]. Such steps help to best fit the decoder's parameters given the calibration data while protecting against over fitting.

1.3.2 Closed-loop Calibration

The concept behind closed-loop calibration is to adjust the decoder's calibration after a closed-loop block of trials, one where the iBCI is used to carry out the task. The process has an iterative nature where the previous calibration block adjusts the decoder used in the next calibration block [20]. Supporting the approach, differences were found in neuronal activity between actual and observed movement [59, 60]. Furthermore, while observation-tuned neurons were found to be predictive of movement direction, strictly action modulating neurons did a better job. Even more recently, such tuning differences, depending upon cognitive context, have also been reported in preliminary studies in clinical trial participants with tetraplegia [61].

Closed-loop decoder calibration has been used successfully to enable control in able-

bodied monkeys [25, 62] and in people with tetraplegia (see Figure 1.5) [26, 63]. An innovation is to run an automated software routine to assist epochs of "brain control" (that is, iBCI-controlled movements, as opposed to hand-controlled movements), thereby increasing trial success. In early iterations, considerable assistance is provided, but with each iteration it lessens until the task is completely governed by brain-control. Not only is the technique thought to improve the decoder, but the user's neural activity appears to change to accommodate the iBCI. In several studies, the response characteristics of units changed as the monkey learned a task [64]. When a tuned decoder was intentionally altered while a monkey was operating an iBCI, units's firing rates changed during continued use of the iBCI [34]. In another closed-loop iBCI experiment, rats improved their task success rate without changing their overt behavior [65]. Thus, use of closed-loop calibration appears to allow both the user and the decoder to update their internal calculations in an effort to converge on stable configurations for useful closed-loop iBCI operation.

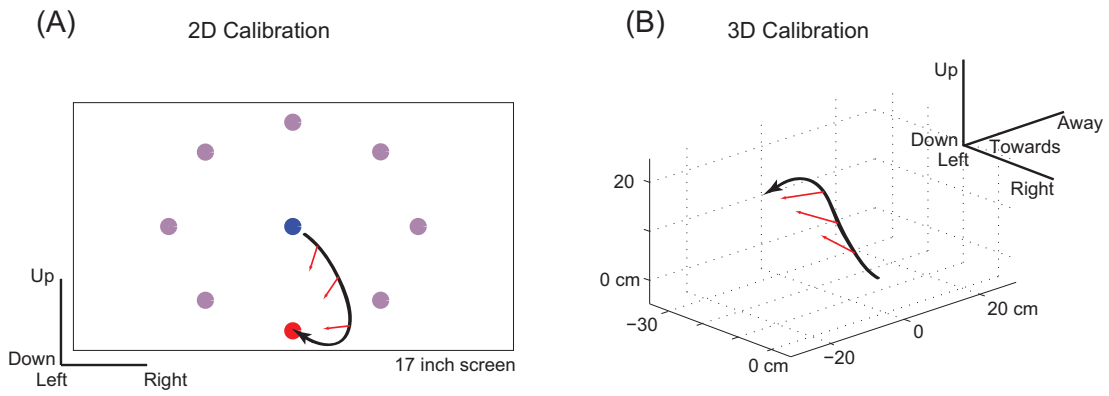

Figure 1.5: Illustration of the closed-loop calibration technique. (a) The black line shows cursor trajectory; the blue and red dots show the starting point and goal, respectively. At each sample time, the direction between the cursor and goal is calculated (red arrows). This direction is then compared with the current cursor heading in order to generate an error signal with which to adjust the decoding tuning parameters. (Printed with permission from [63].) (b) A similar technique can be applied in the 3D case. (Adapted with permission from [26].)

Sometimes, long after a stable mapping has been reached, it changes, making the initial decoding model inaccurate. Fortunately, there are mathematical frameworks to formalize possible paths of deviation and ways to fix this problem. On the efficiency side, a recursive least squares formulation can prevent one from having to reconsider all prior calibration

blocks [66]. Similarly, the problem can be cast in a Bayesian context [67]. These innovations get closer to how best to utilize and efficiently process the closed-loop calibration data to adapt the decoder.

1.3.3 Dimensionality Reduction and Regularization

Several methods seek to remove or discount one or more neural activity features collected during filter calibration in order to make the decoding more accurate. One direct way to go about this is to focus on well-isolated single unit activity (SUA). By comparing changes in waveform characteristics, spike intervals, and directional tuning, one study chose a small subset of 10-15 units from the 75-100 recorded [33]. The calibrated decoder performed well over many days (9 days in one monkey, 19 in another).

Others have employed more automated means, such as cross validation, whereby the calibration data is divided into three sets: training, validation, and assessment. First, the decoder training data is used to build models, i.e. the mathematical mapping between features and motor states. These models are then checked with the validation data set to see which performs best and finally the true test of performance occurs with the assessment data. If models under comparison differ by the features chosen, then a greedy method, akin to forward stepwise regression, can find promising, though typically not optimal, subsets [68]. This strategy yielded reductions in the number of features chosen, in some cases by a factor of four, in 3D reach and grasp in monkeys [42, 69].

Alternatively, the decoder can treat the neural signal features as being driven by a small number of latent variables, which then represent motor intent - a technique called dimensionality reduction. Gaussian Process Factor Analysis (GPFA), for example, both smooths in time as well as finds a smaller set of latent features [70]. Using data from neural ensembles, GPFA was found to improve trajectory reconstruction accuracy over standard methods. Use of dimensionality reduction in real-time decoding reduced the number of variables driving the decoder by approximately 90% in one case and 60% in another [71]. Furthermore, error rates in directional classification dropped dramatically from roughly 20 to 5%. Interestingly, such latent features may generate new theories about fundamental, computational principles of the motor cortex. In a reduced space, characteristic, repeatable

oscillations emerged, suggesting that cell populations in M1 might not encode movement variables per se, but rather serve as a pattern generator to drive downstream computations and eventually the movement itself [72].

These methods become particularly important if more advanced feature extraction methods are used. With LFP histories, the number of features can grow easily into the thousands. Given such a large data set, many of the features display a redundancy which can be taken advantage of to reject a variety of noise sources such as those of a biological nature and recording artifacts. Dimensionality reduction and regularization is one way to be robust to such disturbances and indeed has been successfully wielded to handle feature deluges of as much as 6,400 features in order to achieve stable performance in ECoG work over many days [73].

1.4 MOTOR STATE ESTIMATION

1.4.1 Overview

Given a set of features extracted from the neural signals, computations estimate the intended motor state. Decoded motor states have focused largely on those surrounding arm reaching and grasping, particularly as these functions are listed as most important to restore by people with high cervical spinal cord injury [74]. Specifically, many earlier studies investigated decoding endpoint velocity and/or position in order to control the movement of a computer cursor on a screen [19–21, 24]. For discrete selection, monkeys have chosen a reach target position/direction from a finite list of options [22, 23]. One end goal of the work is to realize the possibility of enabling someone with paralysis or loss of a limb to control a robotic aid or FES-driven limb for reaching and grasping. The relatively high level motor commands of selecting a target or controlling the endpoint trajectory could then be converted to control commands needed to drive the effector.

A standard estimation approach is the linear filter [75] and it has been used for continuous decoding, e.g., [10, 19, 20, 24, 25]. Each feature value specifies the length of a vector of fixed orientation in the motor state space. The vectors are then summed to yield the state estimate. For discrete states, Bayesian classifiers have also proven useful [22,23]. They

typically model the feature distributions conditioned on the discrete states as either Poisson or Gaussian. Both methods have the advantage of being calibrated by exact methods, fast online computation times, and relative transparency in the behavior of the tuned estimator.

1.4.2 More Accurate Estimation Models

In a head-to-head comparison, however, between linear and Kalman filtering, the latter was shown better using several standard performance metrics of cursor control [76]. Interestingly, as Figure 1.6 shows, the Kalman filter did not always yield better cursor trajectories. The algorithm performs the estimation based upon linear, multivariate Gaussian models linking the features to the motor state as well as describing the motor state dynamics. The decoder can run on the same time scale as a linear filter since the results from the linear algebra calculations quickly reach steady state [77]. Of especial note is the fact that the evaluation was done by people with tetraplegia operating an iBCI to control the motion of a computer cursor. Since then, the method has provided a reliable estimation approach, even in trials 1000 days after implantation of the sensor [58]. A variant of the technique recently enabled two people with severe paralysis to perform 3D reach and grasp tasks by way of a robotic arm aid [26], in one case >5 years after sensor implantation.

Careful evaluation of the true functional mapping showed a large fraction (35%) of the unit spike counts to be nonlinearly related to intended motor control states [78] and for this reason, some earlier attempts with iBCIs worked with nonlinear decoders [15]. Another line of research added an auto-regressive element to the feature histories [79, 80]. By combining the two types of model refinements, animals were able to operate iBCIs within an unscented Kalman filter framework, yielding improved performance [81]. Very general mappings between the features and states can be handled by particle filters [39, 82–84] or neural networks [15, 85].

Discrete motor states can also be introduced into the model. The resulting hybrid state space model switches between a series of discrete modes [86]. Computational machinery infers the state transitions during the model fitting process via methods such as expectation maximization (EM). The technique was applied to augment the Kalman filter (95) as well as a point process filter [87]. The augmented set of hidden motor states, like the nonlinear

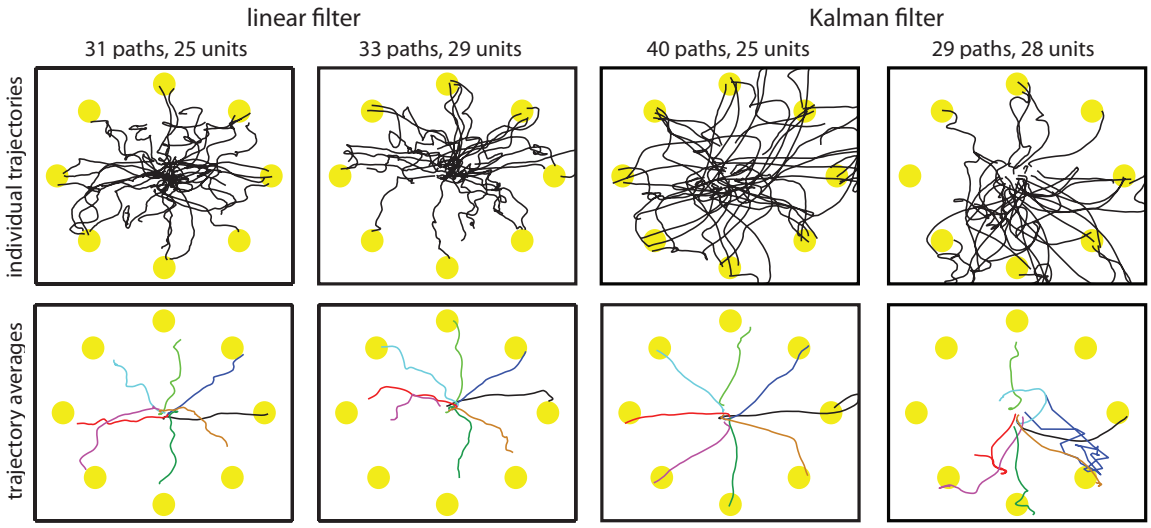

Figure 1.6: A comparison of the paths under brain control using the Kalman filter versus the linear filter. The paths were generated by a person with tetraplegia using velocity-based decoders during the eight-target, center-out task. Performance results are shown from two recording sessions in which both the linear filter and the Kalman filter were sequentially tested. The eight targets were located at 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. Units indicate the number of features employed by the respective decoding filter. (Reprinted with permission from Reference [76].)

models, free the estimator from purely linear assumptions should the calibration data show a clear nonlinear relationship. In terms of metrics such as mean squared error, use of hybrid state models have achieved reductions as high as 22% [88]. However, the results should be treated with caution as the real test lies in closed-loop, head-to-head comparisons [89].

Improving the noise modeling, another line of inquiry makes the more accurate assumption that the spike counts follow a Poisson process (rather than a Gaussian). This is the territory of point process models and they account for the exact timing of the spiking sequences [86,90]. Unlike most linear filter methods which process spike count features within, for example, 30 millisecond bins [25], point processes desire to isolate each unit firing event in time for a typical time bin of 1 millisecond [91]. Decoders running point process models improved decoding offline in animals [88, 92, 93] and in people [91]. Recently, one study successfully reported performance gains online in monkeys [94]. The cost for the improved modeling fidelity, however, is that decoding becomes more computationally intensive.

1.4.3 Choice of Motor Control States

Increasing the degrees of freedom decoded from motor cortical activity would enable more tasks to be performed with an iBCI and/or a greater degree of control. Reports of 3D reach and grasp with a robotic arm in able-bodied monkeys [25] and in people with tetraplegia [26] provide direct evidence of this and relied on expanding the state dimension from two to four (one extra spatial dimension and a variable for grasp). For more precise control, decoders could first determine whether or not the person is attempting a move and in what general direction. Initial creation of such a first stage classifier was reported in monkeys [95]. Even finer stages of motor preparation and execution can be detected with iBCIs, such as shown in macaques in which pre-trial, pre-cue, memory (remembering target location during a delay), and post-saccade phases could be readily decoded [96].

Alternatively, the measured neural activity may better associate with a different set of motor states. For example, a unit could be more tuned to wrist extension rather than general upward motion [97]. Responding to the need, researchers decoded 25 joint angles under natural reach and grasp conditions in monkeys [69] under open-loop conditions. The strategy relied on statistical dependencies between the joint angles, allowing the kinematics states to be compressed down to 10. As the 10 variables were estimated at each time step, a linear transform expanded the estimate to a vector representing all 25 joint angles. Thus, even with a relatively small sample of neurons in motor cortex, iBCIs based on motor cortical recordings should enable complex reach and grasp movements. Forces are also encoded in the stream of neural activity measurements [9, 98]. In planar manipulandum experiments, software simultaneously decoded force and velocity in offline analyses [99]. Estimating intended force (grip force, for example) in people with paralysis will be an important venture, and is likely to benefit from feedback from the effector.

Functional electric stimulation (FES), driven by an iBCI, has led to temporarily paralyzed monkeys regaining control of their muscles. In 2008, monkeys were reported to control wrist torques in closed-loop experiments [28]. More recently, five electrodes stimulated three muscles that drove wrist flexion and grip [30]. Effectively building an artificial bridge from the central nervous system to the muscles, the animals were able to pick up balls and place

them into a receptacle. For people, control of a virtual arm was shown possible by a person with tetraplegia, using a simulation that incorporates the complex, dynamic response behavior of an FES system [31]. These advancements mark key milestones towards restoring limb control to people with paralysis.

While the majority of studies in non-human primates have focused on reaching and grasping, research has also been proceeding in other types of motor control. Going beyond hand grip, individual finger movements (i.e., which finger moved) has been correctly classified better than 80% of the time [100, 101] in offline studies. Leg kinematic offline decoding studies in monkeys have also been performed [102], although full closed-loop locomotion presents interlimb coordination and balance challenges. Rats, with electrodes in the hind limb/trunk region of M1, successfully lightened a load applied to their hips, in time with their gait [103]. While neither study demonstrated closed-loop control over the entire combination of movements needed to enable walking, they do suggest a route to the more modest goals of controlling walking direction and speed as well as making any major load adjustments.

1.4.4 Adaptive Estimation

The relationship between the intended movement of a person with tetraplegia and the relatively tiny ensemble of neural signals used to deduce that intended movement is by no means expected to remain stationary. Although this relationship can be stable over time [33], both the technical factors (e.g., micromotion of the electrode with respect to the tissue, changing characteristics of recording devices) and biological ones (e.g., cortical plasticity) discussed earlier often change the mapping [38,104,105]. Recent research suggests inherent physiological processes of the brain account for the majority of the instabilities [38]. Thus, adaptive decoders have been employed to refine continually the relationship between the recorded neural signals and the expected output. For example, using Bayesian regression methods, decoding performance was maintained in the face of model drift on the time scale of days [67].

Ultimately, the cause of the nonstationarities may arise from a range of cognitive or physiological state variables. For example, spikes rates may modulate by levels of attention.

The amplitude of beta oscillations reduces upon initiation of movement [12] and the phase of the oscillation has been shown to be predictive of spike activity [57]. Thus, directly accounting for movement onset in the estimator may enable it to account for some of the nonstationarities. Similarly, incorporating eye position into the estimation process can raise iBCI performance [106]. Thus, perhaps the observed nonstationarities more reflect the incomplete nature of existing models underlying estimators than changes to the random noise corrupting the neural signal.

1.5 RECENT PROGRESS AND CHALLENGES

Research progress in decoders for iBCIs is advancing rapidly. 2006 saw the reporting of people with paralysis controlling a computer cursor and simple robotic devices. Only six years later, complex 3D reach and grasp movements of assistive robotic devices was achieved (see Figure 1.7) [26, 27]. Furthermore, as has been the case in the neural prosthesis field for 50+ years, publicly funded preclinical research continues to provide vital insights and new directions for clinical investigation focused on developing restorative neurotechnologies for people with paralysis.

At the same time, the number of decodable states is expanding [69] and other locations besides the arm are proving to be amenable to brain-control [103]. The push for additional motor states is being supported by an expansion of the list of features extracted such as those seen with LFPs [52, 73]. The developments promise to increase the capability of robotic aids or prostheses.

Just as important as the degrees of freedom is the quality of control. With the introduction of algorithms such as dimensionality reduction [71], the neural signal features are becoming less contaminated by noise. Noise free signals however will not cure errors in the model, but fortunately model sophistication is increasing as well [76, 92] and so are ways to avoid poor calibration [25, 26]. These efforts fuel the hope of seeing finer control tasks performed in the coming years.

Finally, for the technology to be clinically viable, it would ideally be reliable for very long time periods, with a span of more than a decade being a plausible target. Preliminary

reports, with one person successfully using an investigational iBCI (BrainGate) after 5 years, suggest great potential [58]. When sensor characteristics change with time or the physiological mapping between intended movement and neural activity evolves, adaptive features built into the decoder can mitigate such deviations [67]. The innovation underscores the expectation that iBCI users will soon see implants functioning well for 10 years, and frequent cases where iBCIs are used for days at a time without any recalibration.

The iBCI field is tasked simultaneously with creating better devices for fundamental neuroscience research and with creating viable clinical neurotechnologies. This latter goal requires continued advances in neuroengineering, as well as rigorous clinical evaluation. With iBCI's becoming more capable, precise, and reliable, future research will need to then bridge the gap between demonstrating a capability in a research trial and incorporating the technology into the daily lives of its users. For people who are unable to move or speak due to brainstem stroke or motor neuron disease, the impact of even modest increases in command signal output will be immense. While this and other goals for iBCIs will likely require years of additional research to become reality, recent strides make clear that such technologies are moving steadily from the realm of science fiction toward the domain of devices that support the communication, mobility, and independence of people with neurologic disease or injury.

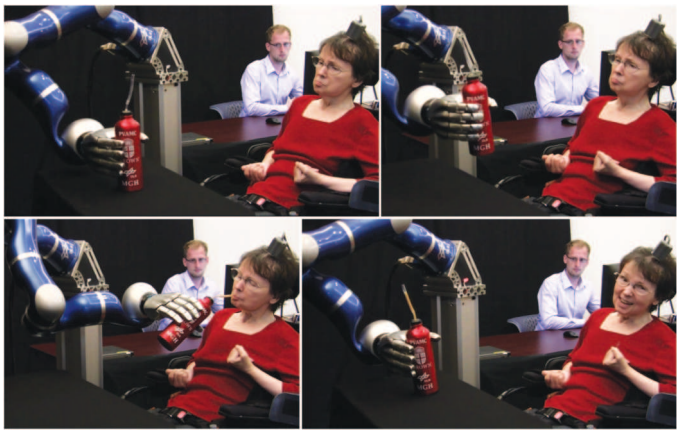

Figure 1.7: Successful iBCI-based neural control of a robotic arm, by a person with tetraplegia, to reach and grasp a thermos of coffee. The microelectrode array recording the neural data had been implanted for more than five years at the time of this achievement. (Reprinted with permission from Reference [26].)

1.6 WORK PRESENTED IN THIS THESIS

In order to address decoding reliability, Chapter 2 attempts to mitigate the nonstationary nature of the neural signals. We look at the commonly extracted feature of unit firing rates and explores how their baselines change over time. Taking advantage of the observed sparsity in the number and frequency of these changes (which we call offsets), we develop an algorithm for estimating the offsets at each time step and then feeds that to the Kalman filter. This multiple offset correction algorithm (MOCA) enables the decoding to adapt to the changing offsets. The algorithm is based upon a maximum likelihood formulation, using an L_0 norm to prevent overfitting. Significant improvement is shown in a retrospective study of people who control a computer cursor with an iBCI. The basis for Chapter 2 derives from a paper we have already published [107].

Despite such adaptation, decoding errors still may occur. If we can detect that such errors have happened, then we can mitigate their impact. The strategy was motivated by discovery of an error related potential in EEG recordings [108]. Chapter 3 thereby estimates the degree of error the decoder is making at any given time. For example, when using an iBCI to select buttons on a computer screen, a person may encounter the situation where the decoder makes a large error, driving the cursor away from the button. We hypothesize that, after the delay it takes for the person to realize the mistake, the realization is reflected in their neural activity. We hypothesize that the recognition of the decoding error can itself be decoded from informative neural signal features, specifically unit firing rates and LFP amplitudes. We explore what states are decodable with iBCI's as well as study the information content within LFP's. A decoder is dedicated to finding cases where a cursor controlled by a person is meant to be driven towards a circular goal, but in fact winds up moving away from the intended target location. An evaluation of the method's effectiveness was performed on closed-looped sessions in a person with tetraplegia.

Taking a cue from the dimensionality reduction of informative features such as spike counts, Chapter 4 considers what can be thought of as dimensionality reduction of information, not before, but after estimation. The research squarely aims to make neural decoding more accurate by combining several simultaneous estimates of the kinematics. We develop

and evaluate a method to mix velocity and position estimates generated by a Kalman filter. Here, we work with open-loop data, where people are instructed to attempt to drive a cursor along a particular trajectory. In such a situation, we can compare what was decoded to what was instructed, to get a sense for the decoding accuracy. The basis for Chapter 4 derives from a paper we have already published [109].

Chapter 5 also uses the open-loop data set in people with tetraplegia, in addition to open-loop data from monkeys. The work develops better estimation models. We challenge the assumption that the relationship between spike counts and hidden kinematic states is a linear one. The novelty is that we do not assume a particular parametric form, instead looking to the nonparametric method of Random Forests. The investigation led us to combine the Random Forest technique with the Kalman filter in a hierarchical fashion.

CHAPTER 2

Correcting Multiple Offsets

2.1 INTRODUCTION

This chapter is derived from a paper we have already published [107]. During iBCI setup, a calibration process is typically used to set the parameters of the decoding algorithm (filter). During calibration, the user attempts a series of instructed movements and the neural signals are recorded. The overt actions of trained non-human primates can be directly measured [17, 19–23]. For people with paralysis, intended actions are assumed to match those required to accomplish the directed tasks [26, 58, 76]. With the inputs and outputs of the filter known, its internal parameters then can be estimated to capture the statistical connection between the neural recordings and intended movement.

However, sometimes during iBCI operation, the neural firing activity or the relationship between the neural signal and intended movement might change. Several studies have reported this phenomenon, called nonstationarity, in which unexpected changes occur in the statistical characteristics of the neural signals [18, 21, 37, 38, 80, 104, 110–112]. For instance, in monkeys, the statistical dependency between kinematics and binned counts of detected action potentials, i.e., spike counts, changed within a few minutes [104]. 84% of units underwent statistically significant changes in mean firing rates in a retrospective analysis of 22 sessions where people operated an iBCI [38]. These neural signal instabilities, combined with a nonadaptive decoding algorithm, can introduce decoding errors, rendering the iBCI

less useful or even unusable until the decoder is recalibrated [38, 80]. shenoy2006

Electrode placement [113], electrode design [114] and how the neural features are extracted from the signals [32, 37, 41] may have an effect on signal stability. However, in one study, much of the intra-day nonstationarity could not be attributed to technical artifacts suggesting that the cause is often physiological [38]. The instabilities may arise from changes in motor or cognitive states not modeled by the decoding algorithm. While modeling more states might improve decoding performance [106], accounting for all the relevant causal factors may not be feasible.

To address this problem, iBCI operation could be halted and the decoding algorithm recalibrated, for example, by the user actively using the iBCI to complete a series of pre-set exercises. Data from this "closed-loop" variant of calibration can be combined with previous calibrations to retune the filter [20, 25–27, 62, 63, 80, 115]. The more time spent calibrating the filter, however, the less time there is available for the user to make use of the iBCI. For a fully automated and practical iBCI, such interruption of iBCI use for recalibration is not desirable.

Ideally, a decoding algorithm would autonomously and "silently" (without disturbing the user) *adapt* to the nonstationarities that occur during iBCI operation. Differing from "closed-loop" filter recalibration techniques, this capability calls for the filter to adjust its tuning parameters without knowledge of the true motor states or objectives. To this end, an adaptive linear filter [116] and point process filters [86,117,118] have been evaluated with simulated data. Dual Kalman filtering was run offline on experimental data from monkeys [119]. In both offline and online conditions, also with monkeys, an unscented Kalman filter updated its parameters via a Bayesian paradigm [67]. In the latter two approaches, the filter's parameters were constantly adjusted by assuming the prior motor state estimates were close to their true values. Adaptive methods have also been developed to deal with nonstationarities in EEG-based classification [120]. Work in the area includes decomposition techniques that assume the observed multivariate time series are a summation of stationary and nonstationary signals [121, 122].

Here, we develop an adaptive Kalman filter that adjusts an otherwise constant term, the offset vector, in the tuning model. For the first time, we present a method that can handle

large, sudden nonstationarities and demonstrate improved performance over a standard, nonadaptive Kalman filter in offline analysis of data from two people with tetraplegia using an iBCI. Underlying the technique is a novel framework that employs penalized maximum likelihood estimation to implement a **M**ultiple **O**ffset **C**orrection **A**lgorithm (hereafter referred to as MOCA) in order to adapt the Kalman filter. The approach is similar to those designed to construct robust filters in the face of unknown inputs [123, 124]. Perhaps most relevant to our approach are filters built to handle large unknown impulses that are sparse in time [125, 126]. They also perform a search over a branching set of possibilities, each a solution from closed-form calculations, to determine if and when an impulse has occurred.

This report presents the general MOCA framework as well as a specific implementation. Methods and data for evaluating MOCA follow, including simulations and offline, retrospective analysis of clinical research sessions. The results demonstrate the improvement that MOCA has over its nonadaptive counterpart.

2.2 DECODING METHODS

2.2.1 Generic Kalman Filter

The Kalman filter [36, 127] has been successfully used in iBCIs [26, 58, 76, 128, 129]. At its core, the Kalman filter assumes a Gaussian-linear model,

$$\boldsymbol{x}_0 \sim \mathcal{N}(\mu_0, P_0) \tag{2.1}$$

$$\boldsymbol{x}_k = A_k \boldsymbol{x}_{k-1} + \boldsymbol{w}_k : \quad \boldsymbol{w}_k \sim \mathcal{N}(0, W_k) \tag{2.2}$$

$$\boldsymbol{z}_k = H_k \boldsymbol{x}_k + \theta_k + \boldsymbol{q}_k : \quad \boldsymbol{q}_k \sim \mathcal{N}(0, Q_k) \tag{2.3}$$

where k is the time step, $\boldsymbol{x}$ is the $d \times 1$ motor state vector, $\boldsymbol{z}$ is the $m \times 1$ neural feature vector, all model parameters reside in the matrices A_k, H_k, W_k, Q_k, μ_0, P_0 and the offset vector θ_k, and $\mathcal{N}(\mu, \Sigma)$ is a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ. The key statistical dependency between the neural features and motor states is captured by (2.3) and is sometimes referred to as the tuning model.

At the current time step, $k = n$, the filter estimates the motor state, $\hat{\boldsymbol{x}}_n$, given the feature

history from the first time step to the present, $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n$ (or alternatively $\{\boldsymbol{z}\}_1^n$). Let $\boldsymbol{x}_k$ conditioned on $\{\boldsymbol{z}\}_1^{k-1}$ be written as $\boldsymbol{x}_k|\{\boldsymbol{z}\}_1^{k-1}$ or $\boldsymbol{x}_{k|k-1}$. Similarly, let $\boldsymbol{x}_k$ conditioned on $\{\boldsymbol{z}\}_1^k$ be written as $\boldsymbol{x}_k|\{\boldsymbol{z}\}_1^k$ or $\boldsymbol{x}_{k|k}$. Using this notation, one way to arrive at the Kalman filter is to maximize the a posteriori probability density (MAP) of $\boldsymbol{x}_{n|n}$,

$$\hat{\boldsymbol{x}}_n = \underset{\boldsymbol{x}_n}{\operatorname{argmax}} \; p\left(\boldsymbol{x}_{n|n}\right). \tag{2.4}$$

Under models (2.1-2.3), $\hat{\boldsymbol{x}}_n$ can be arrived at via recursive calculations [36, 127],

$$\begin{aligned} P_{k|k-1} &= A_k P_{k-1|k-1} A_k^T + W_k \\ \hat{\boldsymbol{x}}_{k|k-1} &= A_k \hat{\boldsymbol{x}}_{k-1} \\ K_k &= P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + Q_k\right)^{-1} \\ P_{k|k} &= \left(I - K_k H_k\right) P_{k|k-1} \\ \hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_{k|k-1} + K_k \left(\boldsymbol{z}_k - \theta_k - H_k \hat{\boldsymbol{x}}_{k|k-1}\right) \end{aligned} \tag{2.5}$$

where, for $k \in \{1, 2, ..., n\}$ as well as initializations $\hat{\boldsymbol{x}}_0 = \mu_0$ and $P_{0|0} = P_0$, we have $\boldsymbol{x}_{k|k-1} \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}\right)$ and $\boldsymbol{x}_{k|k} \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_k, P_{k|k}\right)$.

2.2.2 Standard, Nonadaptive Kalman Filter for iBCI Applications

In the general formulation (2.1-2.3), the matrices and offset vector can change between time points. However, in iBCI applications, these parameters are usually assumed constant in time,

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + \boldsymbol{w}_k : \quad \boldsymbol{w}_k \sim \mathcal{N}\left(0, W\right) \tag{2.6}$$

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \theta + \boldsymbol{q}_k : \quad \boldsymbol{q}_k \sim \mathcal{N}\left(0, Q\right). \tag{2.7}$$

Doing so simplifies the process of learning their values from calibration data. During filter calibration, the participant performs tasks where $\boldsymbol{z}_k$ is recorded and $\boldsymbol{x}_k$ is assumed known over many time steps. θ is set to the feature means over samples whose $\boldsymbol{x}$ values average out approximately to zero. Values for A and H can then be found from linear regression

techniques and the resulting residuals combined to form the covariance matrices W and Q. Details about calibrating the Kalman filter for neural decoding can be found in [128, 129].

As in the generic case, $\hat{\boldsymbol{x}}_n$ is calculated from a recursive calculation, where now all parameters are time invariant. Furthermore, K_k stabilizes over time to its steady state value, K. In the case of iBCI's, with time steps typically separated by 100 milliseconds or less, K_k quickly converges to K [77]. Considering the filter under these steady state conditions, we arrive at our standard, nonadaptive Kalman filter,

$$\hat{\boldsymbol{x}}_k = A\hat{\boldsymbol{x}}_{k-1} + K\left(\boldsymbol{z}_k - \theta - HA\hat{\boldsymbol{x}}_{k-1}\right). \tag{2.8}$$

2.2.3 General MOCA Framework

Changes in neural activity unrelated to movement can introduce decoding errors [38]. In order to approximately model the effects of such nonstationarities in our estimate for $\hat{\boldsymbol{x}}_k$, we relax the assumption of the nonadaptive Kalman filter that θ is constant in time (2.7),

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \theta_k + \boldsymbol{q}_k: \quad \boldsymbol{q}_k \sim \mathcal{N}(0, Q). \tag{2.9}$$

In effect, we go back to the generic Kalman filter model when it comes to the offset vector (2.3). The resulting, recursive Kalman filter estimate of $\boldsymbol{x}_k$ is the same as (2.8) only θ is replaced by θ_k.

The challenge then becomes choosing values for the θ_k's in order to adapt the Kalman filter. In this paper we develop a penalized maximum likelihood approach for estimating the offset vector history, $\{\theta\}_1^n$, namely,

$$\{\hat{\theta}\}_1^n = \underset{\{\theta\}_1^n}{\operatorname{argmax}}\ \ln p\left(\{\boldsymbol{z}\}_1^n | \{\theta\}_1^n\right) - \gamma\left(\{\theta\}_1^n\right) \tag{2.10}$$

$$= \underset{\{\theta\}_1^n}{\operatorname{argmin}}\ -\sum_{k=1}^{n} \ln p\left(\boldsymbol{z}_k | \{\boldsymbol{z}\}_1^{k-1}, \{\theta\}_1^n\right) + \gamma\left(\{\theta\}_1^n\right) \tag{2.11}$$

where, we define $p\left(\boldsymbol{z}_1 | \{\boldsymbol{z}\}_1^0, \{\theta\}_1^n\right) \triangleq p\left(\boldsymbol{z}_1 | \{\theta\}_1^n\right)$ for notational convenience. The penality term γ is necessary, because the standard maximum likelihood estimation severly overfits the offsets. The penalty term can also be used to introduce prior knowledge or other

constraints into the procedure (see Section 2.2.4).

It can be shown (see Appendix A) that (2.11) equates to,

$$\{\hat{\theta}\}_1^n = \underset{\{\hat{\theta}\}_1^n}{\operatorname{argmin}} \frac{1}{2}\sum_{k=1}^{n}(z_k-\nu_k)^T R_k^{-1}(z_k-\nu_k) + \gamma\left(\{\theta\}_1^n\right): \tag{2.12}$$

$$\begin{aligned}
\text{for } k=1 \quad \nu_k &= HA\mu_0 + \theta_k \\
R_k &= H(AP_0A^T + W)H^T + Q \\
\text{for } k>1 \quad \nu_k &= HA\hat{\boldsymbol{x}}_{k-1} + \theta_k \\
R_k &= H(AP_{k-1|k-1}A^T + W)H^T + Q \\
\hat{\boldsymbol{x}}_{k-1} &= A\hat{\boldsymbol{x}}_{k-2} + K\left(\boldsymbol{z}_{k-1} - \theta_{k-1} - HA\hat{\boldsymbol{x}}_{k-2}\right)
\end{aligned}$$

where each $p\left(\boldsymbol{z}_k|\{\boldsymbol{z}\}_1^{k-1},\{\theta\}_1^n\right)$ is $\mathcal{N}(\nu_k,R_k)$, $P_{k|k}$ is computed from (2.5), $\hat{\boldsymbol{x}}_{k-1}$ is now the prior state estimate generated by a Kalman filter under observations for $\{\boldsymbol{z}\}_1^{k-1}$ and a proposed realization of $\{\theta\}_1^n$, and terms not depending upon $\{\theta\}_1^n$ have been dropped [130].

Once $\{\hat{\theta}\}_1^n$ is attained, we can run the Kalman filter, subtracting the estimated offsets from the feature vectors to arrive at an estimate for the current motor state via the recursive calculation,

$$\hat{\boldsymbol{x}}_k = A\hat{\boldsymbol{x}}_{k-1} + K\left(\boldsymbol{z}_k - \hat{\theta}_k - HA\hat{\boldsymbol{x}}_{k-1}\right). \tag{2.13}$$

Under this general strategy, a new θ history is estimated using (2.12) at every time step. Thus, $\hat{\theta}_k$ is not found recursively, but rather it is computed at each time step using the full history $\{\boldsymbol{z}\}_1^n$. Since the homogeneous part of (2.13) is the same as the original nonadaptive Kalman filter, MOCA inherits the stability properties of the nonadaptive Kalman filter. [131].

2.2.4 MOCA Implementation

Constraints and Choice of Penalty for $\{\theta\}_1^n$

We constrain our search over possible θ histories and choose a penalty both to reflect our observation that neural activity is typically stationary but, when it changes, there is a change

in the average level of only a few neural features. This is embodied in the assumption that changes in offsets are *sparse* (e.g. few neurons are affected) and that changes in time occur rarely. The approach differs from incorporating the offsets into the hidden state, where the state transition model is linear and Gaussian. Choosing linear and Gaussian dynamics allows use of the dual kalman filter [132]. However, we find our modeling choices of sparsity and sudden shifts to better reflect the nonstationarties occurring on timescales less than a minute.

The approach also simplifies estimation of the offsets. We assume the offsets stay at their calibrated values, θ, until, at some point in the recent past, they may or may not have made a step change to a new constant value. Let the possible shift occur at $k = n - \tau$ and $\xi \subset \{1, 2, ..., m\}$ contain the indices of the offsets that have shifted. Additionally, let ϕ denote the amount each offset has shifted at $k = n - \tau$. The relationship between the θ_k's, θ, τ, ξ, and ϕ, is described by,

$$\begin{aligned} \theta_k &= \theta + \phi \quad \text{if } k \in \{n-\tau, n-\tau+1, ..., n\} \\ &= \theta \qquad \text{otherwise :} \\ &\text{where } \phi_i = 0 \text{ if } i \notin \xi. \end{aligned} \tag{2.14}$$

For our implementation, τ was set to 5 seconds.

To capture our belief that a few nonstationary offsets are more likely than many at any given point in time, we choose an L_0 penalty,

$$\gamma\left(\{\theta\}_1^n\right) = \ell \tag{2.15}$$

where ℓ is the number of nonstationary offsets. The L_0 penalty causes (2.12) to be equivalent to minimizing the Akaike information criterion [133]. For example, suppose there were 32 neural features, and we assumed only 5 of those features had nonstationary offsets (5 nonzero entries in ϕ). Then by (2.15), $\gamma = 5$.

Other penalization choices include the L_2 and L_1 norm. In order to support our specific choice of L_0 over these more standard options, we set aside the first ten test cases from

sessions with S3 for validation (see Section 2.4). Parameter sweeps were performed for the constant weights applied to the L_2 and L_1 norms. The validation confirmed the L_0 norm as the best choice, but marginally so (1.5% better than the second best).

Search Strategy for Choosing which Offsets are Nonstationary

Now that permissible values for $\{\theta\}_1^n$ are completely defined by ξ and ϕ, we can recast (2.12) as a two step minimization process, where first we propose which offsets are nonstationary and then choose the extent to which they have shifted,

$$\mathcal{E}(\xi, \phi') \triangleq \min_{\xi} \left[\min_{\boldsymbol{\phi'}} \ \frac{1}{2} \sum_{k=1}^{n} (z_k - \nu_k)^T R_k^{-1} (z_k - \nu_k) \right] + \ell \tag{2.16}$$

where ϕ' contains all the nonstationary components of ϕ. For example, if $m = 4$ and $\xi = \{2, 4\}$, then $\phi' = [\phi_2 \ \phi_4]^T$. We can express ϕ in terms of ϕ',

$$\phi = B\phi' \tag{2.17}$$

where B is an m by m identity matrix with columns associated with assumed stationary offset components; that is. $i : i \notin \xi$, removed. For example,

$$\text{suppose } m = 4, \ \xi = \{2, 4\} \text{ and } \phi' = [2.2 \ \ -3.8]^T$$

$$\text{then } B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{and } \phi = [0 \ 2.2 \ 0 \ \ -3.8]^T.$$

Because a complete search over all possible combinations for ξ is intractable, we instead implement the suboptimal strategy of forward stepwise search (FSS) akin to that used in linear regression under similar circumstances [134]. The algorithm initializes ξ to the null set, meaning no offsets have changed. During the first step, we propose adding one of the

```
procedure MOCA
    ξ̂ ← ∅                                  % Initialize nonstationary set to empty
    φ̂′ ← ∅                                 % Start with no offset corrections
    compute E given φ̂′, ξ̂                  % Compute the resulting score
    while ξ̂ ≠ all offsets do               % Advance stepwise
        for all i ∉ ξ̂ do                   % Consider all stationary offsets
            ξ ← ξ̂ ∪ i                      % Propose adding i to ξ̂
            estimate φ′ given ξ
            compute E given φ′, ξ
        if no ξ improves E then
            break
        else
            ξ̂, φ̂′ ← φ′, ξ with best E
    compute x̂_k given φ̂′, ξ̂                % Run the Kalman filter
    return x̂_k
```

Figure 2.1: Overall schematic of the multiple offset correction algorithm (MOCA). At each step in the procedure, an additional offset index, i, is added to the nonstationary set, ξ. The decision is based on improving an objective function, $\mathcal{E}$, tied to maximizing the likelihood. The process terminates when either the step fails to improve $\mathcal{E}$ or all offsets are considered nonstationary. The resulting estimates for the nonstationary offsets, $\hat{\phi}'$, are then used to correct the Kalman filter.

offsets to ξ. For each of the m candidates, the procedure estimates ϕ' (see below) and subsequently computes $\mathcal{E}$. FSS then adds the candidate with the lowest $\mathcal{E}$ to ξ. The search continues, adding only one offset component per step. The process terminates when $\mathcal{E}$ cannot be improved by adding an offset to ξ or ξ contains all offsets (see Figure 2.1).

Estimation of Nonstationary Offsets

For each ξ proposed by the FSS, an optimal ϕ' must be chosen to evaluate $\mathcal{E}$. We can derive an approximate, closed form solution for the inner minimization in (2.16) to yield an estimate for ϕ' (See Appendix B),

$$\hat{\phi}' = \left(\sum_{k=n-\tau}^{n} F_k^T R^{-1} F_k \right)^{-1} \left(\sum_{k=n-\tau}^{n} F_k^T R^{-1} y_k \right) : \tag{2.18}$$

$$F_k \triangleq -HA \left(\sum_{j=n-\tau}^{k-1} S^{j-(n-\tau)} \right) KB + B$$

$$y_k \triangleq z_k - HA\hat{\boldsymbol{x}}_{k-1}^{(0)} - \theta$$

where $S \triangleq (I - KH)A$, $\hat{\boldsymbol{x}}_{k-1}^{(0)}$ is the Kalman filter estimate at time step k under the default assumption that none of the offsets have changed; i.e., $\phi = 0$, and R is a steady-state version of R_k. The latter exists because, as K_k quickly achieves the steady state value of K, then by the iterative equations above in (2.5) and (2.12), R_k also reaches R. Due to (2.18), $\hat{\theta}_k$ is bounded as long as the y_k's are bounded. This fact leads to bounded errors in estimating $\boldsymbol{x}_k$, assuming the original nonadaptive Kalman filter is uniformly asymptotically stable (see Appendix C).

2.3 SIMULATION DATA

Before describing experiments with real iBCI data, we first illustrate the behavior of MOCA using a controlled scenario with simulated neural data.

2.3.1 Cursor Motion Data

In order to generate the synthetic data, the velocities of arm reaching movements performed by an able-bodied person were recorded and fed into a neural activity simulator. We constructed a 2D radial-4 center-out-and-back cursor game using the MATLAB® software platform (Figure 2.2). During each trial, the cursor was moved to within a highlighted, circular target. The game was played for 60 seconds and cursor velocities were sampled at a rate of 10 Hz.

2.3.2 Simulated Neural Activity

The simulator contained a tuning model of the form (2.9) with 32 features, i.e. $n_z = 32$. For each feature, one movement direction, on average, generated the maximum level of activity. We distributed these preferred directions equally over the entire 360 degree range. The model was constructed so that the feature values (mean subtracted firing rates) varied between ± 10 Hz. The Gaussian noise for each feature was independent of the others and their variances were all set to 10 Hz.

The simulator was run in two modes: stationary and nonstationary. Under the scenario with nonstationary offsets, the 5 features most tuned to a rightward velocity had their offsets

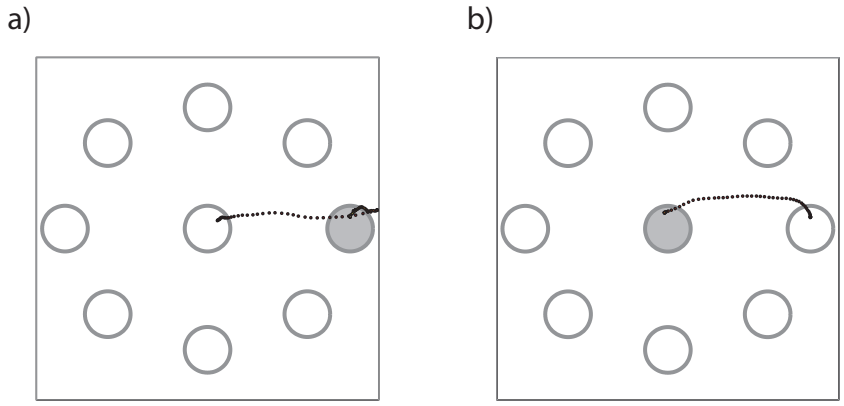

Figure 2.2: Description of the 2D radial-4 center-out-and-back cursor game used to compare the filters. **a)** The objective is to move the cursor (trajectory represented by the black dotted trace) to a randomly chosen, peripheral target (filled in with gray) and then **b)** move it back to the center target.

increased by 40 Hz above their calibrated values throughout the entire simulation. This 40 Hz value was selected to generate a velocity bias that appeared similar in magnitude to the velocity biases occasionally observed in iBCI use.

2.3.3 Evaluation Procedure

Both MOCA and the standard, nonadaptive Kalman filter were run on the synthetic neural activity to estimate the cursor velocity. The models in each filter exactly matched those used by the simulator, except the 40 Hz offset increase when the simulation was run in nonstationary mode. Besides comparing traces of the estimated and true velocities, the mean absolute difference (μAD) between them was computed for both the horizontal and vertical components.

2.4 OFFLINE CLINICAL RESEARCH DATA

2.4.1 Study Participants

Two people with tetraplegia and anarthria due to a pontine stroke enrolled in clinical research studies where they actively controlled a computer cursor via the BrainGate2 iBCI[1] to reach circular targets [58]. One participant in the study, hereafter referred to as S3, is a

[1]CAUTION: Investigational Device. Limited by Federal Law to Investigational Use.

woman who enrolled in the study at age 54. Participant T2, is a 65 year old man. Prior to the sessions examined here, S3 had been working with the iBCI for over 26 months; T2's prior experience was just 2 months.

2.4.2 Implant

The implant consisted of a 10 $\times$ 10 microelectrode array (Blackrock Microsystems, Salt Lake City) inserted into the motor cortex. The hand-arm region [135] was determined by pre-operative MRI. The electrodes penetrated 1.5 mm deep in both S3 and T2 and were spaced 400 μm apart. Since four electrodes were non-active by design, a total of 96 voltage signals were recorded by the array. The array was connected via insulated gold wires to a titanium pedestal that was fastened to the skull. The pedestal functioned as a percutaneous connector through which the voltage signals were transmitted, via a cable, to amplifiers, analog-to-digital converters, as well as signal processing hardware and software on a nearby cart (see [58] for details).

2.4.3 Sessions

Participants operated the iBCI to control a cursor on a computer screen. The trials consisted of either radial-4 center-out-and-back tasks, radial-8 center-out-and-back tasks, or a random step tracking Fitts metric task. While the details of the tasks have been detailed previously [58], we briefly review the key aspects. The purpose of each trial was to move the cursor through the 2D space of the screen to reach the currently highlighted circular target and either dwell on the target for a predefined time period or select it akin to clicking with a computer mouse. For radial-4 and radial-8 variants, the targets were located at the screen's center as well as equally distributed along the periphery. Adhering to a center-out-and-back paradigm, trial sequencing first presented a peripheral target followed by the center target (Figure 2.2). For the Fitts variant, the targets were presented at random locations on the screen and the size of the target randomly varied, but otherwise the tasks followed the same design as the radial-4 and radial-8 tasks.

We analyzed 20 and 22 blocks of trials from sessions with S3 and T2 respectively. Sessions explored ranged over 22 months (792 to 1447 days after implant) for S3, whereas T2

sessions spanned 8 months (68 to 270 days after implant). We chose sessions to encompass a wide range of cursor control performance.

2.4.4 Pre-Filter Signal Processing

After exiting the pedestal, all 96 voltages were immediately passed through a cable to signal processing hardware (Blackrock Microsystems, Salt Lake City). The signals then underwent amplification, an analog bandpass (0.3 Hz to 7.5 kHz), digitization (30 kHz), and highpass filtering (250 Hz). Either time and amplitude windows were used to manually sort spiking units [24] or threshold crossings were employed [26, 39, 105]. For each unit, the number of spikes in nonoverlapping 100 ms time bins were counted. Each component of the feature vector corresponded to the count of a particular unit's spike or threshold crossing.

2.4.5 Evaluation Procedure

Both MOCA and the standard, nonadaptive Kalman filter were run offline using the recorded neural activity features to estimate the intended velocity, separate from any decoding that was done during the session to control the cursor. However, unlike the simulated data case, we did not know the true intended velocity; that is, we did not know the ground truth of how the person intended to move during each sample interval. We did, however, assume the person intended to move towards the target. Thus, we had a plausible value for the velocity's direction, but not its magnitude. At every time step, we computed the angular error, e, between the target direction and the direction of the velocity estimated by the filters. The mean angular error (MAE) across all time steps, $k \in \{1, 2, ..., n_s\}$, in a trial block was used to measure decoding error; that is, $\text{MAE} = \frac{1}{n_s} \sum_{k=1}^{n_s} |e_k|$. Under this metric, the average MAE over all test cases under the nonadaptive Kalman filter was 87.7 degrees. During the sessions, cursor control appeared much better than the measured error because the error tended to fluctuate rapidly between negative and positive values.

Table 2.1: Distribution of corrections made by MOCA during three different situations. Because all features had an average modulation of ±10 Hz, the sizes of the false corrections made to the unperturbed offsets were relatively small.

Simulation mode	stationary sim.	nonstationary sim.	
Offset type	all 32 units unperturbed 0 Hz	27 units unperturbed 0 Hz	5 units perturbed +40 Hz
correction < -2 Hz [a]	0.00%	0.00%	0.00%
$-2 <$ correction < 0 Hz	2.24%	0.05%	0.00%
correction $= 0$ Hz	95.43%	99.93%	0.00%
$0 <$ correction $<= 3$ Hz	2.33%	0.02%	0.00%
$3 <$ correction $<= 38$ Hz	0.00%	0.00%	0.00%
$38 <$ correction $<= 39$ Hz	0.00%	0.00%	1.96%
$39 <$ correction $<= 40$ Hz	0.00%	0.00%	52.42%
$40 <$ correction $<= 41$ Hz	0.00%	0.00%	42.65%
$41 <$ correction $<= 43$ Hz	0.00%	0.00%	2.97%
$43 <$ correction Hz	0.00%	0.00%	0.00%

[a] The table shows the frequency of corrections, binned by value, reported as a percentage.

2.5 RESULTS

2.5.1 Simulation Tests

In both stationary and nonstationary simulation mode, MOCA's offset corrections were recorded over time. When the simulation was in stationary offset mode, meaning the offsets did not depart from their calibrated values, MOCA's corrections were infrequent and minor (Table 2.1). On average, only 1.46 ± 0.57 offsets out of 32 offsets were corrected at any given time. For those offsets that were corrected, the mean correction was relatively small (0.75 ± 0.44 Hz compared to the average modulation range of 20 Hz).

Under nonstationary conditions, MOCA corrected for all five of the increased offsets (Table 2.1 and Figure 2.3). Throughout the simulation, the estimate was close to the true value of 40 Hz in all five nonstationary cases. MOCA made very few corrections to the stationary offsets (0.02 ± 0.13 offsets out of 27 stationary offsets) on average and the

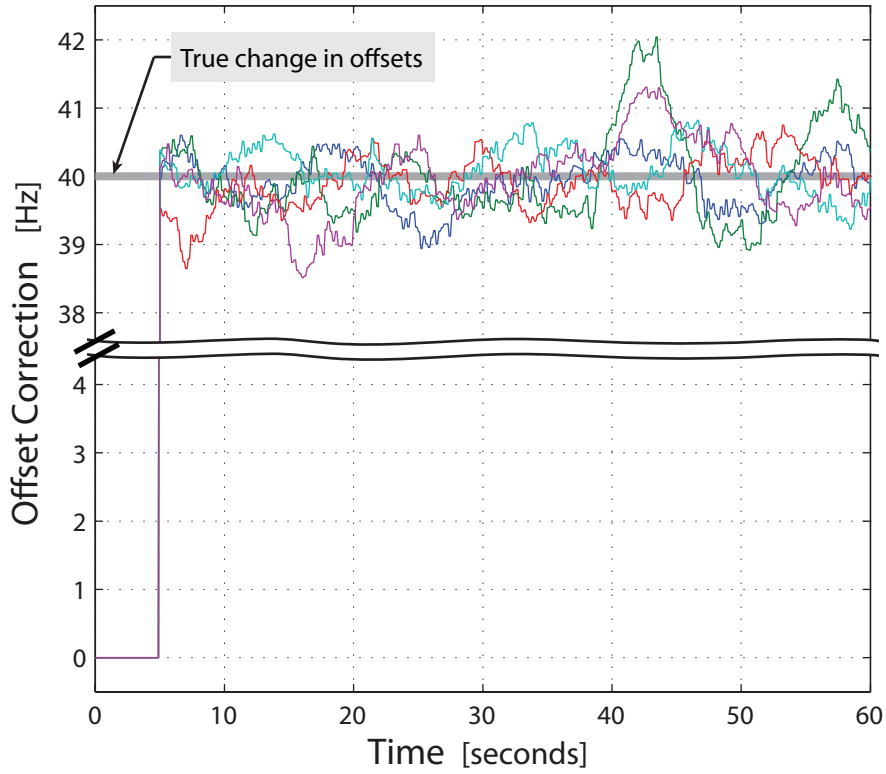

Figure 2.3: MOCA's offset corrections while the simulation ran in nonstationary mode. The colored traces indicate the corrections to 5 offsets which departed from their calibrated values by 40 Hz (highlighted by the gray line). The vertical axis between 4.5 and 37.5 Hz was removed to zoom in on areas of interest. Because MOCA must wait for a history of features to become available, it does not attempt to estimate offsets during the first 5 seconds of the simulation.

associated values were relatively inconsequential (0.03 ± 0.23 Hz vs. an average modulation rate of 20 Hz) (Table 2.1). In addition, MOCA rapidly adapted to the offset changes. After collecting a sufficient amount of neural activity history ($\tau = 5$ seconds), MOCA immediately detected the 40 Hz offset change (estimates at 5 seconds: 39.52, 39.97, 40.14, 40.34, and 40.38 Hz).

In terms of performance, MOCA was similar to the nonadaptive filter during stationary simulation mode, i.e. when no offsets were perturbed. Specifically, the filters produced similar estimates of the velocity. As measured by the mean absolute error between the estimated and actual values, their performance was within 1% of each other.

However, when 5 offsets were increased by 40 Hz from their calibrated values, the velocities estimated by the filters differed substantially. Specifically, while the nonadaptive

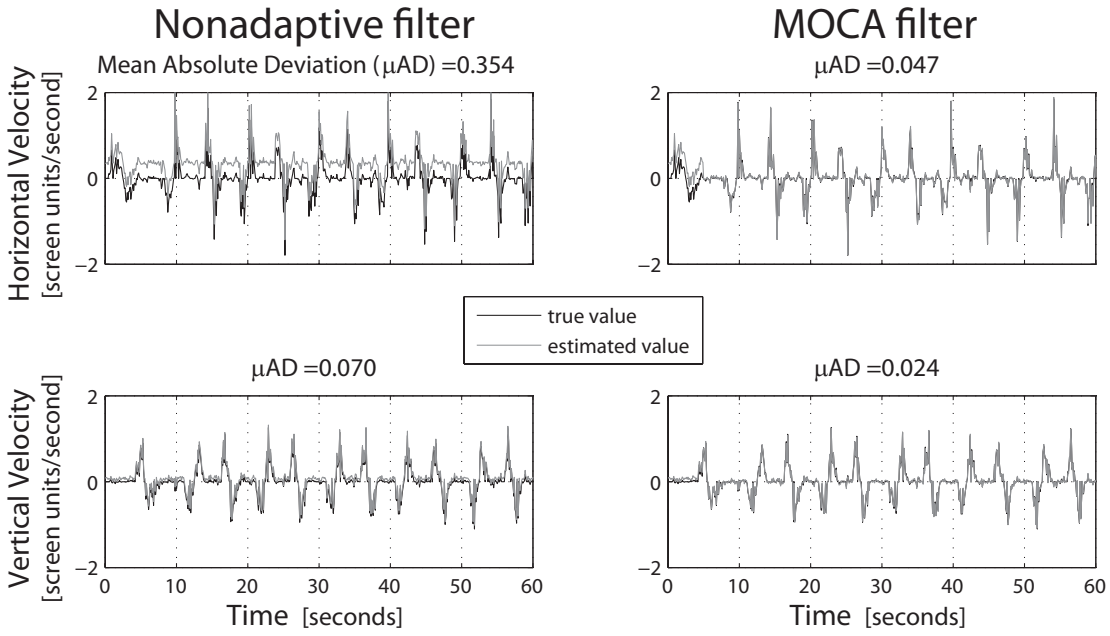

Figure 2.4: Simulation traces comparing the two filters while in nonstationary mode. Black traces denote true cursor velocity over time. Similarly, gray traces provide the respective filter's estimate of the cursor velocity. Mean absolute deviations, μAD, between the true and estimated cursor velocity along each component summarize the overall estimation accuracy. The simulation conditions cover the case where 5 of the offsets were increased from their calibrated values, requiring the decoder to adapt.

decoder's estimates revealed an error bias, MOCA mitigated that bias as shown in Figure 2.4. The mean absolute deviation dropped by nearly an order of magnitude, from 0.354 to 0.047 screen units per second in the horizontal direction under the nonadaptive filter and MOCA respectively. In the vertical dimension, the deviation fell from 0.070 to 0.024 screen units per second (both the width and height of the game area was 1.2 screen units).

2.5.2 Clinical Research Session Tests

MOCA was compared to the nonadaptive Kalman filter using sessions where two individuals operated the investigational BrainGate iBCI (32 test cases). All filters were run offline to generate velocity estimates from the feature histories in a retrospective fashion.

Use of MOCA reduced the mean angular error by 5.1 ± 9.6 degrees ($p < 0.05$, pairwise t-test) (Figure 2.5). MOCA had its greatest beneficial impact in trial blocks where the nonadaptive filter gave relatively large error and did not significantly affect cases where the nonadaptive filter reported relatively low error. Specifically, when the nonadaptive filter

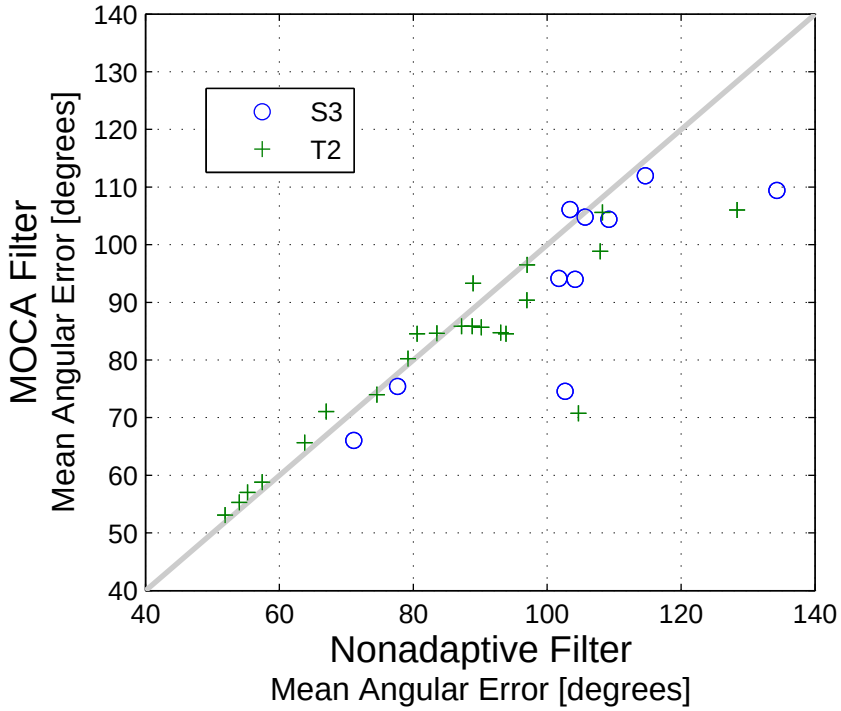

Figure 2.5: Offline comparison between MOCA and the standard, nonadaptive Kalman filter. Each data point in the plot represents a trial block (S3 designated with blue circles and T2 with green crosses). The horizontal position of each data point denotes the mean angular error from the standard filter whereas the vertical position gives the same metric when applying the MOCA variant. The light grey line is provided for reference as data points below the line indicate a lower error under the adaptive decoder.

provided an error above 90 degrees, MOCA yielded an average reduction of $10.2 \pm 10.6\%$ ($p< 0.05$, pairwise t-test). Below 90 degrees, MOCA may have slightly increased error 0.7 ± 2.7 degrees, but the findings were not statistically significant ($p= 0.36$, pairwise t-test).

On average, across all trial blocks, MOCA corrected $19 \pm 12\%$ of the offsets. The fraction of corrected offsets positively correlated with the nonadaptive filter's performance error (Pearson's correlation coefficient = 0.54; $p< 0.05$ t-test). Among those offsets corrected, the level of correction was relatively large. Normalized to the associated feature's standard deviation of modulation, the average was 0.47 ± 0.18, nearly half the level of modulation. When compared to the nonadaptive filter's performance error, no statistically significant correlation was found (Pearson's correlation coefficient = 0.07; p-value = 0.67, t-test). One might assume that as the error of the standard filter increases it is due to more nonstationary

channels. This in turn would suggest more offsets corrected by MOCA. Taken together, the results suggest that MOCA applied a higher degree of correction to the more nonstationary blocks of trials.

MOCA's ability to rapidly adjust modeling parameters differentiates it from other published adaptive filtering methods applied to iBCI recordings. For example, Li, et. al. applied Bayesian regression with a joint formulation to update parameters every two minutes [136]. We tested the Bayesian regression technique on the offline clinical data set and found the best performance to occur when the update interval was reduced to 30 seconds. Despite the optimization, Bayesian regression did not significantly reduce error compared to the nonadaptive filter (2.3 ± 9.6 degrees; p-value $= 0.19$).

2.6 DISCUSSION

In this offline analysis, MOCA provided an improvement in decoding performance when compared with the nonadaptive Kalman filter, resulting in increased robustness to the decoding of nonstationary neural signals. When control was relatively poor with the nonadaptive filter, MOCA often provided reduced error. During relatively good control under the nonadaptive Kalman filter, MOCA demonstrated no significant positive or negative impact. These findings are consistent with the hypothesis that occasionally iBCI control is degraded by nonstationarities that can be approximately modeled by offsets to the baseline firing rate. In these cases, MOCA successfully adapted the filter parameters and mitigated the effects of the nonstationarities. Furthermore, by virtue of the algorithm's design, the adaptation took place in a few seconds.

The offline nature of the analysis presents limitations, but also may lead to an underestimate of the benefits. A decoder's online performance can vary greatly from its offline counterpart [89]. Online, the person controlling the iBCI can try to correct for perceived errors generated by the decoding scheme. We anticipate that the adaptive method developed here would mitigate the nonstationarities sufficiently such that real time corrections made by the iBCI operator would result in a marked improvement in task performance.

MOCA was developed to address a common observation that when a person operates

an iBCI, systematic changes in feature offsets can occur [38]. It is postulated that there are many other types of nonstationary behavior, such as unexpected changes in tuning direction. Modification and expansion of the presented adaptive filtering framework could be used to handle these other cases.

The full space of possible penalty terms has yet to be explored. Specifically, as presented, the L_0 norm is not multiplied by a tunable weighting constant. One can even search more broadly and consider the whole family of penalty L_p terms, where p lies between 0 and ∞.

Finally, MOCA's search to explore which offsets are nonstationary is a partial one. Computationally, it becomes intractable to score all possible subsets of the features (aka powerset) as there are 2^m combinations. Better search strategies or more extensive searches combined with much larger computational power might yield even greater improvement. For example, the small and occasional erroneously assigned corrections by MOCA might be eliminated by backward stepwise search.

MOCA - the presented, adaptive version of the Kalman filter - improved offline decoding performance on average in sessions where two people controlled an iBCI (21 out of 32 test cases reported reduced error). Furthermore, the improvements were the greatest when the effect of nonstationarities were large. The methods are derived from a penalized maximum likelihood formulation, which serves as a formal way to model nonstationarities as offsets and adds an adaptive mechanism to the Kalman filter, already a popular choice for decoding in iBCIs. Thus, beyond the immediate usefulness of the instantiated MOCA algorithm, the MOCA framework provides a foundation for constructing other offset correction algorithms in iBCIs. MOCA may also be useful in applications, beyond iBCIs, when a Kalman filter is applied in the context of nonstationary data. In such cases, the technique can quickly adapt the filter solely from the observations' history.

APPENDIX A

We seek to show that (2.11) equates to (2.12). Examining (2.11),

$$\{\hat{\theta}\}_1^n = \underset{\{\theta\}_1^n}{\operatorname{argmin}} \; -\sum_{k=1}^{n} \ln p\left(\boldsymbol{z}_k | \{\boldsymbol{z}\}_1^{k-1}, \{\boldsymbol{\theta}\}_1^n\right) + \gamma\left(\{\theta\}_1^n\right) \tag{2.19}$$

we see that $\ln p\left(\boldsymbol{z}_k|\{\boldsymbol{z}\}_1^{k-1},\{\theta\}_1^n\right)$ requires evaluation. Let $\boldsymbol{x}_{k|k-1} \triangleq \boldsymbol{x}_k|\{\boldsymbol{z}\}_1^{k-1},\{\theta\}_1^n$ and $\boldsymbol{z}_{k|k-1} \triangleq \boldsymbol{z}_k|\{\boldsymbol{z}\}_1^{k-1},\{\theta\}_1^n$. As mentioned in Section 2.2.1, the current motor state given all prior neural features and the entire offset history follows a multivariate Gaussian distribution, i.e. $\boldsymbol{x}_{k|k-1} \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}\right)$, where $\hat{\boldsymbol{x}}_{k|k-1}$ and $P_{k|k-1}$ can be produced from the recursive Kalman filter calculations (2.5) with $\hat{\boldsymbol{x}}_{1|0} = A\mu_0$ and $P_{1|0} = AP_{0|0}A^T$. Since $\boldsymbol{q}_k \sim \mathcal{N}(0,Q)$ and $\boldsymbol{z}_{k|k-1}$ is a linear function of $\boldsymbol{q}_k$ and $\boldsymbol{x}_{k|k-1}$, i.e., $\boldsymbol{z}_{k|k-1} = H\boldsymbol{x}_{k|k-1} + \boldsymbol{q}_k$, then $p\left(\boldsymbol{z}_{k|k-1}\right) = \mathcal{N}(\nu_k, R_k)$, where $\nu_k = HA\hat{\boldsymbol{x}}_{k-1} + \theta_k$ and $R_k = H(AP_{k-1|k-1}A^T + W)H^T + Q$.

Inserting $\mathcal{N}(\nu_k, R_k)$ into $\ln p\left(\boldsymbol{z}_k|\{\boldsymbol{z}\}_1^{k-1},\{\theta\}_1^n\right)$ yields,

$$\ln p\left(\boldsymbol{z}_k|\{\boldsymbol{z}\}_1^{k-1},\{\theta\}_1^n\right) = \ln\left[(2\pi)^{-m/2}\left|R_k\right|^{-1/2} e^{-\frac{1}{2}(z_k-\nu_k)^T R_k^{-1}(z_k-\nu_k)}\right] = -\frac{1}{2}\left(z_k-\nu_k\right)^T R_k^{-1}\left(z_k-\nu_k\right) + C \quad (2.20)$$

where C is a constant with respect to $\{\theta\}_1^n$.

APPENDIX B

We seek to prove the inner minimization of ϕ' in (2.16) is solved approximately by (2.18). Because we assume a single step change in the offsets at $k = n - \tau$, summation terms in (2.16) associated with $k < n - \tau$ are constant with respect to ϕ' and thus can be removed from the optimization,

$$\hat{\phi}' = \underset{\phi'}{\operatorname{argmin}} \ \frac{1}{2}\sum_{k=n-\tau}^{n}(z_k-\nu_k)^T R_k^{-1}(z_k-\nu_k). \quad (2.21)$$

Next, from its definition, we note that R_k is not a function of ϕ'. Additionally, since K_k quickly achieves the steady state value of K, then by the iterative equations above in (2.5), R_k also reaches a steady state matrix we label as R. Thus, R^{-1} is a precomputed constant,

$$\hat{\phi}' \approx \underset{\phi_*}{\operatorname{argmin}} \frac{1}{2}\sum_{k=n-\tau}^{n}\left(z_k-\nu_k\right)^T R^{-1}\left(z_k-\nu_k\right). \quad (2.22)$$

Thus, $\boldsymbol{\phi}'$ only enters into the objective function through the ν_k's, both directly and implicitly through $\hat{\boldsymbol{x}}_{k-1}$,

$$\nu_k = HA\hat{\boldsymbol{x}}_{k-1} + \theta + B\phi'. \tag{2.23}$$

In order to express each $\hat{\boldsymbol{x}}_{k-1}$ in terms of ϕ', we first rearrange the recursive equation for $\hat{\boldsymbol{x}}_k$ in (2.5),

$$\begin{aligned} \hat{\boldsymbol{x}}_k &= S\hat{\boldsymbol{x}}_{k-1} + K\left(\boldsymbol{z}_k - \theta\right) - KB\boldsymbol{\phi}' : \\ S &\triangleq (I - KH)A. \end{aligned} \tag{2.24}$$

We then can calculate the first few iterations for $\hat{\boldsymbol{x}}_k$ in terms of ϕ',

$$\begin{aligned} \hat{\boldsymbol{x}}_{n-\tau} &= S\hat{\boldsymbol{x}}_{n-\tau-1} + K\left(\boldsymbol{z}_{n-\tau} - \theta\right) - KB\phi' \\ &= \hat{\boldsymbol{x}}^{(0)}_{n-\tau} - KB\phi' : \\ &\quad \hat{\boldsymbol{x}}^{(0)}_{n-\tau} \triangleq S\hat{\boldsymbol{x}}_{n-\tau-1} + K\left(\boldsymbol{z}_{n-\tau} - \theta\right) \\ \hat{\boldsymbol{x}}_{n-\tau+1} &= S\hat{\boldsymbol{x}}_{n-\tau} + K\left(\boldsymbol{z}_{n-\tau+1} - \theta\right) - KB\phi' \\ &= \hat{\boldsymbol{x}}^{(0)}_{n-\tau+1} - SKB\phi' - KB\phi' : \\ &\quad \hat{\boldsymbol{x}}^{(0)}_{n-\tau+1} \triangleq S\hat{\boldsymbol{x}}^{(0)}_{n-\tau} + K\left(\boldsymbol{z}_{n-\tau+1} - \theta\right) \\ \hat{\boldsymbol{x}}_{n-\tau+2} &= S\hat{\boldsymbol{x}}_{n-\tau+1} + K\left(\boldsymbol{z}_{n-\tau+2} - \theta\right) - KB\phi' \\ &= \hat{\boldsymbol{x}}^{(0)}_{n-\tau+1} - (S^2 + S + I)KB\phi' : \\ &\quad \hat{\boldsymbol{x}}^{(0)}_{n-\tau+2} \triangleq S\hat{\boldsymbol{x}}^{(0)}_{n-\tau+1} + K\left(\boldsymbol{z}_{n-\tau+2} - \theta\right) \end{aligned}$$

which leads us to the general relation,

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}^{(0)}_k - \left(\sum_{j=n-\tau}^{k} S^{j-(n-\tau)}\right) KB\phi' \forall \;\; n - \tau \leq k \leq n \tag{2.25}$$

where we note that $\hat{\boldsymbol{x}}^{(0)}_k$ is the Kalman filter estimate at time step k when $\phi = 0$, that is the default assumption that none of the offsets have changed.

Substituting this finding into the definition for ν_k, we have,

$$\nu_k = HA\hat{\boldsymbol{x}}_{k-1}^{(0)} - HA\left(\sum_{j=n-\tau-1}^{k-1} S^{j-(n-\tau)}\right) KB\phi' + \theta + B\phi'. \tag{2.26}$$

In order to further simplify, we again aggregate terms,

$$y_k \triangleq \boldsymbol{z}_k - HA\boldsymbol{x}_{k-1}^{(0)} - \theta \tag{2.27}$$

$$F_k \triangleq -HA\left(\sum_{j=n-\tau}^{k-1} S^{j-(n-\tau)}\right) KB + B, \tag{2.28}$$

in order to arrive at,

$$\hat{\phi}' \approx \underset{\phi'}{\operatorname{argmin}} \frac{1}{2} \sum_{k=n-\tau}^{n} \left(y_k - F_k\phi'\right)^T R^{-1} \left(y_k - F_k\phi'\right). \tag{2.29}$$

To minimize the objective function, $G(\phi') \triangleq \frac{1}{2}\sum_{k=n-\tau}^{n} \left(y_k - F_k\nu_k\right)^T R^{-1} \left(y_k - F_k\nu_k\right)$, we compute the gradient and find the critical point,

$$\nabla G(\hat{\phi}') = \sum_{k=n-\tau}^{n} F_k^T R^{-1} F_k \hat{\phi}' - F_k^T R^{(-1)} y_k = 0 \tag{2.30}$$

$$\hat{\phi}' \approx \left(\sum_{k=n-\tau}^{n} F_k^T R^{-1} F_k\right)^{-1} \left(\sum_{k=n-\tau}^{n} F_k^T R^{-1} y_k\right). \tag{2.31}$$

We know the critical point is a minimum because the second derivatives form the Hessian matrix, $\sum_{k=n-\tau}^{n} F_k^T R^{-1} F_k$ where each term is nonnegative definite, since R is positive definite.

APPENDIX C

To better understand error bounds, we note that (2.18) implies errors in estimating θ_k are bounded, if the y_k's are bounded. Let $\Delta\theta$ denote the error bound on the θ_k's. We can then compare the state estimate under a perfect estimate for the offsets, $\hat{\boldsymbol{x}}_k^{perfect}$, to one where the offset error is at its maximum bound, $\hat{\boldsymbol{x}}_k^{error}$,

$$
\begin{aligned}
\hat{\boldsymbol{x}}_k^{perfect} &= S\hat{\boldsymbol{x}}_{k-1}^{perfect} + Kz_k - K\theta_k \quad \forall k > 0 \\
\hat{\boldsymbol{x}}_k^{error} &= S\hat{\boldsymbol{x}}_{k-1}^{error} + Kz_k - K\theta_k - K\Delta\theta \quad \forall k > 0 \\
e_k &\triangleq \hat{\boldsymbol{x}}_k^{perfect} - \hat{\boldsymbol{x}}_k^{error} \\
e_k &= Se_{k-1} + K\Delta\theta \quad \forall k > 0
\end{aligned} \tag{2.32}
$$

Since $\hat{\boldsymbol{x}}_0^{perfect} = \hat{\boldsymbol{x}}_0^{error} = \mu_0$, we have,

$$
\begin{aligned}
e_1 &= +K\Delta\theta \\
e_2 &= +SK\Delta\theta + K\Delta\theta \\
&\ldots \\
e_k &= \left(\sum_{j=0}^{k-1} S^j K \right) \Delta\theta
\end{aligned} \tag{2.33}
$$

As $k \to \infty$, $e_k \to \sum_{j=0}^{\infty} S^j K\Delta\theta$. Assuming the original nonadaptive filter is uniformly asymptotically stable, then $||S^j||$ exponentially decays as a function of j [131]. By the ratio test for the series then, e_k is bounded as $k \to \infty$.

CHAPTER 3

Estimating Decoding Error

3.1 INTRODUCTION

Throughout the past two decades, various aspects of movement have been decoded from neural signals recorded by iBCIs, including position [19, 21, 24] and velocity [21, 26, 76] of the hand during reaching, force applied by the wrist [98], finger motion [100], as well as walking gait [102]. However, the technology can decode not only kinematic or kinectic variables during movement, but also cognitive states surrounding the action. Firing rates of spikes were found to be predictive of attentional state, specifically whether or not a monkey attended to a reaching task with a manipulandum [84]. In EEG, the error potential can be found during operation of a binary choice communication task [108]. In another experiment, a person's reaction to image discrimination errors was detected, resulting in >20% improvement from online error correction [137]. Closer to the source of the neural activity, ECOG signals revealed the occurrence of two types of movement related errors, with the ability to discriminate between them (>80% accuracy) [138].

Similarly, we consider the case of a person with tetraplegia using an iBCI to control a computer cursor, and investigate whether errors in cursor control can be detected from the neural signals. The cursor movements seek to reach objects on a screen and the error of interest is when the cursor undesirably moves away from the current target. Since cursor motions can range from going straight towards the target to directly away from

it, the work looks at the degree of error, a continuous variable. In six sessions with a participant, informative features from both spiking activity and local field potentials (LFPs) were extracted from the neural signals and employed to estimate the degree of decoding error. We hypothesized that the iBCI user's perception of the error generated a neural activity response. Unfortunately, the accuracy of the estimates, as measured by Pearson's correlation coefficient, were sufficiently weak that error estimates, based upon cursor motion alone, produced slightly better correlations. Thus, we could not determine whether the correlation was derived from the person's perception of the error or the intended movements he used to counter the error.

3.2 NEURAL RECORDINGS AND SESSION DATA

3.2.1 Recordings

Details of the neural recordings and session task have been reported elsewhere [26, 58]. Briefly, a microelectrode array was surgically implanted in the hand region [135], as identified by pre-operative MRI, into an individual with tetraplegia and anarthia as a result of a brain stem stroke. Hereafter referred to as T2, he underwent surgery at age 65. The sessions are part of the BrainGate clinical research trial[1]

The implant consisted of 96 1.5 mm length platinun tipped, silicone electrodes (Blackrock Microsystems, Salt Lake City). The electrodes were arranged in a 10x10 grid and spaced 400 μm apart (the four corners being inactive electrodes). Insulated platinum/gold lead wires connected the array to a percutaneous pedestal. Being fixed to the skull, the pedestal functioned as a connector from which the signals were sent through a cable to signal processing hardware and software.

3.2.2 Session Task

Three clinical research sessions were studied with T2, with two blocks of contiguous trials examined per session, for a total of six blocks used in the evaluation. The experiments were conducted on different days ranging from 67 to 82 days after implantation. During each

[1]CAUTION: Investigational Device. Limited by Federal Law to Investigational Use.

trial within a block, the participant attempted to control a computer cursor, via the iBCI, to reach one of several circular goals on the screen. One goal resided at the screen's center, while four others appeared at peripheral locations (up, down, left, right relative to the target) (Figure 1). At the beginning of a block of trials, the cursor appeared on the center target and a pseudo-randomly selected peripheral target was highlighted in red indicating it as the active target. For the subsequent trial, the center goal became the one to reach with the cursor. This center-out-and-back paradigm continued, where the game insured all peripheral goals were selected in approximately equal number. Trial durations ranged from 1.9 to 15 seconds (9.1 seconds on average). If after a set time limit of 15 seconds, the participant could not reach the goal and dwell within it for fixed amount of time (0.6 to 0.3 seconds), the trial ended and new one began. Within a session, the trials examined were contiguous, with no more than a 0.6 second delay in between them. The setup resulted in anywhere from 189.8 to 298.3 seconds of recording and 17 to 31 trials per block.

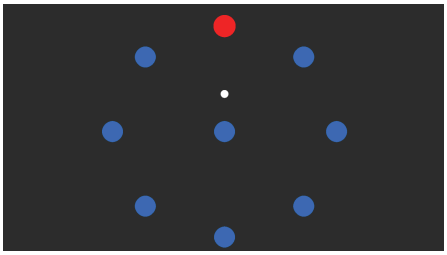

Figure 3.1: Cursor control task. The participant controls the circular cursor (white disk) through the iBCI and attempts to move it to one of five circular targets (the diagonal blue discs are displayed, but not activated as targets). In the picture, the active target is in red and the inactive ones are in blue. Colors have been modified for print clarity.

3.2.3 Closed-Loop Estimation Routine

In order to drive the cursor with his neural activity, T2 worked through a Kalman filter [76, 127, 129]. Extracellular action potentials were detected using a voltage threshold and either sorted using time-amplitude windows or directly passed to a process, where the number of spikes in 100 millisecond non-overlapping intervals was produced. The spike counts thus served as the informative features from the neural signals and were harnessed to estimate the instantaneous cursor velocity. At each session's beginning, the filter was calibrated

by instructing T2 to observe a moving cursor on the screen and attempt movements as if operating a computer mouse that would produce the observed movements. With the instructed cursor motion known and the spiking activity recorded, linear regression provided estimates of the various model parameters inherent in the Kalman filter.

3.2.4 Neurally Controlled Cursor Motions, Game Data, & Error Metric

The neurally controlled cursor trajectories were recorded at the same 100 millisecond rate as the Kalman filter. Additionally, the start and stop times of each trial were logged along with target positions. For the entire data set under analysis, a total of 12,309 time points were recorded. Notably not included in the data was the time and degree of decoding errors made during the trial blocks. The missing information arises, because while the cursor movements commanded by the decoder are provided, the history of desired movements by T2 during the game is unknown.

In lieu of this knowledge, we posit an error metric which assumes the person always wishes to get closer to the goal and any cursor controlled by the decoder that moves away from the goal has a significant degree of error. In fact, we hypothesize that the faster the cursor moves away from the goal, the more the person's neural signals modulate due to the error. These perceptions are assumed to result from an integrate and delay process. Specifically, we assume the participant compares the cursor's present distance from the goal to the same distance a half-a-second before. Furthermore, we assume the perceptual process itself takes half-a-second. The half-a-second assumption was based upon observed reaction times in psychophysic experiments [139]. Thus, the definition for our error metric at time step k, η_k, is

$$\eta_k = (r_{k-\tau} - r_{r-2\tau})/\tau, \tag{3.1}$$

where τ is 500 msec and r_k is the cursor's distance from the goal at time step k (see Figure 3.2 for two examples).

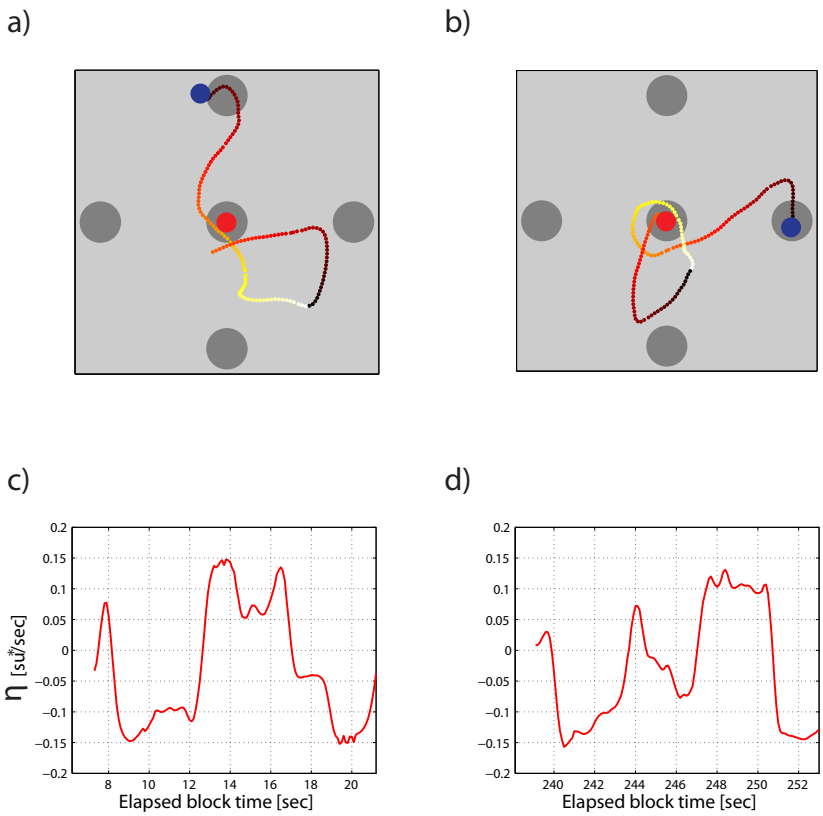

Figure 3.2: Example cursor trajectories and traces of their respective error metric. The trajectories are shown in a) and b) where, in each case, the cursor started the trial in the location denoted by the blue disk and the person attempted to control the cursor to reach and then dwell on the center target (red circle in its center). The cursor trajectory's color cycles through black, red, orange, yellow, and white in order to make it easier to follow its course. The error metric, η, for the case in a) and b) are shown in c) and d) respectively. *Distances are in screen units (su) where the screen is 1.2 su wide by 1.2 su tall.

3.3 ERROR ESTIMATION METHODS

3.3.1 Feature Extraction

The signal processing included 0.3 Hz to 7.5 kHz analog bandpass filtering, 30 kHz analog-to-digital conversion, and 250 Hz digital highpass filtering. Within the 96 filtered, digitized signals, either time-amplitude windows or threshold crossings were setup up to detect putative extracellular actions potentials. Whether the identified spikes were sourced from a single cell or the synchronous activity of many was not clear, and thus we refer to the detected events as unit spikes. We further processed them by counting their number in 300 msec time bins. A spike count vector was created for each time step during the game, with the 300 msec window centered about each time step ($\pm$ 150 msec). Thus, for each time bin, feature extraction yielded a vector of unit spike counts. The number of components in this informative feature vector ranged from 23 to 90, depending upon the session, as some electrodes had more than one unit and others lacked spikes altogether.

Additionally, we extracted normalized signal power within several frequency bands of the local field potentials (LFP) recorded on the 96 channels. The mean signal, using all 96 channels, was subtracted from each 30 kHz sampled, broadband signal and each was digitally low-band pass filtered (250 Hz cutoff) and decimated to 500 samples per second. Next, a short-time fast Fourier transform (SFFT) computed the spectral amplitude in 300 millisecond time windows (40% overlap). Since spectral amplitude scales inversely with frequency, we normalized the amplitude by z-scoring their values over time. Within each window, we averaged the amplitude in four frequency bands: 1-7, 10-40, 45-65, and 70-200 Hz. Because of the 300 msec time window, capturing the power in the lower frequencies can be problematic. In response, we also attempted to measure low frequency amplitude by applying a second order Savitzky-Golay (SG) filter with a smoothing window of 250 msec. As a result, the LFPs produced five features per electrode, four SFFT derived magnitudes and one SG smoothed value, for a total of 480 LFP derived features. Finally, the LFP features were linearly interpolated, in order to determine the feature values during each 100 msec time step of the game.

3.3.2 Regression Approach

Estimation of the error metric was based upon linear regression. Let z_k be the $m \times 1$ feature vector at time step k, then the estimate $\hat{\eta}_k$ is computed as a weighted sum of the components in z_k plus a constant,

$$\hat{\eta}_k = \beta^T z_k + \beta_0, \tag{3.2}$$

where β is a $m \times 1$ vector and β_0 and a constant.

3.3.3 Calibration of the Regression Approach

In order to apply linear regression, we needed to estimate the β's. A standard approach involves calculating the pseudo-inverse with the goal of minimizing the least squared error. However, when using the LFP features, the number of coefficients was relatively large compared to the amount of data used to fit them (481 features vs. 1,519 to 2,387 calibration data points during any given evaluation). Thus, to help prevent overfitting, we employed the lasso method [134, 140], which can be found in the MATLAB® toolbox. Instead of minimizing just the least squared error, lasso also includes a penalty term into the objective function,

$$\frac{1}{2n} \sum_{k=1}^{n} \left(\eta_k - \left(\beta^T z_k + \beta_0\right)\right)^2 + \lambda \sum_{i=1}^{m} |\beta_i| , \tag{3.3}$$

where n is the number of data points and λ is a tuning parameter. Lasso has a tendency to drive some coefficients in β to zero. The enforced sparsity can be viewed as a form of feature selection.

Lambda is selected in the following manner. First, the minimum λ which sets $\beta = 0$ is found; it is the maximum value for lambda considered. Next, 100 possible values for λ are created, descending geometrically, such that the smallest value is 1×10^{-4} times less than the largest. The best candidate is the one that minimizes mean squared error under 5-fold cross-validation within the calibration data (the test data is still held out).

3.4 RESULTS

3.4.1 Informativeness of Spike and LFP Based Features

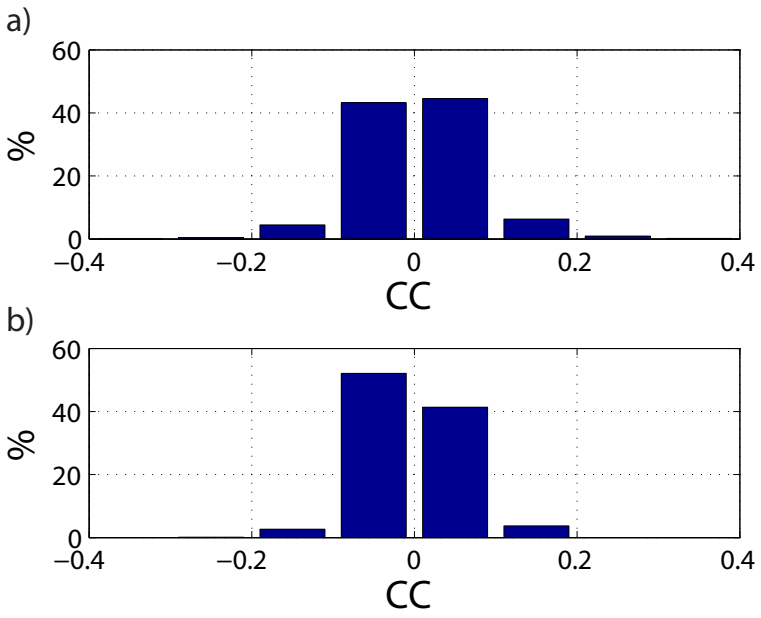

Figure 3.3: Histograms indicating the distribution of statistical association between the error metric and individual a) spike and b) LFP based features. Pearson's correlation coefficient (CC) is the chosen measure of statistical association. Counts are divided by the total number of features considered across all sessions (539 for spikes and 2880 for LFP) to obtain percentages.

Figure 3.3 presents a summary, across the 6 sessions, of how the different spike and LFP features covaried with the error metric, as measured by Pearson's correlation coefficient (CC). The median absolute value for spike derived features was CC=0.038 with upper and lower quartiles of 0.016 and 0.069 respectively. The absolute values of CC's of LFP derived features were in a similar range (median, lower quartile, and upper quartile were 0.017, 0.032, and 0.053 respectively). Out of the 539 spike based features extracted, 8 features (1.5%) were found to have CC's greater than 0.2 or less than -0.2. For LFPs, despite the larger number of features extracted (2880), only 1 feature fell into this category (0.03%). Figure 3.4 gives the firing rates of a representative subset of the units as compared to the error metric during the trial displayed in Figure 3.2a. An example of a typical LFP SFFT based spectrogram and the SG smoothed signal is provided in Figure 3.5 during the same

trial. The results suggest that, individually, the extracted features tend to have a low level of predictive power.

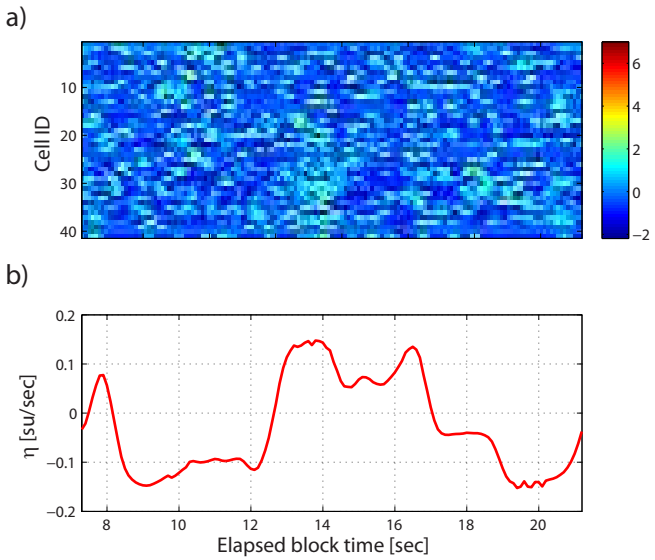

Figure 3.4: Example of spike rate activity and its covariance with the error metric. a) presents the z-scored spike rates of 41 representative units over the trial described in Figure 3.2a. Over the same time, b) gives the trace of the error metric.

3.4.2 Accuracy of the Estimator

Figure 3.6 compares actual traces of the error metric to those estimated from lasso calibrated linear regression, using either spike based or LFP based features. The traces correspond to the two trials described in Figure 3.2. For the traces in Figure 3.6a), the CC's between the actual and estimated spike and LFP based traces were 0.35 and 0.33 respectively. Figure 3.6b) resulted in lower values (spike based CC = 0.11; LFP based CC = 0.25). In both plots, the LFP based estimates appear to fluctuate more rapidly than their spike based counterparts. We posit this is due to the larger number of features typically incorporated into the LFP based estimator compared to the spike based one (309 vs. 78 in Figure 3.6). For both types of features, however, the estimator's ability to track the error metric is relatively low (e.g. coefficient of determination, $r^2, < 0.2$).

The low accuracy seen in Figure 3.6 was the tendency across all sessions. For each of the six blocks, we partitioned the trials into five folds and proceeded, within each block, to hold out one fold as test data while calibrating from the other four. The procedure

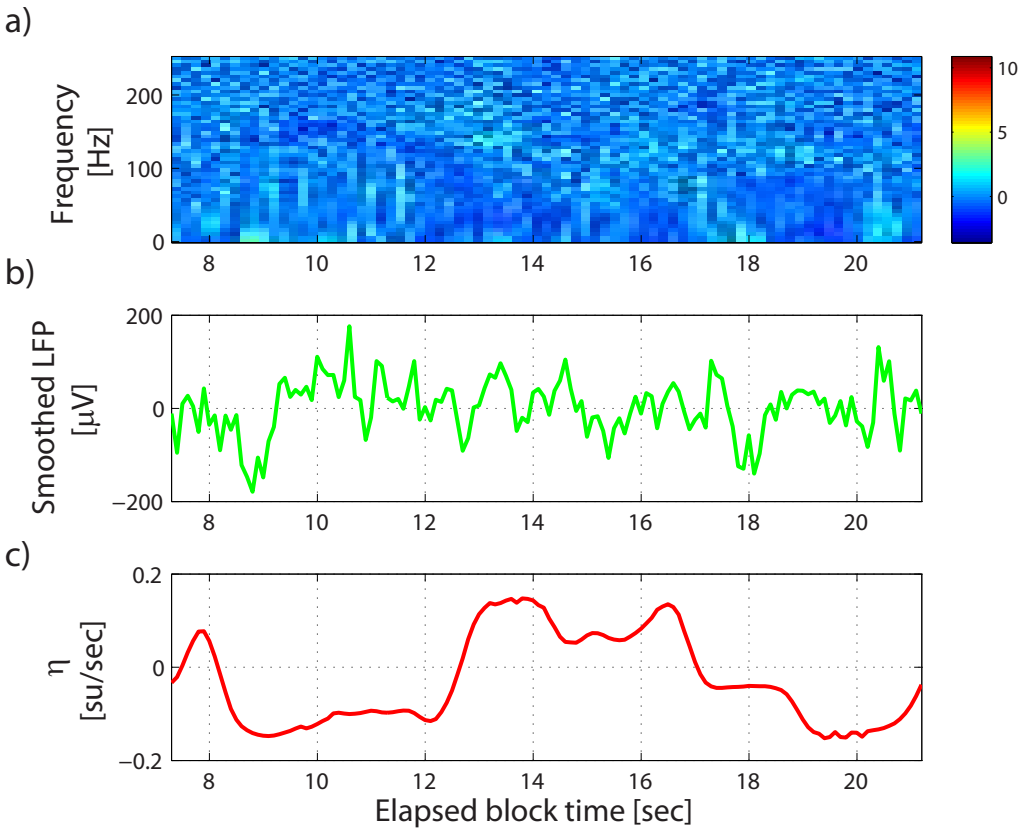

Figure 3.5: Example of LFP features and their covariance with the error metric. a) presents the spectrogram from a representative electrode (z-scored within frequency bands) over the trial described in Figure 3.2a. b) gives the same electrode's Savitzky-Golay (SG) smoothed signal. Over the same time, c) gives the trace of the error metric.

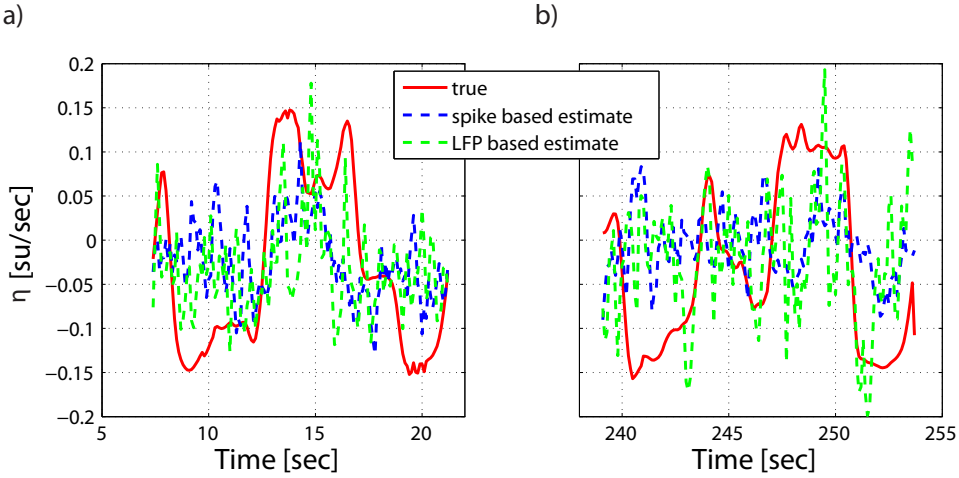

Figure 3.6: Example of error detection accuracy. The trials and the error metric fluctuations during them correspond to those presented in 3.2. a) traces the same error metric as in Figure 3.2a, but now the corresponding estimates of it are also provided. b) gives the true and estimated traces for the trial in Figure 3.2b.

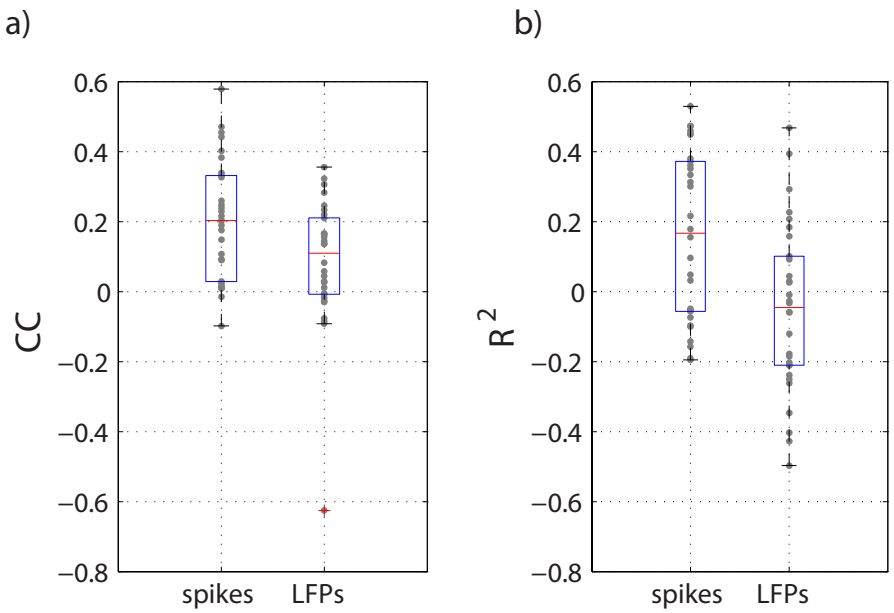

Figure 3.7: Error estimating performance over all 30 Tests. a) Pearson's correlation coefficent (CC) for tests when spike (left) and LFP (right) features were used to estimate the degree of decoding error. Grey dots indicate the individual test values, while box and whisker plots give the range and 25, 50, and 75% quartiles. b) same plot only with the coefficients of determination, R^2, plotted.

yielded a total of 30 tests (6×5), whose results can be seen in Figure 3.7. For spike based features, the median CC was 0.20 (lower quartile = 0.03; upper = 0.33) and median r^2 was 0.17 (lower quartile = -0.06; upper = 0.37). Similarly, estimation using LFP feature types had a median CC of 0.11 (lower quartile = -0.01; upper = 0.21) and a median r^2 of -0.05 (lower quartile = -0.21; upper = 0.10). These values were low compared to that achieved by using cursor position, velocity, acceleration vectors and their magnitudes as the informative features. Using cursor motion instead of neural activity gave a median CC of 0.39 and a median r^2 of 0.27.

3.5 DISCUSSION & CONCLUSION

With median CC values around 0.20 and r^2 values less than 0.20, the presented methods of extracting informative features and decoding lack a strong predictive power. The observation is distinctly different from claiming that the neural signals recorded intracortically from the motor cortex are not predictive of the error metric. Although counting spikes in 300 millisecond time bins, averaging amplitude in frequency bands from an SFFT based spectral decomposition, and SG smoothing are techniques typically employed in the research community [71, 138], they are not the only ones. More information may be gleaned from estimating directly from the point process where the recent history on a millisecond by millisecond scale is considered [91]. We chose to average amplitude within frequency bands that have proven useful in past reports [38, 138], but perhaps each frequency bin isolated by the SFFT should be incorporated, as well as spectral phase along with amplitude.

Use of a linear regression based estimator, with some form of feature selection (lasso in our case), also has a precedent in the literature. Arguable more advanced estimation routines would include the Kalman filter [76] and experimenting with two nonparametric algorithms from machine learning: support vector regression and random forests [141]. Both of the latter algorithms do not assume a linear mapping between the cognitive variables of interest and informative features from the neural signal, rather they each impose a relatively flexible structure (assuming the support vector machine utilizes a nonlinear kernel such as the radial basis function).

An uncertainty that makes estimating the degree of error difficult is the chosen metric itself. We defined the degree of error based upon the speed at which the cursor moves away or toward the target with a time delay. Unfortunately, there is uncertainty about when and to what degree the participant recognizes an error in the cursor's motion. For example, as the cursor starts to drift away from the target at a constant speed, is the error response phasic, where it jumps to a high value at the beginning of the drift, but gradually declines? Does the neurological response to decoding errors modulate only with direction? Going forward, one option would be intentionally to insert brief instances of incorrect cursor motion during a block of trials that were otherwise exhibiting correct cursor motion. The strategy has been used before when recording with electrocortographic arrays [138]. Another direction would be to analyze the data in an unsupervised fashion. Doing so via expectation maximization resulted in competitive performance when EEG drove a P300-based speller [142].

In terms of the evaluation method, confounding factors related to biases in the cursor trajectory might have arisen. For example, during a block of trials, when a decoding error occurred, the decoder might have almost always driven the cursor to the lower left corner of the screen. During those periods of aberrant cursor control, the person might have attempted to move the cursor up and to the right. As a result, any spiking units that tend to increase their firing rates in that intended direction, would appear to be tuned to the presence of decoding error. In order to rule out such confounds, future work should carefully select a subset of trials from the total set, where such statistical dependencies between cursor trajectory and decoding errors do not exist overall in the calibration and test data.

Because of the relatively weak ability to predict the level of decoding error, possibility of a confound, and challenge in labeling when a person perceives an error, the investigation has yet to provide any definitive findings. As discussed in the previous paragraphs though, there exists several potential paths, some of which might lead to demonstrating the presence of decoding error perception in the motor cortex. If such a signal can be pulled out via the electrode recordings, and if that signal is sufficiently clear, then beyond a scientific finding, the technique could alert the decoder controlling the cursor of its failings and help to mitigate its errors.

CHAPTER 4

Mixing Decoder Outputs

4.1 INTRODUCTION

This chapter is derived from a paper we have already published [109]. Here, we explore use of the Kalman filter to control a cursor to reach several targets on a computer screen. As mentioned previously, the cursor is typically driven by decoding for cursor velocity [21,26,76]. However, other motor variables have been decoded in the context of goal directed reaching using iBCIs. One of the first successful closed-loop demonstrations decoded cursor position [19]. Since then, joint torques [98], angles [69] as well as grip action [25] have also been estimated.

Previously, position and velocity information have been shown to be encoded in neural activity at the same time during reach movements [143] and a Kalman filter has simultaneously decoded for both [129]. During decoding of an entire reach movement, factoring in the displayed location of the cursor reduced the time to reach targets with a neurally controlled cursor [115]. Here, similar to [129], a Kalman filter decodes simultaneously for cursor velocity and position. However, instead of controlling the cursor by integrating velocity [76] or linking it directly to the decoded position [129], we mix the velocity and position estimates from the Kalman filter to generate cursor movements. In people with tetraplegia making attempted reaching movements, the strategy results in improved offline reconstructions of movement intention. The boost in accuracy promises to improve iBCIs for the benefit of

those who have lost motor control due to injury or disease.

4.2 NEURAL RECORDINGS AND SESSION TASK

4.2.1 Recordings

The details of the recordings have been described elsewhere [26, 58]. Briefly, two people with tetraplegia, referred to as S3 and T2 participated in the BrainGate2 clinical research sessions[1]. S3 enrolled in the study when she was 54 years old; T2 enrolled when he was 65 years old. A 96 microelectrode array (Blackrock Microsystems, Salt Lake City) was implanted in the hand-arm region [135] of the motor cortex. Each electrode measured 1.5 mm in length and they were evenly distributed in a 10x10 grid on a 4x4 mm platform with 400 μm spacing.

4.2.2 Signal Pre-Processing

A percutaneous connection sent the voltage signals onto various stages of pre-processing including amplification, 0.3 Hz to 7.5 kHz analog bandpass filtering, 30 kHz analog-to-digital conversion, and 250 Hz digital highpass filtering. For each digitized channel, either time-amplitude windows or threshold crossings were setup up to detect extracellular actions potentials, originating from a single cell or a group, i.e. unit spikes. This spiking activity was then time binned (100 msec or 20 msec) to supply a vector of spike counts to the Kalman filter at each time step. Detected spiking units with a mean and standard deviation > 1 Hz were selected for the analysis which resulted between 23 to 90 spiking units depending upon the session.

4.2.3 Session Task

We evaluated the decoders using three sessions with S3 and three with T2. Sessions with S3 (T2) were recorded from 986 and 1003 (67 and 82) days after implantation. We examined open loop blocks of trials. In the trials, a computer program moved a cursor through a series of preset motions on a computer screen. There were five circular targets, one at the

[1]CAUTION: Investigational Device. Limited by Federal Law to Investigational Use.

screen's center and the rest in locations up, down, left, and right relative to the center target. During a trial, the cursor moved from one target to the next in a center-out-and-back fashion, meaning after the cursor completed a trial taking it to one of the peripheral targets, it would move back to the center in the subsequent trial (Figure 4.1a). During each block of trials, all peripheral targets were visited in a pseudo-random order. The cursor headed in a direct, straight line toward each target and its speed followed one of several possible profiles. All were truncated Gaussian curves (Figure 4.1b), but they ranged from 2.5 to 5 seconds.

During the trial blocks, the person was instructed to attempt hand and arm movements as if they were controlling a computer mouse to generate the same cursor motions as they saw on the screen. Although their paralysis prevented any functional arm and hand movements, their intended reaching movements were assumed to be a close match to that displayed on the screen. Supporting this assumption, during a session, the data generated from the blocks were all used to calibrate Kalman filter decoders that were run online in subsequent trials providing better than average neural cursor control.

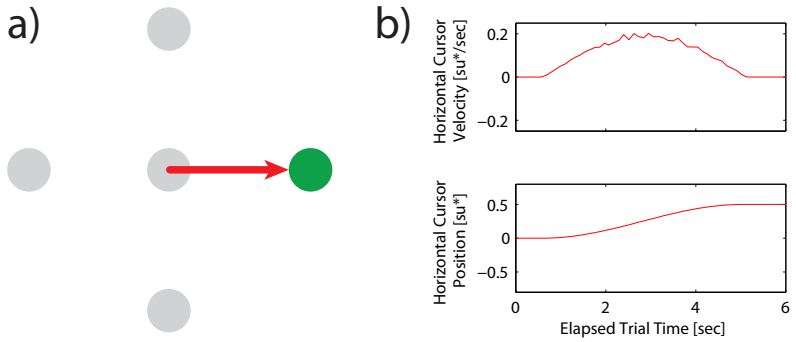

Figure 4.1: Illustration of task and decoded variables. **a)** During a trial, i.e. reach, the trajectory of the cursor (red line) moved from a starting position (grey) and ended at a target (green). The motion was controlled by a computer and the person was instructed to attempt movements as if controlling a computer mouse to match the displayed motion. **b)** The cursor's velocity profile (horizontal component displayed) followed a truncated Gaussian curve. As a result, the target's position relative to the target (horizontal component displayed) changed in a smooth monotonic manner during the reach. *Distances are in screen units (su) where the screen is 1.2 su wide by 1.2 su tall.

4.3 DECODING & CURSOR CONTROL METHODS

4.3.1 Kalman Filter

A typical approach in iBCIs is to decode using the time invariant version of the Kalman filter to estimate kinematic variables at every time step, k. In the context of our analysis, the kinematic vector, $\boldsymbol{x}_k$, describes the intended motion the participant attempts to impart on a computer cursor. The filter estimates $\boldsymbol{x}_k$ from observations of the spike counts at every time step, $\boldsymbol{z}_k$. The change in $\boldsymbol{x}_k$ over time and the relationship between $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ are modeled as

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + \boldsymbol{w}_k : \quad \boldsymbol{w}_k \sim \mathcal{N}(0, W) \tag{4.1}$$

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \theta + \boldsymbol{q}_k : \quad \boldsymbol{q}_k \sim \mathcal{N}(0, Q), \tag{4.2}$$

where $\boldsymbol{x}_0 =$ all zeros and $\mathcal{N}(\mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ. Equation (4.1) is often called the state dynamic model and (4.2) called the tuning model. Model parameters reside in A, W, H, θ, and Q and are estimated via linear regression methods. The calibration procedure uses data from iBCI sessions where the time history of both $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ are assumed known [129].

Given (4.1) and (4.2), the maximum a posteriori probability estimate for velocity at the present time step, n, can be found using the recursive equations that define a time invariant Kalman filter [36, 127],

$$\begin{aligned}
P_{k-1}^{+} &= AP_{k-1}^{-}A^T + W \\
K_k &= P_{k-1}^{+}H^T\left(HP_{k-1}^{+}H^T + Q\right)^{-1} \\
P_k^{-} &= \left(I - K_kH\right)P_{k-1}^{+} \\
\hat{\boldsymbol{x}}_k &= A\hat{\boldsymbol{x}}_{k-1} + K_k\left(\boldsymbol{z}_k - \theta - HA\hat{\boldsymbol{x}}_{k-1}\right),
\end{aligned} \tag{4.3}$$

for $k \in 1, 2, ..., n$ with chosen initializations $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0$ and $P_0^- =$ the zero matrix.

4.3.2 Velocity Based Cursor Control

A standard approach is to solely decode for the two dimensional velocity vector of the cursor, $\boldsymbol{x}_k \triangleq \boldsymbol{v}_k$ [21, 26, 76]. The cursor's commanded location, $\hat{\boldsymbol{c}}_k$, is then controlled by integrating the estimated velocity, $\hat{\boldsymbol{v}}_k$,

$$\hat{\boldsymbol{c}}_k = \hat{\boldsymbol{c}}_{k-1} + \delta \hat{\boldsymbol{v}}_k, \tag{4.4}$$

where δ is the interval between time steps (100 or 20 msec for our data) and $\hat{\boldsymbol{v}}_k = \hat{\boldsymbol{x}}_k$.

4.3.3 Position Based Cursor Control

Alternatively, we can use the four-dimensional kinematic vector $\boldsymbol{x}_k \triangleq \left[\boldsymbol{v}_k^T,\ \boldsymbol{r}_k^T\right]^T$, where $\boldsymbol{v}_k$ is velocity as before and $\boldsymbol{r}_k$ is the two-dimensional position vector of the cursor. It is not obvious how to best use $\hat{\boldsymbol{x}}_k$ to drive $\hat{\boldsymbol{c}}_k$. One possibility is to simply use the cursor position component of $\hat{\boldsymbol{x}}_k$ as in [129], namely,

$$\hat{\boldsymbol{c}}_k = \hat{\boldsymbol{r}}_k, \tag{4.5}$$

which we call position based control. Note, as in Section 4.3.2, one can also only integrate velocity, $\hat{\boldsymbol{c}}_k = \hat{\boldsymbol{c}}_{k-1} + \delta \hat{\boldsymbol{v}}_k$, which turns out to work slightly better than the more standard velocity based control in 4.3.2 (the difference being the inclusion of position in the kinematic variables).

4.3.4 Mixed Velocity & Position Based Cursor Control

Our innovation here is to mix $\hat{\boldsymbol{v}}_k$ and $\hat{\boldsymbol{r}}_k$, both simultaneously estimated from a Kalman filter, to obtain an improved method of commanding $\hat{\boldsymbol{c}}_k$ via

$$\hat{\boldsymbol{c}}_k = \hat{\boldsymbol{c}}_{k-1} + \alpha\delta\hat{\boldsymbol{v}}_k + (1-\alpha)\delta\frac{\|\hat{\boldsymbol{v}}_k\|}{\|\boldsymbol{\Delta}\hat{\boldsymbol{r}}_k\|}\boldsymbol{\Delta}\hat{\boldsymbol{r}}_k, \tag{4.6}$$

where $\alpha \in [0,1]$ is the mixing coefficient, $\|\cdot\|$ denotes magnitude, and $\boldsymbol{\Delta}\hat{\boldsymbol{r}}_k = \hat{\boldsymbol{r}}_k - \hat{\boldsymbol{c}}_{k-1}$. The change in $\hat{\boldsymbol{c}}_k$ is now a mixture of two vectors, one is $\delta\hat{\boldsymbol{v}}_k$ and the other points in the direction of $\hat{\boldsymbol{r}}_k$.

4.4 RESULTS

Figure 4.2 provides example trajectories driven by the three different cursor control strategies using hold-out data. In the trial, position based control appears less accurate than velocity based control. With $\alpha = 0.7$, mixed velocity and position control closely tracks velocity control, but appears more accurate. The better tracking results from the mixing, where position estimates have some predictive value but are not completely correlated with those developed by integrating estimated velocity.

We performed five fold cross-validation on each of the six sessions, resulting in 30 tests. To quantify the performance during a trial, we calculated the rms error (RMSE) in euclidean distance between estimated and actual cursor trajectories [129]. Average RMSE over trials was then calculated for each of the 30 tests and cursor control strategies (Figure 4.3). Mixed velocity and position control improved RMSE by 12.2 $\pm$10.5% and 37.8 $\pm$10.7% over velocity based and position based control respectively (pairwise t-test, $p<0.05$). Compared to velocity control, mixed control did better in 28 out of the 30 tests whereas against position control, mixed control was superior in all trials.

While $\alpha = 0.7$ yielded the lowest RMSE values on average, mixed velocity and position control did well under any setting for α (Figure 4.4). The data show an over reliance on either position or velocity fails to do better than a blending between them. Also, the curve's approximate convexity and single minima suggest a clear trend in positive gains achieved by mixing.

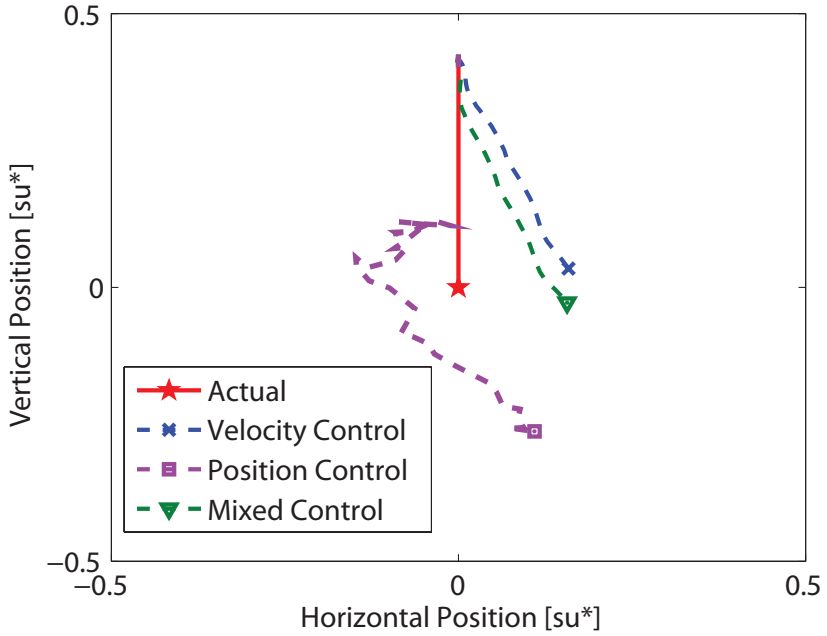

Figure 4.2: Example trial showing the true cursor trajectory during a return from the peripheral target to the center and estimated versions by the three control strategies. Markers such as "x" show the ends of the trajectories. For all decoders, the initial estimated cursor position is set to the true starting position. At the beginning of the trajectory, position control generates a relatively large change in cursor position, whereas velocity and mixed control do not, due to their incremental nature. For mixed control, $\alpha = 0.7$. *Screen units

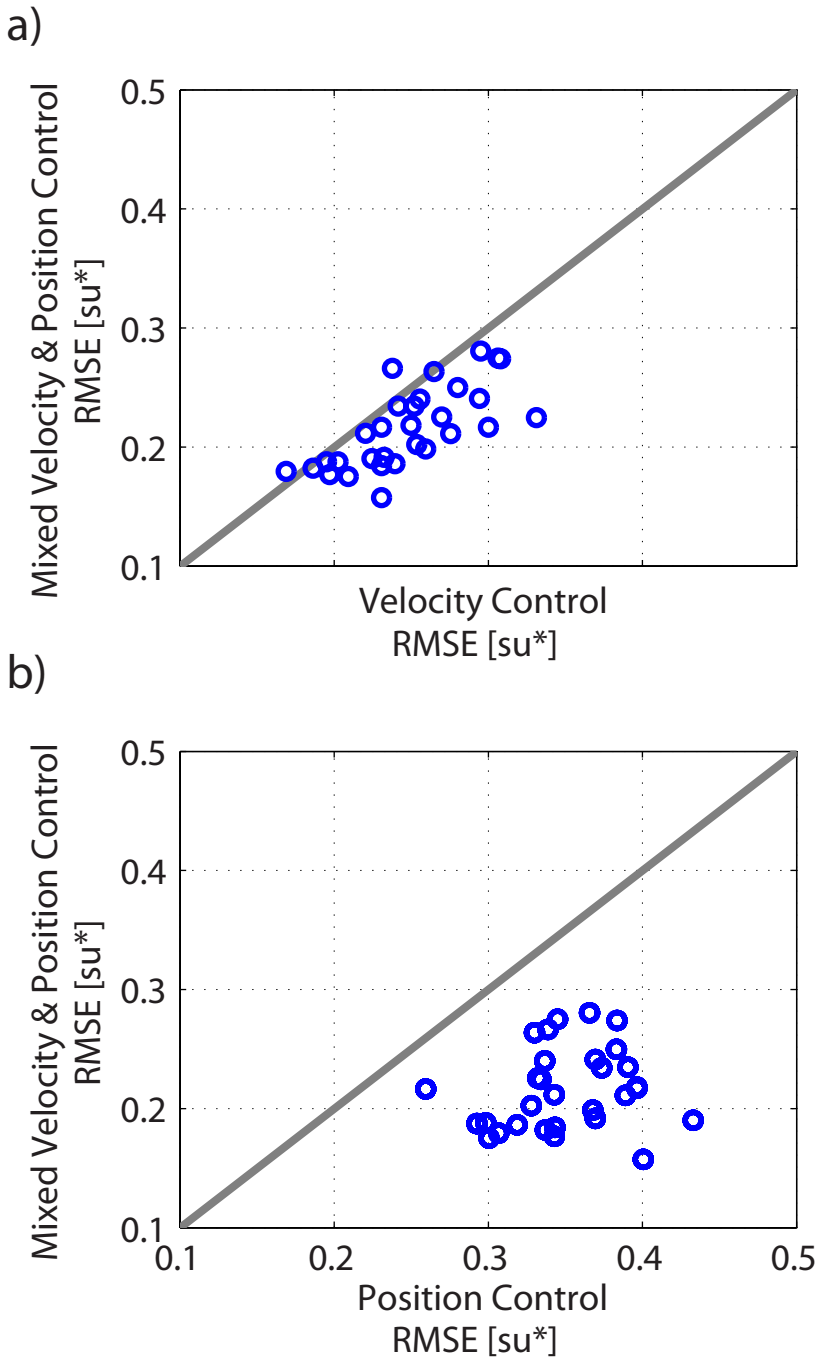

Figure 4.3: Mixing velocity and position estimates improves offline decoding performance. **a)** Mixed control's performance (RMSE) compared to that from velocity based control and **b)** position based control. Each point is a test (30 total) and the grey line denotes where the RMSE values are the same, meaning all points below the line indicate a lower error with mixed control. *Screen units

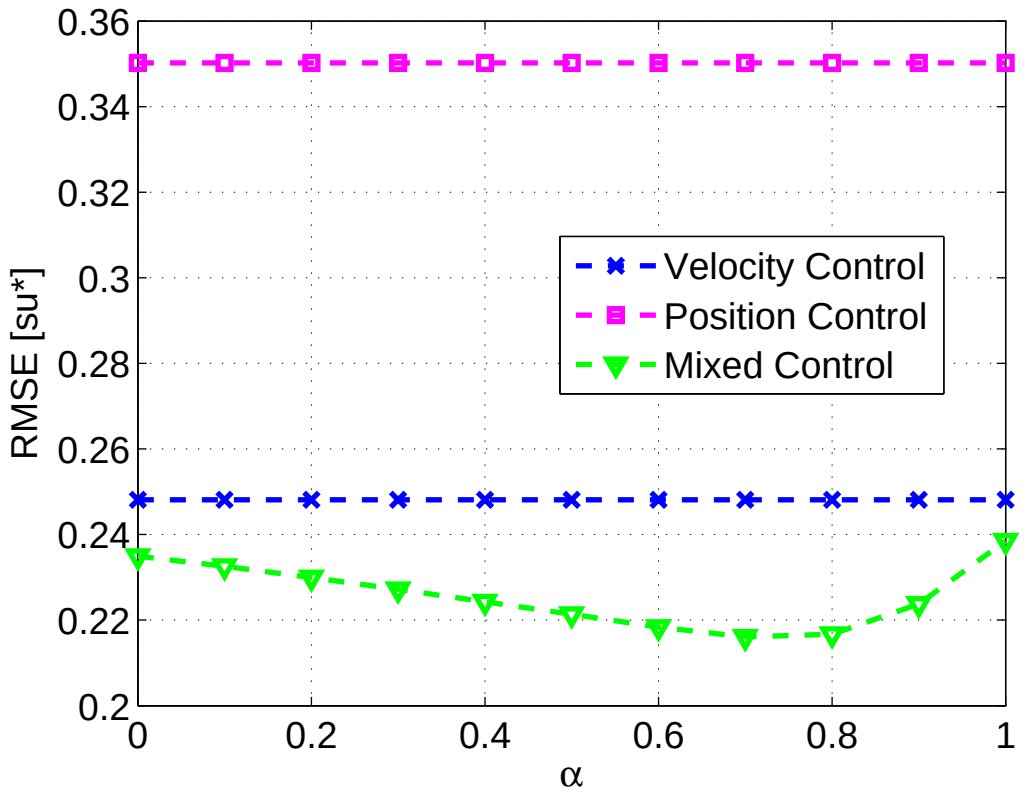

Figure 4.4: Mixed cursor control performance (RMSE) as a function of the mixing coefficient, α. An α close to one is being driven mostly by decoded velocity, while α close to zero is influenced mostly by decoded position. RMSE values for velocity based and position based control shown for reference (they are not functions of α). Note, at $\alpha = 0$ or 1, mixed cursor control is still better than either velocity or position control, since mixed control simultaneously decodes for position and velocity. In contrast, velocity control solely decodes for velocity and position control solely for position. *Screen units

4.5 DISCUSSION

The results show that mixing velocity and position improves cursor control over present methods in an offline analysis. Specifically, average RMSE of the trial cursor trajectories was better than those from prior approaches [76, 129]. Between the velocity based and position based control, the former was better, consistent with prior findings [76]. While $\alpha = 0.7$ gave the best results, the findings are not overly sensitive to this tuning parameter.

One untested sensitivity is dependence of the results on the type of task. During the sessions, the person was instructed to attempt to move the cursor to match that displayed on the screen, where the cursor moved in a smooth fashion over a period of seconds. Thus, we view the task as a variant of pursuit tracking. Differences in the best Kalman filtering approach have been noted between pursuit and step tracking [129]. The most appropriate cursor control strategy may be dependent on the task as well.

A major limitation of the results concerns the offline nature of the decoding. Performance gaps between novel decoders and more standard approaches can narrow when people use the algorithms online within an iBCI to neurally control a cursor [89]. In the online, closed-loop setting, the person has constant feedback about the decoder's ability to estimate their intended movements and thus can attempt to compensate for any decoding errors. However, complete information about the intended movements are also not known, e.g. desired cursor speed. Thus, offline studies often have better information on the actual intended movements of the person, leading to a more direct analysis of decoder inaccuracies.

Nevertheless, a next step is to run the mixed velocity and position based cursor control online in closed-loop trials. Doing so will ascertain whether the decoder can improve task performance. A complementary effort will be to better capture the interplay between intended velocity and position in driving neural activity. Possible future advancements include introducing nonlinearities into the tuning model [144] in order to improve accuracy even further.

CHAPTER 5

Decoding with Nonparametric, Hierarchical Methods

5.1 INTRODUCTION

An iBCI's decoding process occurs in two major steps. First, signal processing techniques extract informative features from the neural signals. Second, software routines estimate motor state variables from those features. The Kalman filter is a common estimator where the features are modeled as being generated from a multivariate Gaussian distribution whose mean is an affine transform of the movement variables [76, 129]. Thus, each feature has a linear connection to the hidden states and the dependency between features are captured in the Gaussian's covariance matrix. In people with tetraplegia, Kalman filtering has enabled control of a computer cursor [24] and, subsequently, control of a robotic arm [26].

In one study, however, 35% of the spiking activity was found to be nonlinearly related to arm kinematics [78]. Capitalizing on these findings, an unscented Kalman filter worked with a quadratic encoding model to improve decoding performance in monkeys [81]. Another tactic taken was to use a switching Kalman filter [87]. When the features have been putative extracellular action potentials, i.e., spikes, their distributions are better characterized as Poisson than Gaussian, which leads to point-process based encoding models [86, 90]. Point process filters have demonstrated their value in animals [88, 92–94] and in people with

tetraplegia [91].

Going even further, a nonparametric model predicted spike times as a function of arm kinematics and the prior spike time history over the entire neural population [145]. The proposed method applied stochastic gradient boosting with regression trees as the base learner. The approach was freed of structural modeling assumptions about how motion and past spikes drive spiking probabilities in the present, one-millisecond-scale time bin. Improvements were seen in the average likelihood compared to Bayesian P-splines and the more common generalized linear models. The gains over the latter are particularly relevant as they represent the standard approach for modeling spike dynamics.

Here, we also employ an ensemble of regression trees, using a technique called random forests, to introduce a nonparametric method, only our estimator predicts arm kinematics from the present and recent history of spiking activity. In a two stage procedure, we combine this nonparametric model with a Kalman filter to account for the dynamics of the kinematic variables. The innovative estimator has direct applicability to iBCI's. We demonstrate the value by evaluating the method's performance offline in two monkeys and two people with tetraplegia. The testing shows use of a nonparametric estimator is an improvement over the standard Kalman filter. Furthermore, the calibration procedure completes in a few minutes, and the online routine can run in real-time, i.e., 100 milliseconds per cycle. The increase in control quality is consistent as very few test cases reporting a contrary result. These characteristics of the hierarchical, nonparametric method point to its promise to advance the state-of-the-art in iBCI decoding.

5.2 EXPERIMENTAL SESSIONS

5.2.1 Data Recordings in People with Tetraplegia

We briefly cover the recordings and sessions here as the full details have been described elsewhere [58]. The neural recordings were taken during experimental sessions, as part of the BrainGate[1] research clinical trial. Both participants, whose data were analyzed for this research, have tetraplegia and anarthia due to brainstem stroke. The participants, hereafter

[1]CAUTION: Investigational Device. Limited by Federal Law to Investigational Use.

referred to as S3 and T2, were a 54-year-old female and 65-year-old male respectively upon enrollment. They both underwent surgery, to implant a microelectrode array (Blackrock Microsystems, Salt Lake City) into their motor cortex.

The microelectrodes were arranged in a 10 by 10 grid where the four corner electrodes were not active by design, resulting in a total of 96 recording channels. The platinum tipped, Parylene-C coated silicon electrodes were spaced 40 micrometers apart and rose 1.5 mm from the silicon base. Additionally, two wires extended from the array to provide two options for a voltage reference. The array was inserted into the hand-knob [135] area of the motor cortex, where prior fMRI revealed increased activity levels to coincide with hand and distal arm motions. The implanted device also included a platinum/gold lead wires that transmitted the signals to a percutaneous connector.

From the connector, the 96 voltage channels immediately passed through shielded cabling to a 10,000 gain amplifier. Subsequently, the signals were analog filtered with a 0.3 Hz to 7.5 kHz bandpass, sampled at 30 kHz per second with 14 bits of accuracy (1 microvolt per bit), and then optically isolated. The resulting digitized signal was then saved for subsequent analysis.

5.2.2 Data Recordings in Monkeys (Macaca Mulatta)

The monkey recordings were part of a past thesis project, and we refer the reader to it for the full details [146]. A male and female monkey (hereafter referred to as AB and LA) were each implanted with two electrode arrays . The arrays were the same 10 by 10 grid, platinum tipped silicone based electrodes mentioned previously (Blackrock Microsystems, Salt Lake City). The only difference was the array length, which measured 1 mm in length. One array was inserted into the upper limb area of M1 (medial to lateral: by the genu of the arcuate sulcus; posterior to anterior: anterior to the central sulcus); the other was located in the arm region of area 2/5 in the sensory cortex. Data recording was carried out by the Cerebus multichannel system (Blackrock Microsystems, Salt Lake City). Like the human recordings, components included a shielded cable, amplifier, optical isolation, and 30 kHz, 14 bit accuracy A/D conversion. The recordings analyzed in this investigation are only from the M1 array.

During the behavioral task, the monkeys sat in a chair and had their head constrained via a fixture. They put their right arms into a custom designed fixture connected to a mechanical exoskeleton (Kinarm from BKIN Systems, Ontario, CA). The device constrained the arm movements to a two dimensional plane parallel to the floor, effectively limiting joint rotations to elbow flexion and shoulder rotation [147]. The joint angles were digitally sampled at 200 Hz and captured by a general purpose computer running Windows XP, where custom software computed hand endpoint position.

5.2.3 Session Task in People with Tetraplegia

During sessions, participants were presented with a contiguous series of trials, which we refer to as a block of trials. During each block, the people were seated facing a 17 inch computer monitor from a distance of roughly 56 centimeters. On the computer monitor were five circular disks, one in the screen's center and the rest arranged in an up, down, left, right fashion relative to the center disk (the diagonal blue discs are displayed, but not activated as targets). These disks were the possible goals to be reached by a computer cursor, which was displayed as a smaller circular dot. Dimensions of the graphics were measured in screen units, where the cursor control area measured 1.2 screen units (su) wide (Figure 5.1)

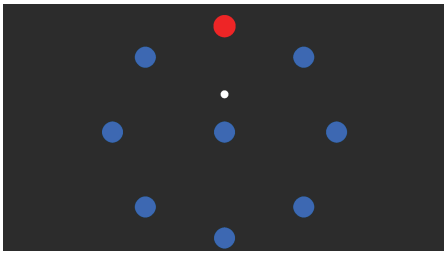

Figure 5.1: Open loop cursor task in people. In this example, a circular cursor (white disk) moves under computer control to the currently active target (shown in red). People are instructed to attempt movements that would drive the cursor through the displayed trajectory on the screen. Colors have been modified for print clarity.

During a block of trials, the cursor moved from one goal to the next in a straight line. For a specific trial, the destination goal turned to red and the computer moved the cursor through a preset trajectory to the target goal. The cursor followed two different speed

profiles, each a truncated Gaussian curve, but differing in their peak speed (0.20 or 0.35 su/second). At the start of the block, the cursor began at the center goal. The cursor then proceeded in a center-out-and-back paradigm; the cursor would move to a peripheral goal, and then return back to the center goal. In a set of eight trials, each peripheral target was visited once in a pseudo-random order and the center target eight times. By doing so, the positions and velocities of the cursor over time were balanced, adding vector-wise approximately to zero.

Meanwhile, the participant was instructed to attempt movements as if he or she were controlling the displayed cursor using a computer mouse. Sessions used in this investigation stretch from 986 (39) days after implantation to 1,330 (68) days for S3 (T2). There were six sessions total, three each from S3 and T2. Within a session the number of blocks analyzed ranged from three to four, yielding a total recording of 316.8 to 361.4 seconds and 68 to 80 trials per session.

5.2.4 Session Task in Monkeys

The Kinarm, through the calculation of hand point position, drove a light-blue cursor dot (1 cm in diameter) on a computer screen (40 by 32 cm) facing the animal, 45 cm distance away. A given trial proceeded in a sequence of events. First, a circular target appeared at the screen's center (a purple disc, 1.1 cm in diameter). The monkey was trained to move the cursor dot to within the vicinity of the target, specifically within 1.5 cm of its center, where upon the target would change to a yellow color. The cursor had to be held near the target for a 1-2 sec hold period. Subsequently, a new purple target appeared, in one of eight locations, equally distributed, all 4.3 cm from the screen's center (see Figure 5.2). The new target appeared for 300 msec and then extinguished. After a pseudo randomly generated hold time (1-1.5 sec), the same peripheral target reappeared and the monkey was trained to move the cursor to it and hold for 1-1.5 sec. As long as the monkey performed all steps of the task thus far and got the cursor within a 1.5 cm error zone of the peripheral target, then a juice reward was provided. Following the first portion of the task, the peripheral target began to move and the monkey was trained to keep the cursor within the target's vicinity, while various events occurred, such as brief disappearances of the target or force

perturbations to the Kinarm.

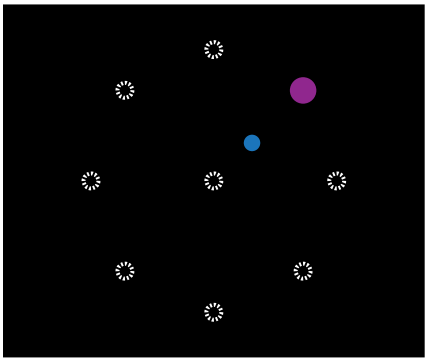

Figure 5.2: Monkey cursor control task. In this example, the monkey controls a circular cursor (light blue disk) via a mechanical exoskeleton, the Kinarm, on its arm. The trained animal attempts to move the cursor to the currently active target (shown in purple). The dashed circles are the locations of the other eight inactive targets.

For the purposes of this investigation, we focus solely on times when the monkey moved the cursor dot over either the center target or from the center target to the peripheral one. All other times during the trial were not included in our investigation. We used four sessions from monkey AB and three sessions from LA. The number of reaching motions to targets, hereafter what we will refer to as trials, ranged from 101 to 248 per session, resulting in a total of 1199 trials. Since each trial took about 1.9 seconds (including a 1-2 second hold), the total recording time used in our evaluation was approximately 2,200 seconds.

5.3 DECODING METHODS

5.3.1 Feature Extraction

For sessions with people, the digitized signals were high-pass filtered (4th order Butterworth filter, 250 Hz cutoff). From the filtered waveform, putative spikes were detected either via simple thresholding (which we refer to as threshold crossings) [26] or time-amplitude windows [24]. The former involved setting a fixed voltage threshold past which dips below it would signify the occurrence of a spike, e.g., -4.5 rms [26]. The time-amplitude windows were manually setup by visually inspecting 1.6 millisecond second samples of the signal. For a given channel, multiple sets of time-amplitude windows could be set with the intent of sorting several spiking sources on the same voltage channel, which we hereafter refer to

as units. The same steps were performed during monkey sessions. However, the principal difference was that spikes were differentiated by using a combination of density grid contour clustering and subtractive waveform decomposition on the first two principle components of the waveform segments [148]. For each unit, software binned the time stamps of its spikes in either 20 or 100 ms non-overlapping intervals. Thus, the feature extraction routines produced a vector of spike counts at every time step, k, which is equivalently an index for the time bins.

5.3.2 Standard Velocity Kalman Filter

The Kalman filter is a common method in iBCIs for estimating the d dimensional intended motor state, $\boldsymbol{x}_k$, from an informative, m dimensional feature vector extracted from the neural signals, $\boldsymbol{z}_k$. In our case, $\boldsymbol{x}_k$ is the intended 2D velocity of the cursor ($d = 2$), and $\boldsymbol{z}_k$ is a vector of spike counts ($m = 49.9$ on average in people, $= 91.9$ on average in monkeys). Two models underlie the Kalman filter,

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + \boldsymbol{w}_k : \quad \boldsymbol{w}_k \sim \mathcal{N}(0, W) \tag{5.1}$$

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \theta + \boldsymbol{q}_k : \quad \boldsymbol{q}_k \sim \mathcal{N}(0, Q), \tag{5.2}$$

where $\boldsymbol{x}_0 = [0\ 0]^T$, θ is a constant offset vector and $\mathcal{N}(\mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ. The first equation (5.1) is the state dynamic model and it governs the change in the hidden motor states over time; the latter equation (5.2) defines the statistical dependency between $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ and we refer to it as the tuning model. The parameters of the models then reside in A, W, H, θ, and Q.

To fit the model parameters, we randomly select a subset of the trials in a session as calibration (aka model training) data. Let n denote the number of time steps in the calibration data, then we define the cursor motion calibration data as $\boldsymbol{X}$, an $n \times d$ matrix where each row is a particular $\boldsymbol{x}_k^T$. Similarly, we define $\boldsymbol{Z}$ as an $n \times c$ matrix of spike count calibration data.

The matrix A captures the relationship between $\boldsymbol{x}_k$ in one time step and the next. Let

$\boldsymbol{X}^{(+)}$ be equal to $\boldsymbol{X}$ with the first row removed and $\boldsymbol{X}^{(-)}$ be equal to $\boldsymbol{X}$ with the last row removed. Then A is obtained using a multivariate form of linear regression,

$$A = \left(\left(\left(\boldsymbol{X}^{(-)} \right)^T \boldsymbol{X}^{(-)} \right)^{-1} \left(\boldsymbol{X}^{(-)} \right)^T \boldsymbol{X}^{(+)} \right)^T \tag{5.3}$$

W is then the error covariance between the actual cursor motion at a given time step and its prediction based upon the previous time step,

$$W = \frac{1}{n-1} \left(\left(\boldsymbol{X}^{(-)} \right) A^T - \boldsymbol{X}^{(+)} \right)^T \left(\left(\boldsymbol{X}^{(-)} \right) A^T - \boldsymbol{X}^{(+)} \right) \tag{5.4}$$

For the tuning model, first we determine the mean firing rate for each unit from the calibration data and collect them into a column vector, θ. Next, we employ multivariate linear regression to estimate H and set Q to the resulting error covariance akin to the development of A and W,

$$H = \left(\left(\boldsymbol{X}^T \boldsymbol{X} \right)^{-1} \boldsymbol{X}^T \boldsymbol{Z} \right)^T \tag{5.5}$$

$$Q = \frac{1}{n} \left(\boldsymbol{X} H^T - \boldsymbol{Z} \right)^T \left(\boldsymbol{X} H^T - \boldsymbol{Z} \right), \tag{5.6}$$

where $\boldsymbol{Z}$ is akin to $\boldsymbol{X}$ with the mean firing rates, θ, subtracted. With the models calibrated, the Kalman filter can be run on the test data, to produce a motor state estimate, $\hat{x}_k$, at every time step.

$$\begin{aligned} P_{k-1}^+ &= A P_{k-1}^- A^T + W \\ K_k &= P_{k-1}^+ H^T \left(H P_{k-1}^+ H^T + Q \right)^{-1} \\ P_k^- &= \left(I - K_k H \right) P_{k-1}^+ \\ \hat{\boldsymbol{x}}_k &= A \hat{\boldsymbol{x}}_{k-1} + K_k \left(z_k - \theta_k - H A \hat{\boldsymbol{x}}_{k-1} \right), \end{aligned} \tag{5.7}$$

where we initialize these recursive set of equations by assuming $\boldsymbol{x}_0$ is the zero vector and P_0^- the all zero matrix. Under the case where the dynamic and tuning models precisely reflect reality, the filter is optimal, both in Bayesian maximum a posteriori sense as well as least squares sense over the range of linear filters [36, 127]. In the context of iBCIs, even a close match between the true model and those estimated in the calibration process is not to be expected. For example, a Poisson distribution is typically a better fit for capturing the modeled noise in the tuning equation. Nonetheless, the models sufficiently capture the observed phenomena that Kalman filter based decoders are one of the few decoders broadly adopted in the iBCI community, especially when people or non-human primates actively use an iBCI to control external devices [26, 81, 115].

5.3.3 Random Forest Regression

Regression trees are well known in the machine learning community [134, 149, 150]. We specifically work with binary trees, that split the multidimensional feature space into rectangular regions. Each node in the tree contains a splitting rule for one for the variables in the feature vector. When decoding a given feature vector, the node's splitting rule indicates which node to visit next (left or right). Leaf nodes are the exception, instead of a splitting rule, they contain a scaler value, which is the tree's output. Figure 5.3a provides an illustrative example of a regression tree and an example of the decoding process; Figure 5.3b represents the tree as rectangular regions in the feature space.

Being nonparametric, regression trees are not so much calibrated as they are constructed. We utilized a standard routine in MATLAB® to build the trees from the data, using the object class RegressionTree, where all parameters were left at their default values. Each node seeks to partition a given region in the feature space and the region's associated calibration data points. The strategy starts with a single root node where the region to split is the entire feature space and all calibration data points are considered. Thereafter, however, subsequent nodes split regions defined by their parent nodes and the data points evaluated are only those points lying in the region. To construct a node, a random subset of the features, a third of the total, are selected. For each feature, the routine searches through all possible ways to partition the region's data points. For each candidate split,

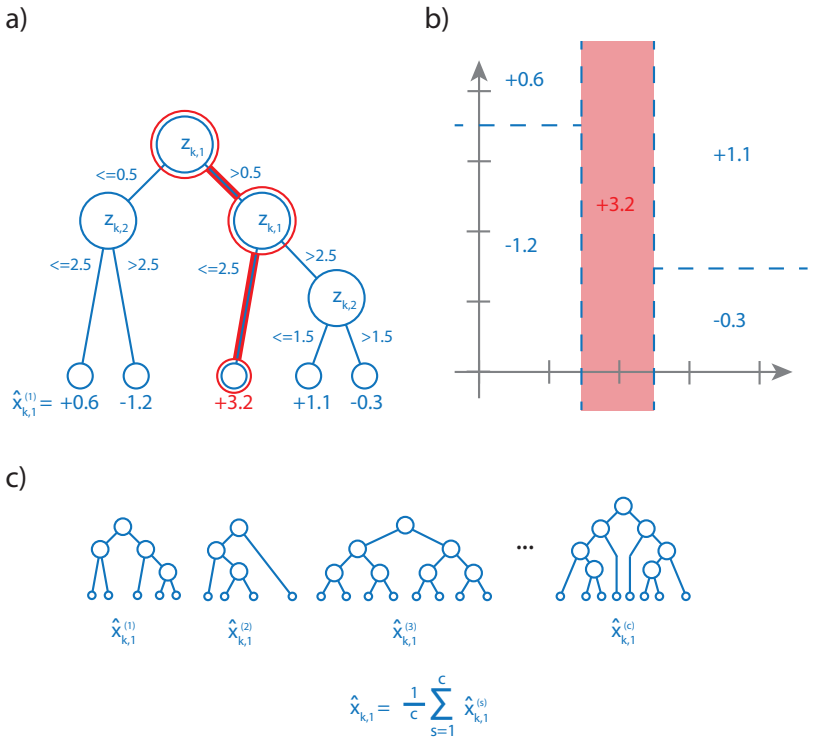

Figure 5.3: Illustration of a regression tree and the random forest algorithm. a) In the example regression tree, each node in the tree contains a splitting rule. Suppose the incoming feature vector was $\boldsymbol{z}_k = [2.1\ 1.3]^T$, then we would follow the path highlighted in red. The splitting rules would lead to an estimate of +3.2. b) the example tree's partitioning of the feature space, with the region associated with $\boldsymbol{z}_k = [2.11.3]^T$ highlighted in red. c) In the random forest algorithm, such a tree is one of many in the group and collectively they can be used to estimate a single dimension of the current motor state ($\boldsymbol{x}_{k,1}$ in this case). Let c be the total number of trees and s be their indexing variable, then the sth tree's estimate can be denoted by $\hat{\boldsymbol{x}}_{k,1}^{(s)}$ and random forest's estimate, $\hat{\boldsymbol{x}}_{k,1}$, is the average of the individual trees' estimates.

the estimate on each side of the split is simply the average of the data points therein. For the region being split, then, the mean squared error is calculated under these two simple estimates. The feature and split with the lowest mean square error is chosen for the node and two new child nodes are created, one for each side of the split. This process continues, ever expanding the tree, until either a created node has less than 10 data points or variance among the node's data points is less than the variance over all data points by a factor of 1 times 10^{-6}. If this stopping rule is met, the node becomes a leaf node and its associated outcome estimate is simply the average of the data points in its region.

Of course, it is not one tree, but a whole forest of them that must be built. Best practice recommends more than 1,000 of them, but in order to keep the calibration procedure less computationally burdensome, we built 100 trees per forest. The overarching strategy to grow the forest is a technique called bagging. In bagging, we sample with replacement from the calibration data. Suppose the calibration data has 10,000 sample points, i.e. observations, then we draw 10,000 samples from the 10,000 observations. Since we sample with replacement, some observations will be sampled more than once, while others not at all. Since each tree in the forest is built from different data, they usually differ in their construction, meaning many of the chosen features and splits are different. Even the structure of the tree (depth or number of nodes under a particular parent node) can widely vary (Figure 5.3c)).

The heterogenity in the trees winds up working to our advantage. Under the bagging technique, when estimating, each tree converts the received feature vector into a single scalar, the value associated with a particular leaf node. Those values are then averaged across the entire forest to yield a single output. Since each tree has likely partitioned the feature space into a different set of rectangular regions, the set of discrete values output by the forest can be quite large.

5.3.4 Two Stage, Random Forest & Kalman Filter Estimator

The use of the random forest technique has been focused on estimating each component of $\boldsymbol{x}_k$ from $\boldsymbol{z}_k$. However, we have yet to capture the dynamics of $\boldsymbol{x}_k$ itself. For example, arm endpoint motion, due to limitations in muscle response times and inertia, enforces

smoothness in the velocity profile over time. Unlike, however, the connection between features and motor states, we presume the relationship between current and prior motor states is a fairly random one. For example, the fine motor motions of handwriting differs drastically from throwing a baseball. We seek to build a decoder that is general purpose for controlling all sorts of arm endpoint motions, not one specialized to a particular task.

In that spirit, we model the dynamics of $\boldsymbol{x}_k$ as a random walk. More formally, we assume it takes the Gaussian-linear form of (5.1). Our expected course of the random walk must be reconciled with random forest estimates. The Kalman filter is an established way to do just that, only this time, instead of using the spike counts as features, we use the output of two random forests as the features. The strategy results in a two stage decoding architecture (Figure 5.4). Thus, unlike the standard Kalman filter decoder described in Section 5.3.2, the Kalman filter here intakes a feature vector with only two components (the first stage horizontal and vertical velocity estimates from the two random forests). Each of the random forests are constructed and run as described in the prior subsection. With the exception of the received features, the Kalman too is calibrated and runs as mentioned in the respective previous section (5.3.2).

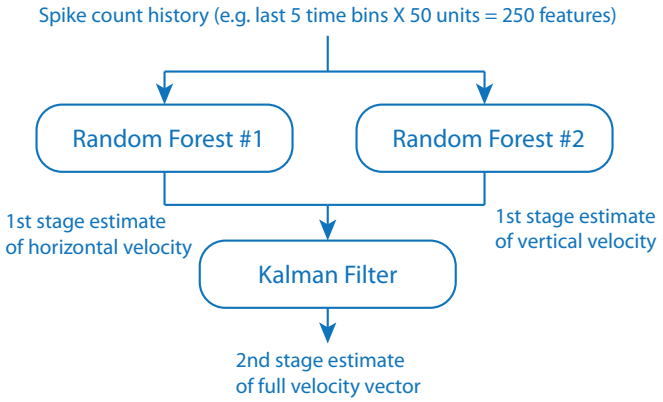

Figure 5.4: Diagram of Two Stage Random Forest & Kalman Filter Estimator

5.4 RESULTS

Figure 5.5 is a representative snapshot of a standard Kalman filter's, random forest's, and two stage decoder's ability to track the true velocity. The decoders are able to track the

overt arm movements of the monkey. In the case of a person's instructed movement, less accuracy is achieved. From the plots, it is not readily apparent which decoder tracks best.

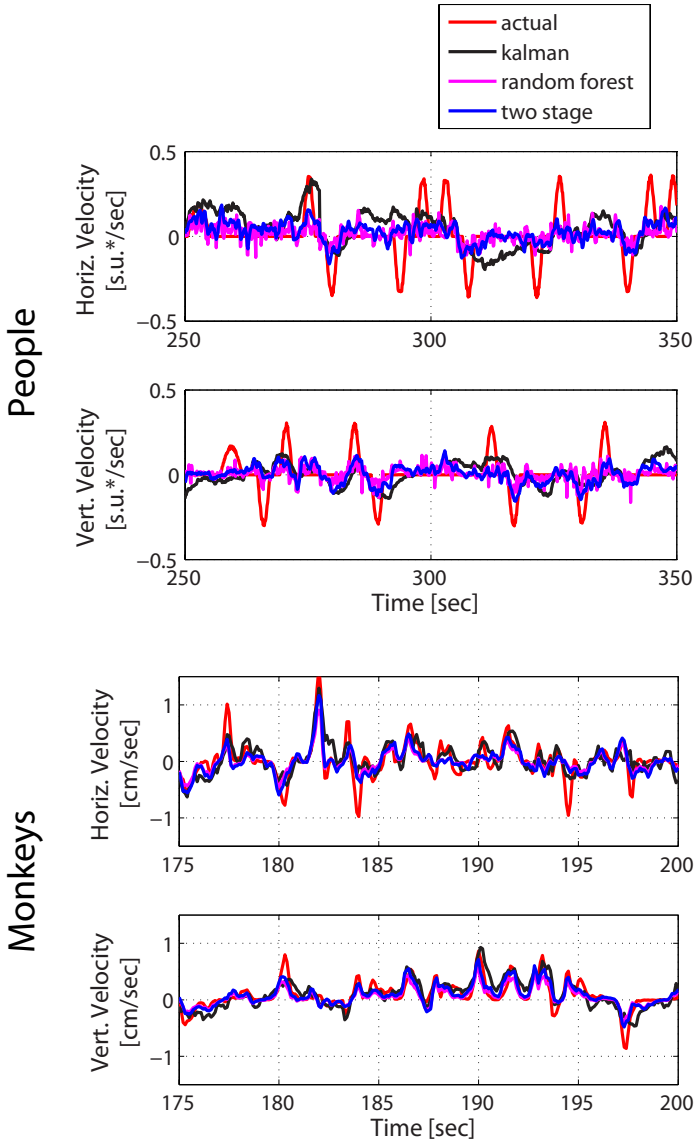

Figure 5.5: Estimates of cursor velocity components from the three decoding approaches (Kalman filter, random forest, the two stage decoder) as compared to the actual values

Five-fold cross-validated-style testing over 6 sessions in people created a total of 30 test cases. Similarly, the 7 sessions in monkeys provided 35 tests. One metric used to predict the accuracy of the results was Pearson's correlation coefficient (CC) between the actual and estimated velocity profile along each component (vertical and horizontal). The other metric was root mean squared error (RMSE), vector distance-wise, between the actual and

estimated velocity vectors over the test. All tests for statistical significance were pairwise t-tests with a $p<0.05$ threshold.

Using this evaluation setup, the standard Kalman filter's performance could be compared to that of the random forest's as well as the two stage strategy's (Figure 5.6 and Figure 5.7). As can be seen from the scatter plots in Figure 5.6, the random forest reduced the RMSE ($15.0 \pm 9.5\%$) relative to the Kalman filter. Although there was an increase, on average, in the CC ($5.1 \pm 20.6\%$), the result was not statistically significant. The picture looked even better for the two stage approach with both CC ($12.4 \pm 18.9\%$) and RMSE ($16.2 \pm 9.8\%$) improved. While the two stage's increase in CC over the random forest was statistically significant, its reduction of RMSE was not.

In monkeys, the story is also promising for the nonparametric decoders. Random forests had better CC's ($14.7 \pm 9.5\%$) and RMSE's ($9.9 \pm 5.4\%$) relative to the Kalman filter. The same held true for the two stage approach in both CC ($15.3 \pm 9.9\%$) and RMSE ($12.5 \pm 5.2\%$) values. Under the monkey data, the two stage's improvement over the random forest was statistically significant in both metrics.

With the two stage approach, we see that all tests, whether in people or monkeys, showed an improvement in RMSE. For CC's obtained from session data in people, 33.3% failed to show improvement, but the average decrease among the group was not large (7.7%). CC's in monkeys looked even better, with the two stage approach failing to increase correlation coefficients only 5.7% of the time, with reductions never more than 4.8%. These statistics all point to the robustness of the improvement the two stage approach had over the standard Kalman filter.

5.5 DISCUSSION AND CONCLUSIONS

Our results show a clear improvement in offline decoding accuracy as measured by CC and RMSE in both people and monkeys. The improvement is greater than 10% and is comparable or better than other reported gains with advanced decoding techniques [80, 81, 85, 94]. Natural questions to ask: Why are these gains happening? Where are they coming from? As mentioned in the introduction, the response of many single units (35%)

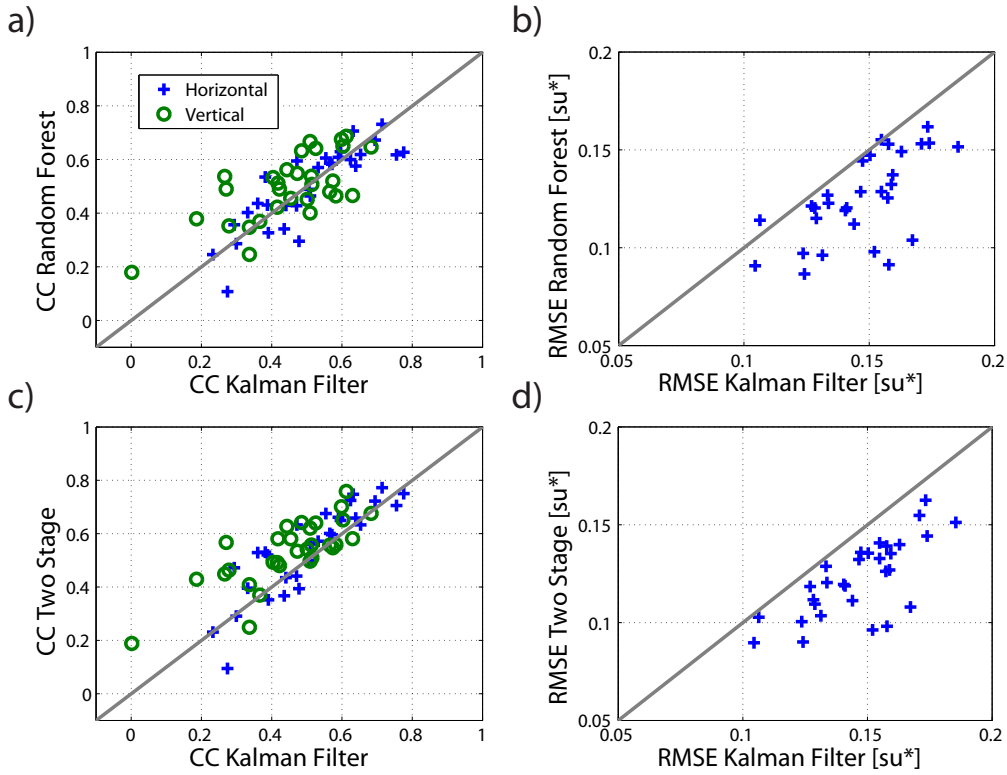

Figure 5.6: Performance of random forest & the two stage strategy vs. the standard Kalman filter over 30 tests across two people. Subplots a) and b) compare the random forest decoder to a standard velocity Kalman filter using Pearson's correlation coefficient (CC) and root mean squared error (RMSE) respectively. Subplots c) and d) do a similar comparison between the two stage random forest and Kalman filter combination against a standard velocity Kalman filter.

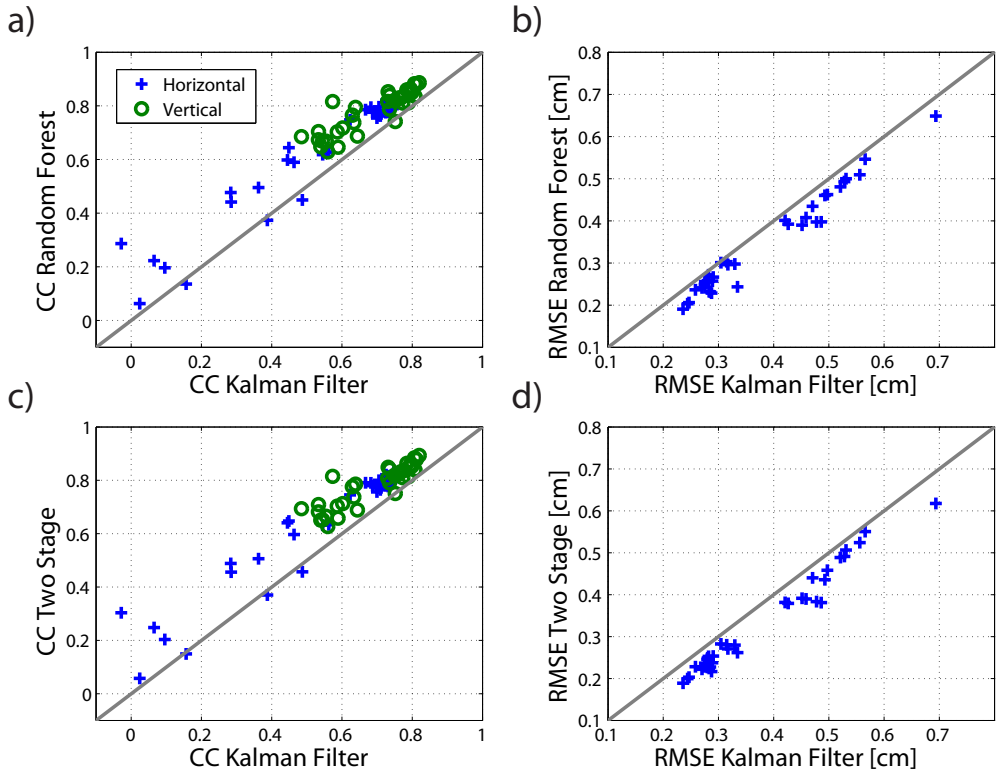

Figure 5.7: Performance of random forest as well as the two stage strategy vs. the standard Kalman filter over the 35 tests in monkeys. Subplots a) and b) compare the random forest decoder to a standard velocity Kalman filter using Pearson's correlation coefficient (CC) and root mean squared error (RMSE) respectively. Subplots c) and d) do a similar comparison between the two stage random forest and Kalman filter combination against a standard velocity Kalman filter.

of the population, has been shown to be superlinear [78]. In the finding, they fit a specific nonlinear curve to the response, but why stop there? One can image a plethora of possible mathematical relationships between unit activity and kinematic behavior, including cubic splines and absolute value functions. The value of the nonparametric approach is that the relationship is built from the data alone, avoiding errors introduced by constraining it to an a priori function.

Arguably, use of a high order polynomial, or piecewise linear function, could approximate the response of a single unit. Where such methods get problematic is in capturing the interactions between unit responses and how those interactions signal, if at all, additional information about the hidden motor state. Indeed, this challenge is representative of a very general problem in machine learning, from whence random forests came about. They come from a class of machine learning approaches called ensemble methods which often strike an effective balance between bias and variance errors [134]. Random forest regression is composed of a large set of regression trees. The nonparametric nature of the trees imbue them with much flexibility in learning the relationship between features and the estimated outcome. This comes at a cost, however, in that they are very sensitive to the particular sample set of data chosen to construct them. To deal with the overfitting, each of the trees is calibrated from a bootstrapped sample of the calibration data. As result, each of the trees tends to be a weak learner, that is its ability to estimate the outcome is low due to overfitting, but still tends to be better than chance. However, when the estimates are combined, the effects of overfitting tend to average out, and the results generalize demonstrably better than any single member in the ensemble. Given enough data and deep enough trees with finer and finer regions on their leaf nodes, the method approximates nearest neighbor methods [151] which is statistically consistent, with errors converging to a multiple of the Bayes rate as the number of samples goes to infinity [152]. For these reasons, random forests have become a top contender in applications where a parametric form to the regression model is not readily apparent [141].

Sometimes, when the mathematical relationship (even an approximate one) between the informative features is known apriori, the structural form of the estimator can be treated as a given; we do not require the method to learn the structure from the data. Indeed, the first

uses of the Kalman filter were in domains such as aerospace, where, for example, an altimeter's response to aircraft's motion is well understood because the design and manufacture of the aircraft and altimeter were completely known. However, when it comes to the central nervous system, the precise mechanisms of action are likely to remain a scientific pursuit for quite some time. Although, as mentioned above, assuming a linear relationship between spike counts and intended velocity is sufficient enough that the Kalman filter [26,81,115], or even the simpler Wiener filter [27] are popular decoders of choice in the iBCI community, we have provided results that support the belief that a technique such as random forests, with nonparametrically derived flexibility, can deliver a better approximate mapping, ultimately delivering a more accurate decoder.

The decoder design is also flexible to the type of features chosen. Recently, there has been much interest in local field potentials (LFPs) and different categories of motor states one can extract from them [41, 42, 46, 47, 49]. Being nonparametric when processing the features, the two stage decoder can handle LFPs and has an established track record for handling the large expansion in features that often result [141]. The architecture can be easily scaled to additional dimensions in the motor states as well. The lack of any specific structural assumptions makes use of random forests very generic.

For direct use in iBCI's as an online decoder, a challenge with nonparametric methods can be their computational complexity. iBCI cycle times are often in the 10's of milliseconds. While the use of bootstrapping, searching for splits, and the need for 100 (or better yet 1,000) trees can make calibration a large number-crunching exercise (several minutes on a 3.4 GHz processor), the number of steps to decode with a random forest is relatively few. Given 100 trees, assuming each tree has an average depth of 10 (in our case), only 1,000 comparisons must be made, a easy computation for today's processors.

Furthermore, we saw that including random forests in a hierarchical scheme led to better performance than decoding by it alone. Decoding in two stages can be seen as a form of nonparametric feature extraction along the same lines as deep belief networks [153]. The value of the 2nd stage was to capture the statistical dependencies between past and present motor states. This functions much like a Bayesian prior, where large changes in velocity over time are unlikely. We chose a straight Kalman filter for this step to avoid over training.

Future work could be to experiment with different decoders (including a random forest variant) for this stage as well.

Another natural next step would be to implement the decoder in real-time, letting a person with tetraplegia use the technology to control a computer cursor, ultimately one of the targeted applications of the technology. Because of a person's ability to compensate for the decoder's shortcomings, an innovative decoder which shows promising breakthroughs in performance offline can wind up unable to repeat the feat online [89]. Nevertheless, our offline analysis is a good first step, enabling one to rapidly discern the effectiveness of an algorithm. Also, it is an open question whether compensating for a less accurate decoder places an undesirable cognitive tax on the user, like a car's steering out of alignment, or the person, via the brain's plasticity, embodies the device and decoder sufficiently that the resulting reprogramming within the central nervous system eliminates such a burden.

Another direction would be to vet out the two stage decoder under more general conditions. Calibration and testing could be done against a broader range of movements beyond the point-to-point reaching studied here. Additionally, the data sets came from only two people, both of which were tetraplegic due to a brain stem stroke. However, other conditions, such as amyotrophic lateral sclerosis (ALS) can result in tetraplegia and we also hope that people with ALS will benefit from the technology. Nonetheless, we show a remarkable similarity in performance improvement between the data sets from people with tetraplegia and able bodied *macaca mulattas*. Investigations that combine sessions from monkeys, where the arm motions are precisely known, and those from people with tetraplegia, who are representative of people who may one day benefit from iBCIs, are rare in the literature.

Beyond showing a promising technical advancement, the results open the door to a better scientific understanding of how movement information is encoded in the motor cortex. Since we did not search over the vast space of possible models, we do not claim that the presented nonparametric, hiearchical decoder is how the biology functionally decodes the neural activity. However, if we can better read out kinematic variables with one decoder vs. another, perhaps the former is closer to the true computational nature. The units the electrodes happen to pick up are not the exact ones that directly encode all the information necessary to represent arm movement, but rather, the unit activity is only a small

subpopulation of a larger ensemble that is processing information broadly associated with arm movement. Even with this more indirect, weaker assumption, the success of the non-parametric decoder suggests that the interaction among unit spiking activity is important to motor cortical processing. The line of reasoning is similar to the work done with mixed selectivity unit activity in the prefrontal cortex [154].

Overall, a two stage hierarchical decoder with multiple random forests in the first step feeding into a single Kalman filter in the second was shown to be an improvement over the standard Kalman filter decoder in an offline analysis. The data was taken both from sessions involving people with tetraplegia and monkeys. The nonparametric aspect of the method provides a flexible, general purpose strategy for iBCI decoding irrespective of the extracted neural signal features or hidden behavioral/perceptual states of interest. The two stage's ability to capture interactions among the neural activity features also lends evidence to the idea that such interactions are fundamental to cortical processing. Thus, future work with the method promises to benefit the fields of both neural engineering and neuroscience.

CHAPTER 6

Conclusion

This thesis has delved into several innovative methods for improving the accuracy and reliability of decoding from neural signals recorded by iBCIs. We have also used a variety of evaluation data sets and methods to show their value, including sessions from volunteer efforts by people with tetraplegia as well as tasks performed by monkeys. We analyzed data both when the user was attempting to control a computer mouse on a screen using an iBCI as well as when they made attempted or overt movements. This mix of data sets, all with an eye towards improving decoding, is unusual in the literature and reflects the different angles at which we pursued decoding improvements.

The rich diversity is also seen in the various methods explored. Chapter 2 sprang from the desire to adapt the decoder, and the research required the development of an original framework to better define the problem. This framework was followed up by creation of an algorithm with a solid mathematical foundation. We arrived at MOCA via a maximum likelihood based derivation, with a central role being played by the L_0 penalty. The success of this method points to a certain level of redundancy in the neural code which we leverage to uncover units undergoing nonstationary behavior. The novel technology has broad applicability, and is not necessarily constrained to the world of iBCIs. MOCA can be directly applied to situations where nonstationarities prevail, but their small number and infrequency gives us a foothold to climb over the challenge.

In Chapter 3, we attempted to estimate the degree of decoding error perceived by the

person from the neural activity. Our findings yielded statistically significant, but fairly weak results, in terms of predictive power. These findings were thought to lack the promise to warrant further investigation. They did however, bring the use of LFPs into the investigation. The weakness of the LFP-amplitude-based detection as compared to the spike-based one suggests the former lacks a uniquely high level of informativeness. Recent publications point to the same trend [49, 155].

In contrast, there was notable success with mixed decoding in Chapter 4. Unlike MOCA, the fundamental algorithm is a simpler set of calculations. In fact, the implementation is a few lines of code. The deeper contribution of the work lies in breaking the equality between what the decoder decodes and how the device is controlled. Intended arm motions (velocity and position) are viewed as distinct from cursor commands. This leaves room for an additional computation step to sneak in between. The idea improved decoding in the offline tests, almost across the board. It also, as a general tactic, could be employed in other iBCI, EEG, or ECoG settings, where perhaps joint angles [69] are further processed to drive a not-necessarily anthromorphically structured robotic arm. Similar ideas could eventually make their way into an iBCI for locomotion. In this way, the gateway has been opened, where a whole host of techniques for various subapplications could be developed.

Chapter 5's use of the random forests in the two stage decoder extracted useful information from feature-to-feature interactions. The success was achieved without any a priori assumptions about the form of those interactions. Instead, the nonparametric algorithm constructed the associations from the data. The implications have immediate use to improve decoding in ongoing projects where a person actively controls an external device [24, 26]. Another advantage of the nonparametric approach is that no assumptions are made about the mathematical connection between the feature and hidden state vectors. Therefore, it can be directly used to decode from other feature types (LFPs) and/or estimate different hidden states (joint rotations).

The findings of Chapter 5 also have neuroscientific value. Here, we see that treating the encoding as a distributed code gives better accuracy. The findings add to the growing weight of evidence that regions of the brain such as the motor cortex convey information via the activity of neural ensembles. The idea we propose is akin to that advocated in other

reports [72, 154] that the spiking of any one neuron in the motor cortex typically lacks any easily discernible computational interpretation. Under this line of reasoning, each neuron is like a pixel on a computer screen, we must move beyond the fluctuation of each pixel, beyond simple pairwise correlations of pixels, and take in the whole screen to get a picture of what is being conveyed about behavior, perception, and/or cognitive state.

However, as good as the random forest was, the addition of the Kalman filter as a second stage made decoding even better. This came about because the random forest itself did not directly account for the dynamics in the hidden motor state. The a priori model of a random walk (approximately what the Kalman filter learns) was sufficient to enforce, it is believed, a level of smoothness in the dynamics, leading to the greater accuracy. More interesting, only after the results of the projects were in hand, did we see that all four of the projects used forms of hierarchy. The hierarchy in the two stage algorithm was apparent, being part of the approach's inception. However, one can see the hierarchy in the mixed approach as well, with the Kalman filter being the earlier stage and the mixing logic a layer of a yet higher stage. The first project too, resulting in the MOCA algorithm, also can be thought of in layers. The MOCA calculations that estimate the offsets are directly supplied with the extracted features and output offset corrections which then feed the Kalman filter. Estimation of error also would be sent to the Kalman filter. Figure 6.1 illustrates one possible way these four research thrusts could unify into a multi-layered hierarchical decoder.

The fact that we are sketching out a picture where banks of decoding modules feed their information to higher layers of decoding is of little surprise. The architecture is reminiscent of recent work with deep belief networks which is an active area of research in both academia and industry. Interestingly enough, deep belief networks have made strides in the field of artificial vision for object recognition [153]. There, some of the layers are specialized, for example, employing convolutional transformations, which are functionally thought similar to how our brains process imagery [156]. A bigger vision would be to make the original extraction of informative features from the neural signal learned partially through machine learning techniques as well. Such a featureless decoding strategy could uncover, as yet undiscovered, information latent in the voltage fluctuations streaming in

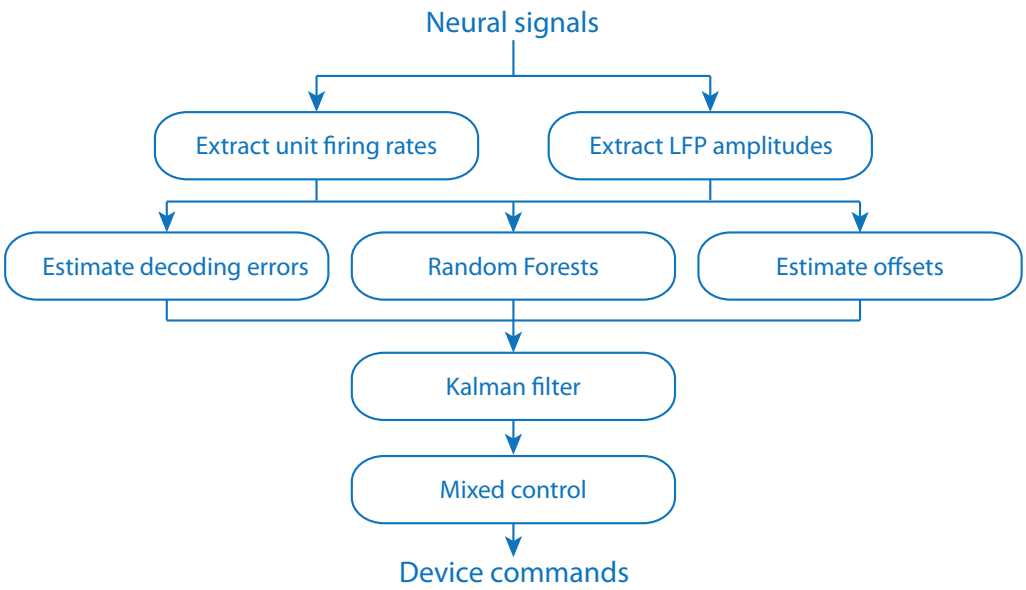

Figure 6.1: Illustration of how the various research projects in this thesis could be unified into a hierarchical decoder.

from the microelectrodes.

An immediate use of the technologies invented in this thesis lie in improving the accuracy and consistency of iBCIs for motor control. The next natural step is to evolve the algorithms for and evaluate them under real-time conditions. By real-time we mean having a person use an iBCI to control a computer cursor or other external device where one or more of the novel algorithms in this thesis partake in the decoding. While the offline setting is an effective test kitchen to cook up new ideas and rapidly iterate to refine them, real-time evaluation is a better measure of effectiveness. Fortunately, the developed methods can support the required calculation deadlines (1 to 100 msec). Also, the three methods are feature agnostic in the event that future decoding not only works off of spike counts, but also adds features derived from local field potentials, for instance. Regarding the software implementation itself, the code does not rely on proprietary subroutines which may be difficult to port to other languages or difficult to further improve. For example, the solution to eigensystems is sufficiently difficult that developing the routines from scratch is not advised [157]. The algorithms are sufficiently transparent to be implementable in typical programming languages of choice.

As these algorithms get evaluated, however, it does not mean that their refinement will

stop. For the MOCA algorithm, the suboptimal greedy search is within its core. Forays can be made to improve the search, such as backward stepwise search to remove offset components. With the mixed control, we propose opening up the disconnect even further between what is decoded and what controls the cursor. Added variables, such as joint angle, or functions of kinematic variables, such as the natural log of the horizontal velocity, could be added to the list of hidden states. Doing so would make the mixture step more complicated and the Kalman filter might need to handle nonlinear equations and thus change to an extended or even unscented version. Right now the random forest works with 100 trees to insure calibration is feasible in a few minutes. Another option is to setup the algorithm to run in parallel on a large cluster and send the results back to the iBCI cart or future embedded system. Doing so might enable 1,000 or even 10,000 trees to be included in the ensemble.

We propose a future course in the science direction as well. When models for how nonstiationarities enter into the tuning model or how interactions contain information about latent motor states improve decoding performance, the gains may be because those models capture how the nervous system itself decodes. Therefore, we have the opportunity to shed light on how the brain processes information. The investigation path would be similar to work done in the prefrontal cortex to understand mixed selectivity [154]. The MOCA algorithm thus could be used as a tool to probe more carefully the dynamics of fast nonstationarities (over a few seconds), a topic not yet well covered in the literature. The mixed controller could help to represent the downstream transformations necessary to covert the information best associated with the recorded neural population with commands that ultimately move a muscle's motor unit. This fact may become even more apparent when an FES system is used to move the participant's own arm. For example, suppose we can connect the FES to activate muscles, but find joint velocities give people the best control, then we start to have evidence that it is joint angles that are most closely connected with the information contained within the neural signals we record. Such insights can occur with random forests as well. Using them as an investigational tool can reveal the true distributed nature of the motor cortex, or even other cortical regions.

In these ways, the contributions of the thesis positively impacts two communities: the

neural engineering field and the scientific one. Furthermore, the algorithms developed are general enough to accommodate future renditions of the decoder in this rapidly developing field. Even with MOCA, where the derivation on the exact algorithm is based upon the Kalman filter, the maximum likelihood framework with L_0 norm can still be applied should a different decoder be used. The mixed control and two stage control are even more agnostic to the core decoder used. The implication is that the techniques are unlikely to be found obsolete in the near term. Even more broadly, as mentioned above, they open up new ways of thinking about decoding neural signals and the ideas will hopefully find their way into a multitude of future work. The prospect is that, in the years to come, our scientific knowledge and technical know how will be sufficient enough to restore freedom of movement to individuals who have lost the ability.

Bibliography

[1] M. L. Homer, A. V. Nurmikko, J. P. Donoghue, and L. R. Hochberg, "Sensors and decoding for intracortical brain computer interfaces," *Annual review of biomedical engineering*, vol. 15, pp. 383–405, 2013.

[2] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the United States: 2005 to 2050," *Archives of Physical Medicine and Rehabilitation*, vol. 89, no. 3, pp. 422–429, 2008.

[3] "One degree of separation: Paralysis and spinal cord injury in the United States," *CDR Foundation*, 2009.

[4] A. B. Jackson, M. Dijkers, M. J. DeVivo, and R. B. Poczatek, "A demographic profile of new traumatic spinal cord injuries: change and stability over 30 years," *Archives of Physical Medicine and Rehabilitation*, vol. 85, no. 11, pp. 1740–1748, 2004.

[5] J. N. Mak and J. R. Wolpaw, "Clinical applications of brain-computer interfaces: current state and future prospects," *Biomedical Engineering, IEEE Reviews in*, vol. 2, pp. 187–199, 2009.

[6] G. Schalk and E. C. Leuthardt, "Brain-computer interfaces using electrocorticographic signals," *Biomedical Engineering, IEEE Reviews in*, vol. 4, pp. 140–154, 2011.

[7] E. V. Evarts, "Relation of pyramidal tract activity to force exerted during voluntary movement," *J Neurophysiol*, vol. 31, no. 1, pp. 14–27, 1968.

[8] E. Fetz, "Operant conditioning of cortical unit activity," *Science*, vol. 163, no. 3870, pp. 955–958, 1969.

[9] D. R. Humphrey, E. Schmidt, and W. Thompson, "Predicting measures of motor performance from multiple cortical spike trains," *Science*, vol. 170, no. 3959, pp. 758–762, 1970.

[10] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, no. 4771, pp. 1416–1419, 1986.

[11] J. F. Kalaska and D. J. Crammond, "Cerebral cortical mechanisms of reaching movements," *Science*, vol. 255, no. 5051, pp. 1517–1523, 1992.

[12] J. N. Sanes and J. P. Donoghue, "Oscillations in local field potentials of the primate motor cortex during voluntary movement." *Proceedings of the National Academy of Sciences*, vol. 90, no. 10, pp. 4470–4474, 1993.

[13] S. L. BeMent, K. D. Wise, D. J. Anderson, K. Najafi, and K. L. Drake, "Solid-state electrodes for multichannel multiplexed intracortical neuronal recording," *Biomedical Engineering, IEEE Transactions on*, no. 2, pp. 230–241, 1986.

[14] E. M. Maynard, C. T. Nordhausen, and R. A. Normann, "The Utah intracortical electrode array: a recording structure for potential brain-computer interfaces," *Electroencephalography and clinical Neurophysiology*, vol. 102, no. 3, pp. 228–239, 1997.

[15] M. D. Burrow and D. Humphrey, "Cortical control of a robot using a time-delay neural network," in *International Conference on Rehabilitation Robotics*, 1997.

[16] D. R. Humphrey, L. Hochberg, M. Burrow, and J. Dugger, "Cortical control of neural prosthetic devices, NIH/NINDS contract N01-NS-1-2308," Tech. Rep., 1997.

[17] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, vol. 2, no. 7, pp. 664–670, 1999.

[18] P. R. Kennedy, R. A. E. Bakay, M. M. Moore, K. Adams, and J. Goldwaithe, "Direct control of a computer from the human central nervous system," *Rehabilitation Engineering, IEEE Transactions on*, vol. 8, no. 2, pp. 198–202, 2000.

[19] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.

[20] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.

[21] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biology*, vol. 1, no. 2, pp. 193–208, 2003.

[22] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, no. 5681, pp. 258–262, 2004.

[23] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain-computer interface," *Nature*, vol. 442, no. 7099, pp. 195–198, 2006.

[24] 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*, vol. 442, no. 7099, pp. 164–171, 2006.

[25] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[26] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.

[27] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. C. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, "High-

performance neuroprosthetic control by an individual with tetraplegia," *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.

[28] C. T. Moritz, S. I. Perlmutter, and E. E. Fetz, "Direct control of paralysed muscles by cortical neurons," *Nature*, vol. 456, no. 7222, pp. 639–642, 2008.

[29] E. A. Pohlmeyer, E. R. Oby, E. J. Perreault, S. A. Solla, K. L. Kilgore, R. F. Kirsch, and L. E. Miller, "Toward the restoration of hand use to a paralyzed monkey: brain-controlled functional electrical stimulation of forearm muscles," *PloS One*, vol. 4, no. 6, p. e5924, 2009.

[30] C. Ethier, E. Oby, M. Bauman, and L. Miller, "Restoration of grasp following paralysis through brain-controlled stimulation of muscles," *Nature*, vol. 485, no. 7398, pp. 368–371, 2012.

[31] E. Chadwick, D. Blana, J. Simeral, J. Lambrecht, S. Kim, A. Cornwell, D. Taylor, L. Hochberg, J. Donoghue, and R. Kirsch, "Continuous neuronal ensemble control of simulated arm reaching by a human with tetraplegia," *Journal of Neural Engineering*, vol. 8, no. 3, p. 034003, 2011.

[32] M. D. Linderman, G. Santhanam, C. T. Kemere, V. Gilja, S. O'Driscoll, B. M. Yu, A. Afshar, S. I. Ryu, K. V. Shenoy, and T. H. Meng, "Signal processing challenges for neural prostheses," *Signal Processing Magazine, IEEE*, vol. 25, no. 1, pp. 18–28, 2008.

[33] K. Ganguly and J. M. Carmena, "Emergence of a stable cortical map for neuroprosthetic control," *PLoS Biology*, vol. 7, no. 7, p. e1000153, 2009.

[34] B. Jarosiewicz, S. M. Chase, G. W. Fraser, M. Velliste, R. E. Kass, and A. B. Schwartz, "Functional network reorganization during learning in a brain-computer interface paradigm," *Proceedings of the National Academy of Sciences*, vol. 105, no. 49, pp. 19 486–19 491, 2008.

[35] J. Wolpaw and E. W. Wolpaw, *Brain-computer interfaces: principles and practice*. Oxford University Press, 2012.

[36] A. Gelb, *Applied Optimal Estimation*. Cambridge, US: MIT Press, 1974.

[37] C. A. Chestek, V. Gilja, P. Nuyujukian, J. D. Foster, J. M. Fan, M. T. Kaufman, M. M. Churchland, Z. Rivera-Alvidrez, J. P. Cunningham, S. I. Ryu, and K. V. Shenoy, "Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex," *Journal of Neural Engineering*, vol. 8, no. 4, p. 045005, 2011.

[38] J. A. Perge, M. L. Homer, W. Q. Malik, S. Cash, E. Eskandar, G. Friehs, J. P. Donoghue, and L. R. Hochberg, "Intra-day signal instabilities affect decoding performance in an intracortical neural interface system," *Journal of Neural Engineering*, vol. 10, no. 3, p. 036004, 2013.

[39] G. W. Fraser, S. M. Chase, A. Whitford, and A. B. Schwartz, "Control of a brain–computer interface without spike sorting," *Journal of Neural Engineering*, vol. 6, no. 5, p. 055004, 2009.

[40] D. R. Humphrey, "Systems, methods, and devices for controlling external devices by signals derived directly from the nervous system," 2001, US Patent 6,171,239.

[41] E. Stark and M. Abeles, "Predicting movement from multiunit activity," *The Journal of Neuroscience*, vol. 27, no. 31, pp. 8387–8394, 2007.

[42] J. Zhuang, W. Truccolo, C. Vargas-Irwin, and J. P. Donoghue, "Decoding 3-d reach and grasp kinematics from high-frequency local field potentials in primate primary motor cortex," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 7, pp. 1774–1784, 2010.

[43] W. Malik, S. Stavisky, D. Bacher, J. Simeral, W. Truccolo, E. Brown, J. Donoghue, and L. Hochberg, "Decoding multiunit activity in neural interfaces for individuals with tetraplegia," in *Society for Neuroscience Meeting Planner*, 2010.

[44] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currentseeg, ecog, lfp and spikes," *Nature Reviews Neuroscience*, vol. 13, no. 6, pp. 407–420, 2012.

[45] J. Bouyer, M. Montaron, and A. Rougeul, "Fast fronto-parietal rhythms during combined focused attentive behaviour and immobility in cat: cortical and thalamic localizations," *Electroencephalography and Clinical Neurophysiology*, vol. 51, no. 3, pp. 244–252, 1981.

[46] C. Mehring, J. Rickert, E. Vaadia, S. C. de Oliveira, A. Aertsen, and S. Rotter, "Inference of hand movements from local field potentials in monkey motor cortex," *Nature Neuroscience*, vol. 6, no. 12, pp. 1253–1254, 2003.

[47] J. Rickert, S. C. de Oliveira, E. Vaadia, A. Aertsen, S. Rotter, and C. Mehring, "Encoding of movement direction in different frequency ranges of motor cortical local field potentials," *The Journal of Neuroscience*, vol. 25, no. 39, pp. 8815–8824, 2005.

[48] H. Scherberger, M. R. Jarvis, and R. A. Andersen, "Cortical local field potential encodes movement intentions in the posterior parietal cortex," *Neuron*, vol. 46, no. 2, pp. 347–354, 2005.

[49] A. K. Bansal, W. Truccolo, C. E. Vargas-Irwin, and J. P. Donoghue, "Decoding 3d reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials," *Journal of Neurophysiology*, vol. 107, no. 5, pp. 1337–1355, 2012.

[50] N. F. Ince, R. Gupta, S. Arica, A. H. Tewfik, J. Ashe, and G. Pellizzer, "High accuracy decoding of movement target direction in non-human primates based on common spatial patterns of local field potentials," *PloS One*, vol. 5, no. 12, p. e14384, 2010.

[51] A. K. Bansal, C. E. Vargas-Irwin, W. Truccolo, and J. P. Donoghue, "Relationships among low-frequency local field potentials, spiking activity, and three-dimensional reach and grasp kinematics in primary motor and ventral premotor cortices," *Journal of Neurophysiology*, vol. 105, no. 4, pp. 1603–1619, 2011.

[52] R. D. Flint, E. W. Lindberg, L. R. Jordan, L. E. Miller, and M. W. Slutzky, "Accurate decoding of reaching movements from field potentials in the absence of spikes," *Journal of Neural Engineering*, vol. 9, no. 4, p. 046006, 2012.

[53] A. Gunduz, P. Brunner, A. Daitch, E. C. Leuthardt, A. L. Ritaccio, B. Pesaran, and G. Schalk, "Decoding covert spatial attention using electrocorticographic (ECoG) signals in humans," *NeuroImage*, vol. 60, no. 4, pp. 2285–2293, 2012.

[54] E. J. Hwang and R. A. Andersen, "Brain control of movement execution onset using local field potentials in posterior parietal cortex," *The Journal of Neuroscience*, vol. 29, no. 45, pp. 14 363–14 370, 2009.

[55] D. Rubino, K. A. Robbins, and N. G. Hatsopoulos, "Propagating waves mediate information transfer in the motor cortex," *Nature Neuroscience*, vol. 9, no. 12, pp. 1549–1557, 2006.

[56] S. N. Baker, "Oscillatory interactions between sensorimotor cortex and the periphery," *Current Opinion in Neurobiology*, vol. 17, no. 6, pp. 649–655, 2007.

[57] J. Reimer and N. G. Hatsopoulos, "Periodicity and evoked responses in motor cortex," *The Journal of Neuroscience*, vol. 30, no. 34, pp. 11 506–11 515, 2010.

[58] J. D. Simeral, S. P. Kim, M. J. Black, J. P. Donoghue, and L. R. Hochberg, "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array," *Journal of Neural Engineering*, vol. 8, no. 2, p. 025027, 2011.

[59] D. Tkach, J. Reimer, and N. G. Hatsopoulos, "Observation-based learning for brain–machine interfaces," *Current Opinion in Neurobiology*, vol. 18, no. 6, pp. 589–594, 2008.

[60] J. Dushanova and J. Donoghue, "Neurons in primary motor cortex engaged during action observation," *European Journal of Neuroscience*, vol. 31, no. 2, pp. 386–398, 2010.

[61] J. Feldman, B. King, W. Truccolo, L. Hochberg, and J. Donoghue, "Decoding neural representations of action from motor cortex ensembles during action observation in humans with tetraplegia," in *Society for Neuroscience Meeting Planner*, 2011.

[62] A. L. Orsborn, S. Dangi, H. G. Moorman, and J. M. Carmena, "Closed-loop decoder adaptation on intermediate time-scales facilitates rapid bmi performance improvements independent of decoder initialization conditions," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 20, no. 4, pp. 468–477, 2012.

[63] B. Jarosiewicz, N. Y. Masse, D. Bacher, S. S. Cash, E. Eskandar, G. Friehs, J. P. Donoghue, and L. R. Hochberg, "Advantages of closed-loop calibration in intracortical brain–computer interfaces for people with tetraplegia," *Journal of Neural Engineering*, vol. 10, no. 4, p. 046012, 2013.

[64] A. M. Green and J. F. Kalaska, "Learning to move machines with the mind," *Trends in Neurosciences*, vol. 34, no. 2, pp. 61–75, 2011.

[65] A. C. Koralek, X. Jin, J. D. Long II, R. M. Costa, and J. M. Carmena, "Corticostriatal plasticity is necessary for learning intentional neuroprosthetic skills," *Nature*, vol. 483, no. 7389, pp. 331–335, 2012.

[66] W. Wu and N. G. Hatsopoulos, "Real-time decoding of nonstationary neural activity in motor cortex," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 3, pp. 213–222, 2008.

[67] Z. Li, J. E. O'Doherty, M. A. Lebedev, and M. A. L. Nicolelis, "Adaptive decoding for brain-machine interfaces through Bayesian parameter updates," *Neural Computation*, vol. 23, no. 12, pp. 3162–3204, 2011.

[68] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer, 2009.

[69] C. E. Vargas-Irwin, G. Shakhnarovich, P. Yadollahpour, J. M. Mislow, M. J. Black, and J. P. Donoghue, "Decoding complete reach and grasp actions from local primary motor cortex populations," *The Journal of Neuroscience*, vol. 30, no. 29, pp. 9659–9669, 2010.

[70] M. Y. Byron, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Gaussian-process factor analysis for low-dimensional single-trial analysis of

neural population activity," *Journal of Neurophysiology*, vol. 102, no. 1, pp. 614–635, 2009.

[71] G. Santhanam, M. Y. Byron, V. Gilja, S. I. Ryu, A. Afshar, M. Sahani, and K. V. Shenoy, "Factor-analysis methods for higher-performance neural prostheses," *Journal of Neurophysiology*, vol. 102, no. 2, pp. 1315–1330, 2009.

[72] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "Neural population dynamics during reaching," *Nature*, 2012.

[73] Z. C. Chao, Y. Nagasaka, and N. Fujii, "Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys," *Frontiers in neuroengineering*, vol. 3, 2010.

[74] K. D. Anderson, "Targeting recovery: priorities of the spinal cord-injured population," *Journal of neurotrauma*, vol. 21, no. 10, pp. 1371–1383, 2004.

[75] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time Signal Processing*. Prentice Hall Upper Saddle River, 1999, vol. 5.

[76] S. P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, and M. J. Black, "Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia," *Journal of Neural Engineering*, vol. 5, no. 4, pp. 455–476, 2008.

[77] W. Q. Malik, W. Truccolo, E. N. Brown, and L. R. Hochberg, "Efficient decoding with steady-state kalman filter in neural interface systems," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 19, no. 1, pp. 25–34, 2011.

[78] L. Paninski, S. Shoham, M. R. Fellows, N. G. Hatsopoulos, and J. P. Donoghue, "Superlinear population encoding of dynamic hand trajectory in primary motor cortex," *The Journal of Neuroscience*, vol. 24, no. 39, pp. 8551–8561, 2004.

[79] J. Fisher and M. J. Black, "Motor cortical decoding using an autoregressive moving average model," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the.* IEEE, 2006, pp. 2130–2133.

[80] L. Shpigelman, H. Lalazar, and E. Vaadia, "Kernel-arma for hand tracking and brain–machine interfacing during 3D motor control," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1489–1496, 2009.

[81] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. Nicolelis, "Unscented kalman filter for brain-machine interfaces," *PLoS One*, vol. 4, no. 7, p. e6243, 2009.

[82] Y. Gao, M. Black, E. Bienenstock, S. Shoham, and J. Donoghue, "Probabilistic inference of hand motion from neural activity in motor cortex," *Advances in Neural Information Processing Systems*, vol. 1, pp. 213–220, 2002.

[83] A. Brockwell, A. Rojas, and R. Kass, "Recursive bayesian decoding of motor cortical signals by particle filtering," *Journal of Neurophysiology*, vol. 91, no. 4, pp. 1899–1907, 2004.

[84] F. Wood, J. P. Donoghue, M. J. Black *et al.*, "Inferring attentional state and kinematics from motor cortical firing rates," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the.* IEEE, 2006, pp. 149–152.

[85] D. Sussillo, P. Nuyujukian, J. M. Fan, J. C. Kao, S. D. Stavisky, S. Ryu, and K. Shenoy, "A recurrent neural network for closed-loop intracortical brain–machine interface decoders," *Journal of neural engineering*, vol. 9, no. 2, p. 026027, 2012.

[86] L. Srinivasan, U. T. Eden, S. K. Mitter, and E. N. Brown, "General-purpose filter design for neural prosthetic devices," *Journal of Neurophysiology*, vol. 98, no. 4, pp. 2456–2475, 2007.

[87] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue, "A switching kalman filter model for the motor cortical coding of hand motion," in

Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE, vol. 3. IEEE, 2003, pp. 2083–2086.

[88] V. Lawhern, W. Wu, N. Hatsopoulos, and L. Paninski, "Population decoding of motor cortical activity using a generalized linear model with hidden states," *Journal of Neuroscience Methods*, vol. 189, no. 2, pp. 267–280, 2010.

[89] S. M. Chase, A. B. Schwartz, and R. E. Kass, "Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain–computer interface algorithms," *Neural Networks*, vol. 22, no. 9, pp. 1203–1213, 2009.

[90] E. N. Brown, L. M. Frank, D. Tang, M. C. Quirk, and M. A. Wilson, "A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells," *The Journal of Neuroscience*, vol. 18, no. 18, pp. 7411–7425, 1998.

[91] W. Truccolo, G. M. Friehs, J. P. Donoghue, and L. R. Hochberg, "Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia," *The Journal of Neuroscience*, vol. 28, no. 5, pp. 1163–1178, 2008.

[92] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects," *Journal of Neurophysiology*, vol. 93, no. 2, pp. 1074–1089, 2005.

[93] K. Nazarpour, C. Ethier, L. Paninski, J. M. Rebesco, R. C. Miall, and L. E. Miller, "Emg prediction from motor cortical recordings via a nonnegative point-process filter," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 7, pp. 1829–1838, 2012.

[94] M. M. Shanechi, Z. M. Williams, G. W. Wornell, R. C. Hu, M. Powers, and E. N. Brown, "A real-time brain-machine interface combining motor target and trajectory

intent using an optimal feedback control design," *PloS One*, vol. 8, no. 4, p. e59049, 2013.

[95] C. Kemere, G. Santhanam, M. Y. Byron, A. Afshar, S. I. Ryu, T. H. Meng, and K. V. Shenoy, "Detecting neural-state transitions using hidden markov models for motor cortical prostheses," *Journal of Neurophysiology*, vol. 100, no. 4, pp. 2441–2452, 2008.

[96] M. Campos, B. Breznen, and R. A. Andersen, "A neural representation of sequential states within an instructed task," *Journal of neurophysiology*, vol. 104, no. 5, pp. 2831–2849, 2010.

[97] S. Kakei, D. S. Hoffman, and P. L. Strick, "Sensorimotor transformations in cortical motor areas," *Neuroscience Research*, vol. 46, no. 1, pp. 1–10, 2003.

[98] A. H. Fagg, G. W. Ojakangas, L. E. Miller, and N. G. Hatsopoulos, "Kinetic trajectory decoding using motor cortical ensembles," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 5, pp. 487–496, 2009.

[99] R. Gupta and J. Ashe, "Offline decoding of end-point forces using neural ensembles: application to a brain–machine interface," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 254–262, 2009.

[100] S. Acharya, F. Tenore, V. Aggarwal, R. Etienne-Cummings, M. H. Schieber, and N. V. Thakor, "Decoding individuated finger movements using volume-constrained neuronal ensembles in the m1 hand area," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 1, pp. 15–23, 2008.

[101] J. Baker, W. Bishop, S. Kellis, T. Levy, P. House, and B. Greger, "Multi-scale recordings for neuroprosthetic control of finger movements," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE.* IEEE, 2009, pp. 4573–4577.

[102] N. A. Fitzsimmons, M. A. Lebedev, I. D. Peikon, and M. A. Nicolelis, "Extracting kinematic parameters for monkey bipedal walking from cortical neuronal ensemble activity," *Frontiers in Integrative Neuroscience*, vol. 3, 2009.

[103] W. Song and S. F. Giszter, "Adaptation to a cortex-controlled robot attached at the pelvis and engaged during locomotion in rats," *The Journal of Neuroscience*, vol. 31, no. 8, pp. 3110–3128, 2011.

[104] S. P. Kim, F. Wood, M. Fellows, J. P. Donoghue, and M. J. Black, "Statistical analysis of the non-stationarity of neural population codes," in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on.* IEEE, 2006, pp. 811–816.

[105] V. Gilja, C. A. Chestek, I. Diester, J. M. Henderson, K. Deisseroth, and K. V. Shenoy, "Challenges and opportunities for next-generation intracortically based neural prostheses," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 7, pp. 1891–1899, 2011.

[106] A. P. Batista, B. M. Yu, G. Santhanam, S. I. Ryu, A. Afshar, and K. V. Shenoy, "Cortical neural prosthesis performance improves when eye position is monitored," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 1, pp. 24–31, 2008.

[107] M. L. Homer, J. A. Perge, M. J. Black, M. T. Harrison, S. S. Cash, and L. R. Hochberg, "Adaptive offset correction for intracortical brain computer interfaces," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 2013.

[108] G. Schalk, J. R. Wolpaw, D. J. McFarland, and G. Pfurtscheller, "EEG-based communication: presence of an error potential," *Clinical Neurophysiology*, vol. 111, no. 12, pp. 2138–2144, 2000.

[109] M. L. Homer, M. T. Harrison, M. J. Black, J. A. Perge, S. S. Cash, G. Friehs, and L. R. Hochberg, "Mixing decoded cursor velocity and position from an offline kalman filter improves cursor control in people with tetraplegia," in *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on.* IEEE, 2013, pp. 715–718.

[110] J. Wessberg and M. A. L. Nicolelis, "Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys," *Journal of Cognitive Neuroscience*, vol. 16, no. 6, pp. 1022–1035, 2004.

[111] U. Rokni, A. G. Richardson, E. Bizzi, and H. S. Seung, "Motor learning with unstable neural representations," *Neuron*, vol. 54, no. 4, pp. 653–666, 2007.

[112] J. A. Donoghue, "Stability of continuous multi-electrode recordings in motor cortex during control of a neural interface system," Brown University Senior Honors Undergraduate Thesis, 2009.

[113] C. A. Chestek, A. P. Batista, G. Santhanam, B. M. Yu, A. Afshar, J. P. Cunningham, V. Gilja, S. I. Ryu, M. M. Churchland, and K. V. Shenoy, "Single-neuron stability during repeated reaching in macaque premotor cortex," *Journal of Neuroscience*, vol. 27, no. 40, pp. 10 742–10 750, 2007.

[114] G. Baranauskas, E. Maggiolini, E. Castagnola, A. Ansaldo, A. Mazzoni, G. N. Angotzi, A. Vato, D. Ricci, S. Panzeri, and L. Fadiga, "Carbon nanotube composite coating of neural microelectrodes preferentially improves the multiunit signal-to-noise ratio," *Journal of Neural Engineering*, vol. 8, no. 6, p. 066013, 2011.

[115] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and S. K. V., "A high-performance neural prosthesis enabled by control algorithm design," *Nature Neuroscience*, vol. 15, no. 12, pp. 1752–1757, 2012.

[116] T. Gürel and C. Mehring, "Unsupervised adaptation of brain machine interface decoders," *Frontiers in Neuroscience*, vol. 6, no. 164, 2012.

[117] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Computation*, vol. 16, no. 5, pp. 971–998, 2004.

[118] U. T. Eden, W. Truccolo, M. R. Fellows, J. P. Donoghue, and E. N. Brown, "Reconstruction of hand movement trajectories from a dynamic ensemble of spiking motor

cortical neurons," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 2. IEEE, 2004, pp. 4017–4020.

[119] Y. Wang and J. C. Principe, "Tracking the non-stationary neuron tuning by dual kalman filter for brain machine interfaces decoding," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 1720–1723.

[120] P. Shenoy, M. Krauledat, B. Blankertz, R. P. Rao, and K.-R. Müller, "Towards adaptive classification for bci," *Journal of neural engineering*, vol. 3, no. 1, p. R13, 2006.

[121] P. von Bünau, F. C. Meinecke, F. C. Király, and K.-R. Müller, "Finding stationary subspaces in multivariate time series," *Physical Review Letters*, vol. 103, no. 21, p. 214101, 2009.

[122] D. A. Blythe, F. C. Meinecke, P. von Bünau, and K.-R. Müller, "Explorative data analysis for changes in neural activity," *Journal of neural engineering*, vol. 10, no. 2, p. 026018, 2013.

[123] C.-S. Hsieh, "Robust two-stage kalman filters for systems with unknown inputs," *Automatic Control, IEEE Transactions on*, vol. 45, no. 12, pp. 2374–2378, 2000.

[124] P. K. Kitanidis, "Unbiased minimum-variance linear state estimation," *Automatica*, vol. 23, no. 6, pp. 775–778, 1987.

[125] P. Sanyal. and C. Shen, "Rapid estimation by detecting probabilistically unknown impulse inputs," in *Decision and Control, 1972 and 11th Symposium on Adaptive Processes. Proceedings of the 1972 IEEE Conference on*, vol. 11. IEEE, 1972, pp. 248–252.

[126] P. Sanyal and C. Shen, "Bayes' decision rule for rapid detection and adaptive estimation scheme with space applications," *Automatic Control, IEEE Transactions on*, vol. 19, no. 3, pp. 228–231, 1974.

[127] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation.* New York: Wiley-Interscience, 2001.

[128] W. Wu, A. Shaikhouni, J. R. Donoghue, and M. J. Black, "Closed-loop neural control of cursor motion using a Kalman filter," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 2. IEEE, 2004, pp. 4126–4129.

[129] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a kalman filter," *Neural Computation*, vol. 18, no. 1, pp. 80–118, 2006.

[130] A. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge, UK: Cambridge University Press, 1991.

[131] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* New York: Academic Press, 1970.

[132] Y. Wang and J. C. Principe, "Tracking the non-stationary neuron tuning by dual kalman filter for brain machine interfaces decoding," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE.* IEEE, 2008, pp. 1720–1723.

[133] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[134] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer-Verlag, 2001.

[135] T. A. Yousry, U. D. Schmid, H. Alkadhi, D. Schmidt, A. Peraud, A. Buettner, and P. Winkler, "Localization of the motor hand area to a knob on the precentral gyrus. a new landmark," *Brain*, vol. 120, no. 1, pp. 141–157, 1997.

[136] Z. Li, J. E. O'Doherty, M. A. Lebedev, and M. A. L. Nicolelis, "Adaptive decoding for brain-machine interfaces through Bayesian parameter updates," *Neural Computation*, vol. 23, no. 12, pp. 3162–3204, 2011.

[137] L. C. Parra, C. D. Spence, A. D. Gerson, and P. Sajda, "Response error correction-a demonstration of improved human-machine performance using real-time eeg monitoring," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 11, no. 2, pp. 173–177, 2003.

[138] T. Milekovic, T. Ball, A. Schulze-Bonhage, A. Aertsen, and C. Mehring, "Error-related electrocorticographic activity in humans during continuous movements," *Journal of Neural Engineering*, vol. 9, no. 2, p. 026007, 2012.

[139] G. Der and I. J. Deary, "Age and sex differences in reaction time in adulthood: results from the united kingdom health and lifestyle survey." *Psychology and Aging*, vol. 21, no. 1, p. 62, 2006.

[140] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[141] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 96–103.

[142] P.-J. Kindermans, D. Verstraeten, and B. Schrauwen, "A bayesian model for exploiting application constraints to enable unsupervised training of a p300-based bci," *PloS one*, vol. 7, no. 4, p. e33758, 2012.

[143] N. G. Hatsopoulos, Q. Xu, and Y. Amit, "Encoding of movement fragments in the motor cortex," *The Journal of Neuroscience*, vol. 27, no. 19, pp. 5105–5114, 2007.

[144] W. Wang, S. S. Chan, D. A. Heldman, and D. W. Moran, "Motor cortical representation of position and velocity during reaching," *Journal of Neurophysiology*, vol. 97, no. 6, pp. 4258–4270, 2007.

[145] W. Truccolo and J. P. Donoghue, "Nonparametric modeling of neural point processes via stochastic gradient boosting regression," *Neural Computation*, vol. 19, no. 3, pp. 672–705, 2007.

[146] N. Rao, "Cortical dynamics of movement planning, execution, and sensory integration in parietal-motor networks," Ph.D. dissertation, Brown University, 2012.

[147] S. H. Scott, "Apparatus for measuring and perturbing shoulder and elbow joint positions and torques during reaching," *Journal of Neuroscience Methods*, vol. 89, no. 2, pp. 119–127, 1999.

[148] C. Vargas-Irwin and J. P. Donoghue, "Automated spike sorting using density grid contour clustering and subtractive waveform decomposition," *Journal of Neuroscience Methods*, vol. 164, no. 1, pp. 1–18, 2007.

[149] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[150] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*. springer New York, 2006, vol. 1.

[151] Y. Lin and Y. Jeon, "Random forests and adaptive nearest neighbors," *Journal of the American Statistical Association*, vol. 101, no. 474, pp. 578–590, 2006.

[152] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[153] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[154] M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi, "The importance of mixed selectivity in complex cognitive tasks," *Nature*, 2013.

[155] J. Perge, S. Zhang, W. Malik, M. Homer, J. Donoghue, and L. Hochberg, "Reliability of directional information in human motor cortex multiunit spikes and local field potentials," *Journal of neural engineering*, submitted.

[156] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 994–1000.

[157] W. H. Press, B. P. Flannery, and S. Teukolsky, "Numerical recipes in C: The art of scientific computing," 2002.