UNIVERSITY OF CALIFORNIA, SAN DIEGO

Exploratory studies of human sensorimotor learning with system identification and stochastic optimal control

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Engineering Sciences (Mechanical Engineering)

by

Charles Alexander Simpkins, Jr.

Committee in charge:

Professor Emanuel Todorov, Chair
Professor Raymond de Callafon, Co-Chair
Professor Seana Coulson
Professor Frank Talke
Professor George Tynan

2009

The dissertation of Charles Alexander Simpkins, Jr. is approved, and it is acceptable in quality and form for publication on microfilm:

Co-Chair

Chair

University of California, San Diego

2009

DEDICATION

To my parents.

EPIGRAPH

If Life were Only an Equation

...But then I found an answer
The world of simple functions
Was only the beginning to
The world of simple logic
Beyond pi's function:
You...

C. Alexander Simpkins, Sr.

1995

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The author would first like to thank his advisors, Prof. Emanuel Todorov and Prof. Raymond A. de Callafon. Their guidance, encouragement, and support have made this work possible. Not only have I learned the specifics of control theory, motor control (not to mention the differences between *biological* and *mechanical* motor control theory), research and development, but I have also learned a great many lessons regarding teaching, mentoring, and life.

The author would next like to thank the rest of his committee (in no particular order) - Prof. Frank Talke, Prof. George Tynan, and Prof. Seana Coulson. I am honored to have been advised by such great individuals. I have taken courses from most of you (and from any whom I haven't, I know your students and the positive words they speak of you) and the world is lucky to have the contributions of you all as teachers, researchers, and advisors. I hope to continue to develop and grow as you have all suggested in various ways over my entire life.

The author would like to thank his family. I would not exist without you - mind, body, or spirit. You have made this possible. Your support and encouragement have driven me forward through the best of times and the worst of times. In fact, I amend the last statement to say "...and the most challenging of times," for times have never been bad thanks to you and your presence always in my life. I love you all so much.

The entire Movement Control Lab (in no particular order), Weiwei Li, Dan Liu, Ben Huh, Miro Enev, Yuval Tussa, Rika Yatchak, and Mike Kelley, (past and present) needs mentioning as you have all been like a family. The Natural Computation Group is also included in this thanks. I look forward to our continued involvement and collaborations in the future. The input, stimulating discussions,

and humor have made the times 'living in the lab' great. Mike Kelley (part of both the MCL and the SICL) has been particularly supportive and I have enjoyed working with him on our many projects. The light-heartedness he brought into the lab, and our friendship has helped make it a very pleasant place to be. The author has received strong support and encouragement from Joanne Jao, part of the NatComp Group as well, and would like to thank her. Our closeness has been wonderful, and I am looking forward to witnessing and being connected with the great works you do.

The System Identification and Control Laboratory (SICL) has been a source of excellence and experience both theoretically and culturally. The lab meetings have been very interesting in particular and provided a forum for gaining experience with not only one's own project, but also others. The guidance and challenges Prof. Raymond de Callafon provided during these meetings stimulated intellectual progress for everyone. In particular the author wishes to acknowledge Gabriel Graham who has continued to be a friend since our first meeting. It has been wonderful to see and be part of your evolution from a gifted student to an excellent engineer.

The author would like to thank Dr. Nate Delson as well. I have always considered you to be an additional advisor. You have given input during many decisions along the way, and have provided many opportunities. I hope I have contributed to your work at UCSD in a good way, and that I continue to do so in the future.

The author would like to thank Chris Cassidy. You have also been an advisor and friend during the past several years. Our conversations have always been stimulating, and your support and input have been very helpful as well. I hope we continue to work together as time passes.

The technical and administrative staff of the MAE and Cognitive Science department at UCSD have been supportive as well. The university has excellent advisors and technical support staff for nearly all challenges students face, and I appreciate the guidance I received.

To Prof. Allen M. Schneider with whom I began my graduate work, thank you for your input over the years. Though you were not my advisor for the whole of my graduate time, I came a long way with your input. Your commitment to teaching is very admirable, and I feel that I had an opportunity to work with one of the pioneering masters of UCSD.

To anyone I have not mentioned, anyone who was helpful or encouraging, thank you. It is too many to mention by name but I am very thankful to all my professors for the time they gave me and which I pass on to the new generation as it goes to infinity (positive divergence). I am glad to have known you and look forward to continuing to be connected in the future. This is one of the great aspects of not only academia, but also modern society - wherever we are in the world, we can remain connected.

For all that I have learned and continue to learn, I am eternally grateful. Thank you all. Thank you UCSD.

The material contained in Chapter 3 was, in part, originally published in the proceedings of the *IEEE American Control Conference, 2008*, by Simpkins, Alex, de Callafon, Raymond, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The co-authors listed in this publication directed and supervised the research which forms the basis for this chapter.

The material contained in Chapter 4 was, in part, accepted for publication in the proceedings of the accepted for publication in the *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2009*, by Simpkins, Alex, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

The material from Chapter 6 is, in part, submitted for publication in the *IEEE Conference of Robotics, Science, and Systems, 2009*, by Simpkins, Alex, Kelley, Mike, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The second co-author listed in this publication directed and supervised the research which forms the basis for this chapter and Mr. Kelley was a research associate who participated in part of the development of the work in this publication.

VITA

2001	B.S. Applied Mechanics and Engineering Sciences and Psychology, University of California, San Diego
2001-2004	Teaching Assistant, Department of Mechanical and Aerospace Engineering and Department of Psychology, University of California, San Diego
2001-2004	Research Assistant, Department of Mechanical and Aerospace Engineering, University of California, San Diego
2003-2009	Mechanical Engineer, 3D Engineering Corporation, Inc.
2004	M.S., University of California, San Diego
2004-2006	Teaching Assistant, University of California, San Diego
2004-2006	Research Assistant, Department of Mechanical and Aerospace Engineering, University of California, San Diego
2006	Associate in the Department of Cognitive Science, Course instructor for Cognitive Science 109: Modeling and Data analysis
2007	Associate in the Department of Cognitive Science, Course instructor for Cognitive Science 109: Modeling and Data analysis
2009	Ph.D. University of California, San Diego

PUBLICATIONS

A. Simpkins, M. Kelley, and E. Todorov. Design and control of biologically inspired modular robots for locomotion, manipulation, and virtual reality interaction. Submitted to*Robotics, Science, and Systems,* 2009.

A. Simpkins, R. de Callafon, and E. Todorov. Optimal tradeoff between exploration and exploitation. *American Control Conference*, IEEE Computer Society, 2008.

A. Simpkins and E. Todorov. Optimal tradeoff between exploration and exploitation in human movement. *14th Southern California Nonlinear Control Workshop*, 2007.

E. Todorov and A. Simpkins. Optimality of active sensing. *Computational and Systems Neuroscience Workshop,* 2007.

A. Simpkins and E. Todorov. Practical numerical methods for stochastic optimal control of biological systems in continuous time and space. Accepted for publication in *The Conference of Adaptive Dynamic Programming and Reinforcement Learning*, 2009.

FIELDS OF STUDY

Major Field: Engineering

Studies in Control Theory, System Identification, Mechanical Design, Optimal Control.
Professors Raymond de Callafon and Emanuel Todorov

Studies in Cognitive Science - Motor Control.
Professor Emanuel Todorov

Studies in Mechanical Engineering Design.
Professor Nathan Delson

ABSTRACT OF THE DISSERTATION

Exploratory studies of human sensorimotor learning with
system identification and stochastic optimal control

by

Charles Alexander Simpkins, Jr.

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)
University of California, San Diego, 2009

Professor Emanuel Todorov, Chair
Professor Raymond de Callafon, Co-Chair

Biological sensorimotor control systems possess the ability to achieve control objectives under circumstances which would challenge even the most masterful control engineer - high dimensionality, noise, redundancy, uncertainty, continuously changing tasks, and delays. Even the control objective itself may be uncertain and require some exploration to determine. Therefore by studying these systems, science and engineering can benefit from the knowledge that will inevitably be gained. Stochastic optimal control has been successfully used to model human sensorimotor behavior in many contexts, and, combined with Bayesian inference, the framework can be extended to address problems where sensorimotor learning must take place.

This dissertation contributes a suitable model for sensorimotor learning applicable to a broad class of problems (including, high dimensional, nonlinear, and stochastic contexts). The second contribution is solution methods based on function approximation and techniques for practical application of these methods for producing globally (approximately) optimal controllers. These controllers are compared (favorably) to human subjects performing exploration/exploitation tasks involving redundancy and uncertainty.

The active exploration framework developed here can be applied to artificial systems for creating systems capable of anticipating what a task is and how to achieve the goals. The third contribution is robotic designs for testing, studying and simulating sensorimotor learning and control. Preliminary results demonstrate that the designs succeed in possessing the required capabilities for mimicking biological systems. This design methodology is highly effective regarding modularity, backdrivability, compliance, and responsiveness.

Previous studies have shown that the sensorimotor system simplifies control by combining individual muscle activations into synergies, allowing fewer high level control knobs mixed into complex behaviors. This notion of ’synergy’ is redefined to include visual and motor synergies. A measure of dimensionality of synergies and variability of the sensorimotor system is created by presenting subjects with a stereoscopic image of a complex hand posture for them to match while measuring their hand postures, trajectories from a nominal posture, and repetitions. Methods are also presented to examine the sources of motor variability. Results differ from other recent work. Task difficulty is found to affect the number of synergies available.

Chapter 1

Introduction

Biological systems, especially humans, have the capability of achieving control objectives under tremendously adverse conditions such as high dimensionality, uncertainty, noise, and redundancy[56]. Reaching for a few pieces of popcorn while watching a movie appears to be a trivial task, however from the control perspective this is a strongly nonlinear high dimensional uncertain task which involves unobservable parameters which must be learned. Here one is using his or her fingers as both actuators and sensors, optimally combining the flow of sensory information with control actions to achieve an objective (eating popcorn). Researchers have studied the ability of the sensorimotor system to perform learning[87][71][19][51][107][75], but have failed to present a unifying theory which can potentially address the task-dependent characteristics of sensorimotor learning.

Recently, sensorimotor control models have made a radical shift towards optimality as a driving force behind goal-directed movements[98][101]. A theory of goal-directed movement based on stochastic optimal control has begun to unify the field and combine the traditional planning and execution stages of movement in a more intelligent framework which simplifies without *oversimplifying* the sensorimotor system[97].

This theory of stochastic optimal control [94][12] can be extended by combining it with optimal estimation [32][2] in such a way that the uncertainty becomes part of the decision making process. This dissertation presents new models, experiments, and artificial systems which can be used to describe the process of sensorimotor learning using a combination of stochastic optimal control and Bayesian inference[52][49], as well to test those descriptions. Similar to the realization that stochastic optimal control can be used as a basis of a unifying theory of sensorimotor control, the extension presented here has the potential to describe many seemingly conflicting results and unanswered questions regarding sensorimotor learning.

In order to test this model, unique robotic systems and techniques for developing new robots were formulated and are elucidated in this dissertation. Though robots with similar degrees of freedom have been developed[80][79], the unique backdrivability, high efficiency drive systems, low mass, and modularity distinguish the design from other designs in existence. The characteristics of the robot were chosen so that the dynamics would have fundamental similarities to biological systems, such as passivity[66][74]. Though such compliant characteristics can be 'simulated' by measuring force and performing feedback[23], or adding springs for partial compliance[22], the inherently non-backdrivable nature of the system would influence the formulation of a control system (and the system would be very low bandwidth in order to address compliance, since outside the actuator bandwidth the system would become stiff). Here the ultimate goal is to create a model which encapsulates both human (or other biological system) sensorimotor control and learning, and test/compare approaches with control designs implemented in realtime on the modular robots.

From an engineering perspective, studies of nature have often led to tremendous innovations. Each of the techniques presented can be applied in a strict engineering setting. For example, methods for developing more compliant robots, as well as engineering for safe human interaction with robots will need to be developed as robots become more integrated with the average human's life. Assistive robotics where a robot will not only be able to perform tasks but recognize a need and act based on that need will require solid active exploration models and methods, as well as systems which have passive dynamics.

1.1 Problem formulation

The overall problem (which will be addressed and narrowed in chapters 3 and 4) for the learning component of this dissertation can be stated in the following way:

- Create an artificial system which can perform automatic feedback control successfully under similar conditions to which biological sensorimotor systems successfully achieve a goal in tasks which require sensorimotor learning.

To frame this in a controls problem statement:

Problem 1.1 *Consider a known or unknown, potentially unstable plant which contains a feedback controller, and may have unobservable parameters which are integrated with the control objective. Design a controller which can improve the performance of the system online by reducing uncertainty in an optimal fashion.*

Additional constraints are:

Constraint 1.1 *Solve problem 1.1 such that it is possible to implement the design in cases where systems possess characteristics of redundancy and high dimensionality.*

Indeed problem 1.1 with the constraints imposed in 1.1 form a challenging control problem. And once again, addressing this type of challenge successfully is one of the many things at which biological systems excel. If well understood, the knowledge gained will benefit engineering as well as contribute to scientific understanding of sensorimotor learning.

In order to understand how the sensorimotor system addresses this challenge, a second problem addressed in this dissertation (in chapter 5) is

Problem 1.2 *Study the characteristics of the sensorimotor system's behavior from an input-output perspective.*

Many have postulated that the sensorimotor system simplifies the control problem by coupling several joints together in what are referred to as 'synergies.' However they only define synergies in the sense of output combinations, where an input-output perspective may be more helpful in an input-output system. The following questions are all addressed in chapter 5:

Question 1.1 *What is a more appropriate definition of the notion of synergy than output-only?*

Question 1.2 *What is the maximum dimensionality of the human sensorimotor system within the context of hand manipulation (a high dimensional but measurable context)?*

Question 1.3 *In what way does the sensorimotor system do well and in what way does it fail?*

Question 1.4 *What are the characteristics of these failures (noise)?*

Question 1.5 *Does task difficulty affect the dimensionality available? What might explain this?*

Question 1.6 *What is the most task-relevant information required to perform a sensorimotor task (vision or proprioception)?*

Question 1.7 *What kind of trajectory is followed in the sense of task performance and is it affected by the task condition?*

Question 1.8 *What are appropriate mathematical methods to answer these questions?*

These are fairly exploratory questions, and the questions, methods, results, and conclusions are discussed in depth.

A third problem addressed in this dissertation (in chapter 6) is:

Problem 1.3 *Design a robotic manipulator upon which the above theories and methods can be implemented in realtime for the purpose of being a testbed to perform experiments comparing performance with biological sensorimotor systems.*

Design constraints for this robot are:

Constraint 1.2 *The device should be modular in order to be useful for many different experimental designs, with minimal difficulty setting the robot up in each particular configuration.*

Constraint 1.3 *The robot is to be used for manipulation and locomotion experiments, and thus must have as high an output-force-to-weight ratio as possible while retaining a size similar to a human finger.*

Constraint 1.4 *The robot must possess passive dynamics similar to a biological system - namely backdrivability from any point of contact.*

Constraint 1.5 *The robot must be robust enough to withstand shocks, and strong enough to withstand 50N linear force in any direction without structural failure.*

Constraint 1.6 *The robot joints must have maximum stiffness and minimum friction.*

Constraint 1.7 *The control algorithm must possess the capacity to be adjusted online.*

Constraint 1.8 *It must be possible to collect the data from the robot for later analysis.*

The challenges faced by designing this robotic manipulator are difficult, but not insurmountable. Modern microprocessor technology allows an embedded system to be manufactured and implemented in such a way that any reasonable (in a computational sense) controller can be applied and data can be transmitted at a high rate over a wireless link to any laptop[64]. Due to space and mass constraints, adding an encoder to the flat pancake brushless motors to be used is not feasible. These motors must be electrically commutated in order to control the torque they generate and move the joint, and thus a new method of measuring rotor position and commutating these motors is needed. This commutation algorithm must run at a maximum update rate in order to possess appropriate resolution to produce smooth responses in the motor, and thus in the robot. Commutating brushless motors is not new[7]. But the combination of a new way of measuring position and estimating parameters online with other methods of commutation is new. These motors possess particularly challenging characteristics for control due to their size and thus very fast time constants, as most existing algorithms update at a rate of at most 1kHz. We are required to update at the maximum hardware limit of near 50kHz, and thus computational constraints threaten to limit the complexity of our control algorithms. This issue is discussed and addressed successfully.

1.2 Organization of this dissertation

Chapter 1 introduces this work, the scope, problems, questions, and challenges. Several preliminary assumptions and definitions are laid out, and the parallel between biological and dynamical systems is presented.

Chapter 2 prepares a background review of the literature, giving a context and motivation to the presented work.

Chapter 3 presents the exploration/exploitation theoretical framework, beginning with an introduction to exploration/exploitation methods, and active sensing in general.

Chapter 4 presents methods of approximating the solution to the stochastic optimal control problem described in the previous chapter. Results and comparisons of these solutions to discrete solutions over a grid are presented and discussed.

Chapter 5 discusses a new model of synergies and a theory of sensorimotor coordination which redefines the literature with a new interpretation of data, as well as results from new experiments which support the claims.

Chapter 6 describes several robotic designs and design methodologies for studying and simulating human sensorimotor behavior. This includes methods of safely interfacing humans and robotic devices, a tripedal robot prototype, and a unique modular robot design with exceptional passive dynamics. This chapter covers not only the design of the modular robot, but also the commutation methods employed for controlling the small brushless motors, as well as a unique estimation scheme for estimating unknown parameters of the motor and robot.

Chapter 7 discusses some preliminary results and future work which will be continued after this dissertation is complete. This includes a new sensor module design for hand manipulation experiments which addresses several shortcomings of current motion capture devices.

Chapter 8 concludes the dissertation with several remarks and a discussion of present results and directions.

Chapter 9/Appendices consist of appendices of this work such as derivations the results of which are used throughout the work, but may appear in the standard literature and so are not presented as new work. These include the derivation of Principal Component Analysis and the stochastic Hamilton-Jacobi-Bellman equation. Some of these techniques/theories are not prevalent in mechanical engineering at this time, though the underlying mathematical techniques are, and so brief explanations assist with readability of the dissertation for some audiences.

1.3 Preliminaries - definitions

The audience for this dissertation may have varied backgrounds from different disciplines. The following definitions attempt to familiarize the reader with the essential terminology used throughout this dissertation.

1.3.1 System

We begin with the definition of a **system** as we will refer to the word in this dissertation. A system will be defined as *any set of physical or conceptual* *components which are defined, and that are measurable or not measurable.* We assume the physical world to exist, not only within the human mind. This world is consistent across observers but may include randomness (to be defined later) thus influencing experiments to have slightly varying results.

We also adhere to the axiom stated by E.L. Thorndike that anything that exists exists in some quanity and anything that exists in some quantity can be measured[15]. It is this philosophical position which motivates the attempts to model and predict human learning and behavior within this research.

Systems will include inputs, outputs, and internal states which may be observable, partially observable, or non-observable.

The system encapsulates the components of interest which we intend to model, which will be referred to hereafter as the **Plant**.

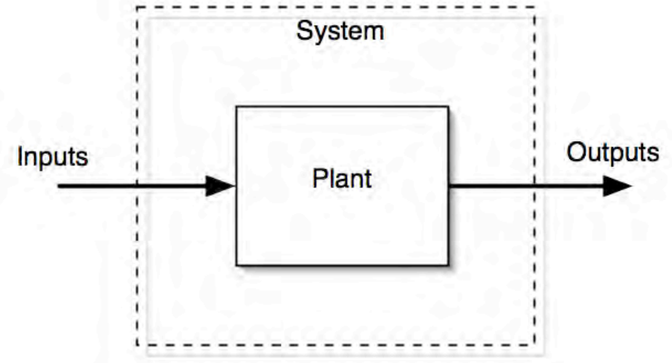


Figure 1.1: Relationship between inputs, outputs, the plant, and the overall system.

1.3.2 Model

A **model** will be defined as a mathematical or symbolic description which approximates some real quantity or system. We draw from the view propounded in Lennart Ljung's influential book[61]. There it is stated that a model description of a true system is never the *same* as the true system. This is a matter for philosophical debate as to its truthfulness, however, we take the pragmatic perspective in this dissertation that models are considered good models which approximate or estimate the system behavior well enough to fulfill some clearly defined purpose[81][13].

1.3.3 Feedback control

Feedback control is the process by which states of a system are observed in some way then fed back into the input to a device or algorithm (hereafter referred to as the 'controller' or 'control') designed to compare these states to some desired state or trajectory (the 'reference'), then act upon the system in order to cause the system to behave in the desired fashion. The reader is referred to [30] for a good overview of feedback and feedforward control.

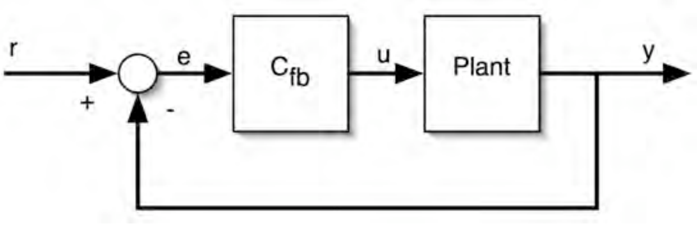


Figure 1.2: Relationship between inputs, outputs, the plant, and the overall system.

1.3.4 Feedforward control

Feedforward control is the process by which states of a system are not observed, rather assumptions are made about the state, or a forward model is used, and a reference (r) input leads to a mapping into a control action injected directly to the system (u).

Typically, feedforward control may be used to determine a desired trajectory, and then is combined with feedback control to drive the system to the desired trajectory.

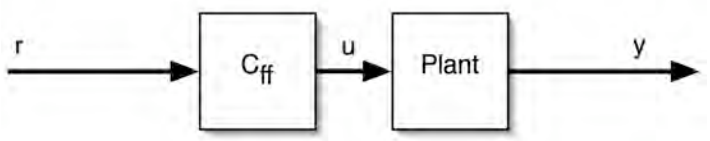


Figure 1.3: Relationship between inputs, outputs, the plant, and the overall system.

1.3.5 Human sensorimotor learning modeled as nonlinear time varying feedback control systems

A preliminary discussion of the concept of learning

Learning is defined in this dissertation as an alteration of the (potentially unobservable) parameters of a system model or the formulation of a nonparametric model in order to improve performance of a task relevant to that system. The system can be as varied as a computer program that adapts to its user, a robotic arm that adapts to unobservable altering task parameters, or a human being who learns the task of playing a violin. Once a system is defined, it can be modeled with inputs and outputs, as well as dynamical equations with possibly unknown parameters, which may change through the course of a task or situation. This alteration of certain basic parameters allows us to quantify learning as a continuous process. Models of learning will be discussed further in chapter 2.

The quantification of the components of a biological system within the context of estimation and control[86][19] has led to vast increases in the quality of fit and usefulness of models for learning theory. These components can be broken down into actuators, sensors, plant, signals (input, output, error, reference, and control), and the controller. An introduction to these components follows in the next few sections.

Actuators and sensors

Computational models of the human musculoskeletal system have existed for over 100 years[36][72]. They can be broken down into three main categories: classical mass/ spring/ damper models, ODE models, and PDE models. Some of these models are more applicable than others due to computational complexity. Each of the components of the neuromusculo-skeletal system can be described from a mechanical engineering perspective. Skeletal mammalian muscle is divided into fascicles, which are bundles of smaller fibers called muscle fibers. The physiology of the motor unit of the muscles is as follows

Muscle fibers are specialized multi-nucleated cells with a 50-100 μm diameter and a 2-6cm length. One motor neuron is connected to 100-1000 muscle fibers, with one muscle fiber connected to one motor neuron. The quantity of motor units per muscle varies tremendously (consider your finger muscles versus your arm bicep). A typical muscle is composed of hundreds of thousands or even millions of individual contractile fibers connected in parallel and series. There is a synapse, ACh (Acetylcholinesterase), called the endplate that is the connection between the neuron and muscle fiber. Every action potential causes a large, slowly propagating muscle contractile spike. This facilitates recording EMG (electromyograms, electrical signals which parallel muscle activation). Signals to and from the brain and spinal chord can be understood within a standard framework of actuators and sensors. Motor axons exit the spinal chord through ventral roots, whereas sensory axons enter the spinal chord through separate dorsal roots. Muscle fibers contain parallel myofibrils, Calcium delivery mechanisms and stabilizing connective tissues. Within the myofibrils are small (1.5-3.5 μm) sacromeres, separated by Z disks.

These sacromeres have thin filaments on each end, and thick filaments in the middle. The thin filament is composed of actin, troponin, and tropomyosin. The thick filaments are composed of myosin.

Contraction takes place by a process of attachment and detachment of myosin heads in a recurring manner on adjacent thin filaments. There are five stages to muscle contraction, after the signal has been received to activate.

A. The muscle fiber at rest has all the myosin heads of the thicker filaments in a prepared position (like a spring) with ADP (adenosine diphosphate bound at the ready. The troponintropomyosin complexes on the thin filaments do not have bound Ca2+ and are blocking the binding sites on the actin

B. When activation occurs of the muscle fiber, the Ca2+ lo is released from the sacroplasmis reticulum[47] and binds to some of the tropomyosin sites. The thin filament shape alters, exposing actin-binding sites. This allows the myosin heads to attach, forming cross-bridges between thin and thick filaments.

C. These attached myosin heads undergo a rotation which exerts longitudinal forces that cause the thin and thick filaments to overlap and shorten the muscle fiber.

D. At the end of this cross-bridge contraction, fresh ATP (adenosine triphosphate) binds to the myosin head that subsequently detaches.

E. The myosin head is cocked again into position for attachment to another binding site using chemical energy released by dephosphorylation of the ATP to ADP (still bound in position)[47].

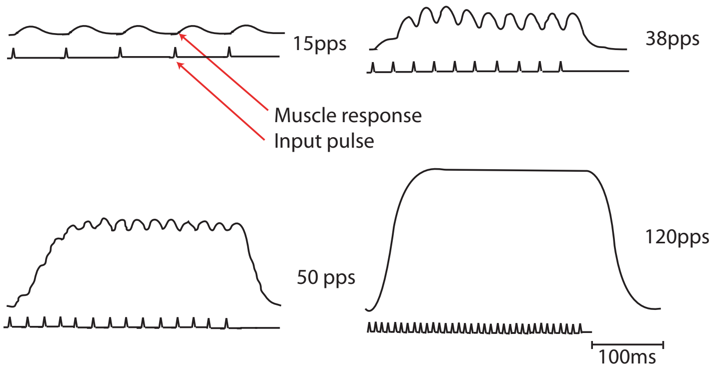


Figure 1.4: Muscle contraction is frequency dependent. As the stimulation frequency increases, the muscle does not have time to fully relax in between stimulations, so forces vary between zero and maximum tension. The full contraction is referred to as 'tetanization.'

If muscles are stimulated at a slow enough rate, discrete twitches are evident, with a rise and fall shape. As stimulation rate is increased, the muscle does not relax entirely between stimulations. Force builds as stimulation rate increases, until a peak is achieved at which the muscle is fully contracted with no relaxation between stimulations. This is referred to as tetanic contraction. Muscle force is dependent on velocity and on length of the muscle fibers. The relation is shown below.

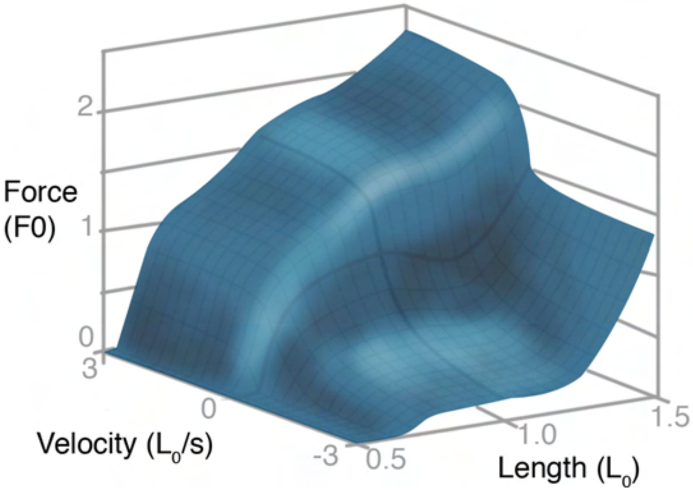


Figure 1.5: Tetanic muscle force capability as a function of velocity and length.

Mathematical models of force production

There are several mathematical models of muscle force production. The most computationally expensive and therefore least often used in online studies is the sliding filament hypothesis, by Huxley[40]. It specifies how the cross-bridge distribution evolves as a function of neural input, length and velocity in a partial differential equation.

The second method is a balance between accuracy and computational speed. It is a model by Zahalak called the distribution moment model that utilizes Gaussian distributions to model the cross-bridge distribution[111]. Ordinary differential equations are derived describing the mean and variance of the Gaussian. The complexity is computationally efficient but difficult for many to understand in the field, so it is not often used.

The third and most popular method is empirically based and involves curve-fitting. This is referred to as a Hill model, and it frequently satisfies requirements for simulations and modeling. Many variants on the model exist, a few of which are:

$$F = A \tag{1.1}$$

$$F = A + l * s + v * d$$

$$F = A * s * d$$

The current modern Hill model contains models for each relation of a muscle, such as the parallel elastic elements, thick filament compression, force-length relationship, force-velocity relationship, etc[31]. All of these elements sum into the final equation for total contractile element force,

$$F_t = F_{PE} + F_{CE} \tag{1.2}$$

Where the first term is the total parallel elastic force, and the second term is the total active contractile force.

Sensors are integrated into the muscle group in the form of spindles that carry position and velocity information, and in the form of golgi tendon organs that react linearly with force.

One important consideration that factors heavily into the structure of the motor control system of the human body is that, due to physiology, changes in joint angles bring about changes in the moments which muscles apply to joints. The moment arm of a muscle is defined as the length between a perpendicular line from the center of action of a muscle and the center of rotation of the joint that the muscle is acting upon.

A second important consideration is that one muscle does not normally act upon only one joint. Often one muscle is linked to multiple joints, and each joint

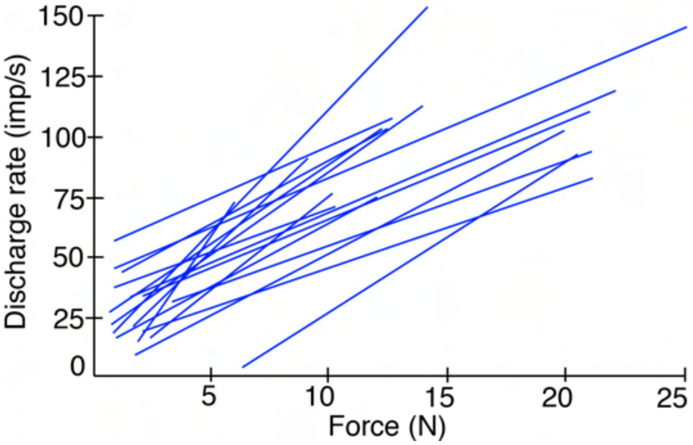


Figure 1.6: Golgi tendon discharge frequency vs. measured force.

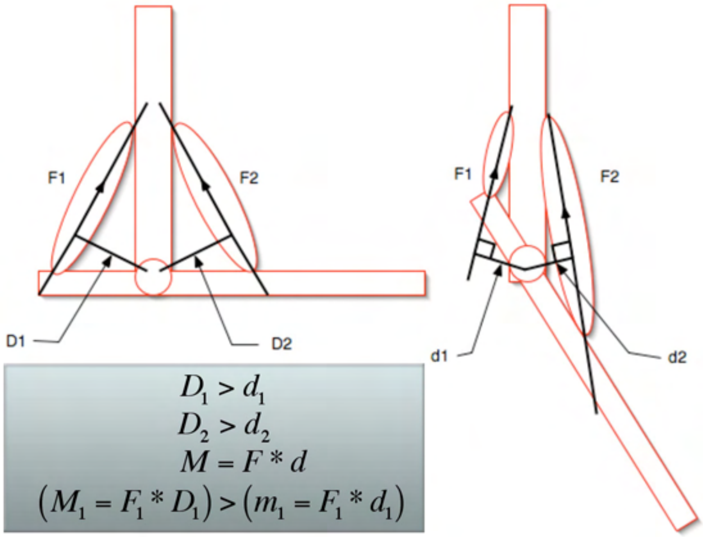


Figure 1.7: The basic concept of changing moment arms in human muscle actuators.

has multiple muscle groups as an input. This complicates the conceptualization of muscles as actuators, and as such complicates the resulting models, but not impossibly so. It must merely be accounted for. Joint torques (moments) are often used as a means of discussing biomechanics due to the traditional parallel with robotics, where actuators are often servomotors which act in an angular fashion. When using kinematic computational models, it may be more parsimonious to deal with joint torques than with linear forces, since that is a relatively mature science. However, a method of control which can handle redundancy[70] is necessary to eventually understand not only how the brain controls the body, but also how the brain performs sensorimotor learning. Having a good model of sensorimotor learning will not only improve the theoretical explanations of empirical results from experiments, but will also aid in the development of new therapies for brain damaged individuals who must relearn how to control these complex systems. It will also aid in the development of control algorithms for prosthetic limbs and other assistive technologies.

This discussion strikes home the notions that

- The complex biological system can be thought of as an engineering plant.
- Models need to be developed for sensorimotor control and learning which can handle the tremendous redundancy and high dimensionality of this plant.
- To include learning these models must be time varying.

Chapter 2

Background

2.1 Theories of sensorimotor control - from classical to modern computational theories

Before addressing approaches to sensorimotor learning, we must first address sensorimotor control. Theories of sensorimotor learning imply prior conceptions of sensorimotor control.

2.1.1 Traditional theories of motor control

Traditional theories of sensorimotor control have focused on the very difficult to model problem of how the brain controls the body to perform tasks by looking for simplifying assumptions or patterns in movement trajectories, muscle activation patterns, postural primitives, and others. Some of these theories are equilibrium point control[25][33], and the virtual spring attractor[38][104]. Both of these theories and other similar theories cannot explain endpoint errors that are not attributable to sensory-motor misalignment[59]. There was a fundamental lack for some time of researchers addressing the driving force behind actions. There was also a lack of a single unifying theory of motor control that could potentially address the broad capabilities of the motor system. This is because the biological system (including the brain, body, and all associated sub-systems) is so adaptable to a variety of tasks and goals within those tasks that this becomes a very convoluted issue - no one solution seemed to fit all cases.

Typically there was a resulting assumption in the literature for a long time that there is a separation between planning and execution stages of movement (in engineering design, this can be compared to the difference between the 'over-the-wall' style of engineering versus 'concurrent' engineering practices - traditional theories being parallel to over-the-wall style). Within these theories, for a movement to occur, first a stage of planning a trajectory would take place, then the movement would be executed using some servo control method[29][106][35]. This allowed a notion that if only the fundamental simple primitives could be found[68], the proper trajectory-generating curves derived, or true muscle synergies observed, the problem of how sensorimotor control works could be solved. Researchers all 'knew' that the problem of sensorimotor control was too challenging a problem, and the only way to begin to address the problem was to construct a simplifying model.

2.1.2 Modern theories of motor control

All models no matter how complex or accurate are simplifications of the true system[61]. However, it is important that a model capture an appropriate level of complexity to address the problem at hand in order to be useful. It is possible to observe and model aspects of the sensorimotor system for the purpose of creating an artificial system to perform some engineering task in a way the biological system might, but in order to consider a theoretical framework which can potentially explain the richness of behavioral data in biological systems (or to engineer systems which can operate upon a variety of tasks autonomously), one must step back and consider one fact, as researchers eventually did: *The biological organism is goal-directed*[97]. The goal may vary from moment-to-moment, task-to-task[60].

Ultimately there was no evidence of a separation between planning and execution stages in sensorimotor control, and thus a model in which planning and execution are integral was required. These two aspects of sensorimotor control can both be addressed within the framework of stochastic optimal control theory. Here one can define a notion of the sensorimotor system which will include the dynamics of biomechanics, delays, signals, and all the other required aspects, as well as a clearly defined goal which will plan optimal trajectories. This theory has shown tremendous promise for explaining a wealth of seemingly conflicting results. This theory can also address the issue of cognitive penetrability[45][78] - the idea that the higher levels of cognitive processes can interfere significantly with reflexes[54] and planned motor movements 'online.' Optimality can also account for dimensional change in movement patterns.

In most of the motor control literature there is a belief, based on a mistaken experimental task, that the dimensionality of movement, coordinated hand movement for example, is much lower than is actually the case. These subspaces have been explored in the literature but typically consider combinations of muscle activation patterns or canonical grips, and fail to include all the dimensionality of the motor system as a result. Thus a method of understanding the coupling that occurs in the motor system without oversimplifying (i.e. without ignoring the actual dimensionality of movement that the sensorimotor system is capable of) is needed, and will be addressed later in this dissertation. A fundamental aspect of this coupling is that it is very likely that the brain operates in a synergy space rather than in the high dimensional task space. Thus, if this is the case, it is likely that there is some transformation between sensory input, synergy space, and motor output which is again high dimensional. According to the minimal intervention principle, the dimensionality of the synergy space would be highly task-dependent. Indeed this appears to be the case and a clear set of experiments to explore this dimensional transformation, as well as a complete redefinition of the notion of synergy is included in a chapter of this dissertation.

2.2 Theories of learning - from classical to modern computational theories

There are many cases where uncertainties play a role not only as disturbances - such as the multiplicative uncertainties prominent in some components of biological motor systems[58], but also as a driving force for actions. For example, consider a case where to achieve a goal some tool must be used which maps the biological system's actions to an output relevant to the task at hand such as controlling a robotic manipulator with a joystick. Some learning is required to properly control the position of the robotic manipulator. Without some knowledge of the mapping from joystick position to manipulator position, proper control of the manipulator is difficult if not impossible. Therefore some adaptation or active learning must take place to acquire that information in order to achieve the control objective.

Learning as a concept has been studied for many years, more recently being translated to computational models which can be implemented in control systems and for modeling biological sensorimotor systems. Biological systems possess a unique capability of achieving a control objective when uncertainties relative to the task are so large that there is no way to make a control policy with constant parameters that can achieve the same objective. Other times biological systems operate under conditions of continuous rapid variation of system parameters. We will briefly now review some of those relevant theories and move to computational theories which will prove useful in the coming chapters.

2.2.1 Classical Conditioning

Many readers are likely to be aware of the classic experiment Ivan Petrovich Pavlov (1849-1936) performed where a bell is rung and a dog salivates. Consider that when meat powder is placed in a dogs mouth, salivation occurs. The food is what is referred to as the unconditioned stimulus. The salivation is the unconditioned reflex or unconditioned response. Then a stimulus that is initially unrelated is presented within an appropriate time frame as the food is presented. This can be any arbitrary stimulus, a bell ring, for example. After many repetitions, the unconditioned stimulus can be removed (i.e. the food), and the unconditioned response (salivation) will occur by the conditioned stimulus, and the response is referred to now as the conditioned response [37].

$$\begin{aligned} US- &> UR \qquad (2.1)\\ (CS,US)- &> UR \\ CS- &> CR \end{aligned}$$

2.2.2 Operant Conditioning

Operant conditioning is a process of behavior modification in which the likelihood of a specific behavior is increased or decreased through positive or negative reinforcement each time the behavior is exhibited, so that the subject comes to associate the pleasure or displeasure of the reinforcement with the behavior. In operant conditioning, behavior operates on the environment producing various effects.

Operant conditioning, sometimes called instrumental conditioning or instrumental learning, was first extensively studied by Edward L. Thorndike (1874-1949). Thorndike's most famous work investigated the behavior of cats trying to escape from puzzle boxes. When first constrained in the boxes the cats took a long time to escape. With experience however, ineffective responses occurred less frequently and successful responses occurred more quickly enabling the cats to escape in less and less time over successive trials. In his Law of Effect, Thorndike theorized that successful responses, which produced satisfying consequences were "stamped in" by the experience and thus occurred more frequently. Unsuccessful responses, those which produced annoying consequences, were stamped out and subsequently occurred less frequently. In short, some consequences strengthened behavior and some consequences weakened behavior. This effect involves a strengthening of the association between the response and its effect.

Operant conditioning was named by psychologist B.F. Skinner (1904-1990). He used a similar idea to Thorndike's Law of Effect with the notion of reinforcers. Reinforcers are those events that strengthen a response. The two kinds of reinforcement include positive reinforcement and negative reinforcement. Positive reinforcement occurs when a behavior (response) is followed by a pleasant stimulus that rewards it. In the Skinner box experiment, positive reinforcement is the rat pressing a lever and receiving a food reward. Negative reinforcement occurs when a behavior (response) is followed by an unpleasant stimulus being removed. In the Skinner box experiment, negative reinforcement is a loud noise continuously sounding inside the rat's cage until it presses the lever, at which time the noise ceases[91][90].

2.2.3 Modeling Social Learning Theory

Bandura proposed an explanation of learning acquisition as modeling, a conceptualization of cognition, which posits that humans learn in the social world (i.e., from observing actors, or models, and replicating, or matching, their behavior). Part of three studies using the famous Bobo doll experiments in the mid-1960's proposed a contiguity theory as an alternative to the established reinforcement theories of imitative learning. In his contiguity model, an observer witnesses a demonstration of sequenced behavior, then acquires perceptual and symbolic responses, and, at some later point, replicates the modeled behavior[8].

2.2.4 Prism Adaptation

It has long been known that putting an image of the environment onto the retina of the eye is just a small piece of the visual perception system. A striking example of this can be seen in the phenomenon of prism adaptation. Prism adaptation experiments have been conducted since at least the 1800s. These experiments involve changing the usual correspondence between the light hitting the eye and the environment, by placing prisms (among other optical devices tested) in front of the eyes[63].

In the simplest case, this can shift the direction of light rays by a constant angle so that, at first, objects in the world seem to be moved by that amount. But over a relatively short period of time, one can become used to the perturbation, so that one can move around with nearly as much ease as before (behavioral adaptation), and the world may even begin to "look right" (experiential adaptation). Because it is easier to measure the behavioral changes, most experiments have concentrated on how motor tasks, most commonly reaching, are affected by prism adaptation.

In addition to altering the apparent position of objects, prisms have side effects, such as changing straight edges to curves (primarily affecting vertical lines, given the prisms in this experiment) which can cause the world to appear ”warped” as one moves through it. Another alteration is ”chromatic aberration”, the classic ”prism effect” where white light separates into spectral colors, especially around light sources. It seems to be possible to adapt to these effects as well.

2.2.5 Degrees of Freedom (Dimensional) Change During Learning

Bernstein (1967) proposed a model of learning called developmental progression [10] that has had a major influence on the field of learning. This theory accounts for the fact that many human limbs have redundant degrees of freedom. For example, the human arm moves in three-dimensional space, but has seven degrees of freedom. Though the redundancy provides adaptability, it complicates formulation of a model. He suggested that acquisition of new skills occurs in three stages:

1. Initial freezing of the number of DOF to a minimum
2. Gradual lifting of all restrictions on DOF
3. Utilizing and exploiting reactive phenomena arising in movement control

Though this theory has been discounted by many, it has stimulated a rich literature of discussion and experimentation [5]. This view is ultimately too narrow, as in some cases degrees of freedom are higher initially, and then reduced to the minimum necessary for a task [71]. One very useful aspect to this model is the suggestion that learning is developmental, not fixed [43].

2.2.6 Modern computational theories of learning

Modern computational theories of learning have extended the results of these pioneers to computational models. The main focus has been on the methods of learning which most directly lend themselves to mathematical modeling - learning based on behavioral models. These are: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning an error signal is provided for feedback based on the difference between actual outputs and desired outputs, in order to allow the system to update its parameters in some way to improve performance (there are many algorithms in existence which use the error signal in various ways, such as gradient descent). Unsupervised learning does not have any error signal presented to the learning algorithm. Such algorithms are often used to find patterns in data, or to create associative memory patterns. Finally, reinforcement learning is a process by which an action is performed and feedback is given in the form of 'good' or 'bad' in some way.

Typically reinforcement learning is one of the most popular ways to build control policies, however it traditionally consists of discretizing the statespace and solving for the optimal policy using dynamic programming. Unfortunately, this method suffers from Bellman's 'curse of dimensionality.' Essentially what this 'curse' consists of is the fact that the number of states required to solve a problem satisfactorily (and thus, the amount of memory required) increases exponentially with linear increase in dimensions and resolution. For example, in the simplest redundant case of active exploration to be presented in the following chapter, the state space is eight dimensional. If one discretized this state space with a grid size of fifty in each dimension, which is still fairly rough, the state space would require a storage of at least 4e13 states, which would require, using double precision floating point storage space, $1.6e5$GBytes of memory. Additionally, for each state the problem must be iteratively solved until convergence. The computation time and memory are not feasible using current technology for even the simplest redundant problem, let alone the high dimensional problems. Considering biological systems (since the brain can be considered to be a massive parallel computational machine), if slightly higher dimensional problems such as hand manipulation are considered, a discretization size of fifty for the twenty-two degree of freedom human hand would require more than the 10^{15} synapses in the central nervous system for a single task, let alone the myriad of tasks and capabilities of the human being behaviorally and cognitively. Therefore though reinforcement learning itself is a useful concept and in low dimensional cases discretization methods are useful, here they do not provide the best parallel with biological systems.

Recently, methods of reinforcement learning based on continuous time and space are beginning to be explored[20]. These involve some method of function approximation. Because of their continuous nature, these methods are much less sensitive to the curse of dimensionality, and are gaining favor among the reinforcement learning researchers.

In engineering literature, one of the single most significant contributions of the last century in optimal estimation (which can be considered a form of learning) has been the Kalman Filter[46][92][2]. Recent work has shown evidence that the central nervous system uses probabilistic models during sensorimotor learning to optimally combine prior knowledge with new information in a Bayesian way[50]. The Kalman filter is optimal in a two-norm sense, and so provides a useful framework, combined with stochastic optimal control, for developing a model of active learning[1].

[1] Active learning will be more clearly defined in the next chapter regarding exploration/exploitation tradeoffs

By implementing the above theory (combining Bayesian inference and stochastic optimal control) in continuous time and space, continuous function approximators can be combined with a form of reinforcement learning to arrive at approximate methods for stochastic optimal control which can effectively model biological systems operating in tasks where exploration/exploitation tradeoffs are necessary. From the perspective of system identification theory, these approximators are referred to as nonlinear black (or gray) box models, with parameter estimation based on the method of collocation.

2.3 Back-drivable robotic systems

The word *robot* was first used in literature by a Czech writer, Karel Čapek in 1921 in the play Rossum's Universal Robots. The word *robota* literally means work, labor, or servitude. The word *robot* now defines a broad class of mechanisms which perform many services in society from entertainment to basic labor in assembly lines. However, the majority of robots have very different design requirements from those discussed in this dissertation and there are far too many of all classes to mention presently.

Designing artificial systems for testing models of biological control systems as well as for interfacing with humans requires similarity in certain key design parameters of the robot to biological systems. This includes passive dynamics, appropriate inertial characteristics, safety for interfacing with biological systems, and in the case of locomotion[79], modularity or compactness of overall size and components.

There are very few robots which are specifically engineered for back- drivability via passive dynamics. In engineering, often this is an undesirable trait. Consider a milling machine cutting through a block of aluminum. One would not wish the aluminum to cause deviations in the path of the mill bit. These are often referred to as *haptic* robots - robots designed specifically to interface directly with a human. One of these is the MIT Manus, a planar S.C.A.R.A. two degree of freedom robot developed initially at MIT for psychophysical experiments. Another, and perhaps the most popular is the Phantom [96], which is a six degree of freedom robot which uses cable drives to achieve high efficiency and low backlash. A third is the Delta Haptic robot [18], which is also an up to six degree of freedom robot, and uses a unique delta structure to provide motion in three-dimensional space, with an optional three degree of freedom end which adds the other three degrees of freedom. All of these robots are tied to a base, and have large external electronics and amplifiers, as well as a tether which connects to a personal computer. Typically they can produce around twenty-five Newtons of endpoint force. Thus these robots can be used for manipulation tasks as well, but are not suitable for locomotion or modular experimental designs. They are not capable of carrying their own weight, and cannot be freely mobile.

Most robots require some transmission to increase the torque capability of the actuators rather than use direct drive. This is because in general, the torque-to-weight ratio of actuators is simply not high enough to perform most functions required in robotics. When mobility is considered significant, the requirements become very stringent, and so motors, which typically have a much higher maximum speed and a greater bandwidth than required, can be geared down by some transmission to produce the larger required torques for the weight of the motor[80]. Mobile robots based on biological locomotion (and general mobile robots as well) tend to use gears or pulleys which increase friction and backlash in the system. Consider Little Dog developed by Boston dynamics. This is a quadruped with DC motors[105][93] and a geartrain. Backdrivability is limited but it is a testbed for learning and locomotion algorithms in general. Big Dog is the older cousin of Little Dog but uses hydraulic actuators. These are not designed for precision nor are they backdrivable given the current Big Dog design, though Big Dog is a dynamic robot - meaning that it stores and reuses energy, allowing it to take advantage of dynamic effects such as jumping. Cable drives are notoriously difficult to design and assemble, often requiring many parts, tensioners, and suffering from wear effects such as the cable stretching, and slippage. Thus most engineering projects use more traditional methods of drive systems. However, cable drives are quite ancient, dating back to the times of Leonardo da Vinci (several cable-drive type transmissions are clearly displayed in his works of mechanical design). It is quite rare to see a compact cable drive implemented with a low parts count. Additionally, very little technical information regarding cable drives appears to be part of the robotics literature, and there appear to be few if any technical books detailing cable drive design methods. However they are capable of over 99% efficiency, and virtually eliminate backlash and frictional effects (if the correct cable is selected for the application). Therefore the author decided to consider cable drives for our robotic designs, and depart from the standard engineering practice due to the unique design parameters for our robotic systems.

An additional challenge is that we also want to explore locomotion as well as manipulation, so a compact robotic design which can output multiple Newtons of force and potentially carry its own weight is required.

Chapter 3

Exploration/exploitation

3.1 Introduction

As previously introduced, in many cases a system must be controlled which posesses large uncertainties. These uncertainties may include kinematic or dynamic maps which, if unknown, prevent the computation of an optimal (or any, for that matter) control policy, and may lead to instabilities due to control action (or lack of action). It is possible therefore to inject control actions into the system which cause the system to reveal the unknown characteristics during control actions, and thus improve feedback characteristics.[9][24][28][109].

In other cases, the uncertainties may be so large that achieving a control objective may not even be possible. Thus some exploratory actions must take place in order to exploit this knowledge for feedback or feedforward control. This can be done in an optimal manner, balancing exploratory actions whose aim is to reduce uncertainty with exploitation actions whose goal is to achieve the control objective.

This can also be done in a non-optimal manner with switching controllers which 'switch' between a random exploration mode and a CE-based feedback control mode. In such a case some care must be taken in order to avoid problems during transitions.

Active exploration is a powerful approach for dealing with uncertainty. It is widely used throughout biology - examples include movable eyes, ears and whiskers, fingers used to explore surface properties, and muscle spindles with tunable sensitivity. The importance of active exploration is also increasingly recognized in engineering.

The challenge is to incorporate information gains and control actions within the same formalism and define a notion of utility which is equally applicable to both.

In principle, the general framework of stochastic optimal control under uncertainty can be applied. In this framework one uses an optimal/Bayesian estimator which computes the posterior probability density over the state space at each point in time, and an optimal feedback controller which maps probability densities into actions. Although probability densities are infinite-dimensional objects one can rely on finite-dimensional estimators such as the Kalman filter which is optimal in a 2-norm setting. Interestingly the variance of the noise does not affect the optimal control law, thus defeating the purpose of active exploration. This property (known as the separation principle, and controllers based on this are often referred to as certainty equivalence, or CE, controllers) is normally considered a virtue, but from the present perspective it is a deficiency. A modified formalism is needed, where the estimator is still finite-dimensional but the control law is sensitive to uncertainty and generates actions aimed at reducing uncertainty. Here we develop such a formalism.

Active exploration can be classified into two categories. The first includes actions that only affect the flow of sensory feedback, thus having indirect consequences on achieving control objectives. Eye movements are a good example. The second category includes actions that not only affect the flow of sensory feedback but also have direct consequences on control objectives. Finger movements which simultaneously sense and manipulate objects are an example of such actions. This second category, which involves an interesting tradeoff between optimization of sensory information and achieving the control objective, is the focus of the present work.

Exploration and exploitation taking place simultaneously (through online estimation and control) is necessary in these types of problems where the plant is partially-observable and non-stationary. There are adaptive dual control (ADC) techniques which approach this problem by generating a 'cautious' control signal for tracking, with an additional excitation signal that accelerates parameter estimation [109], [9]. A unique characteristic of the present method is that it does not generate a reduced gain (cautious) tracking signal (which ADC controllers do), instead triggering pseudo-random actions when needed which are combined with a nominal tracking signal to produce an optimal action. Additionally, most of the approaches in the ADC literature center around discrete time systems, whereas here we develop an elegant continuous time method (there are some continuous time ADC methods such as the bicritical method but it approximates the original problem, whereas here the original problem is solved). Task-relevant versus task-irrelevant uncertainty, an important but rarely addressed distinction, is also considered in this dissertation. Finally, it is emphasized that the development of this model for control actions is centered around modeling human exploration/exploitation control actions.

3.2 Making the problem fully observable

3.2.1 Example of exploration task

We develop our method via an example, which is later used to compare the optimal solution to experimental data obtained from human subjects. The task can be thought of as tracking a screen target using an uncertain computer mouse: we do not know how the mouse works and the way the mouse works changes all the time. More precisely, let $h\left(t\right) \in \Re^{n_h}$ denote a state variable which is fully observable and controllable (say the position of the hand), and $s\left(t\right) \in \Re^{n_s}$ denote a target which needs to be tracked. The unknown mapping from h-space to s-space has a hidden state $m\left(t\right) \in \Re^{n_s n_h}$ which undergoes Brownian motion. The mapping is assumed to be a linear projection from hand to target space

$$h \rightarrow M\left[m\right]h. \tag{3.1}$$

Here M is a linear operator which reshapes the vector $m\left(t\right)$ into an n_s-by-n_h projection matrix. For example, if the case $n_s = 2$ and $n_h = 3$ that operator is

$$M\left[m\right] = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \end{bmatrix}. \tag{3.2}$$

3.2.2 Formulation as a stochastic optimal control problem

The dynamics are assumed linear-Gaussian:

$$dh = u(t)dt, \qquad ds = d\omega_s, \qquad dm = d\omega_m, \tag{3.3}$$

in which $u\left(t\right) \in \Re^{n_h}$ is the control signal which is generated in order to modify the hand state and $\omega_s\left(t\right)$ and $\omega_m\left(t\right)$ are Brownian motion processes with covariances Ω_s

and Ω_m respectively.[1] We assume that $h(t)$ and $s(t)$ are directly observable, while $m(t)$ is not directly observable and needs to be estimated. The observation process is modeled as

$$dy = M\left[m(t)\right]h(t)dt + d\omega_y, \tag{3.4}$$

where $y(t) \in \Re^{n_s}$ corresponds to the integral of the (noisy) cursor position. dy is the n_s-dimensional observation/measurement vector, and $d\omega_y$ is the corresponding measurement error. In this observation model $m(t)$ is the unknown state and $h(t)$ plays the role of an observation "matrix". In usual estimation problems the latter would be fixed, but here the subject can control it directly by changing hand position. This makes apparent the exploratory nature of hand movements in our setting. To put this observation model in a more familiar form, define the n_s-by-$n_s n_h$ matrix $H\left[h(t)\right]$ such that $H\left[h(t)\right]m(t) = M\left[m(t)\right]h(t)$. For example, if the case $n_s = 2$ and $n_h = 3$ the operator H is

$$H\left[h\right] = \begin{bmatrix} h_1 & h_2 & h_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_1 & h_2 & h_3 \end{bmatrix}. \tag{3.5}$$

With this notation, we rewrite (3.4) as

$$dy = H\left[h(t)\right]m(t)dt + d\omega_y. \tag{3.6}$$

Now suppose the prior over the initial state of the mapping is Gaussian, with mean $\widehat{m}\left(0\right)$ and covariance $\Sigma\left(0\right)$. Then the posterior over $m\left(t\right)$ remains Gaussian for all $t > 0$. Given the additive, Gaussian white noise model, the optimal estimate of the mean and error covariance of the map is propagated by the well-known Kalman-Bucy filter [2], [92]:

[1] In this study we apply deterministic hand dynamics and assume a first-order model where the control signal corresponds directly to hand velocity. A more sophisticated biomechanical arm model (in the case of modeling biomechanical systems) can be used but is not necessary in order to capture the trade-off of interest. Any dynamical system can be substituted - linear or nonlinear, low or high dimensional, for these dynamics, and this method will create useful solutions in either case.

The general linear dynamical system to which a standard Kalman-Bucy filter is applied is of the form:

$$\begin{aligned} \dot{m}(t) &= F(t)m + G(t)u(t) + L(t)v(t) \qquad (3.7)\\ m(0) &= m_0 \end{aligned}$$

with measurement equation (3.6), where $\mathbf{m}$ is an n-dimensional state vector, $u(t)$ is the p-dimensional controlled input vector, v is the s-dimensional disturbance input vector, $dy(t)$ is the r-dimensional observation/measurement vector, and $d\omega_y$ is the corresponding measurement error. The matrices F, G, L, and $H\left[h\right]$ are of appropriate sizes. The standard derivation for this continuous time system results in

$$\begin{aligned} d\widehat{m}(t) &= (F(t)\widehat{m}(t) + G(t)u(t))dt \qquad (3.8)\\ &+ K(t)\left[dy - H\left[h\right]\widehat{m}(t)dt\right]\\ \widehat{m}(0) &= \widehat{m}_0\\ K(t) &= \Sigma(t)H\left[h\right]^T R_y^{-1}(t)\\ d\Sigma(t) &= (F(t)\Sigma(t) + \Sigma(t)F^T(t) + L(t)\Omega_h(t)L^T(t)\\ &- \Sigma(t)H\left[h\right]^T R_y^{-1}(t)H\left[h\right]\Sigma(t))dt\\ \Sigma(0) &= \Sigma_0 \end{aligned}$$

In our particular example, F and G are zero, since the dynamics are assumed purely Brownian motion, and L is the appropriately sized identity matrix. Now the above equations become

$$
\begin{aligned}
d\widehat{m} &= K\left(dy - H\left[h(t)\right]\widehat{m}(t)dt\right), && (3.9)\\
\widehat{m}(0) &= \widehat{m}_0, && (3.10)\\
K &= \Sigma(t)H\left[h(t)\right]^{\mathsf{T}}\Omega_y^{-1}, &&\\
d\Sigma &= \Omega_m dt - K(t)H\left[h(t)\right]\Sigma(t)dt, &&\\
\Sigma(0) &= \Sigma_0, && (3.11)\\
& && (3.12)
\end{aligned}
$$

with measurement equation (3.6), and $H\left[h(t)\right]$ appropriately sized. The properties of the noise and disturbances do not change over time.

The mean and covariance of the state estimate is $\widehat{m}(t)$ and $\Sigma(t)$, respectively, and $d\omega_y$ is a white, zero-mean Gaussian random process as well, with covariance Ω_y. $d\omega_m$ and $d\omega_y$ are assumed to be uncorrelated[2].

Then consider the innovations sequence, where the innovation is defined as

$$
\begin{aligned}
d\omega_{\widehat{m}} &= dy - d\widehat{y}, && (3.13)\\
&= dy - H\left[h(t)\right]\widehat{m}(t)dt, &&\\
&= H\left[h(t)\right]m(t)dt + d\omega_y - H\left[h(t)\right]\widehat{m}(t)dt, &&\\
&= d\omega_y - H\left[h(t)\right]\Delta(t)dt, &&
\end{aligned}
$$

where $\Delta(t) = \widehat{m}(t) - m(t)$. $d\widehat{y}$ is a Gaussian random variable, with

$$d\widehat{y} \sim N(H\left[h(t)\right]\widehat{m}(t)dt, H\left[h(t)\right]\Sigma H\left[h(t)\right]^T + \Omega_y). \qquad (3.14)$$

[2] Note that $K\left(t\right)$ is the filter gain matrix. It is clear that K is a deterministic function of the other quantities and does not have to be propagated through time explicitly. Unlike usual estimation problems where $K\left(t\right)$ can be precomputed, here it needs to be computed online because we do not know in advance how the state $h(t)$ and thereby the observation matrix $H\left[h(t)\right]$ will evolve over time.

The innovation sequence is dependent on dy and $d\widehat{y}$, however since the true measurement is given, the only variable which contributes to the uncertainty of $d\omega_{\widehat{m}}$ is $d\widehat{y}$, and

$$d\widehat{\omega} \sim N(H\left[h(t)\right]\Delta(t)dt, H\left[h(t)\right]\Sigma(t)H\left[h(t)\right]^T + \Omega_y). \tag{3.15}$$

The Kalman filter can be written in innovations form by expressing $\widehat{m}\left(t\right)$ as another stochastic process:

$$d\widehat{m} = Kd\omega_{\widehat{m}}. \tag{3.16}$$

Here $\omega_{\widehat{m}}\left(t\right)$ is a standard Brownian motion process with unit covariance. The advantage of the innovations form is that we are now dealing with a fully observable system where $\widehat{m}\left(t\right)$ and $\Sigma\left(t\right)$ act as state variables. The latter is a symmetric matrix, therefore it is uniquely defined by its upper-triangular part. Let $\sigma\left(t\right) \in \Re^{n_s n_h (n_s n_h + 1)/2}$ be the vector of upper-triangular elements of $\Sigma\left(t\right)$. Similarly to M and H above, we will define the linear operators f and F which transform between the vector and matrix representations of the covariance, namely $\sigma(t) = f\left[\Sigma(t)\right]$ and $\Sigma(t) = F\left[\sigma(t)\right]$.

We can now define the composite state vector of our system which includes the mean and covariance (a measure of uncertainty) of the estimates:

$$x(t) = \left[h(t); s(t); \widehat{m}(t); \sigma(t)\right], \tag{3.17}$$

and write its stochastic dynamics in the control-affine form [94]

$$dx = \left(a\left(x\right) + Bu\right)dt + C\left(x\right)d\omega. \tag{3.18}$$

In the next three equations explicit time-dependence is temporarily omitted for clarity (e.g. $h(t) \rightarrow h$). The uncontrolled dynamics $a\left(x\right)$ are needed to represent the evolution of the covariance matrix:

$$a\left(x\right)=\left[\begin{array}{c}0\\0\\0\\f\left[\Omega_m-F\left[\sigma\right]H\left[h\right]^{\mathsf{T}}\Omega_y^{-1}H\left[h\right]F\left[\sigma\right]\right]\end{array}\right].\tag{3.19}$$

The controlled dynamics Bu capture the evolution of the hand state:

$$B=\left[\begin{array}{cccc}I&0&0&0\end{array}\right]^T.\tag{3.20}$$

The noise-scaling matrix $C\left(x\right)$ captures the dependence of the innovation process on the filter gain matrix, as well as the covariance of the target drift:

$$C\left(x\right)=\left[\begin{array}{cccc}0&&&\\&\sqrt{\Omega_s}&&\\&&F\left[\sigma\right]H\left[h\right]^{\mathsf{T}}\Omega_y^{-1}&\\&&&0\end{array}\right].\tag{3.21}$$

Here $\sqrt{\Omega_s}$ denotes the symmetric matrix square root, and $\omega\left(t\right)$ is a vector of standard Brownian motion processes with unit covariance.

The main idea behind our work is to define a sensible cost function for the control task, and then induce an indirect cost over exploratory actions by considering how they affect uncertainty. Since we have a tracking task, the obvious state-dependent cost rate is

$$\left\|M\left[m(t)\right]h(t)-s(t)\right\|^2.$$

This cost rate depends on the true state of the mapping $m(t)$ which is not part of our composite state vector x. However we know that $m(t)$ has a Gaussian distribution with mean $\widehat{m}(t)$ and covariance $F\left[\sigma(t)\right]$, and both $\widehat{m}(t)$ and $\sigma(t)$ are part of $x(t)$.

Therefore we can compute the cost rate by taking an expectation over $m(t)$. Using the identity $H\left[h(t)\right]m(t) = M\left[m(t)\right]h(t)$, we have (again omitting time-dependence):

$$\begin{aligned} q\left(x\right) &= E\left(\left\|M\left[m\right]h - s\right\|^2\right), \qquad (3.22)\\ &= \left\|M\left[\widehat{m}\right]h - s\right\|^2 + \operatorname{tr}\Big(H\left[h\right]^{\mathsf{T}} F\left[\sigma\right] H\left[h\right]\Big), \end{aligned}$$

where $tr(\cdot)$ denotes the trace. We also incorporate a quadratic control cost, to obtain the following cost rate:

$$\ell\left(x, u\right) = q\left(x\right) + \tfrac{1}{2}\left\|u\right\|^2. \qquad (3.23)$$

Thus we have transformed our partially-observable tracking problem to the fully-observable non-linear stochastic optimal control problem defined by equations (3.18) and (3.23).

The cost terms defined in $q(x)$ each have a unique significance. The first term in (3.22) represents a simple tracking cost, quantifying the control objective to minimize the distance between the cursor and target. The second term in (3.22) represents an uncertainty cost. This comes into play only when the system moves into a region where there is larger uncertainty. The advantage of this term is the triggering of random 'exploratory' actions which by nature reduce the uncertainty and allow the first cost term to once again dominate. The final term in (3.23) is a quadratic control cost to penalize overly large control.

The cost function for the set of problems discussed in this paper is taken to be an infinite horizon discounted cost, since there is not an expected final time for the behavior.

3.3 Experimental data and comparison to model predictions

Determining the problem is challenging, but solving the preceding problem proves challenging as well. The solution methods follow in the next chapter, titled 'Numerical Methods for Approximately Optimal Control.' The remainder of this chapter is devoted to presentation of experimental data and model predictions.

3.3.1 Description of experiment

Subjects sat in front of a computer screen and held a small 3D sensor in their hand (Polhemus Liberty, 240Hz sampling rate). At the beginning of each session the subject was asked to place his or her hand in a convenient central location; this location was recorded and used to center the 3D data during the session. In the current experiments $n_h = 2$ and $n_s = 1$. The target shown on the screen was a circle whose center remained stationary but its diameter $s(t)$ fluctuated randomly. The mapping $m(t)$ also fluctuated randomly. The subject controlled another circle whose diameter was continuously updated to equal $m(t)^T h(t)$. Circles were presented as opposed to bar graphs to avoid other properties of the human perceptual system from distorting performance. The simulation, data recording, and screen update rate was 50Hz.

The reason for visualizing the target and cursor states with the size of a circle, as opposed to a bar graph, is because we did not want this visualization to have a directional aspect. If it did (say the bar moved vertically), then certain directions of hand movement in 3D space would be special in the sense that they would correspond directly to what is happening on the screen. In this case subjects are likely to exhibit a strong prior for mappings m which preserve spatial compatibility. Since we do not yet understand these priors and cannot model them accurately, we designed the

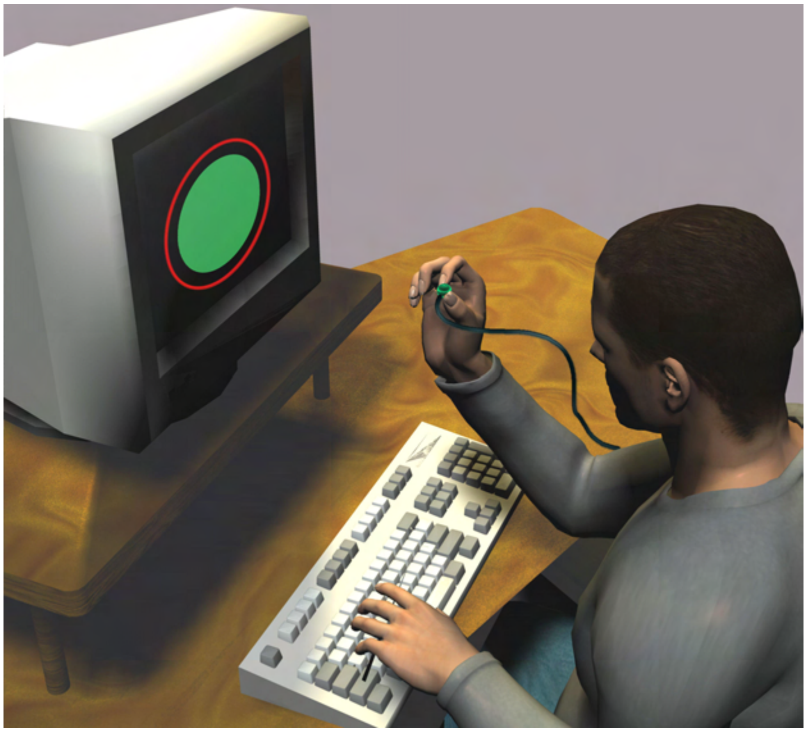


Figure 3.1: Experimental setup. The subject sits in front of a monitor, holding a 3D absolute position sensor (Polhemus Liberty). Here $n_h = 2$, $n_s = 1$. The center-stationary green circle is the target, whose diameter ($s(t)$) fluctuates according to damped Brownian motion, and there is a concentric 'cursor' (red circle) whose diameter is $z(t) = m(t)^T h(t)$, with $m(t)$ fluctuating according to an independent damped Brownian motion. The subject is then told to move however possible to try to keep $z(t) = s(t)$. Trials are 1 min., started by a key-press and sampling rate is 50Hz.

experiment so as to discourage subjects from using priors. Indeed, no particular setting of the mapping m seemed easier or more difficult than other settings.

There were two types of 1 minute trials. In the first 5 (training) trials subjects were given feedback of $m(t)$, displayed as a 2D vector on the screen. This feedback could be used to simplify the task. This vector told subjects that if they move their hand in the direction of the vector the hand movement will affect the size of the circle. (movement in orthogonal directions does not affect the size). This condition was included to familiarize subjects with the idea of the experiment as well as to provide some initial training in an easier task. The remaining 20 trials were performed without direct feedback of $m(t)$. Here subjects operated in the same setting as the model: they could observe $h(t)$, $s(t)$, and cursor circles, but had to infer the mapping in order to track the target. The mapping $m(t)$ and the target $s(t)$ fluctuated over time as second-order damped Brownian motions, namely

$$ds = vdt, \qquad dv = -\beta vdt + d\omega. \tag{3.24}$$

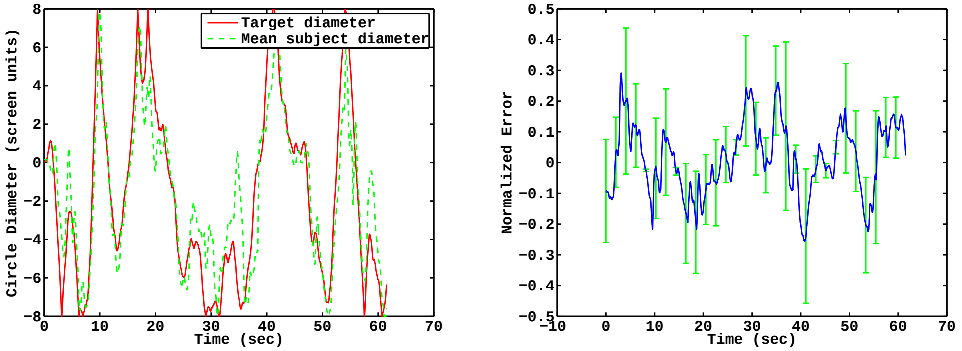
Initial experimentation revealed that a simple Brownian motion is not sufficiently smooth to result in a sensible task that subjects are comfortable with. Each data record included $m(t)$, $h(t)$, and $s(t)$. Each trial was of one minute duration. Data was recorded continuously at 50Hz. (the third coordinate was saved for additional analyses even though it does not affect the task)

3.3.2 Data and Results

Human Subjects

In Figure 3.2(a) and 3.2(b) it is clear that the human is capable of tracking the reference trajectory of the target diameter while still performing exploration when needed. When the map's velocity changed or for some other reason they lost the map, they exhibited sudden large exploratory movements and then tracked for another period of time. The exploration appears error- as well as tracking-driven.

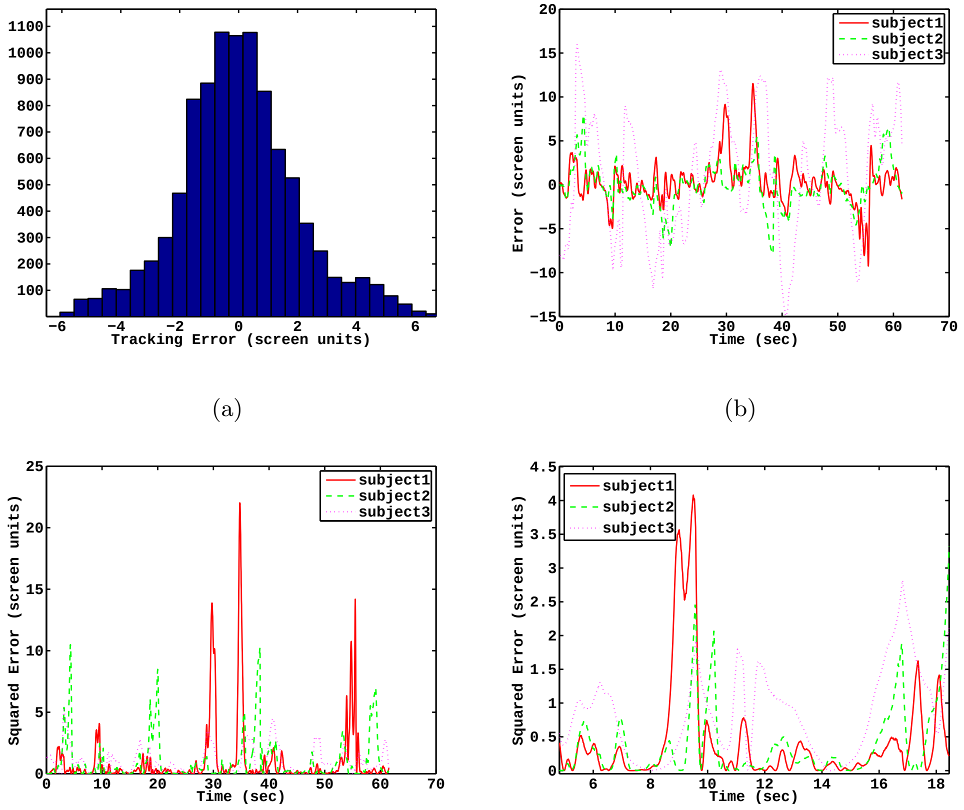
Figure 3.3(a) represents a histogram of the tracking error during a typical trial. Here one can see that the Gaussian assumption upon which the Kalman estimator is based holds up well when applied to modeling the human performance. The significance of Figure 3.3(b), 3.3(c), 3.3(d) is that different individuals tended to have similar patterns of error, though the amplitude varies across subjects.



(a)

(b)

Figure 3.2: (a) Plots the target circle diameter vs. the average human circle diameter during a typical hidden trial. Note that the human can track the circle diameter even in the case of an unobservable fluctuating parameter (the map). (b) Shows the average human error over one particular trial case with error bars representing the Standard Deviation for one subject. The error is normalized to be between zero and one.



(a)

(b)

(c)

(d)

Figure 3.3: (a) Is a histogram of average human tracking error over several trials. It is clear that this error distribution is quite Gaussian, confirming that the assumption of Gaussian noise for the Kalman filter is reasonable. (b) Shows the tracking error for three subjects during the same trial as plot a. Not the similarities in pattern over subjects for plot d, e, and f. Plots (c) and (d) show the mean-squared-error over the same trial. Plot (d) is zoomed in to reveal a better sense of fine detail.

Model performance and predictions

Often both the model and human exhibited exploratory movements at similar times.

It is notable that the FASC outperforms the human subject in terms of tracking the target. This is not surprising since a human subject has delays between perception, processing and action, as well as biomechanical limitations. The model is not approximating any of those processes, and so should outperform humans.

Though we cannot measure (given the current experimental setup) the human exploratory cost function, we can compute the associated tracking cost. The average human tracking cost for 4 subjects over 24 trials was $2.6e4$. This is higher than the FAS control ($1.1e4$), but within the same order, whereas the control which does not adapt or explore had an average cost of $6.2e5$ which was the highest of control systems compared. The basic feedback controller, when given a carefully tuned gain, had an average tracking cost of $3.3e4$, which is also comparable with humans, but when poorly initialized, or presented with highly rapid map changes (which were part of the experiments), the CE control diverged frequently The CE control did not exhibit exploratory movements. Instead if the gain was too low it failed to track sufficiently or to excite the system enough to estimate $m(t)$, and with a sufficient gain, deviations which occurred due to map estimation errors led to explosive instability.

Controller performance

The controller's performance can be characterized by the behavior of the two components of the cost function (Fig. 3.5(b)), and by summing the total average cost per trial. The NAC performed worst ($cost_{total} = 6.3e5$), followed by the RANDC (this does not diverge as far as the NAC, so the overall tracking cost is lower) with a gain of 50 ($cost_{total} = 1.6e5$), the CE-based control ($cost_{total} = 4.7e4$ or $4.2e7$, depending on stability), and then the best performance was achieved by the FASC ($cost_{total} = 1.5e4$). This is attributed to the cost term that triggers random movements when entering state space with large uncertainty, which happens periodically, as can be seen in Fig. 3.5(b). In Fig. 3.5(b) it is clear that exploration cost increases when tracking cost is low, and then tracking cost increases during the resulting exploratory actions, leading to a decrease in uncertainty and thus exploration cost. This whole behavior is periodic since the map and target velocity continuously change, requiring new information input at varying speeds (i.e. when $m(t)$ has a lower velocity - Fig. 3.4(b) - tracking can be achieved with less exploration, and as uncertainty increases due to rapid changes in $m(t)$, tracking suffers during exploratory movements).

The estimator parameters converge to a close enough approximation of the true mapping to achieve good tracking. A poor estimate would result in large control and possible instability. The FAS improves estimates with larger control inputs, so this acts to counter uncertainties, instabilities, and errors. Fig. 3.5(d) shows the FAS control actions, which appear to be near mirror images at certain time periods (e.g. - 0-15sec). Comparing that figure with Fig. 3.5(c) it is clear that the control in one axis becomes negative when the map is negative, and a resulting desired cursor output is positive.

Map estimation and uncertainty reduction

The most information can be gained by moving in such a way as to cover as much of the unknown space as possible in a random way. Indeed, it seems clear that replacing the controller with a white noise signal generator of arbitrary gain could yield the closest estimate of the map.

However, exploring the entire subspace is neither necessary nor desired in our case. The measure of uncertainty can be broken down into relevant parts to the task and irrelevant parts in the following way. The task-relevant uncertainty is measured by the trace term in (3.22), whereas the irrelevant uncertainty can be quantified by finding the orthogonal complement of $h(t)$, defined by $h(t)^{\perp}$. Then $h(t)^{\perp}$ is substituted in the trace term in (3.22), and the two uncertainty histories are computed over several trials (Fig. 3.5(e), 3.5(f), and Table 3.1). These measures were compared for the FASC, CE and RANDC (the latter using a gain of 50, which approximates human bandwidth).

What is clear is that the FASC reduces the relevant uncertainty the most of all methods compared. The FASC emphasizes reducing the h-space uncertainty (which is relevant in this task, effectively ignoring irrelevant uncertainty). The CE controller does not achieve as much overall reduction in uncertainty as the FASC since its design does not include an exploratory component[3]. Only the white noise control reduces both types of uncertainty; it does not differentiate between the two types of uncertainty. However, the RANDC does not achieve as large of a reduction of the task-relevant uncertainty as the other two methods. By distributing control action over the whole space, without unlimited gain, the RANDC wastes much of its resources reducing task-irrelevant uncertainty.

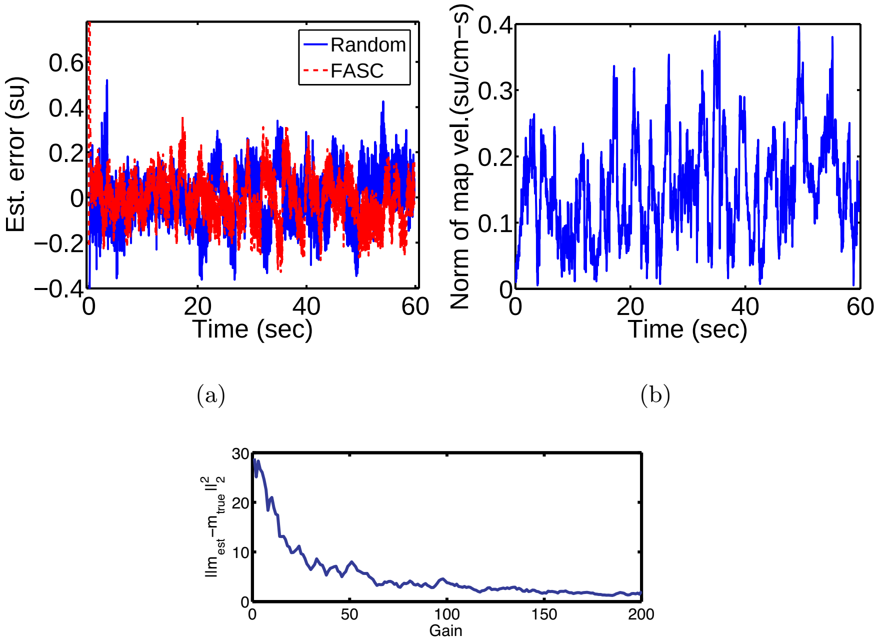
3.3.3 Conclusion

We introduced a method of approximating optimal control laws in problems which involve a trade-off between exploration and exploitation. This was done by treating the Kalman-Bucy filter dynamics as part of the plant dynamics, incorpo-

[3] Additionally, the CE gain affects this measure heavily - low gain reduces uncertainty less, but high gains lead to instability which yield poor numerical results.

Table 3.1: $\sqrt{Norm}$ of uncertainty quantities plotted in Fig. 3.5(e) and 3.5(f), which is a Standard Deviation quantity.

	h-space	$h^{\perp}$-space
FASC	8.8	36.7
CE	10.2	81.2
RAND	13.7	14.7

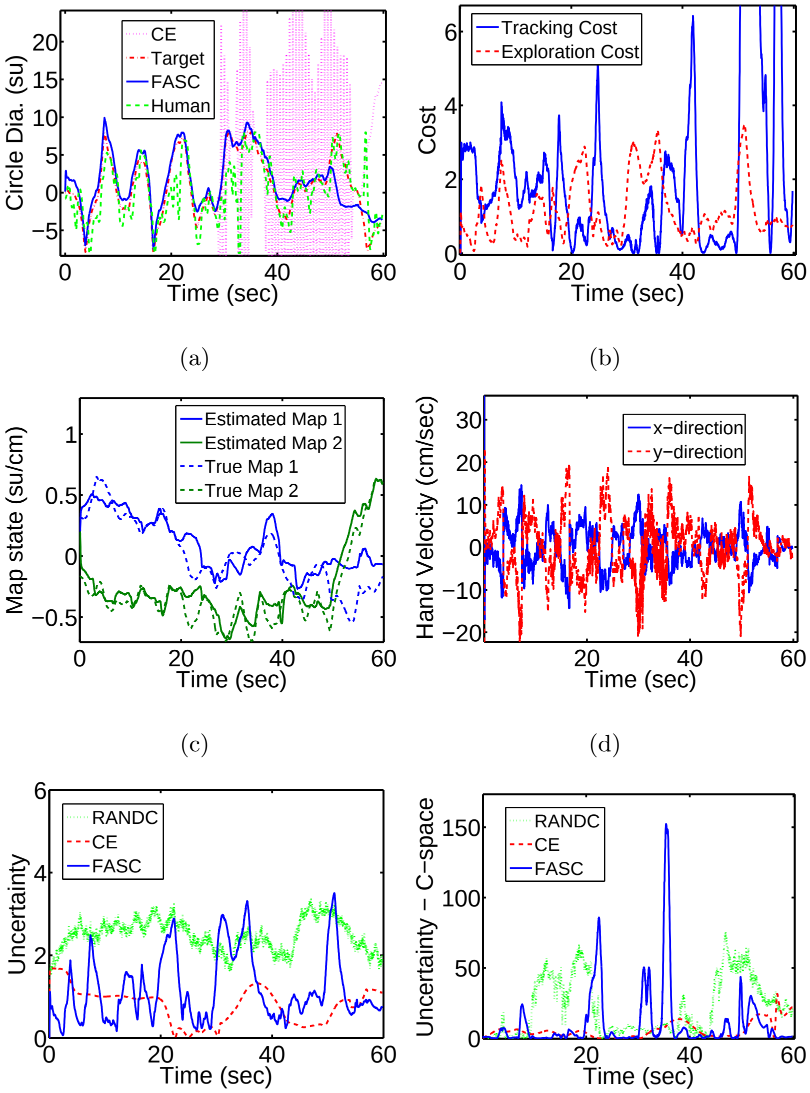


(a)

(b)

(c)

Figure 3.4: (a) Estimation error for FAS scheme, with a noise level of 0.1su, and estimation error for random signal generator, with a gain of 50. (b) $||\dot{m}(t)||_2^2$, giving a measure of where the map is changing more rapidly. (c) Map estimation error (over 1 trial), $\|m_{est} - m_{true}\|_2^2$ vs. maximum amplitude of white noise 'controller.' The estimation improves with increasing white noise amplitude.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.5: (a) A section of a 60 second trial displaying human subject data, FAS, and proportional feedback controller tracking. (b) Represents the two portions of the cost - the tracking and exploration costs. Plot (c) shows the true and estimated map. (d) Shows the FAS control actions. Plot (e) and (f) show two measures of uncertainty - (e) is in h-space, and is given by the trace term in 3.22, while (f) is the same, but in $h^{\perp}$-space.

rating the estimation uncertainty in the state vector, and applying a basis function approximation scheme to the Hamilton-Jacobi-Bellman equation. Our method produced control laws which outperformed other controllers. It not only succeeded in achieving low tracking error, but also did so with nonlinear time-varying system parameters and large uncertainties. This approach shows promise in situations where active exploration can reduce uncertainty. Additionally, this method distinguishes between and reduces task-relevant versus task-irrelevant uncertainty. That is advantageous in real-world situations where there are limited control resources.

The method was also compared to human data from an uncertain tracking task similar to the one being modeled. Although these comparisons are very preliminary, we already see interesting similarities which will be pursued in future work. In fact our motivation for developing the method is to study human behavior. Stochastic optimal control combined with Bayesian estimation is emerging as the leading theoretical framework for understanding sensorimotor function in the brain [98]. However model-data comparisons are presently limited to the few simple tasks where we can compute what behavior is optimal. New methods such as the one developed here, which allow us to extend optimality principles to more interesting tasks, can accelerate progress in the field of sensorimotor control. Of course such methods also are likely to find engineering applications.

Acknowledgment

The material contained in this chapter was, in part, originally published in the proceedings of the *IEEE American Control Conference, 2008*, by Simpkins, Alex, de Callafon, Raymond, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The co-authors listed in this publication

directed and supervised the research which forms the basis for this chapter.

Chapter 4

Numerical methods for approximately optimal control

4.1 Introduction

Solving the exploration/exploitation problem is challenging in and of itself, as the PDE is second order, nonlinear, and high dimensional. Discrete methods can be used in simple low dimensional cases, but above four dimensions, the number of states becomes large, and a fine enough grid cannot be generated. Thus, continuous function approximations are a way to approximate the solution to the original problem.

This chapter will first discuss a discretization methodology to convert the problem into a Markov Decision Process, and then solve the approximation. Next, the continuous function approximator will be developed, and several issues related to function approximation in high dimensions will be explained. The discrete solution will be used as a comparison with the continuous solution (in low dimensional cases) in order to verify optimality.

These methods are applied to model experimental human data with an exploration/exploitation tradeoff task. Results are presented and compared to humans, and other control methods are compared as well (including adaptive dual control, which has limitations in this case).

Finally, the FAS is applied to control of an inverted pendulum (single link for comparison purposes), and an inverted pendulum with an uncertain and wandering mapping.

4.2 Discretization

An alternative approach to approximating a solution to the problem is first to approximate the problem and then solve the approximate problem exactly. This is done by discretizing the problem as a Markov Decision Process (MDP)[48][95][21][108] which captures the optimal tradeoff, and solving it exactly with dynamic programming. It is then shown that the Markov Chain approximation is a locally consistent approximation to the continuous time problem.

The derivation of the transition probabilities and interpolation time follows closely the general finite difference method of [53]. Consider the controlled process $x(\cdot)$, $x \in \Re^r$ satisfying the following stochastic differential equation:

$$dx = b(x, u(x))dt + \sigma(x)dw \tag{4.1}$$

and a discounted cost of the form

$$W(x, u) = E_x^u \left[\int_0^\infty e^{-\beta s} k(x(s), u(x(s)))ds \right], \tag{4.2}$$

$$W(x,u) = g(x), \text{for x=0,B.} \tag{4.3}$$

$$W(0) = W(B) = 0$$

$$x \in (0,B)$$

The boundary and functions $k(\cdot)$ and $g(\cdot)$ will play only a formal role in the development of the transition probabilities and interpolation times, so their values are not significant here. Additionally, one can reflect some (or all) of the boundaries in the case of a cylindrical state space such as an inverted pendulum where position repeats every 2π radians with essentially the same mathematical development.

Let us define the covariance matrix $a(x) = \sigma(x)\sigma(x)^T = \{a_{ij}(x)\}, i,j = 1,...,r$. We also define e_i as the unit vector in the i^{th} coordinate direction, the uniform h-grid $\Re_h^r = \{x : x = h\sum_i e_i n_i : n_i = 0, \pm 1, \pm 2, ...\}$, and the state space of the Markov chain as $S^h = \Re_h^r$. We limit the discussion to uniform grids for clarity.

We define the differential operator from Ito's rule:

$$\mathcal{L}^\alpha = \sum_{i=1}^{r} b_i(x,\alpha)\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{r} a_{ij}(x)\frac{\partial^2}{\partial x_i \partial x_j} \tag{4.4}$$

The above is applied to W, which yields

$$\mathcal{L}^{u(x)}W(x,u) + k(x,u(x)) = 0 \tag{4.5}$$

We obtain the transition probabilities by solving the above equation with finite differences. A convenient property of this problem is that the matrix $a_{ij}(x)$ is diagonal (i.e. $a_{ij} = 0$ for $i \neq j$).

The finite difference approximation to the second derivative is:

$$f_{x_i x_i}(x) = \frac{f(x+e_i h) + f(x - e_i h) - 2f(x)}{h^2} \tag{4.6}$$

The finite difference approximation to the first derivative is given by the one-sided approximation:

$$f_{x_i}(x) = \begin{cases} \frac{f(x+e_ih)-f(x)}{h} & \text{if} \quad b_i(x,\alpha) \geq 0 \\ \frac{f(x)-f(x-e_ih)}{h} & \text{if} \quad b_i(x,\alpha) < 0 \end{cases} \tag{4.7}$$

Now we substitute (4.6) and (4.7) into (4.5), define the solution as $W^h(x,u)$, and after some algebra obtain the finite difference equation

$$\begin{aligned} W^h(x,u) = \sum_y e^{-\beta\Delta t^h(x,u(x)}p^h(x,y|u(x))W^h(x,u) \\ + k(x,u(x))\Delta t^h(x,u(x)) \end{aligned} \tag{4.8}$$

Where $p, \Delta t$, and Q are defined as

$$p^h(x, x \pm e_ih|\alpha) = \frac{a_{ii}/2 + hb_i^{\pm}(x,\alpha)}{Q^h(x,\alpha)} \tag{4.9}$$

$$\Delta t^h(x,\alpha) = \frac{h^2}{Q^h(x,\alpha)} \tag{4.10}$$

$$Q^h(x,\alpha) = \sum_i \left[a_{ii}(x) + h|b_i(x,\alpha)|\right] \tag{4.11}$$

and, assuming $\Delta t^h(x,\alpha) = O(h)$, we set $p^h(x,\alpha) = 0$ for $y \in S_h$ not of the form $x \pm he_i$ for an arbitrary i. For each x and α the probabilities sum to one, and all are non-negative. Together, (4.9), (4.10), (4.11) are referred to as the transition probabilities for our controlled Markov chain, where their local consistency with $x(\cdot)$ is checked by

$$E_{n,x}^{h,\alpha}\Delta\zeta_n^h = b(x,\alpha)\Delta t^h(x,\alpha) \tag{4.12}$$

$$\begin{aligned} cov^{h,\alpha}_{n,x}\Delta\zeta^h_n &= a(x)\Delta t^h(x,\alpha) + O(h)\Delta t^h(x,\alpha) \\ &+ \left[O(\Delta t^h(x,\alpha))\right]^2 \end{aligned} \tag{4.13}$$

The optimal control for this MDP is computed by Dynamic Programming, specifically approximation in value space using Jacobi iteration [53],[11].

The cost functional and optimal cost functional are written in vector form as

$$W^h(u) = R^h(u)W^h(u) + C^h(u) \tag{4.14}$$

and

$$V^h = \min_{u(x)\in U}\left[R^h(u)V^h(u) + C^h(u)\right] \tag{4.15}$$

respectively, where

$$\begin{aligned} r^h(x,y|\alpha) &= e^{-\beta\Delta t^h(x,\alpha)}p^h(x,y|\alpha) \\ R^h(u) &= \{r^h(x,y|u(x)); x,y \in G^0_h\} \\ W^h(u) &= \{W^h(x,u), x \in G^0_h\} \\ V^h &= \{V^h(x), x \in G^0_h\} \\ C^h(x,u) &= k(x,u(x))\Delta t^h(x,u(x)) \\ C^h(u) &= \{C^h(x,u), x \in G^0_h\} \end{aligned} \tag{4.16}$$

Since we work with a fixed grid size, we will omit the dependence on h for notational clarity:

$$W(u) = R(u)W(u) + C(u) \tag{4.17}$$

$$V = \min_{u(x)\in U} \left[R(u)V(u) + C(u)\right] \tag{4.18}$$

Three assumptions will be needed:

A1. *$r(x, y|\alpha), C(x, \alpha)$ are continuous functions of α for each x and y in S.*

A2. *(i) There is at least one admissible feedback control $u_0(\cdot)$ such that $R(u_0)$ is a contraction, and the infima of the costs over all admissible controls is bounded from below. (ii) $R(u)$ is a contraction for any feedback control $u(\cdot)$ for which the associated cost is bounded.*

A3. *If the cost associated with the use of the feedback controls is $u^1(\cdot)$, ..., $u^n(\cdot)$, ... in sequence is bounded, then*

$$R(u^1)\cdots R(u^n) \rightarrow^n 0$$

Theorem 1. *Let $u(\cdot)$ be an admissible feedback control such that $R(u)$ is a contraction. Then for any initial vector W_0, the sequence W_n defined by*

$$W_{n+1}(x, u) = \sum_y r(x, y|u(x))W_n(y, u) + C(x, u(x)) \tag{4.19}$$

converges to $W(u)$, the unique solution to (4.17). Assume (A1) - (A3). Then for any vector V_0, the sequence recursively defined by

$$V_{n+1}(x) = \min_{u(x)\in U} \left[R(x, u)V_n(x) + C(x, u)\right] \tag{4.20}$$

converges to V, the unique solution to (4.18)

The control policy which leads to the minimum cost is the optimal control policy for the approximating Markov Chain.

4.3 Continuous function approximation-based methods

One approach to approximating the solutions to continuous optimal control problems is to discretize them. However discretization methods such as in [53] and [12] are only feasible in low-dimensional spaces, while the problems we are dealing with tend to be rather high-dimensional. In particular, the dimensionality of the augmented state x is

$$n_x = n_h + n_s + n_s n_h + n_s n_h \left(n_s n_h + 1\right)/2. \tag{4.21}$$

For $n_h = 2$ and $n_s = 1$, which is the simplest redundant problem and corresponds to the experiments described below, we have $n_x = 2 + 1 + 2 + 3 = 8$. For $n_h = 3$ and $n_s = 2$, corresponding to a mapping from 3D hand space to a 2D screen, we have $n_x = 3+2+6+21 = 32$. Thus we have to focus on continuous function approximation methods - which may lack theoretical guarantees in terms of convergence and error bounds, but in practice turn out to have very appealing properties.

The method begins with the continuous stochastic dynamical equation defined as in (3.18)-(3.23).

Consider an infinite horizon discounted cost formulation, with discount factor $\alpha > 0$. The optimal value function for our problem satisfies the Hamilton-Jacobi-Bellman (HJB) equation for stochastic systems:

$$\alpha v(x) = \min_u \left\{ q(x) + \frac{1}{2}\|u\|^2 + (a(x) + Bu)^T v_x \right. \tag{4.22}$$
$$\left. + \frac{1}{2} tr(C(x)C(x)^T v_{xx}) \right\},$$

where the subscripts denote partial derivatives. The minimization in (4.22) can be performed in closed form to yield the optimal feedback control law

$$\pi(x) = -B(x)v_x(x). \tag{4.23}$$

Substituting (4.23) into (4.22) and dropping the *min* operator we arrive at the minimized HJB equation

$$\begin{aligned}\alpha v(x) &= q(x) + a(x)^T v_x(x) \\ &+ \frac{1}{2} tr(C(x)C(x)^T v_{xx}(x)) - \frac{1}{2}\|\pi(x)\|^2.\end{aligned} \tag{4.24}$$

Using (4.23) and (4.24) we now construct a function approximation scheme based on the collocation method [16] to approximate a continuous time optimal control law. We begin with a general linear (in the parameters, nonlinear in the state) function approximator

$$v(x, w) = \sum_i w_i \phi^i(x) = \phi^T(x) w, \tag{4.25}$$

where $\{\phi^i\}$ is a set of predefined features and w_i are corresponding (to be determined) weights. Function approximation is a broad topic, and many choices are available for the set $\{\phi^i(x)\}$. The reader is referred to [88] for a survey of techniques. It is possible to approximate any given function to any desired accuracy given a sufficient number of terms[61]. However, with sensible choices of terms an equivalent quality of fit is obtained using many less terms. Since we have a tracking task, we first choose to include the two cost terms. We then choose the set of all quadratic terms of the form x_r, and $x_r x_s$ to generally fit the function, with a set of Gaussians to introduce corrections about the quadratic.

Before we can substitute (4.25) into (4.24), the first and second derivatives of $v(x)$ must also be computed:

$$v_x(x, w) = \sum_i w_i \phi^i_x(x) = \phi^T_x(x) w, \tag{4.26}$$

$$v_{xx}(x, w) = \sum_i w_i \phi^i_{xx}(x) = \phi^T_{xx}(x) w. \tag{4.27}$$

The idea is to reduce this nonlinear partial differential equation to an equation of the form

$$\mathbf{M}w = \mathbf{d}, \tag{4.28}$$

which we can solve as a linear least squares problem, then recompute **d** and iterate until appropriate convergence criteria are met.

We get to (4.28) by rewriting (4.24) as

$$\begin{aligned}\alpha v(x) - a(x)^T v_x(x) - \frac{1}{2}tr(C(x)C(x)^T v_{xx}(x)) \\ = q(x) - \frac{1}{2}\|\pi(x)\|^2,\end{aligned} \tag{4.29}$$

Then substituting (4.25), (4.26), and (4.27) into (4.30) and simplifying for w, defining **M** and **d**

$$\begin{aligned}\mathbf{M} = \Big\{M_{j,i} = \Big(\alpha\phi^i(x_j)^T - a(x_j)^T\phi_x^i(x_j)^T \\ - \frac{1}{2}tr\{C(x_j)C(x_j)^T\phi_{xx}^i(x_j)^T\}\Big), \forall i, j\Big\},\end{aligned} \tag{4.30}$$

$$\mathbf{d} = \Big\{d_j = q(x_j) - \frac{1}{2}\|\pi(x_j)\|^2, \forall j\Big\}. \tag{4.31}$$

FAS initialization

The function approximation scheme is initialized in the following way:

1. Generate a set of vectors $\{x_j\}$ and Gaussian centers $\{c_{i_g}\}$ which span the space of interest (in this case the boundaries of the space are chosen to be equivalent to the experimental boundaries) and $n_j > n_i$. Make sure that at least one x_j equals each c_i. If $S \in \Re^r$ is the state space,

$$x_j = \text{rand}_j\{S\}, \qquad c_{i_g} = \text{rand}_{i_g}\{S\}. \tag{4.32}$$

2. Compute $\phi^i(x_j)$, $\phi^i_x(x_j)$, $\phi^i_{xx}(x_j)$ and store the results for all i, j.

3. Initialize w^0 in a sensible way :

$$w^0 = \begin{cases} 0 & i \neq i_{cost} \\ 1 & i = i_{cost} \end{cases}, \tag{4.33}$$

where i_{cost} are the two locations of the cost terms included in the function approximator. Then we initialize our control policy as

$$\pi^0(x_j) = -B\phi_x(x_j)^T w^0. \tag{4.34}$$

FAS Iteration

Given a set of control actions $\pi(x_j)$ for every j, the update from iteration k to k+1 is as follows:

1. Substitute the constraints $\phi^i(x_j)$, $\phi^i_x(x_j)$, and $\phi^i_{xx}(x_j)$ into (34), and $\pi^k(x_j)$ into (4.31) to obtain one constraint on $\mathbf{w}$ for every j.

2. Find the least-squares solution to (4.28) and assign it to w^{k+1}. For the new setting of the function approximator, compute $\pi^{k+1}(x_j)$ for every j using (4.23)

3. Stop if the stopping criterion is met. Many criteria are possible, and the one used in the present results is

$$e^k = \frac{1}{n_j}\|Mw^k - d\|_2^2, \qquad de^k = e^k - e^{k-1}, \tag{4.35}$$

$$\text{if}(\{e^k < \gamma\}|\{de^k < \beta\})\rightarrow\text{break}, \tag{4.36}$$

where $\gamma = 10^{-3}$ and $\beta = 10^{-5}$ are tolerances . We also test for divergence:

$$\text{if}(\{(de^k) > \lambda\}|\{\text{isnan}(de^k) == \text{true}\})\rightarrow\text{break}, \tag{4.37}$$

where 'isnan' is a test for invalid numbers, and $\lambda = 10^{-3}$ is a positive constant which is arbitrary, but can be on the order of one.

4.4 Application problems

4.4.1 Single link inverted pendulum problem

Consider a bar-shaped pendulum with limited torque. When the maximum torque the actuator can exert is less than $mgl/2$, (where m is the mass of the pendulum bar, l is the length, and g is the acceleration due to gravity), the pendulum must undergo multiple swings in order to attain the necessary momentum to swing to a vertical position. Additionally, the controller must anticipate the peak of the arc and begin deceleration at the appropriate time in order for the pendulum to avoid over-rotating and falling again.

The dynamics of the system are given by

$$J\ddot{\theta} + H\dot{\theta} + G(\theta) = \tau. \tag{4.38}$$

In this case $J = ml^2$ is the inertia of the link, $H\dot{\theta}$ is the velocity-dependent friction, $G = mglcos(\theta)$ is the torque due to gravitational force, and τ is the torque applied to the pendulum externally (via an actuator such as a motor). This can be arranged in the optimal control framework we set out in the following way. Define the state x as

$$x = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T \tag{4.39}$$

$$a(x) = \begin{bmatrix} \dot{\theta} \\ -J^{-1}(H\dot{\theta} + G) \end{bmatrix} \tag{4.40}$$

$$B = \begin{bmatrix} 0 \\ J^{-1} \end{bmatrix} \tag{4.41}$$

$$u = \tau \tag{4.42}$$

Now we can write the dynamics in the standard form. Here we make the problem deterministic to simplify comparisons between 'what to do and what not to do steps:' This makes the second derivative drop out, leaving us with

$$\dot{x} = a(x) + Bu(x), \tag{4.43}$$

with parameters $m = 1kg$, $l = 1m$, $g = 9.81m/s^2$, and $H = 0$.

The cost rate was determined by a combination of a velocity penalty, a control energy penalty, and a position penalty (for angles other than $\pi/2$). All trials had random initial conditions, and were considered successful if the pendulum achieved a vertical orientation for an indefinite period of time ($t > 15sec$), achieving the vertical position in under 10 seconds (this time is arbitrary depending on the degree of under-actuation). The cost rate is then of the form:

$$\ell(x, u) = k_\theta(\theta - \pi/2)^2 + k_{\dot{\theta}}(\dot{\theta})^2 + \frac{1}{2}u^2, \tag{4.44}$$

where the k's are gains which can be adjusted to tailor behaviors if desired. In all example problems presented in this paper, the constants are all the same value for simplicity.

4.4.2 1-DOF Pendulum problem with an uncertain wandering mapping. Solving partially observable nonlinear exploration/ exploitation problems to which the separation principle does not apply

Consider now the same problem with a slight change- there is an unobservable and continuously wandering mapping between the observation of the base angle and the actual angle. In physical terms this could be interpreted as the base to which the pendulum is attached undergoing a continuously random rotation about a parallel axis with the pendulum base. This is similar to our previous study in [89]- except

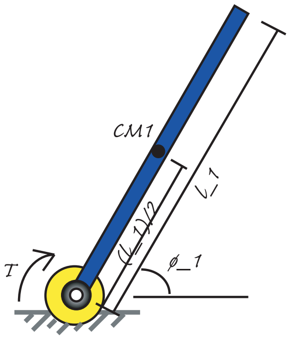


Figure 4.1: Basic diagram of 1-link pendulum. The state space is a cylinder with position range ±π

that the motion is not damped, but instead undamped Brownian motion. Now we have a partially observable problem, since the cost function includes the angle. By taking the expectation of the uncertain term, and augmenting the state with the mean and covariance of the estimated quantity, one can create a fully observable, but higher dimensional problem solvable with our FAS scheme.

$$\begin{aligned}\ell(x,u) \approx & E(k_\theta||\theta-\pi/2||^2 + k_{\dot{\theta}}||\dot{\theta}||^2 + \frac{1}{2}||u||^2) \\ = & k_\theta(||\widehat{m}\theta-\pi/2||^2 + \theta^2\Sigma) + k_{\dot{\theta}}||\dot{\theta}||^2 + \frac{1}{2}||u||^2.\end{aligned} \tag{4.45}$$

The observation process is given by

$$dy = m(t)\theta(t)dt + d\omega_y. \tag{4.46}$$

Assuming the prior over the initial state of the mapping is Gaussian, with mean $\widehat{m}(0)$ and covariance $\Sigma(0)$, then the posterior remains Gaussian for all $t > 0$ for $m(t)$. Given the additive white noise model (where the properties of the noise and disturbances do not change over time, and $d\omega$ is a white, zero-mean Gaussian noise process with covariance Ω_y), the optimal map estimate is propagated by the Kalman-

Bucy filter[46][2][92],

$$\begin{aligned} d\widehat{m} &= K\left(dy - \widehat{m}(t)\theta(t)dt\right), \qquad (4.47)\\ K &= \Sigma(t)\theta(t)^{\mathsf{T}}\Omega_y^{-1},\\ d\Sigma &= \Omega_m dt - K(t)\theta(t)\Sigma(t)dt. \end{aligned}$$

The mean of the estimate is $\widehat{m}(t)$ and the covariance is $\Sigma(t)$. Ideally we would like to augment our state with $m(t)$, but we can only estimate m, so now our composite state vector will be

$$x(t) = [\theta(t); \dot{\theta}(t); \widehat{m}(t); \Sigma(t)], \qquad (4.48)$$

and our stochastic dynamics can be written in the form of (4.43), with uncontrolled dynamics representing the pendulum and the evolution of the covariance matrix,

$$a(x) = \begin{bmatrix} \dot{\theta} \\ J^{-1}(-H\dot{\theta} - G(\theta, \dot{\theta})) \\ 0 \\ \Omega_m - \Sigma^2\theta^2\Omega_y^{-1} \end{bmatrix}, \qquad (4.49)$$

controlled dynamics,

$$Bu = \begin{bmatrix} 0 & J^{-1}\tau & 0 & 0 \end{bmatrix}, \qquad (4.50)$$

and finally the noise-scaling matrix,

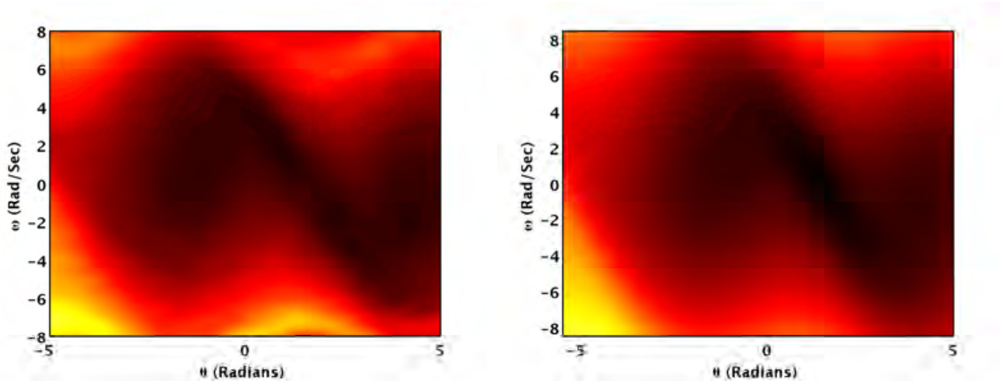
$$C(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma\theta\Omega_y^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (4.51)$$

Now we have a nonlinear stochastic optimal control problem defined by (4.43), (4.45), and (4.48)-(4.51). An approximation to the optimal control policy can be created using our FAS algorithm.

4.4.3 Application problem results

Single link inverted pendulum swing-up

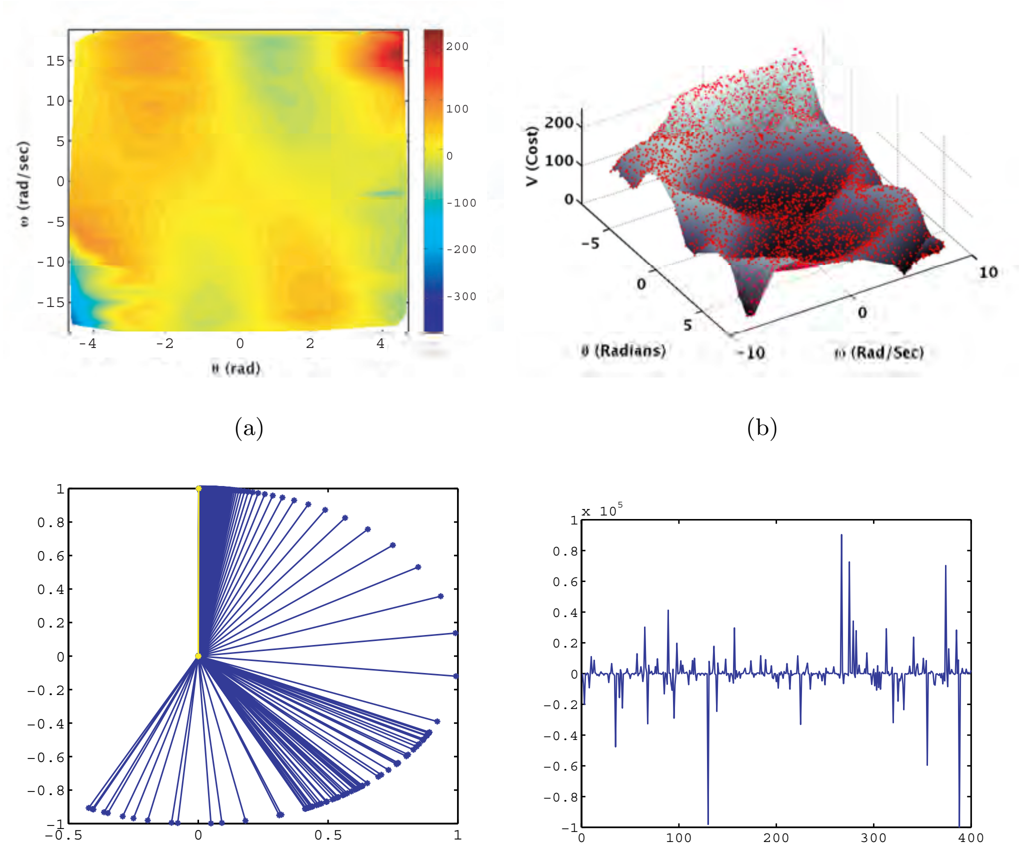
The single link pendulum swing-up task results show that the FAS can indeed perform nonlinear control for a nontrivial problem quite effectively. Given one hundred trials with random start points, the average time to vertical was under five seconds, with a final error under 1e-3 Radians. A typical policy function representation is shown in Figure 4.3(a), using one hundred basis functions.



(a)

(b)

Figure 4.2: (a) Shows the cost surface over the position-velocity state space which is computed by the continuous function approximation scheme using 1000 Gaussian basis functions. (b) Shows the same but it is computed as a discrete 100x100 grid Markov Chain Approximation Method. Many fewer basis functions can be used, but even with this seemingly large number of basis functions, the memory and computation time required to solve the problem are in favor of the FAS method - e.g. 6 seconds for the FAS versus 5809 seconds for the MDP.



(a)

(b)

(c)

(d)

Figure 4.3: (a) Shows the surface of the control action space. (b) Shows the random cloud of points used to fit the weights and thus a surface defining the cost function in the method of collocation. (c) Shows a typical swing-up trial, with the initial position in this case at -0.89 Radians. The final error is 1e-3 Radians, and the swing-up time is under 5 seconds (measured as $t_{s.u.t.} = t(\text{error} < 0.001\text{rad})$ (d) Shows a plot of typical weight values if the Gaussian variance is slightly too large for the distance between Gaussian centers. Note that the weights have large opposing values to balance each other.

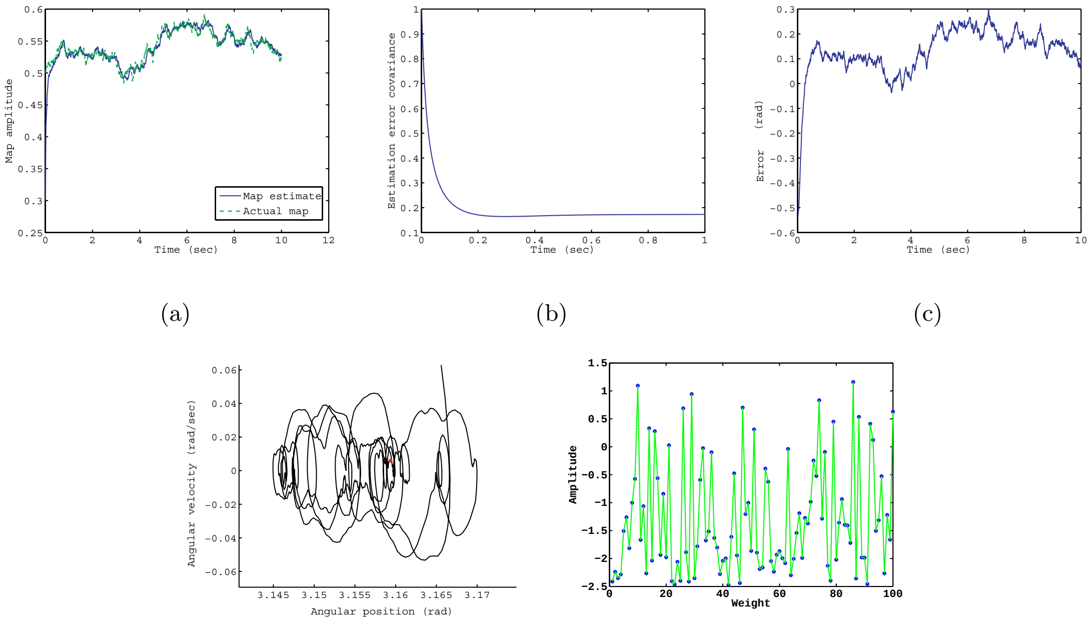
Single link inverted pendulum swingup with uncertainty

The pendulum was still able to perform the experimental task when driven by the uncertain mapping, in addition to the pendulum swing-up challenge. This problem is four dimensional, which is difficult to solve with discretization methods, yet our FAS could solve this with one hundred basis functions (this suggests fewer bases could be effectively used on the previous problem, but minimal basis function application was not the goal). The Kalman filter effectively estimated the unobservable mapping (Figure 4.4(a)), keeping the pendulum swing-up task possible. The parameter is only shown fluctuating in a small range, but positive or negative values are acceptable and posed little problem for the FAS algorithm during experimentation.

Figure 4.4(d) shows exploratory actions being injected into the system by the policy after convergence to the vertical position. This is done to highlight the pseudorandom behavior triggered by the covariance term. By this time the map parameter was being tracked well by the FAS algorithm, and so the actions are small due to the covariance term being small (Figure 4.4(b)).

4.5 Limitations of function approximation and computational methods for optimal solutions

Though the method of collocation with radial basis functions is an effective way to approximate an arbitrary function, there are limitations. One such limitation is that the number of Gaussians, given a fixed variance and location of centers required to approximate a given function may be very large[61]. This problem can be mitigated by including relevant features of the system being studied (to be described



(a)

(b)

(c)

(d)

(e)

Figure 4.4: (a) Map versus estimate for pendulum swing-up problem with uncertainty. (b) Estimation error covariance. Note the rapid drop in uncertainty during this trial. (c) Position error $(m(t)\theta - \pi/2)$. (d) Small exploratory movements in position-velocity space. (e) An example of the typical numerical fit achieved in two iterations. The normalized fit error is 1.1e-14.

in depth below).

A second limitation is that, in the stochastic case, it is challenging to know when a true minimum is reached versus some local minimum far from the solution due to noise. Additionally, one does not wish to fit the noise of a system, only the relevant features. There are several approaches to mitigate these types of problems. This is the well-known issue of generalization and overfitting (namely, we want to increase the generalization of our function fit, and reduce overfitting).

4.5.1 Large number of parameters

If one uses some insight into the system being modeled and then creates some features which have an appropriate structure, function approximation can be performed with many less parameters, and thus less computational expense. In this way many less Gaussians can be used. A common feature to use if the system has global and local fluctuations is a polynomial term such as a quadratic (all terms $x_a * x_b$ and x_a)

It can be shown that there is a relationship between the number of required Gaussians to achieve appropriate level of coverage in all dimensions by

$$n_g = \prod_i \frac{p_i}{\sigma_i} \tag{4.52}$$

where σ_i is the Standard Deviation of each Gaussian (assuming fixed widths) in dimension i and p_i is the function space width in dimension i. Thus, if the order of p is large and small fluctuations are expected as well as large global ones, it may prove useful to have a nonuniform Gaussian variance in each dimension (by having sets of variances), or include a global quadratic feature. The global quadratic's second order term weight matrix is symmetric, and thus uniquely determined by its upper triangular portion. The general equation is

$$y_q = A + B^T x + x^T C x, \tag{4.53}$$

and

$$C = C^T. \tag{4.54}$$

To find the weights C, it helps numerically to compute the coefficients for the upper or lower triangular parts of C, then rebuild the matrix later for simulation by

$$\begin{aligned} C &= C_L + C_U, \qquad (4.55) \\ C_L &= C_U^T. \end{aligned}$$

4.5.2 Generalization/overfitting

The traditional method of maximizing generalization and minimizing overfitting is by using validation sets. This consists of splitting the randomized collocation points (or randomly regenerating more sets), then testing the computed fit on the new data set, since a good fit over a space should be an effective minimum for all points within the space, unless noise has been fit. In this case, it is best to reduce the number of Gaussians or to compute an average over data points.

4.5.3 Numerical errors (differentiation, condition number, and sparsity)

Numerical errors are likely to arise in many stochastic problems due to the second order term. Differentiation is an un-smoothing process and so functions with analytical derivatives are always preferable to numerical differentiation if possible. Fortunately Gaussians and quadratics are analytically differentiable. One must take care with the second order derivatives however as dimensionality is such that the matrices become tensors, and during implementation this is typically a point where bugs delay results. Not only must the equations be correctly computed on paper, but a double check of the actual code will prevent many headaches.

Once the algorithm has been implemented, a computational check of the **M** matrix condition number gives a measure of the probable accuracy of the least squares operation to be performed. If necessary, computational accuracy can be improved without a significant loss of efficiency in cases where the condition number is moderate to poor. See, for example, [77],[26].

Finally, if the widths of the Gaussians are too large, the weights will tend to alternate in a large and numerically (the more Gaussians that overlap and the wider the overlap) unstable fashion. Thus, considering Section (4.5.1) and the following section regarding simple visualization and numerical overlap computation will be significant. If the widths of the Gaussians are too narrow, there will be space that cannot be covered by the function approximator (no matter what weight is computed, the approximator will output zero). This results in a sparse coefficient matrix (**M**), and so is easily checked. The general rule is that some overlap is preferable to open space if good approximation is desired.

4.5.4 How to determine unkown basis function constant parameters - centers and widths

In order to determine if the state space is completely covered by the Gaussians, numerical computations can be confirmed with visualization methods. One effective scheme for setting the width of the Gaussians is to compute the mean absolute distance between centers and set the covariance to approximately 68% of the maximum value. The simplest way to check that the Gaussians cover all the space with no gaps is to plot the 2σ radius relative to the center of each Gaussian as a circle about center points, plotting two dimensions at a time (4.6). One can then extend this to include false color representation with natural mappings which use the natural processing of the human perceptual system to help map the 'quantity' of covered space as colors (black = no cover, green = one Gaussian, move towards red as more and more overlap occurs, see figure 4.5).

4.5.5 Measures of numerical fit quality

Numerical consistency of the computed fit can be checked by using the weights to attempt to recompute the right side of the $Mw = d$ fit equation. The two results d and $\hat{d}$ are plotted and compared. The number

$$\Big|\Big|\frac{Mw - d}{\max(||d||_1)}\Big|\Big|_2 \tag{4.56}$$

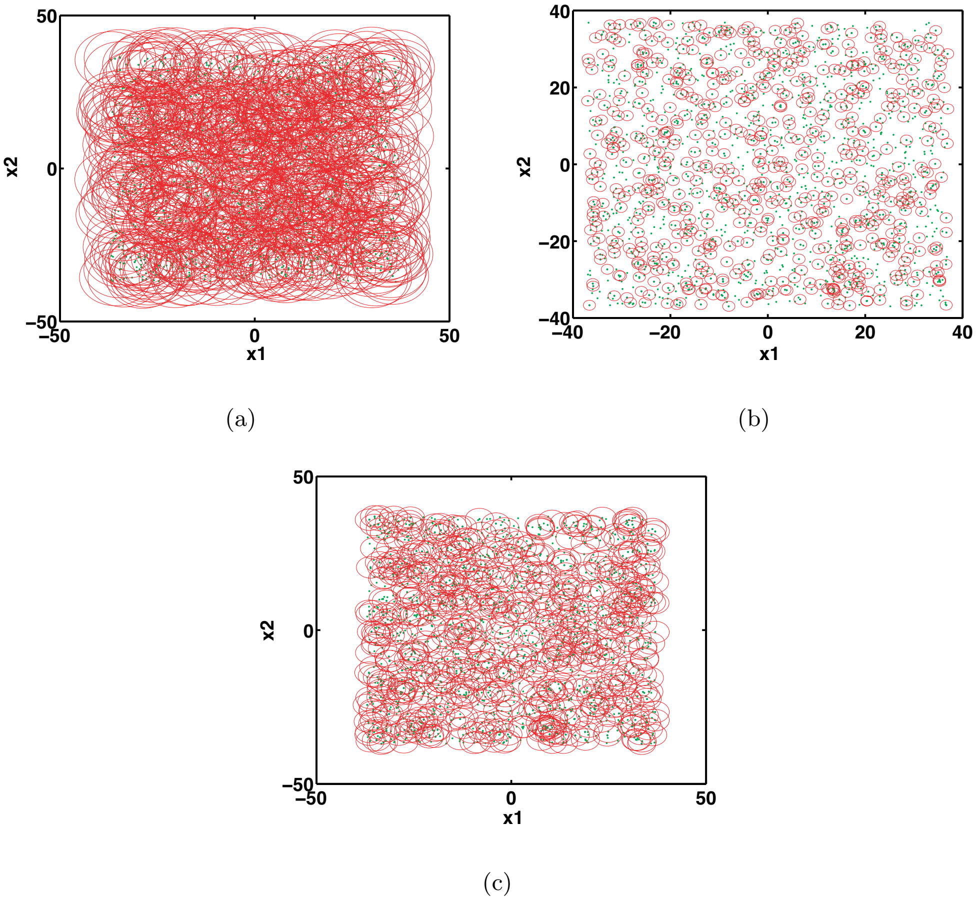
gives a single value measure of fit quality. The L_1 norm in the bottom of the fraction scales the fit to essentially a proportion of fit. Otherwise the numerical value returned by this check is deceiving since each iteration tends to grow.

4.5.6 Numerical stability

In order to assure numerical stability of the weights, one can normalize the weights to one or another constant maximum after computing the least squares fit. What this does is prevent the next iteration's d from exponentially increasing in the norm of the amplitude. Then the maximum amplitude scale of the w's can be determined by another least squares fit after convergence is achieved.

4.5.7 Visualize the value function and control

When a reasonable solution has been determined, or one that appears reasonable, it is important to visualize the resulting value function and control actions



(a)

(b)

(c)

Figure 4.5: A low dimensional visualization of the space coverable by the Gaussians for a function fit. The red dots represent the points covering the space, and the red circles represent the Standard Deviation of the centers of the Gaussians over the space scaled to minimize overlap. Too much redundancy results in an indeterminate problem.

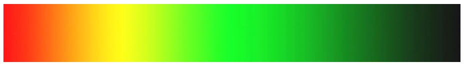


Figure 4.6: One typical colormap for Gaussian coverage: green is appropriate overlap, yellow is some overlap, red is too much overlap. In other words, black = no cover, green = one Gaussian, move towards red as more and more overlap occurs.

mapped over the state space. Though the system may be high dimensional, most real-world applications have a low dimensional interpretation which is possible. For example, in our recent work [89], we applied this methodology to an eight dimensional problem. A reasonable control solution could be visually confirmed versus an unreasonable solution by creating a 3d surface plot of the cost versus two-dimensional states in slices. An unreasonable or poor solution would yield no consistent shape, merely an appearance of noise, whereas a reasonable solution would yield fairly smooth shapes. A smooth shape may not be the numerically optimal solution possible, but in practice a smooth shape was produced only when a good fit was found. Reasonable solutions tend to be logical.

Similarly, the control response surface provides a positive measure of consistency. Both surfaces should be of reasonable maximum amplitude, though the control action is more intuitive in terms of the programmer's being able to recognize a reasonable state-dependent action (i.e. if a robot arm is being controlled, and given a particular state, if a control action which is near infinite is computed despite a penalty on control energy, it is likely that the designer is witnessing an inconsistent solution).

4.5.8 Simulation performance and repeated measures

The most important measure of a control's goodness of fit may be in terms of performance. This means that a control should be computed, then a short simulation performed and a performance criterion evaluated, such as the mean-square-error or 2-norm of the error between reference and actual output.

Given a stochastic system, it is important to perform repeated solution attempts using the same parameters in order to confirm a particular solution. For example, one setting of initial conditions may lead to a poor solution in one solution attempt, and a good solution in another. But averaged over several solution attempts given the same parameters/initial conditions, a good versus bad solution is evident since in general the problem will be solved well for good settings. It is in this way that repeated measures are essential in stochastic problems for determining a solution's validity.

4.6 Conclusion

In this and the previous chapter we addressed several topics related to function approximation in continuous time and space applied to nonlinear stochastic optimal control problems suitable for modeling biological systems. The FAS algorithm can effectively approximate optimal policies for active exploration-type problems, as we demonstrated with the pendulum on a randomly rotating base problem. The fact that the FAS algorithm can produce a viable control policy in the latter case, using the state augmentation method will be very useful for modeling sensorimotor learning and control. In our previous paper we showed that this method can effectively deal with redundancy, a common issue in motor control, as well as higher dimensional systems.

The shortcomings of these methods are due to the significant human effort (parameter adjustment) that must take place to implement them effectively. This chapter also deals with many of these shortcomings and suggests, where possible, numerical methods such as using performance criteria to pose optimizations over those tuning parameters. This can reduce the manual tuning that makes implementing many reinforcement learning and approximately optimal control policies difficult and time consuming.

Another future direction of this work is to combine global and local methods. Previously, iterative quadratic approximation methods were developed in our laboratory [103], [57]. The local methods suffer from the need for good initialization, but are very effective when in moderately close proximity to a solution. Thus it is reasonable to suggest that an effective algorithm would use the global method to initialize the local method and provide a check at each time step.

In the near future, we will be implementing these control policies in several novel robots which the author has developed to further explore the benefits of active exploration and to model human sensorimotor learning.

In some senses, all learning can be reduced to the estimation of observable or unobservable functions and parameters. Optimal control has been successfully applied in many simple settings for modeling sensorimotor control. This extension to redundant and unobservable systems is very powerful. In this context, estimation and control not only coexist, but they are also intermixed, driving each other to achieve an otherwise impossible control objective. A methodology such as the one presented here, which specifically makes use of the uncertainty rather than attempting to average it out, allows a more broad range of problems to be addressed.

Acknowledgment

The material contained in this chapter was, in part, originally published in the proceedings of the *IEEE American Control Conference, 2008*, by Simpkins, Alex, de Callafon, Raymond, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The co-authors listed in this publication directed and supervised the research which forms the basis for this chapter.

The material contained in this chapter was also, in part, accepted for publication in the proceedings of the accepted for publication in the *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2009*, by Simpkins, Alex, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

Chapter 5

Synergies and coordinated control

5.1 Introduction

It has been postulated that there may be a set of muscle synergies[14] which allow the simplification of the control of redundant degrees of freedom in the human sensorimotor system [10], [55]. In the original theory, these synergies would be fixed and thus task independent [83]. Several researchers have postulated that there may be a set of canonical postures for types of grips - see refs in [83]. However, it has been shown that the number and exact structure of the synergies is task-dependent, and is better described by an optimal control approach [101]. Additionally, studies have suggested that hand synergies in general have low dimensionality [83]. It may in fact depend on the type of task, as a study involving manipulation tasks [100] has found a much higher number than previous studies which focused on grasping tasks, and in the current study it is considered whether visuomotor matching synergies vary from manipulation (and grasping) task synergies. The visuomotor matching synergy characteristics are explored with several techniques not only to compare to other types of synergies, but also to characterize visuomotor matching synergies and characteristics of human performance on this type of task (mainly in terms of precision and accuracy).

A bottleneck in the sensorimotor system may explain or predict certain qualities of performance. A model for such a bottleneck is also considered (Figure 5.1) and fit to the data.

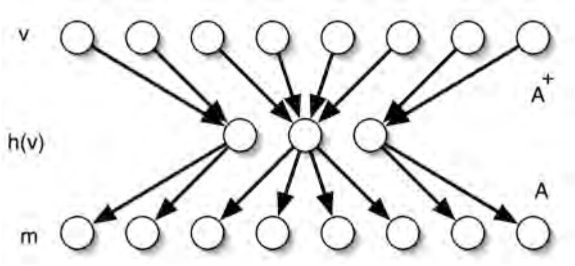


Figure 5.1: Simple model of sensorimotor bottleneck which may be responsible for limitations of performance. v represents sensory inputs (vision of a hand in space for example), A and A^+ represent a transformation and inverse transformation to some internal hidden states h, and m represents the final motor outputs resulting in hand positioning.

Several of these analyses require absolute angle data to facilitate comparisons. The CyberGlove by VTI Technologies (CyberGlove; Virtual Technologies, Inc, Palo Alto, CA), used in these experiments, measures a voltage which corresponds to absolute angle of several joints in the human hand. However, the CyberGlove's measurements vary across (but not within) subjects, and so must be calibrated for each user.

The typical calibration procedure involves instructing the subject to perform a series of postures such as making a fist, putting their hand on a flat surface, gripping a ball, etc. The subject's highly approximate joint angles are used to perform a simple calibration. This method suffers from poor reliability, and the joint angles a person performs vary significantly, and thus are difficult to relate to the voltage measurements. Constraining a human hand to specific joint angles is difficult. It is more straightforward to have the subject command the joint angles themselves.

Another method of calibration is simply normalizing data for each subject to 0-1 range or unit variance. Though this is useful for statistical tests such as Principal Component Analysis, only having data in these ranges limits the analysis possible. Another potential confound at times is that comparison of one joint to another is no longer possible since the subspace is elongated or compressed to have the same range in all directions.

The current experiment can thus be used to perform a calibration procedure automatically. Once it has been characterized what is the best precision and accuracy an average human can achieve using this method, we can use posture matching to perform a more robust calibration procedure.

5.2 Materials and Methods

5.2.1 Experimental setup and procedure

Eight Subjects, a mixture of male and female, in early twenties, wore a right-handed VTI Technologies 22 joint Cyberglove which contains proprietary bend sensors. The Cyberglove measured voltages corresponding to joint angles. Determining the joint angles required a subject-specific calibration, described later (voltage to angle is constant within subjects but not between subjects due to different fits). Data was recorded from the thumb's metacarpal-phalangeal joint (MCP), abduction between thumb and index finger (ABD), interphalangeal joint (IJ), and angle of thumb rotation passing through the trapeziometacarpal joint (TMJ). For the other

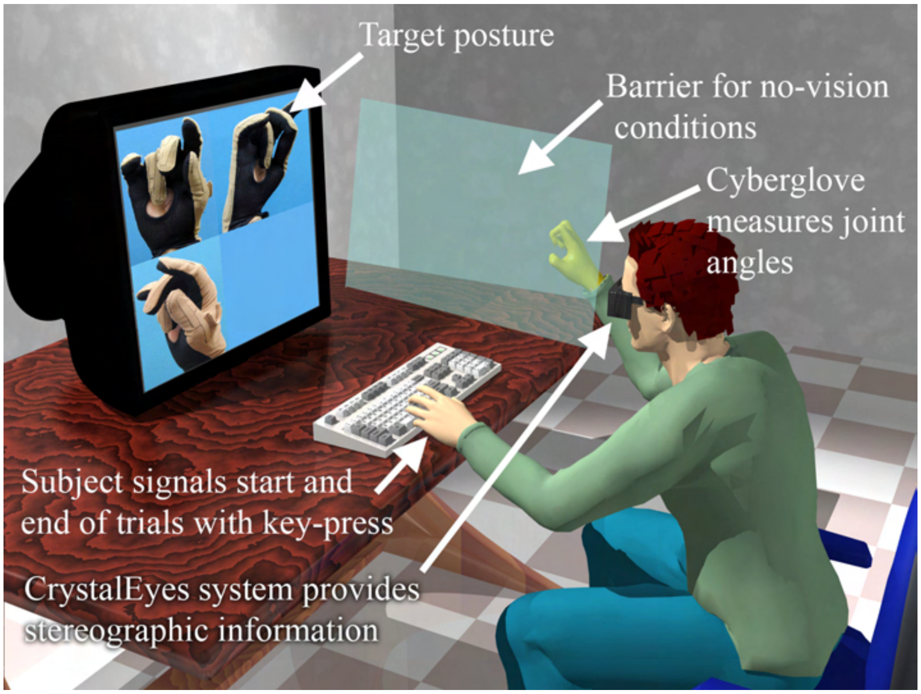

Target posture
Barrier for no-vision conditions
Cyberglove measures joint angles
Subject signals start and end of trials with key-press
CrystalEyes system provides stereographic information

Figure 5.2: Image of experimental setup. Subject sits in front of a stereographic image system, with an optional barrier (depending on vision or no-vision condition) which obscures their hand. Postures are presented on the screen and the subject attempts to match those postures, with no time constraints.

four fingers, data was recorded from the MCP, ABD, proximal-interphalangeal joints (PIJ), and distal-interphalangeal joints (DIJ). Resolution of the CyberGlove was approximately 0.1°

Subjects sat in a chair facing a high speed computer monitor which presented stereoscopic information to the subject. This was achieved by using the StereoGraphics CrystalEyes Workstation model CE-WS, coupled with a high performance CRT monitor set at a refresh rate of 160Hz. The subjects were presented with images of a hand in various postures. The subjects were told to practice the posture for as long as desired, and when ready, to open their hand to a flat position (See Figure 5.3(a)), press a key, then move their hand to match the posture on the screen, press the key again, and repeat for three iterations (total) per posture. Each key-press that began a trial was signaled by a tone, with a different tone signal for the second key-press to signify the end of the trial. Data was captured continuously during each trial from the start key press to the end key press at 100Hz sampling rate and 8-bit resolution. The captured data was buffered in memory and written to a file at the end of each subject session. This file included all information about the current experiment it included data for - subject number, condition, number of images, etc.

There was a vision condition, and a no-vision condition, where the subjects could view or not view, respectively, their hand to compare to the image on the screen. In the no-vision conditions, the subject's hand was obscured by a barrier placed adjacent to the computer display.

The target images were comprised of fifty meaningful hand postures and fifty randomly generated hand postures which were technically within human joint space (to keep more of the postures within human capability, the meaningful postures were randomly combined and reshuffled to generate the random postures).

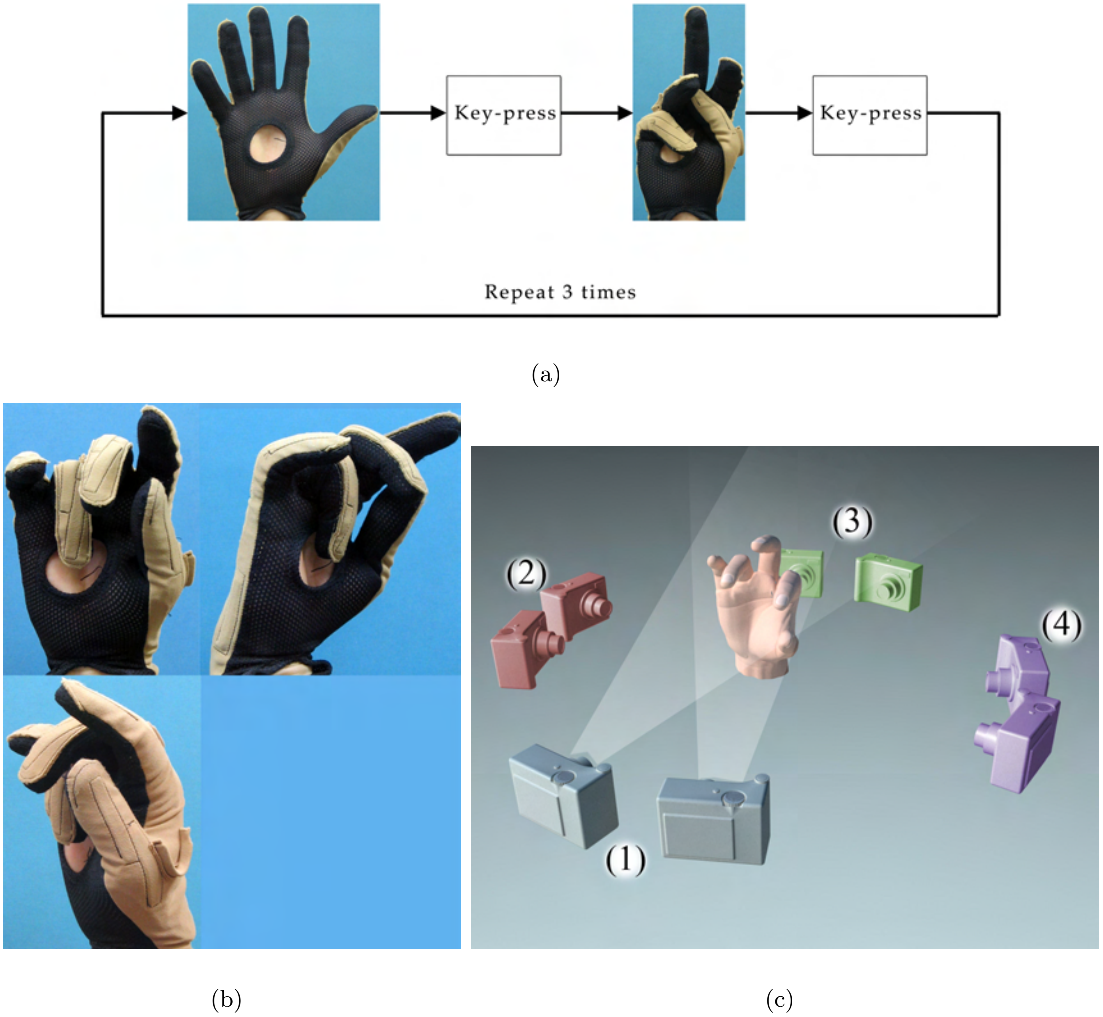


Figure 5.3: (a) Flow diagram of operations sequence of a single trial for subjects. (b)Sample trial stimulus image for subject to match (non-stereographic version). Note the VTI proprietary bend sensors sown into the glove, used to measure joint angles. (c) Pictorial representation of the multiple camera positions recording each hand posture, used to produce the 3D images of the hand. Each number represents one of the camera angles. Note the dual camera placement (and camera perspective depicted by light cone) for recording stereoscopic information.

The postures were first rendered using Poser 5, then a human experimenter wearing a Cyberglove attempted to match those computer generated postures, and stereoscopic photographs were taken of his hand at several camera positions (Figure 5.3(c)), and the Cyberglove joint angles at those postures were recorded, as well as a calibration routine. The stereoscopic image on the screen showed additional 3D information by displaying three different angles (see Figure 5.3(b)).

Each subject had the order of the conditions randomly shuffled. Subjects were given as much time as they desired between trials. The four conditions were meaningful-vision, meaningful-no-vision, random-vision, random-no-vision.

5.2.2 Data analysis

The data was analyzed using canonical correlation analysis (CCA), modified factor analysis (FA), Principal Component Analysis (PCA), and an estimate of the Signal to Noise Ratio (SNR). The residuals for the matching task were computed (how closely the subjects matched the template data), as were joint by joint correlations to the template data. Statistical tests for significance were performed on the results of the aforementioned methods, where appropriate, using a 2-way analysis of variance (ANOVA). ANOVA's were performed on the duration of each trial, path length of each trial trajectory, trajectory straightness, and the SNR.

Pooling the data

Each analysis technique was performed on the subject data for the following methods:

- (SBS) Subject by subject.
- (SA) Subject averaged (before or after data analysis).

- (P) Pooling all subject data.
- (NUP) Pooling all subject data after normalizing each subject's data to unit variance.
- (ONUP) Pooling all subject data after normalizing and orthogonalizing each subject's data to unit variance (see description below about orthogonalizing).

The pooled orthogonalized data yielded the best comparison between the comparison set and the ONUP subject data. Therefore, the other results will generally be omitted for clarity. For the SNR and residuals the data was processed in both normalized space and degree space.

Normalization and Orthogonalization

The Unit Variance Normalization was achieved by dividing each joint column of data by the Standard Deviation of that column.

The orthogonalization was performed as follows for data input to the FA and CCA analyses. PCA was performed on NUP data, as orthogonalizing the data results in each variable contributing equally to the overall variance in the data.

Define

$$s = \frac{1}{2}\Big(\mathrm{Cov}(x) + \mathrm{Cov}(y)\Big) \tag{5.1}$$

we can then orthogonalize x and y by

$$x' = xs^{-1/2} \tag{5.2}$$

$$y' = ys^{-1/2} \tag{5.3}$$

and one can verify this by checking that

$$\mathrm{Cov}(x') \approx I \tag{5.4}$$

Principal Component Analysis

principal components analysis (PCA) has been used as a method of measuring the dimensionality of data [83], [100]. Here we use it both to estimate the dimensionality of the posture space and of the sensorimotor noise, as well as to observe which joints contributed most to overall variability accounted for by each principal component.

Postures

The principal components of the postures were determined by pooling the data together for all subjects, after averaging over the three trials per posture, and scaling appropriately per subject (all three different forms were computed - angle, normalized 0-1, and normalized to unit variance).

Motor noise

The principal components of the motor noise were determined by taking the covariance of the three observations for each mean-centered posture, which gave a 20x20 covariance matrix for each of the 50 postures for the four conditions, so for each subject the pooled covariance matrix was 20x20x200. The subject data was pooled, yielding a 20x20x1600 covariance matrix. PCA was then performed on this matrix, yielding a measure of the characteristics of motor variability in the human sensorimotor system.

Canonical Correlation Analysis

Canonical correlation analysis (CCA) [3], [62] was performed on all four conditions (meaningful vision, meaningful no-vision, random vision, random no-vision) between the subject data and the comparison set (the data from the experimenter who generated the posture image photographs). To facilitate comparison, the comparison set was also normalized to unit variance and orthogonalized.

There is a fundamental indeterminacy in the canonical correlation coefficient vectors, since rotating the vectors by pi radians results in the same magnitude sum. In order to use these coefficients for generating posture deviations that can be appropriately compared between conditions, a method of flipping the coefficient vectors when appropriate was devised as follows:

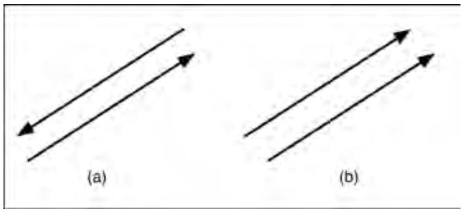


Figure 5.4: Graphical representation of vector quantities. By considering the angle between the two vectors in (a), one can determine if one vector should be flipped in its direction in order to facilitate comparison (b).

For each joint (one to twenty) and condition (one to three) the dot product was taken between the coefficients of the current joint and condition with the coefficients of the current joint and next condition. The dot product of two vector quantities is

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos(\theta) \tag{5.5}$$

Thus the angle between the two vectors is given by

$$\theta = \cos^{-1}\Big(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\Big) \tag{5.6}$$

The state of the vector which produced the smallest angle was chosen as the direction of the vector. In this way the coefficients were aligned and thus allowed better comparison for visualization as hand posture deviations, and as numerical plots.

The canonical correlation coefficients which have a high value may represent processes or features shared by both data sets, with the intensity of the sharing reflected by the value of the coefficient. Thus plots and posture deviation visualizations clarify which components of those vectors contribute the most to the correlation.

Modified Factor Analysis

A modified Factor Analysis model was fitted to the data as well. The model maps inputs i to outputs o using hidden states h[99].

$$h = Ai + \eta \tag{5.7}$$

$$o = Bh + \nu \tag{5.8}$$

η and ν are both zero mean Gaussian noise processes with covariance matrices Q and R respectively. The matrix A represents a mapping from input to hidden states, and B represents a mapping from hidden states to outputs. As with standard factor analysis methods, R is diagonal. The expectation maximization algorithm to determine A, B, Q, and R proceeds as follows:

E-step:

Define

$$K = QB^T(R + BQB^T)^{-1} \tag{5.9}$$

Then

$$\bar{h} = Ai + K(o - BAi) \tag{5.10}$$

Define the covariance of h as S.

$$S = Q - KBQ \tag{5.11}$$

M-step:

We update the model parameters according to the following:

$$A = \frac{1}{N}\sum_{j=1}^{N} \bar{h}i^T \Big(\frac{1}{N}\sum_{j=1}^{N} ii^T\Big)^{-1} \tag{5.12}$$

$$B = \frac{1}{N}\sum_{j=1}^{N} o\bar{h}^T S^{-1} \tag{5.13}$$

Similar updates for Q and R can easily be derived.

Posture deviation visualization

Each analysis, PCA, CCA, and FA was visualized by using the component coefficients, appropriately scaled, to command postural deviations about a mean posture for a 3D realistic human hand model. This facilitated a comparison between analysis results, and recognition of patterns of significant results in terms of joints.

ANOVA

A 2-way Analysis of Variance (ANOVA) was performed on many of the analyses. The factors being subject, joint, meaningful vs. random, and vision vs.

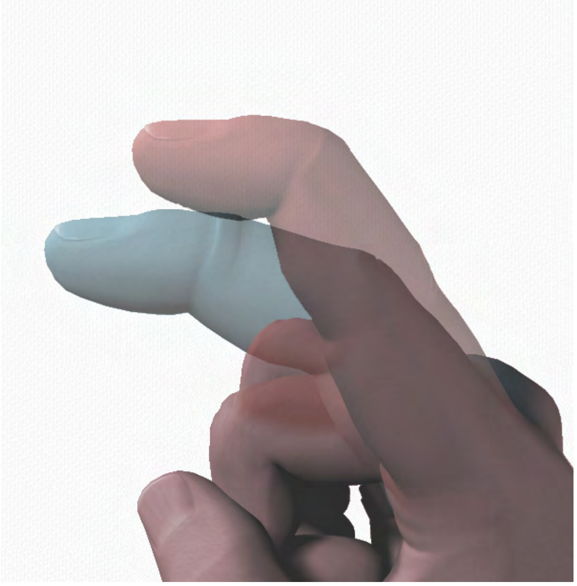


Figure 5.5: Example of postural deviation (both images superimposed for comparative purposes, and only one finger deviating for clarity).

no-vision conditions. The ANOVA was performed on the data converted into joint angles, as well as a second ANOVA on the data normalized to unit range (for all cases). The subject factor was treated as a random variable since it is a random sample from the population. This determined levels of significance and what variable is correlated with significant changes in the data. ANOVAs were performed on the residuals, trajectory straightness, duration of movement, and the Signal to Noise Ratio.

Trajectory straightness

The trajectory characteristics of the joint vector for each posture, condition, trial, and subject were analyzed by two methods.

Method 1 - Max deviation: The first took the maximum normal deviation from a straight line trajectory (See Figure 5.6) in twenty dimensional joint space, then dividing by the length of the straight line vector A:

$$D_i = \frac{\left|\left|B_i - \frac{B_i \cdot A}{A \cdot A} A\right|\right|_2}{||A||_2} \tag{5.14}$$

The subscript i denotes the ith vector B computed along the trajectory. The maximum deviation is then taken to be

$$D_{max} = \max(D_i) \tag{5.15}$$

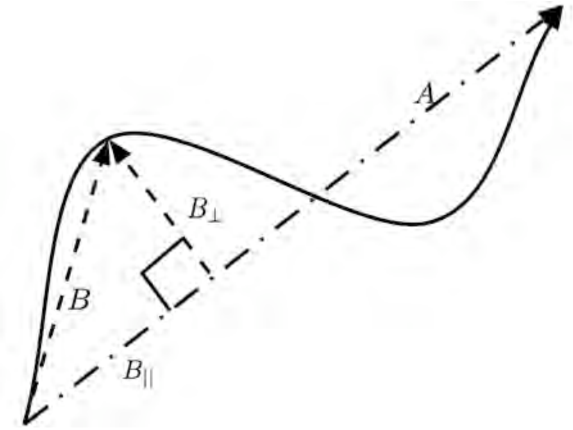


Figure 5.6: Graphical depiction of vector used to measure straightness. The solid line represents the actual trajectory, the dotted, dashed, and dash-dotted vectors are used to calculate trajectory straightness (see text)

Method 2 - Duration: The second trajectory analysis (See Figure 5.7) method centers around computing the length of the trajectory with respect to a straight trajectory. In order to eliminate adding distance due to jitter or noise during a static posture (i.e. at the end of the trajectory, where a finger might be experiencing slight muscular tremors), the maximum joint velocity, V_{max} for each trial was computed. Then the distance accounted for was cut off at $V_{cutoff} = 0.1 * V_{max}$

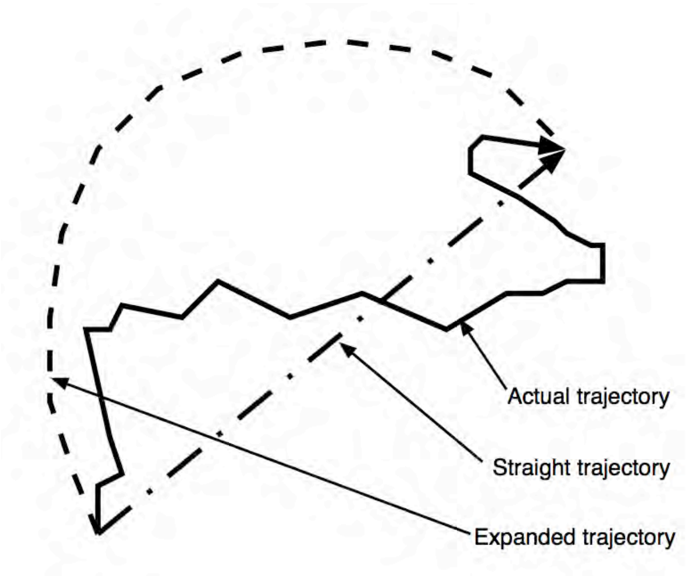


Figure 5.7: Graphical depiction of vector used to measure straightness, method 2. The overall distance travelled during the trajectory can be considered as a single deviation curve (dashed line) from the straight line (dash-dot)

Signal to Noise Ratio (SNR)

An estimate of the Signal to Noise Ratio was computed by an algorithm according to the following pseudo-code (s=subject, j=joint, p=posture, r=repetition, c=condition, ovp=over postures, ovr=over repetitions):

```
for each s,c and j,
 signal(s, c, j) = var$_{ovp}$( mean$_{ovr}$ (angle(s, c, j, p, r)))
 noise(s, c, j) = mean$_{ovp}$ (var$_{ovr}$ (angle(s, c, j, p, r)))
 SNR(s,c, j) = $\sqrt{\frac{\text{signal}}{\text{noise}}}$
end
```

Mapping cyberglove space into joint angle space

In order to map the cyberglove a/d counts into joint angles suitable to control the 3D hand model in Poser, a procedure needed to be derived which would recover the abductions for each finger from the bend sensors in the cyberglove which were configured slightly differently from the model. The following procedure was performed.

1. Rearrange the raw cyberglove data into the joint arrangement for the cyberglove as follows:

   (4, 1, 2, 3, 5, 11, 6, 7, 8, 11, 9, 10, 12, 15, 13, 14, 16, 19, 17, 18)

2. Loop through the joints (one to twenty), scaling and translating the data according to the subject specific calibration, determined by either a minimization algorithm or least squares fit.

   $\theta_{\mathtt{N_j}} = \mathtt{a_0} + \theta_{\mathtt{cyberglove}} * \mathtt{a_1}$

3. Now recover the abduction angles for each individual finger from the Cyberglove's coupled measurement

   $\mathtt{tmp} = (\mathsf{180} - \theta_{\mathsf{10}} - \theta_{\mathsf{14}} - \theta_{\mathsf{18}})/\mathsf{2}$

   $\theta_{\mathsf{6}} = \mathtt{tmp} + \theta_{\mathsf{10}} + \theta_{\mathsf{14}} + \theta_{\mathsf{18}} - \mathsf{90}$

   $\theta_{\mathsf{10}} = \mathtt{tmp} + \theta_{\mathsf{14}} + \theta_{\mathsf{18}} - \mathsf{90}$

   $\theta_{\mathsf{14}} = \mathtt{tmp} + \theta\mathsf{18} - \mathsf{90}$

   $\theta_{\mathsf{18}} = \mathtt{tmp} - \mathsf{90}$

   $\theta_{\mathsf{6}} = -\theta_{\mathsf{6}} + \mathsf{25}$

   $\theta_{\mathsf{10}} = -\theta_{\mathsf{10}} + \mathsf{10}$

   $\theta_{\mathsf{14}} = -\theta_{\mathsf{14}} - \mathsf{5}$

   $\theta_{\mathsf{18}} = -\theta_{\mathsf{18}} - \mathsf{20}$

The data is then in joint angle space suitable for generating postural hand deviations from the Poser 6 3D rendering program.

5.3 Results

5.3.1 Dimensionality results and characteristics

The PCA yields an estimate of the dimensionality of the postures or motor noise, depending on the input to the algorithm. The plot shown in Figure 5.8(b) gives an estimate of dimensionality of the posture data. For example, if all joint data contributed equally to the overall variance in the data, those charts would have all bars the same length. Instead we see a falloff. We can determine the number of dimensions of the synergies by choosing a cutoff, and computing the number of principal components which account for that amount of overall variance. We choose, as has been done previously (and is fairly standard) [100], 95% and 85%.

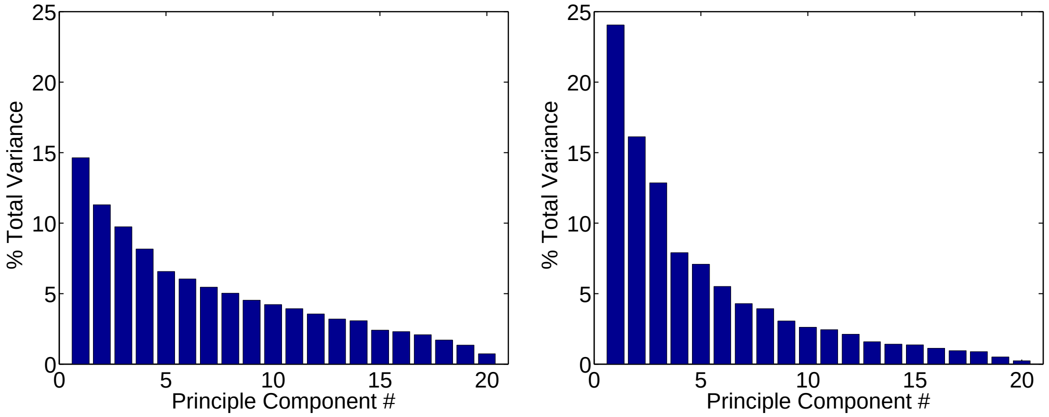
In order to compare to previously published findings for manipulation tasks, we summarize the PCA results in a table of similar form to Table 1 and Table 3 from [100]. Though PCA of absolute angle data tends to underestimate dimensionality due to scaling differences for different joints [83], we see only a slight reduction in synergy counts here.

Table 5.1: Synergy counts in position space for motor noise.

	Angle	0-1	Unit Var
MV	15 10	17 13	16 12
MNV	15 11	17 13	16 12
RV	16 12	17 13	17 13
RNV	15 11	16 13	16 12
POOLED	16 12	17 13	17 13

The PCA of the motor noise (PCAMN) is a different measure than in [100]. One significant result of this PCAMN is that the dimensionality is much higher than expected, being, when we average over all conditions and methods of scaling, 16.2 and 12.2 components (when choosing quantity accounting for 95% and 85% variability, respectively). The angle data is smaller, as expected since some joints in joint angle space have smaller overall variance in magnitude. However, the number of synergies only differs by roughly one (See Table 5.1). Another significant feature to note is the fact that there is strong similarity in all conditions.

The average number of synergies for this posture matching task (over all conditions and for the data pooled) is 12.6 and 8.3 for 95% and 85% of the variance in the data accounted for, respectively. Table 5.2 shows the number of synergies broken down by condition and for all the data pooled as well.



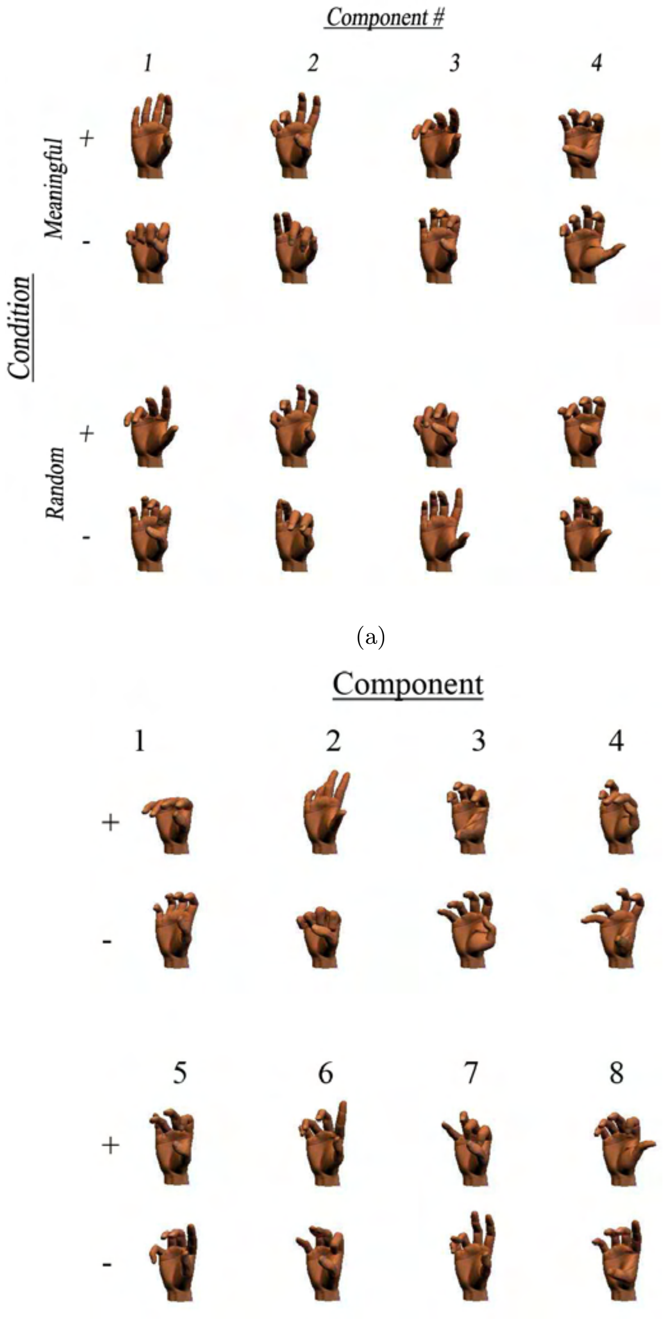
(a)

(b)

Figure 5.8: Eigenspectrum of the principal components for motor noise and postures. (a) PCA of Covariance of motor noise. (b) PCA of postures Note that the motor noise has a much more flat spectrum than the postures. This is the spectrum for the normalized 0-1 analysis. Other analysis results are similar and thus are omitted for clarity.

Table 5.2: Synergy counts in position space for posture matching performance.

	Angle	0-1	Unit Var
MV	12 8	13 8	12 8
MNV	12 8	13 8	11 7
RV	13 10	15 10	13 8
RNV	13 9	15 10	11 7
POOLED	12 9	14 10	12 8



(a)

(b)

Figure 5.9: (a)PCA posture deviations for task, vision and no-vision conditions pooled. (b) PCA posture deviations for motor noise, all conditions pooled since no significant differences existed in the data. The first 8 components are displayed.

What is clear from Figure 5.9 (both parts a and b) is that, though the condition of the task did correspond to significant differences in the results of the analysis, the synergies show slightly similar shapes. Consider the first four PC's for the task space (Figure 5.9(a)). For the first component, the MPJ's, PIJ's and DIJ's for each finger (except the thumb) generally have a large coefficient, though the meaningful condition coefficients are all in the same direction, whereas the random condition coefficients follow a slope. The thumb TMJ, MPJ and IJ show contributions as well, with little from the ABD. The second PC shows more similarity between conditions, with the index and middle finger MPJ's and PIJ's, as well as the ring and little finger PIJ's showing large contributions. The abductions show little contribution to the first few principal components. Generally the increase in variability, which will be presented for the residuals portion of the experiment, for the random conditions, shows in the PC synergies having a slightly different shape than the meaningful conditions. Higher dimensionality of the synergies result in many differing postural synergy shapes, with no one group of joints entirely dominant, though the first principal component accounts for the largest amount of the variability in the data and many of the joints recorded show strong contribution to the component, though abductions are the lowest contributors.

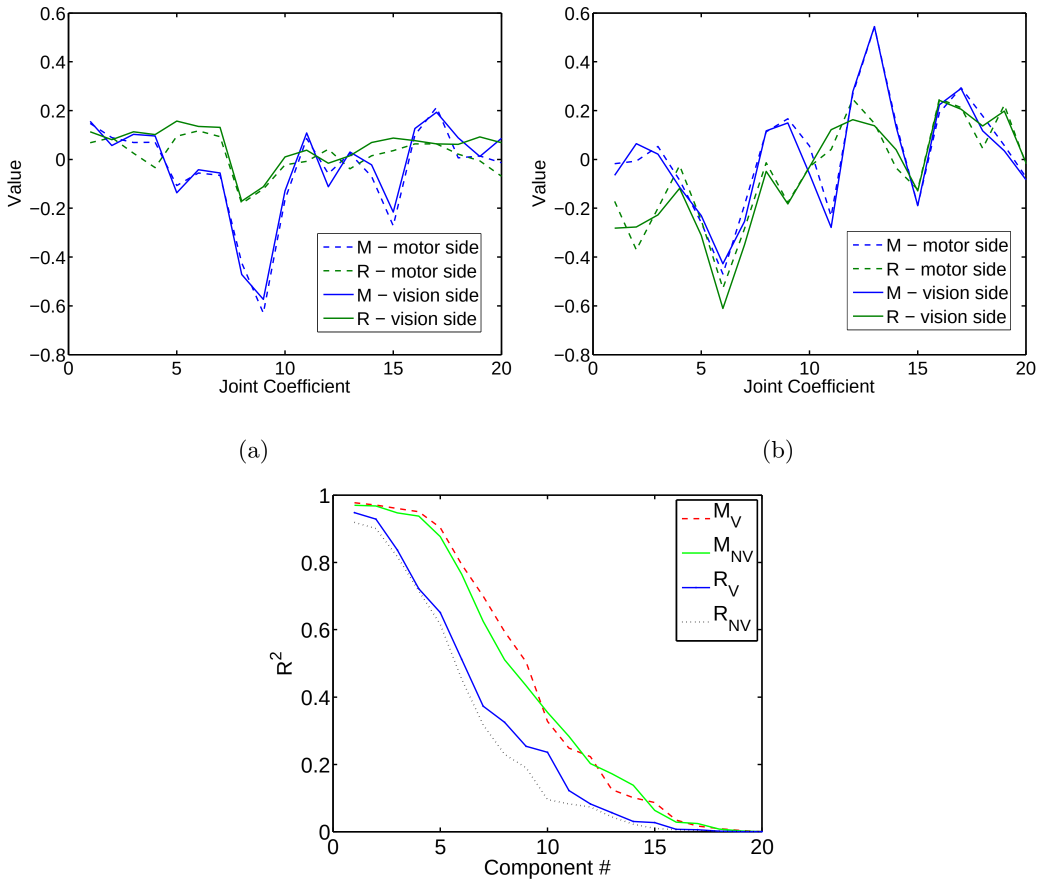
The dimensionality of the motor noise (Figure 5.9(b)) is nearly twice the dimensionality of the task space PC's, and shows a similar variety of hand synergy shapes. The lack of significant differences with all four conditions suggested pooling all conditions for clarity, and thus twice the number of synergies are displayed. The first PC is still the largest contributor (by a small margin) of variability in the data and so we will point out some of its features. First, the MPJ's are large contributors, with all the same sign, and second, the PIJ's and DIJ's are large contributors, with the same signs as each other, but opposite from the MPJ's. The ABD's have little contribution to the first and second PC's, though they contribute further down in the group of PC's. Thus though they contribute little to the postural synergies, they also have smaller variability. The first PC also shows little variability of the joints of the thumb as well. Other PC's show large contributions of the thumb joints to variability of the data, however.

5.3.2 CCA

The most immediate result to note about the CCA is that the vision and no-vision conditions show little difference, while meaningful and random conditions do show a difference, though the pattern is somewhat similar. In addition, the random conditions have a consistently lower correlation than the meaningful conditions.

The input and output side correlation vectors were compared, as a general check, by multiplying the transpose of one by the Moore-Penrose Pseudoinverse other. The more one side is the inverse of the other, the closer the result should be to the scaled identity matrix (See Figure 5.11(a)). Here we see that the first few components are moderately close to inverses of each other, with a decrease in relationship as we consider higher and higher components (less diagonality, more randomness). This is reasonable since the correlation decreases for higher CCA components.

Referring to Figure 5.11, we consider first the vision/no-vision differences. The first several canonical correlation components (CC's) show the minimal difference between vision and no-vision conditions. As the correlation becomes worse (for higher CC's), there is more and more difference between conditions observed. However there remains a stronger similarity between vision and no-vision conditions than between meaningful and random conditions. The first CC shows that there is the

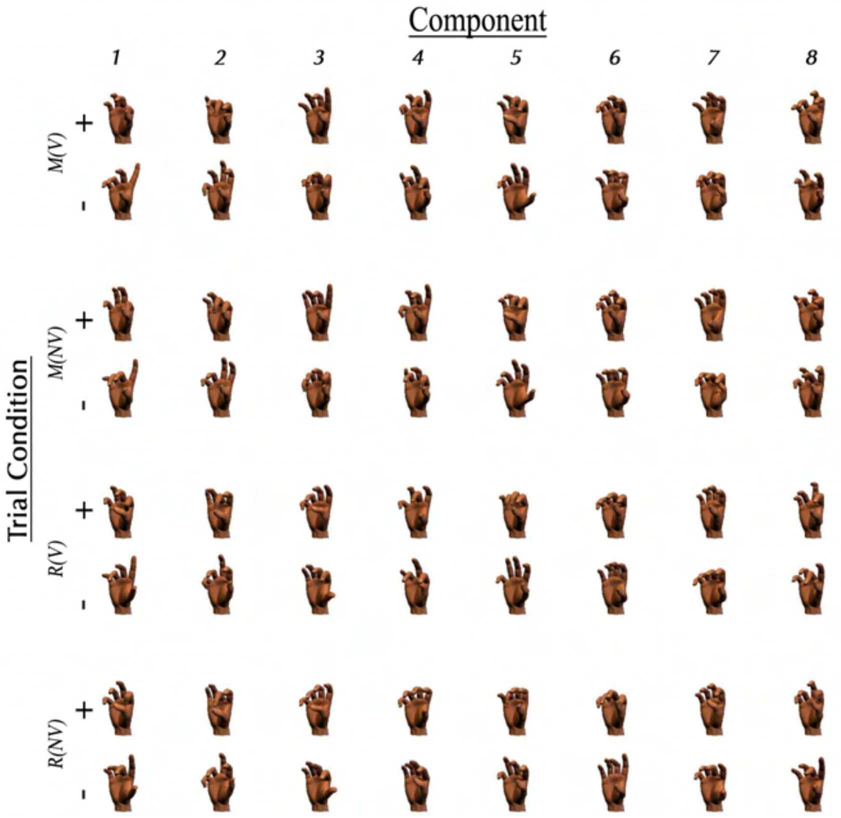


(a)

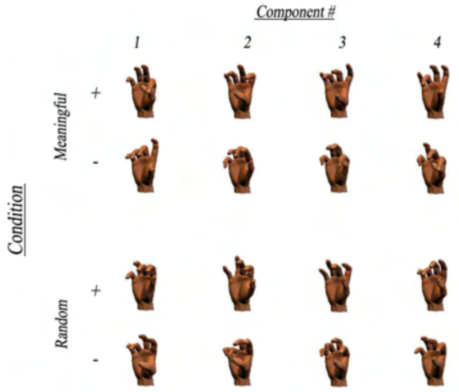
(b)

(c)

Figure 5.10: (a) and (b) First and second canonical correlations components, respectively, displaying the similarity between vision and motor sides. The plots are organized thumb, index, middle, ring, small fingers (in groups of four joints per finger). (c) Canonical correlations, for othogonalized pooled data, normalized to unit variance. Again we see more difference between meaningful and random conditions than vision and no-vision conditions.



(a)



(b)

Figure 5.11: (a)CCA posture deviations for ONUP data, vision and no-vision conditions NOT pooled to emphasize similarity between vision and no-vision conditions. (b) FA posture deviations, ONUP data vision and no-vision conditions pooled.

largest correspondence between subject and template data in the joints of the index finger, with decreasing contribution as one considers in turn the middle, ring and little fingers. The thumb ABD joint corresponds well in the first CC as well. The random conditions are similar to the meaningful conditions, with a smaller difference between the middle, ring, and little fingers than in the meaningful conditions. The thumb has a larger coefficient for its TMJ joint in the random than in the meaningful conditions. Again for this analysis, similar to our PCA results, high dimensionality makes for several possible synergistic correlations. The second CC shows the largest coefficient for the middle finger MPJ and PIJ, with decreasing coefficients for the MPJ's and PIJ's of the other fingers. The thumb has a significant contribution from each of its joints in all conditions of the second CC. A pattern which appears in the most correlated posture deviations is that generally one or two fingers dominate with decreasing contributions from other fingers or joint groups in a fairly smooth pattern.

5.3.3 Joint by joint correlations

The abduction degree of freedom was consistently lower than all other joints in joint by joint correlation to the template data. The highest was the second joint from the tip of the finger, the PIJ. All joints were reasonably high in correlation (ie no joints were near zero correlation) to the template data. Another important point is that all five fingers exhibit similar performance properties in terms of correlations with each joint of the template data.

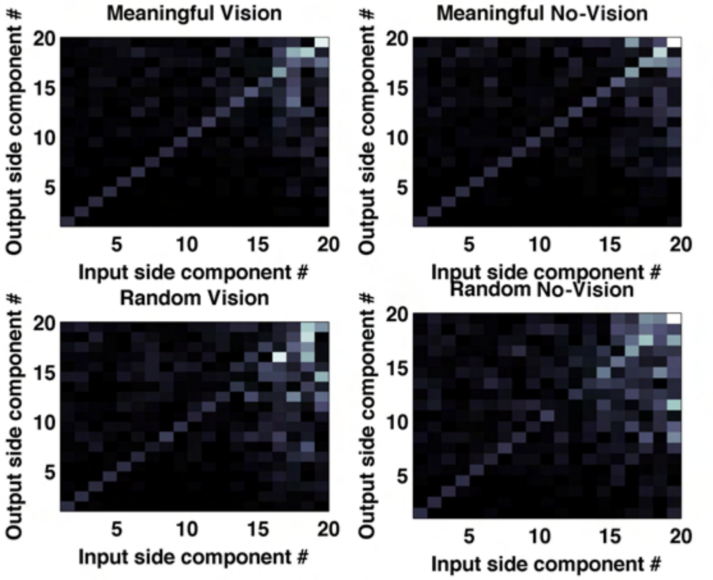


Figure 5.12: Canonical correlation vectors (shown here for the ONUP data) should represent the transformation (A) from input to a new variable U maximally correlated with a new variable V and transformation (B) from motor side to a new variable V maximally correlated with U. This is tested by multiplying $A^T * (B^T)^\dagger$ (to get a matrix which, if each transformation were the inverse of the other, would be the identity matrix). Here we see that as we compare the canonical correlation vector corresponding to further canonical correlation components, the matrices are less and less inverses of each other, which is expected since correlation reduces for each successive canonical correlation component.

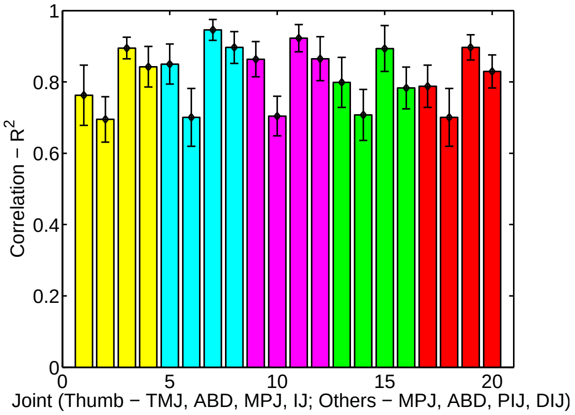


Figure 5.13: Joint by joint correlations of subject data vs. template data. The correlations are averaged over subjects and conditions, and the chart is organized by finger (the color/order is yellow/thumb, cyan/index, purple/middle, green/ring, red/little). TMJ - angle of thumb rotating across palm, ADB - Abduction, MPJ - joint where the finger meets the palm, PIJ - joint second from finger tip, DIJ - joint closest to finger tip, IJ - outer thumb joint.

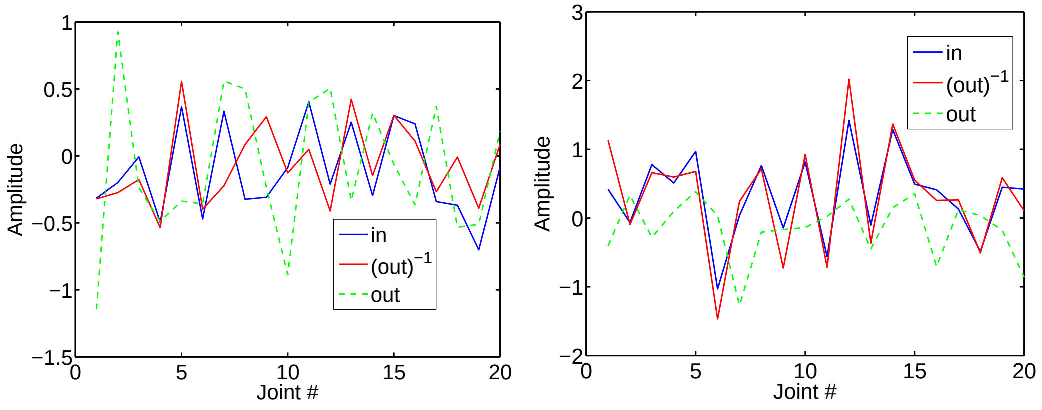
5.3.4 FA

The factor analysis model should identify two transformations which are pseudoinverses of each other. Therefore as a check, we multiply the input and output factor loading matrices and observe how close the resulting matrix is to the (scaled) identity matrix. Here we do not see the falloff of the diagonality but a fairly consistent relationship which is quite diagonal. Additionally a comparison of the first two factor loadings (input side) with the first column of the pseudoinverse of the output side factor loadings shows strong similarity.

We consider the posture deviation visualizations of the factor loadings in Figure 5.11(b), in which the vision and no-vision conditions were pooled for clarity. Similarly to the CCA first CC, the index finger has the largest coefficient for the first component in the meaningful condition, with slightly decreasing contribution from the other fingers and joints. Additionally, the other finger coefficients are all in the opposite direction from the index MPJ and PIJ, except for the thumb, which has a fairly high contribution as well in its TMJ and MPJ joints. Additionally, the PIJ joints of the other fingers in the meaningful conditions contribute the most of the other joints.

5.3.5 Mapping from cyberglove to joint angles

The order of the joint numbers was arranged in the following way. Thumb - TMJ, ABD, MPJ, IJ. Then for the index, middle, ring and little the order follows, finger by finger as MPJ, ABD, PIJ, DIJ. See Figure 5.16. Though the correlations for the abduction joints were not as high as the other joints, all joint correlations were acceptable. Most comparisons relate subject to template data, and it is clear that all subjects exhibited a similar pattern of correlations for the mappings.



(a) (b)

Figure 5.14: (a)FA factor 1, meaningful condition, (b) FA factor 1, random condition. Both plots are generated from data which have vision and no-vision conditions pooled.

5.3.6 Residuals - how well did subjects actually perform the task?

It is notable to consider that different joints of the hand have differing ranges in absolute angle coordinates. Thus it is more appropriate when comparing one joint to another to consider normalized joint space (0-1 or unit variance). The residuals are provided in the Table 5.3, Table 5.4, and in a bar chart of the pooled conditions in Figure 5.17.

Notable is that in both cases (degrees and normalized to unit variance), the vision and no-vision conditions are not significantly different ($F_{(1,28)} = 0.91, p =$ 0.3699), whereas the meaningful and random conditions are significantly different ($F_{(1,28)} = 16.84, p = 0.0042$). In short, subject accuracy is reduced significantly for the random conditions vs. the meaningful conditions, and slightly for no-vision vs. vision conditions. Similar to what was found in [83], mean Standard Deviation

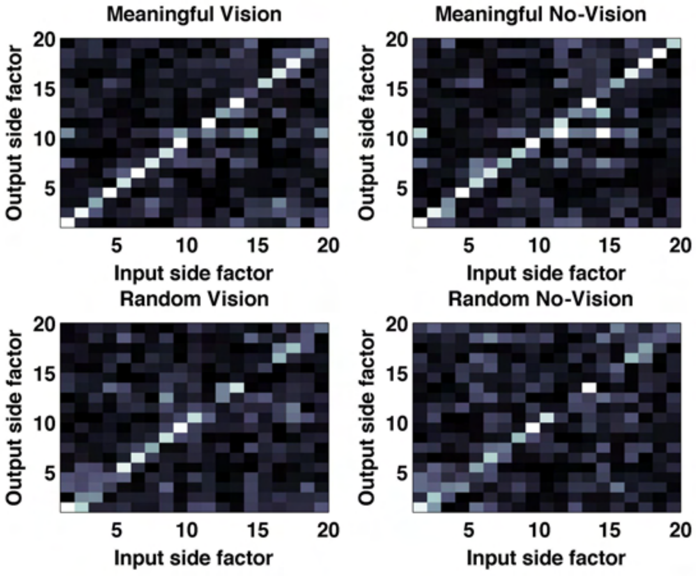


Figure 5.15: Factor loadings (shown here for the ONUP data) should represent the transformation from input to output (A) and output to input (B). This is tested by multiplying $A^T * B^T$ (to get a matrix which, if each transformation were the inverse of the other, would be the identity matrix).

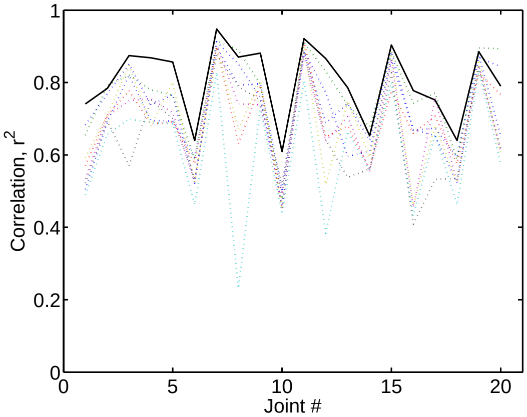


Figure 5.16: Mapping to angles, joint by joint correlations for the linear fit mapping cyberglove data into joint angles. Raw cyberglove data from each subject (solid line is the template data) is mapped into joint angle space and then compared to the template joint angle data to compute the correlation.

(averaged over subjects and trials) was within the 5-10° range [84] of grasping final postures prior to biomechanical coupling with an object.

Each finger has a fairly similar pattern of performance. That is, the ABD has the highest residual, or nearly the highest, and the lowest is generally the PIJ. The thumb has a highest residual for the ABD and TMJ, with the MPJ and IJ the lowest. All fingers are within roughly 20% accuracy at worst, with best accuracy being slightly less than 10%.

5.3.7 Duration

The durations of the trial trajectories (the time which passed from start of the trial until the velocity has dropped to 10% of maximum) are summarized in Table 5.7. Though none of the factors are significant at the 0.05 level, M-R

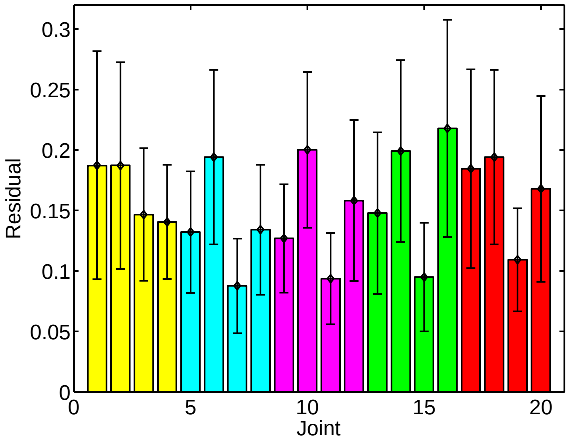


Figure 5.17: Residuals for all conditions, trials, repetitions, and subjects pooled, results presented by finger, with Standard Deviation errorbars. Data normalized to unit range prior to pooling to facilitate comparisons. Joint arrangement the same as Figure 5.13.

Table 5.3: Absolute values of normalized residuals, collapsed over subjects, trials, and joints. Mean (Standard Deviation).

	Vision	No-Vision
Meaningful	0.13 (0.18)	0.14 (0.18)
Random	0.17 (0.23)	0.18 (0.24)

Table 5.4: Absolute values of residuals in degree space, collapsed over subjects, trials, and joints. Mean (Standard Deviation).

	Vision	No-Vision
Meaningful	13.14 (7.49)	13.37 (7.84)
Random	15.43 (8.82)	16.63 (9.31)

Table 5.5: Normalized residuals, collapsed over subjects, trials, and joints. Mean (Standard Deviation).

	Vision	No-Vision
Meaningful	0.00 (0.18)	0.01 (0.18)
Random	-0.02 (0.23)	-0.01 (0.24)

Table 5.6: Residuals in degree space, collapsed over subjects, trials, and joints. Mean (Standard Deviation).

	Vision	No-Vision
Meaningful	0.27 (7.49)	0.12 (7.84)
Random	-0.16 (8.82)	-0.23 (9.31)

($F_{(1,28)} = 4.12, p = 0.0706$) as a factor is closer to significant than V-NV ($F_{(1,28)} = 0.59, p = 0.4686$) as a factor. In addition, vision conditions took slightly longer than no-vision conditions; and though the difference was not statistically significant, it was consistent for all conditions.

Table 5.7: Table of mean (Standard Deviation) duration time values, in seconds, for each condition, pooled over subjects and trials.

	Vision	No-Vision
Meaningful	2.65 (1.47)	2.17 (1.45)
Random	3.79 (2.54)	3.36 (2.51)

5.3.8 Signal to Noise Ratio

As can be seen from Figure 5.18, the meaningful conditions have a higher SNR ($F_{(1,28)} = 38.89, p = 0.0004$) than the random conditions, whereas the vision and no-vision conditions do not appear to differ significantly ($F_{(1,28)} = 4.85, p = 0.0634$). Though vision and no-vision differences do not appear to be significant, there appears to be more difference between vision and no-vision conditions for random versus meaningful conditions.

Another result is that each finger, with the exception of the thumb, follows a similar pattern of SNR per joint. The PIJ has the highest SNR of all joints for each finger (except the thumb), while generally the ABD joints have the lowest SNR. The thumb's highest SNR is in its MPJ, and lowest is close between the IJ and TMJ.

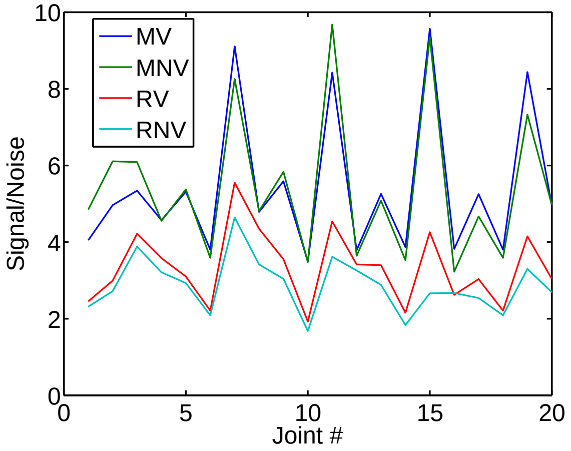


Figure 5.18: Signal to Noise Ratio for each joint and condition, averaged over subjects, repetitions, and trials.

5.3.9 Trajectory path length

The trajectory path length does not vary significantly for vision vs. no-vision conditions. It varies significantly at the 0.05 level , but not the 0.01 level for meaningful vs. random conditions ($F_{(1,28)} = 6.06, p = 0.0329$). Joints ($F_{(19,28)} = 0, p = 1$) and subjects ($F_{(7,28)} = 1, p = 0.4992$) are not significant factors for path length. The way in which the path length varies is that the length is greater in the case of random than in the case of meaningful conditions.

This is essentially showing that during the random posture trials, subjects took a less direct route to move from the start to finish posture than for meaningful condition trials.

5.3.10 Maximum deviation

The maximum deviation method of analyzing the trajectories is somewhat difficult to interpret, since it only gives information about distance from a linear trajectory, and when the data is scaled, the subspace becomes elongated in a potentially undesirable fashion.

However, meaningful and random conditions were significantly different, while vision and no-vision conditions were not. These results are consistent with our other results and so contribute to consistency.

5.4 Discussion

Comparing Table 5.2 with Table 1 and 3 of [100], we see that the number of synergies for the posture matching performance is on average higher than the number of synergies for the manipulation and individual joint control tasks (6.5 and 8.7, respectively, using the authors' method of counting. If one counts synergies from their table neglecting the 15 joint data, the individual task synergy count more closely matches the current study). This finding could represent a difference between manipulation tasks and matching (including grasping) tasks (i.e. that the difference is that our matching task is higher dimensional than the manipulation tasks). This hypothesis is reasonable considering that the difference is small between the number of synergies measured during individuated joint manipulation tasks and the posture matching tasks of the current experiment. Generally many of the postures in this experiment require recruitment and precise positioning of nearly all joints, as opposed to a manipulation task where the goal is to achieve the task, and the motor system can determine a more optimal solution to manipulating an object than using all joints. If the motor system could achieve the task just as well with only a few joints, that was acceptable, whereas here the goal is to match each posture as closely as possible with all joints.

The fact that the number of synergies for the motor noise is so high is surprising, since it is approximately double what has been previously found. The fact that the motor noise has a similar number of synergies across conditions is expected if we really were measuring motor noise, which is likely to be coupled to the system, and only weakly task-dependent. The system does not change between conditions, so the motor noise should be similar if that assumption holds. Though the number of synergies did not change, the Signal to Noise Ratio did drop in the random conditions as opposed to the meaningful conditions, and there was a slight drop in no-vision, random condition SNR. This supports the suggestion that under more difficult tasks, integrating all sensory information becomes more important than in simpler tasks. It is also possible that the meaningful postures show some practice effects from outside the experiment. But that would not account for the difference in vision vs. no-vision conditions for the random conditions unless in the case of the meaningful condition, the subject did not use visual feedback. This is not likely since subject accuracy did reduce slightly for no-vision conditions (though not statistically significantly), and canonical correlations were reduced for vision vs. no-vision conditions slightly as well, regardless of whether the condition was meaningful or random. The dominant factor across all the results is that meaningful or random conditions determined change, and that for the number of synergies, there was no significant difference in all conditions, only the characteristics of those synergies.

The fact that the subjects were able to match the postures to within twenty percent shows that a human can match hand postures reasonably well by looking at another human, and when one considers the raw residual angle averaged over subjects, trials, and joints, human subjects were able to match the overall posture space to within the precision of the Cyberglove system (roughly 0.5 degrees). However, the variability in the subject's accuracy was non-zero, roughly 20% of the joint ranges. Subjects performing significantly less well on the random conditions may simply mean that the random conditions were slightly more challenging, and that human precision drops overall slightly as the task becomes more challenging. What 'challenging' means exactly is complicated to address, but for current purposes we consider challenge to refer to a multidimensional surface where one dimension is independent dimensionality of posture and the other is novelty of posture (how often in life one makes the hand posture). The meaningful postures represent postures which subjects are likely to have practiced in their lives. It should be noted that removing time as a variable should contribute to more equivalent processing of the postures. Under time pressure, posture performance for meaningful postures may improve over random postures since a refined internal model may already exist in the human brain, and less overall processing would be necessary, and total computational time, stimulus to final posture, would be less. This would be an interesting topic to pursue further in the future.

Additionally, though there was a significant difference statistically, the meaningful and random posture residuals were within the same order of magnitude, which suggests fine differences in the information processing rather than massively different pathways. A sensorimotor bottleneck (See Figure 5.1) where the dimensionality of the synergies is less than the full dimensionality of the target posture space would explain the inverse relationship between challenge and accuracy. The higher dimensional the postures, the more information is lost, after a certain number of dimensions.

The motor system has no reason to encode information differently for different fingers, except perhaps the thumb which serves a slightly different purpose than the other fingers. Thus, considering Figure 5.17, the normalized residuals are similar for each finger, supporting this hypothesis. The abduction has the smallest absolute range of all the joints recorded, which may explain the higher % error.

The lack of significant difference between vision and no-vision conditions is worth considering carefully. In every single angle the data was analyzed from, the vision-no-vision difference was not significant. One would expect that visual feedback would influence human ability to match a posture. Indeed, each subject reported during a post-experimental interview that they felt they had performed significantly better in vision versus no-vision conditions. Some reported that they had done very poorly in the no-vision trials, when clearly each subject's performance was comparable in each instance. This suggests that proprioceptive inputs are dominant over visual inputs in the sensorimotor feedback loop, at least for this type of task.

This would fit well with models of Parkinson's disease (PD), where subjects lose proprioceptive inputs to their motor system and thus must rely on their visual feedback for guiding their sensorimotor system to perform tasks and attempt to coordinate muscle activity [1]. In that case, the visual channel, which has not evolved as a sole input to the motor system, is insufficient for equivalent performance of the variety of tasks the motor system is capable of. Visible tremors (in vision conditions), or in the no-vision conditions for some patients, complete incapability of performing a control task, demonstrate the fundamental quality of proprioceptive inputs to perform most motor tasks. In addition, it is logical that the motor system would evolve to depend more heavily on proprioceptive inputs, since those function in light, darkness, under water, and in other circumstances which may inhibit visual inputs, and proprioceptive inputs would be necessary to coordinate muscle activation patterns [85]. It should be noted that PD has complex effects on the motor system and motor coordination, and the reader is referred to the wide body of literature on PD for further information.

Another possibility to consider is that the subjects somehow ignored their ability to visually compare and correct postures during vision conditions, and so vision and no-vision conditions would be comparable in performance because essentially the subjects were performing the same task as in no-vision conditions. The subjects were given as much time as they desired to practice each posture, and thus they might have used the visual feedback during practice, but not during the actual trial repetitions. Thus their motor performance could be similar to no-vision conditions potentially.

The concept that subjects treated all trials as no-vision trials is unlikely since the vision versus no-vision differences were consistent with all analysis methods - not significantly different, but residuals slightly worse in no-vision (less accuracy), correlation slightly lower, trajectory maximum deviation slightly smaller, number of synergies slightly lower, duration slightly shorter, and Signal to Noise Ratio slightly lower. If subjects were performing all trial conditions the same, there would be

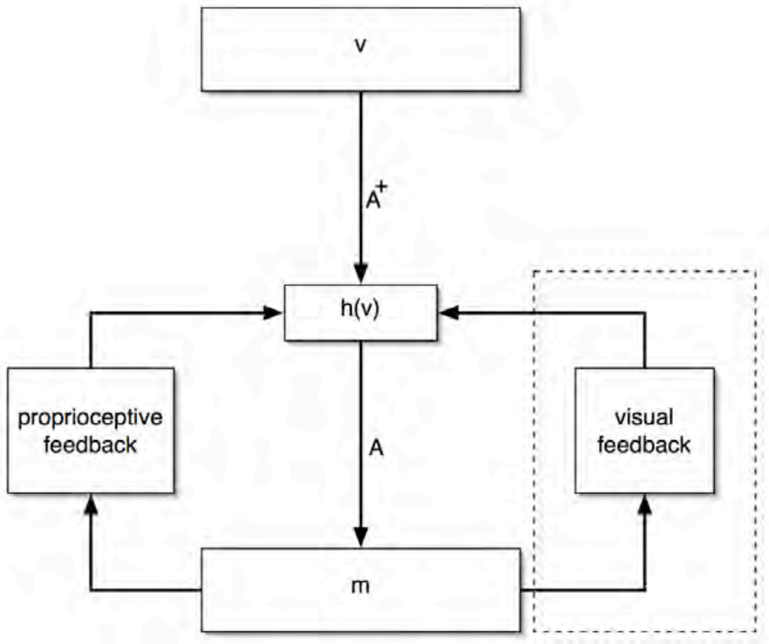


Figure 5.19: FA feedback model, depicting contributions from visual channel and proprioceptive channels, and possible explanation for lack of vision/no-vision differences in subject performance.

less consistency in the vision-no-vision comparisons. In addition, subjects reported attempting to use correction and comparison during vision conditions.

Considering the first few canonical correlation components (Figure 5.10), which have near unity correlation, we see, when using the posture deviation visualizations (Figure 5.11(a)), which joints and fingers were most related between the target and subject data. The primary relations in the first two components are in the first and second fingers and thumb, which are often viewed in the literature [83] as primary digits for many types of tasks (especially grasping). There is also a 'peak pattern' of high correlation, with a peak moving from the middle (first component) to the opposite ends (second component).

The FA posture deviations (Figure 5.11(b)) and factor coefficient plots (Figure 5.14) show a somewhat similar pattern, which may be significant. Our FA computes a set of hidden states, and the transformation to from input, and from hidden to output, which are common to the pooled template and subject data. One interpretation of this is that the subjects do succeed in the matching task. The CCA, which computes a vision and motor side matrix of coefficients (but the two are correlated with each other, rather than pseudo-inverses of each other) shows that a linear transformation can be determined which brings about a correlation very near one - that the subject joint angles can be mapped onto the template joint angles very closely. Additionally, the similarity between vision and no-vision, meaningful and random conditions suggest that the visuomotor channel that this task is being processed through is the same, they are merely performing less well in the random condition than meaningful. The results of our FA model show that a common set of factors related to both the template and subject data are related by a very similar transformation. Thus with this model, subjects transform into a synergistically-based internal coordinate system which is used as a reference for final postures.

The fact that the current study has determined a higher number of synergies than previously found yields further support to the minimum intervention principle. In this task, a small number of synergies may not have been sufficient to achieve the task, and thus the motor system modeled as an optimal control system, automatically employs a higher dimensional synergistic control than for a task which does not explicitly require independent control of maximal degrees of freedom.

The unexpected lack of (significant) vision vs. no-vision differences can be reconciled by considering the strength of proprioceptive feedback as dominant in the sensorimotor system. Additionally, the proposed factor analysis model suggests that addition of noise to the mapping process to hidden body-centered synergy states and noise added to the motor transformations which result in output commands may account for the nonzero residuals (and Standard Deviation of residuals) in the motor system outputs compared to the target posture.

Chapter 6

Robotic designs for studying and simulating human sensorimotor behavior

6.1 Introduction

6.1.1 Contents of the chapter

Due to the length of this chapter, it is broken up into multiple sections. The first section presents a short description of haptic issues and approaches to solving these problems. A suggestion for a watchdog methodology is presented.

The second section of the chapter is devoted to the design methodologies developed for producing robotic designs which are useful for studying sensorimotor learning. These methodologies are presented in the context of a specific robot design. The design sections are the motivation, design inspirations, a risk reduction design, requirements, friction mitigation methods, strength/mass tradeoffs, design for manufacturability/assembly, cable drive transmission, and final leg design.

The third section, modeling and results, consist of dynamic modeling of the bldc motor, block commutation method and experiments, sinusoidal commutation method and theory, Hall Sensor-based position estimation, implementation in an embedded system, kinematic modeling and decentralized PID control method, and finally results for the robot.

6.1.2 Scope of the chapter

The development and required theory for creating the modular robot leg design are presented, along with preliminary results validating the robot leg's capabilities.

6.2 Design for safe human-robot interaction and effective haptics

6.2.1 Design considerations for human-robot mechanisms, from mechetronics to safety

Designing a haptic virtual reality (HVR) system to interact with and be coupled to humans is challenging in several respects. The first and foremost challenge is finding the balance between safety and bandwidth/perturbation strength of the human-robot closed loop system. Several different approaches are often employed, but each has certain limitations. Three of the most typical approaches are discussed along with their advantages and disadvantages.

The first and most obvious approach is to create a device which is small and weak, with minimal risk because maximum force and velocity is below risk level for average human beings. The disadvantage to this approach is that this limits the application of such HVR systems. A system designed for robot-assisted neurorehabilitation would have little assistive capability.

A second approach which has safety advantages is to use completely passive systems which use the human being to generate all forces, and the perturbations are performed with a series of sensitive braking systems. This has the disadvantage of only being able to passively apply forces. Thus if one wished to use the robot to assess physical capabilities of a brain damaged individual who could perform minimal controlled movements, this system would have defficiencies.

A third approach is to use a robotic system which physically dissassembles under a load which is too large. The Delta Haptic robot [18] uses such a mechanical design, and works fairly well, with a medium force capability. However one can still potentially generate dangerous velocities in a subject.

Thus some other mechanism, mechanical, electronic, or some combination thereof, is necessary. This 'watchdog' methodology is presented here as a means of adding a factor of safety to any HVR system. This allows designing a system with higher bandwidth and force production capabilites without the inherent dangers of exceeding limitations of the individual subject for which an experiment is being performed (in terms of range of motion, maximum velocity/acceleration, forces, and any other parameters necessary to keep within tolerances).

6.3 Modular robot for locomotion, hand manipulation, and virtual reality interaction

Each leg/finger of the robot possesses three-degrees of freedom. The limbs are actuated by brushless dc flat pancake motors which are closely mounted near

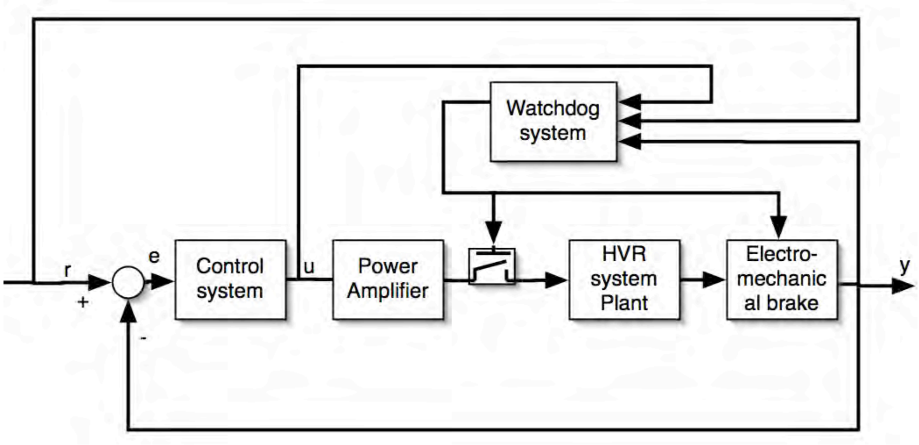


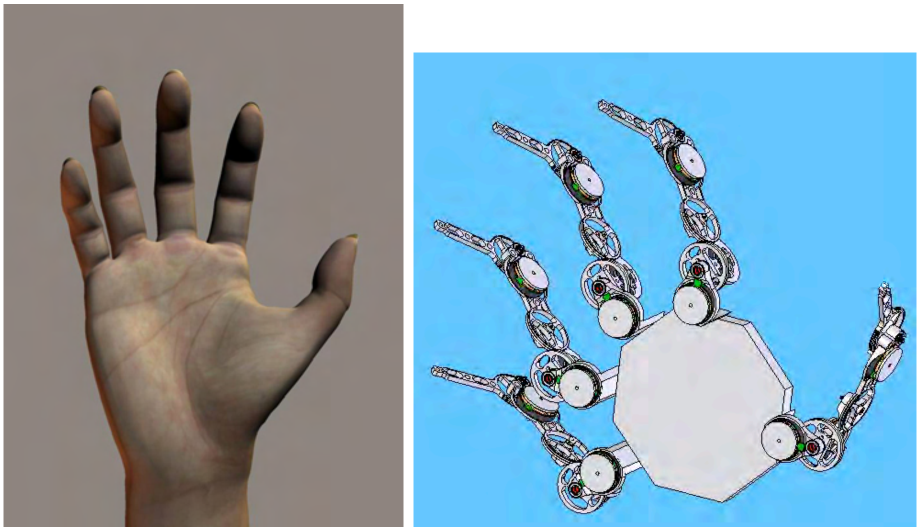
Figure 6.1: A safety method for human-robot interaction. A watchdog timer 'oversees' all important variables, and has the ability to cut power to the actuators and safely stop the system if values fall outside acceptable parameters.

each joint, and connected through no-backlash, low friction cable drives. The cable drives use ultra-low-stretch wire with 7x49 strand stainless steel coated cables.

6.3.1 Motivation for design decisions

The design of the robot leg is driven by the desire for a truly bio-mimetic backdrivable mechanical system. Very few, if any robots currently in existence are truly backdrivable. As is shown in Figure 6.4, non-backdrivability seriously alters the way a system by nature interacts with its environment. Most haptic robots, such as the Delta Haptic robot [18] and the Phantom [96] have friction, torque ripple, inertia, a static base or other problems which limit their applicability to bio-mimetic experimentation (or for bio-mimetic locomotion)[1]. This can be mitigated

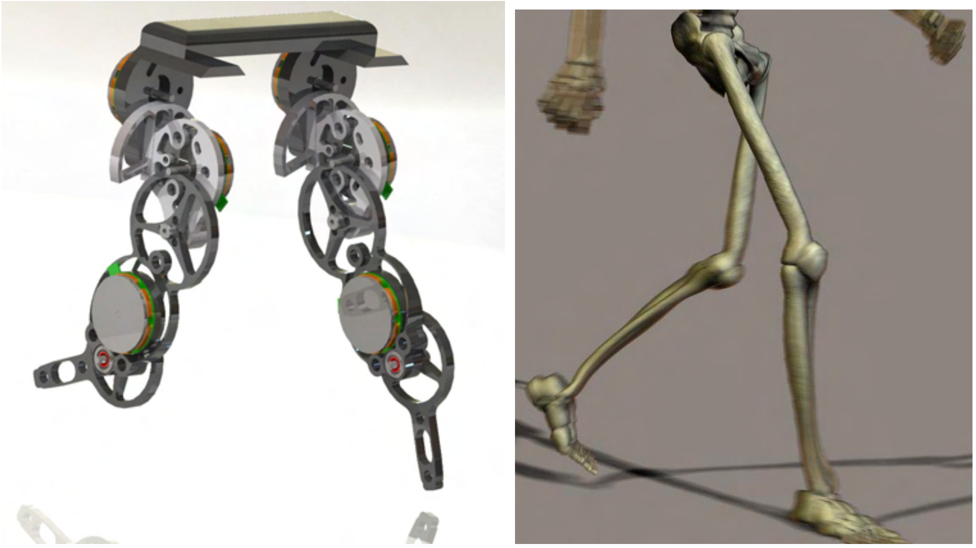
[1] It should also be mentioned that these are not mobile robots, being designed in such a way as the majority of the hardware is external to the mechanism and far too heavy for it to move



(a)

(b)

Figure 6.2: Depiction of (a) a human hand, and (b) a configuration of five ModBots arranged as fingers for an artificial hand.



(a)

(b)

Figure 6.3: (a) Bipedal implementation of the ModBot during a typical walk locomotion cycle (simulated, 'feet' not included in image). (b) Human skeleton during a typical walk locomotion cycle.

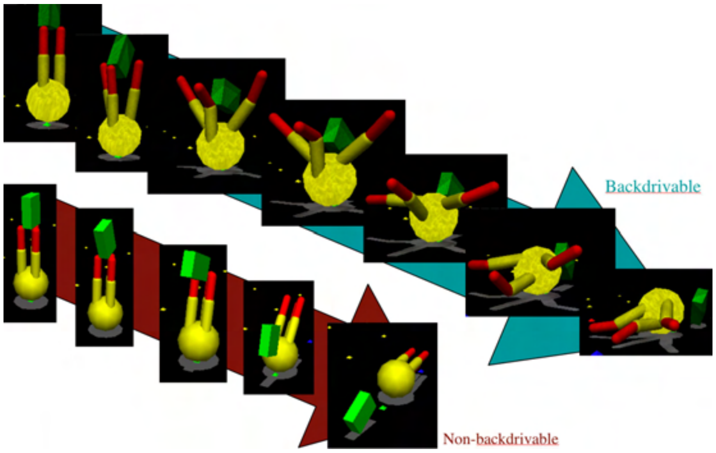


Figure 6.4: This depicts a block colliding with a multi-jointed system. One system is backdrivable, the other is not. The two systems (backdrivable and non-backdrivable) interact with the environment in fundamentally different and complex ways. It is difficult to force the non-backdrivable system to behave in the same way as the backdrivable system.

with control of endpoint forces and friction/intertial compensation schemes, but there are inherent limitations due to actuator saturations, bandwidth limitations, power consumption, etc. In addition, one would have to instrument every surface with a force-sensitive device in order to make the system be backdrivable from any point of contact with the outside environment. That may not be either feasible or possible given current technology, though nanomaterials are a promising technology - the author is in contact with researchers developing paintable nano-surfaces which act as a force-sensitive resistor, but presently such technologies are in their infancy. The aim here is to make the passive dynamics of this robot meet specifications.

6.3.2 Design components

The main design components and issues to be dealt with in this next section are:

- Mechanical

  Number of degrees of freedom/design inspiration

  Mass/length/strength requirements

  Range of motion (workspace)

  Dynamics/speed

  Onboard or offboard motors

  Drive system

  Friction minimization

  Modularity

  Joint type(s)

Materials

DFM/DFA (design for manufacturability/assembly

Service life

Rigidity

End effector design

Mechanical and electrical coupling to a base

- Electrical

  Onboard or offboard electronics

  Sensors - measuring position, velocity, acceleration, force

  Wireless link with high level control system (hierarchical control design)

- Control

  Control methodology

  Design for functionality (manipulation and locomotion)

A major issue which molds the outcome of this robot design is modularity. This is a single robot leg which is designed to mechanically and electrically couple to an arbitrary base to form a new machine. In this way, the 'leg' can serve the purpose of a finger or a leg for manipulation experiments or locomotion experiments. In some cases, both regimes may be simultaneously explored. However, significant challenges appear when a designer attempts to include actuators, electronics, and drive systems which fulfill the purposes of back-drivability and other bio-mimetic characteristics while retaining a minimal mass in order to carry all components with the system in a modular way.

6.3.3 Design inspiration

The robot design was inspired by several components present in nature. One can observe that the human finger has three degrees of freedom - 3R. A human leg (not including the ankle and foot) has four degrees of freedom - a ball joint at the hip gives three of them. However, the third rotation at the hip would not be necessary for locomotion, and would be an additional degree of freedom to control during manipulation experiments. In this version of the robot leg the author decided to use a three degree of freedom leg. A future robot will include more degrees of freedom. Additionally, the modularity of the robot design and drive mechanism make for addition of more degrees of freedom during assembly fairly trivial. The size of each finger is slightly larger than that of a large adult male finger (so that electronics and motors may be included on-board).

6.3.4 Risk reduction

A three-legged non-modular robot (dubbed *TRPS* - Tripedal Robot Prototype System) was designed and built out of acrylic material and small self-contained servos in order to test the overall joint design and number of degrees of freedom. The electronics consisted of a basic stamp microprocessor coupled to three custom-designed and built (by the author) motor drivers to drive all the dc motors and provide pwm signals to the servos. The basic stamp received reference commands from a National Instruments NI USB-6008 data acquisition card and both labview and matlab software configurations, depending on the experiment. This design, shown in Figure 6.5, was capable of supporting its own weight and posessed some back-drivability in most of the servos. However, the base servos were not backdrivable and thus were prone to damage. The main results of TRPS can be summed up in the following way:

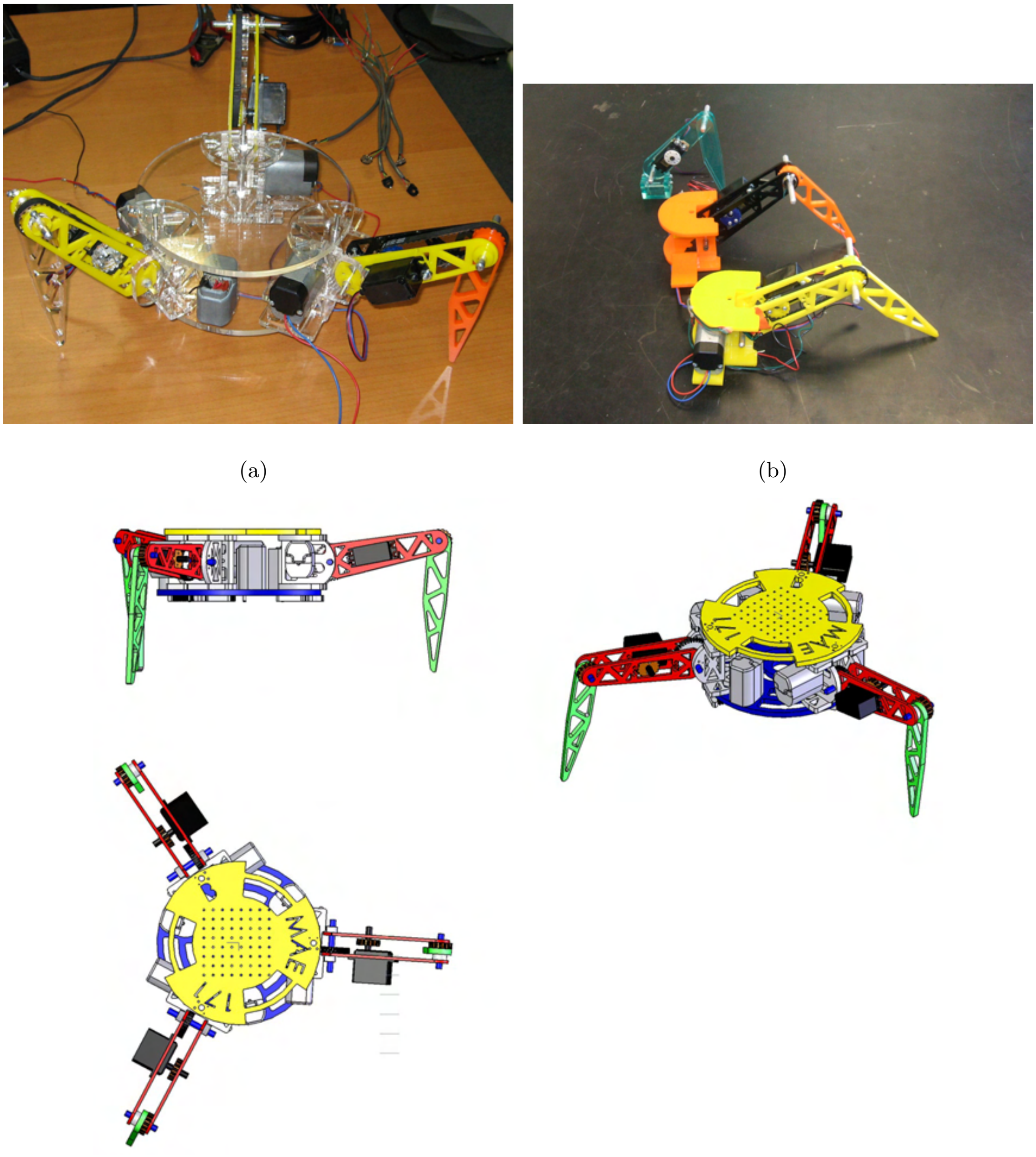
- Backdrivability is important for creating an interactive robot capable of not only acting in an environment, but using information gained by those actions in order to improve a control policy.

- Backdrivability created by compliant control poses limitations due to the actuators in terms of bandwidth, force, stability, and smoothness. Additionally, compliance can only be created by external measurements, and thus prohibitive numbers of sensors may be need to be integrated into the system, which requires tremendous bandwidth or compression in order to convey the information for feedback. Additionally, if this is a high resolution signal over a surface, the signal becomes high dimensional, and creating feedback control loops around this is computationally expensive and complicated, though not impossible to implement.

- Gears are not an appropriate transmission system for a backdrivable robot. Transmissions based on gears suffer from backlash, friction, and other issues which limit the transparency of the gear reduction system. Ideally a direct drive would be implemented, eliminating these issues, but other transmissions must be considered to efficiently transmit loads and forces bidirectionally (output-input, input-output). The robot destroyed its gears in a matter of minutes no matter how carefully the external forces were applied to the system.

6.3.5 Length/mass/force requirements

The approximate size of the robot leg was decided to be determined by the length of an average human finger. Initially the leg design was to be much



(a)

(b)

(c)

Figure 6.5: (a)Photograph of the prototype TRPS, with electronics removed for clarity. (b)Three versions of the leg from the initial to the final revision. (c) Top, front, and isometric views of TRPS. Here one can see that the robot can be used for locomotion or basic manipulation tasks.

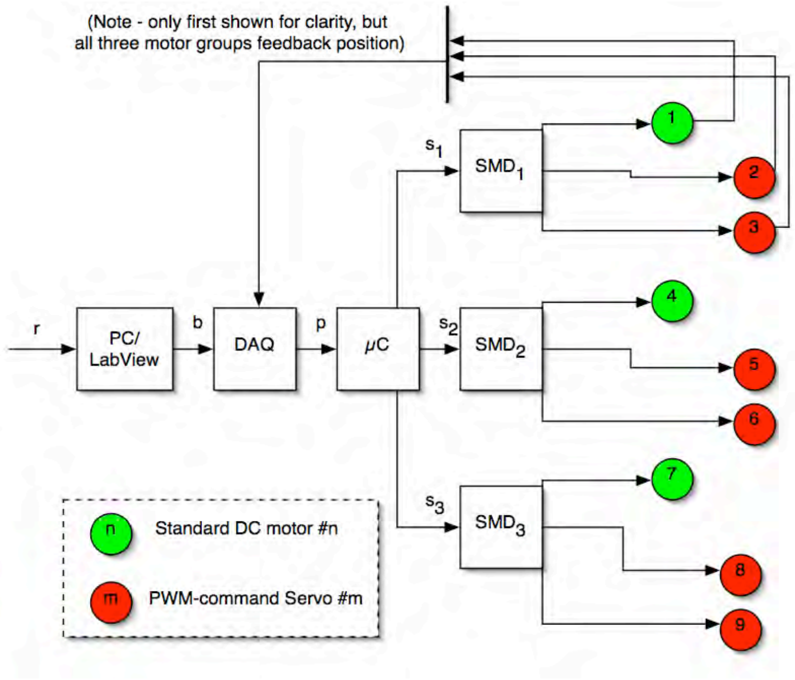


Figure 6.6: Block diagram for Tripedal Robot Prototype System (TRPS). The DAQ was a NI USB-6008 connected to provide analog outputs to a microprocessor which takes in the analog value and converts it to a motor torque command, which is then passed serially to the serial motor drivers (SMD). The motor drivers interface directly with DC motors and provide PWM pulses to two servos per driver. Position is measured with high accuracy low hysteresis potentiometers.

larger, but strength-inertia-weight ratios of brushless motors drove the design to an optimal size. In addition, some of the desired experiments involved human hand-sized manipulators.

For manipulation, only minimal forces in addition to balancing the mass of the leg (minus the portions constrained at the base - which comprise the bulk of the mass of the leg) are required, though the additional force means faster dynamic response. For locomotion, each leg must be able to support its entire weight plus as much as possible given the constraints to be elaborated upon in the next sections.

Therefore a maximal torque-weight ratio is required for this design. Figure 6.7 shows the evolution of the mass of each main component of the ModBot.

The motors were selected based on the constraints imposed by the mass of the robot leg and the fact that this robot is to be used as a manipulator (hand, etc) *and* as a leg with onboard actuators. The selection of motors is vast, but can be quickly narrowed down by comparing specifications in terms of requirements. Here we require minimal mass, high torque-weight ratio, and minimal inertia if the robot is to be used for manipulation, and if it is to be back-drivable. Thus a useful set of plots for making the decision is given in Figure 6.8.

6.3.6 Range of motion

The range of motion of each joint was determined to be symmetrical (constraints could be imposed in software or by adding removable mechanical stops), with a range of usable space of approximately 180 degrees. The cable drive allows a slightly larger range but the value determined is a minimum for each range. For locomotion applications this allows not only standard bio-mimetic control approaches, but also symmetric locomotion (i.e. the robot falls down, then its joints rotate such

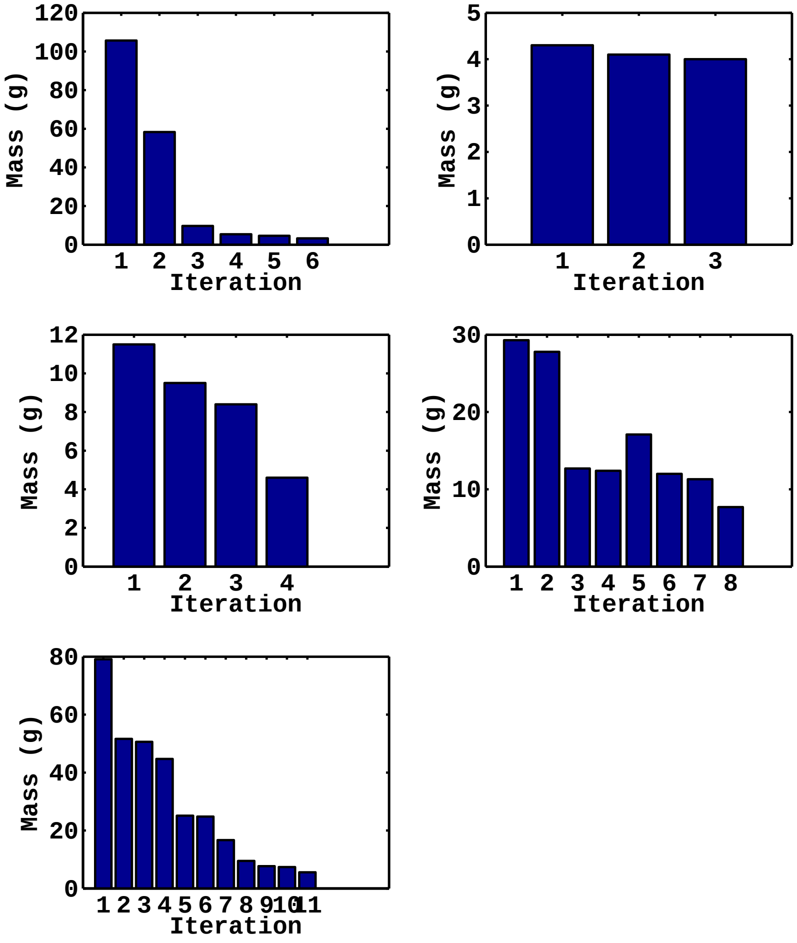


Figure 6.7: Bar plots showing the reduction in mass for each iteration of each component. Some components were developed in later versions of the robot, thus they had less reduction in mass (they were designed at a more evolved state of the robot).

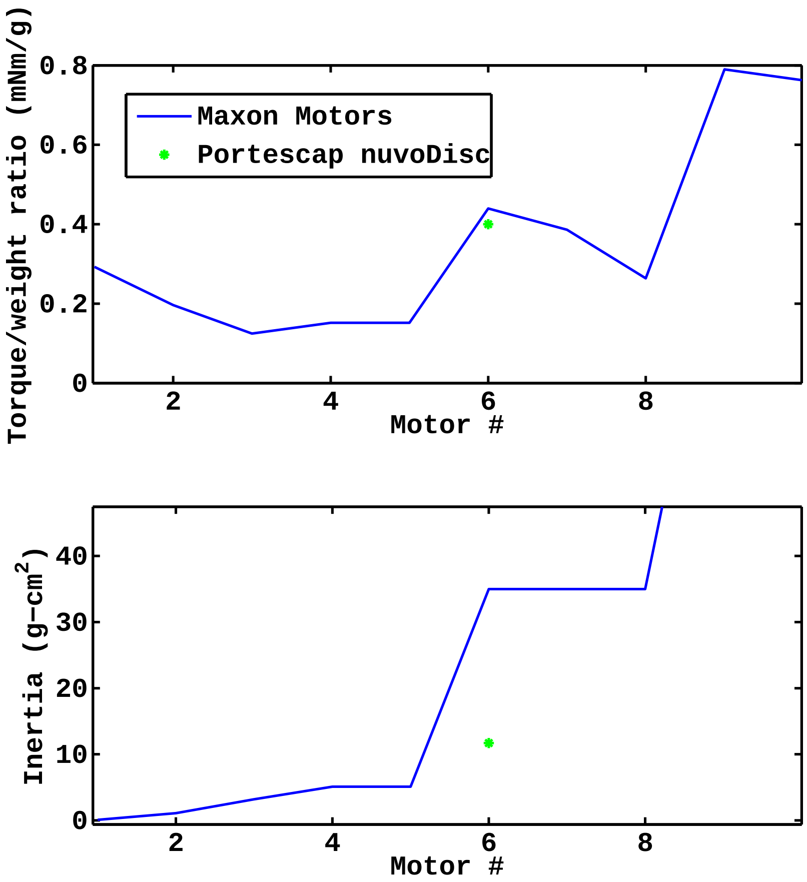
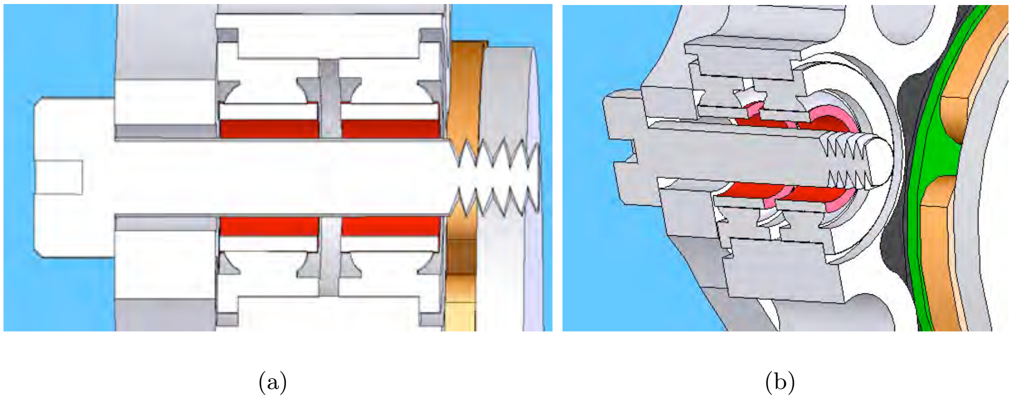


Figure 6.8: The motors were selected based on many properties, but the most key properties are mass, rotor inertia, and torque. Here it is clear that in terms of both torque-weight ratio and rotor inertia, the Portescap nuvoDisc brushless DC flat pancake motor is superior to the Maxon flat brushless DC motors. It has almost the same torque-weight ratio, but four times less inertia than the comparable Maxon motor.



(a) (b)

Figure 6.9: Section view of double ball bearing joint with nut removed for clarity. The use of precision high speed ABEC7 bearings, and all components pressed into place creates a very stiff (each bearing is rated to 250lbs. radial load) joint with an extremely low friction coefficient.

that the inverted orientation of the robot is then considered upright).

6.3.7 Friction

Minimizing friction is clearly a must for each joint of this design. Friction causes nonlinearities, unexpected behavior, and biological systems tend to have very minimal friction. Thus, since we are mimicking biology, it is obvious that we should minimize friction. In order to achieve low friction, without sacrificing stiffness or stability, a double-bearing/shaft with press fit design was created (Fig. 6.9). ABEC-7 rated ball bearings were used for high precision, low friction, high stiffness, and high potential bandwidth.

In Subsection 6.3.8 we discuss situations where friction is not minimal and some mitigation must be performed actively (via estimation and control). This essentially can be done in many ways. The one pursued here is one of model reference control, where the system has various sub-loops within the control structure which are designed to push the dc motor plant towards ideal behavior, leaving the higher level control to act on an ideal servo system. We present in this section data and results demonstrating reduction of undesirable friction effects in motors with very strong friction properties. The added loops require minimal computation time and thus are realizable in online embedded systems.

6.3.8 Friction compensation of a brushless y-configuration motor using hierarchical model reference control design methods

Overview

In this subsection we discuss situations where friction is not minimal and some mitigation must be performed actively (via estimation and control). This essentially can be done in many ways. The one pursued here is one of model reference control, where the system has various sub-loops within the control structure which are designed to push the dc motor plant towards ideal behavior, leaving the higher level control to act on an ideal servo system. We present in this section data and results demonstrating reduction of undesirable friction effects in motors with very strong friction properties. The added loops require minimal computation time and thus are realizable in online embedded systems[65].

Introduction

In an ideal world we would live without friction when we don't desire it (friction happens to be necessary for life - just consider walking on ice - now imagine if everything was impossible to grip!). However it often comes to be that one must

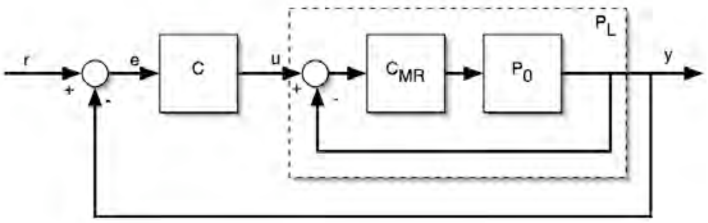


Figure 6.10: Model reference control design.

design and use systems which have non-ideal quantities of friction. Friction as a phenomenon comes in several forms - Static, Dynamic, Stribeck, Coulomb, Viscous, etc. See [73][4] for a good review of friction models.

One approach to mitigating effects of friction in an actively controlled system is to model the friction in the system and then use that model to push the system towards more ideal behavior (Figure 6.10). Then a higher level controller can send commands to the servos without having to compensate for friction effects. This allows a linear design for the higher level control system, with the lower level system being a nonlinear, but stable feedback loop. This is referred to as a model reference control design, combined with hierarchical feedback control structure. There are many variations upon this structure, such as placing the feedback control in parallel with the friction estimator.

Friction theory

Friction is a force which acts to oppose the relative motion of two surfaces in contact (or surfaces in contact with a fluid).

There are several types of friction which exist in different dynamic regimes. The two main friction splits are static and kinetic friction. Within the kinetic friction

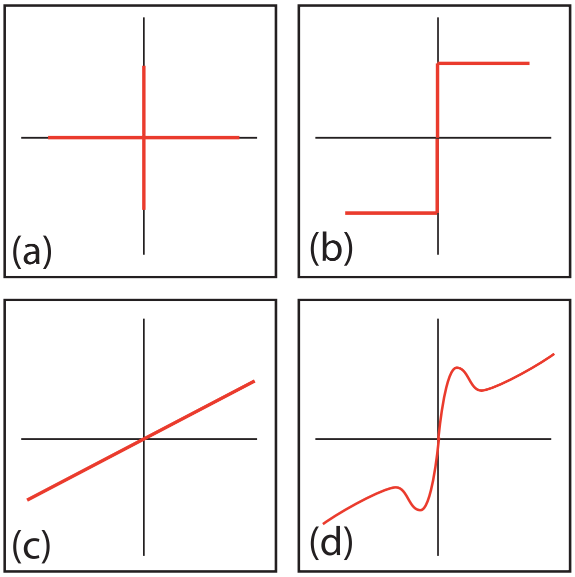


Figure 6.11: Models of friction. a, b, and c are, respectively, Stiction, Coulomb, and Viscous. d takes into account that friction is not discontinuous, and so captures the velocity-dependent friction components. This is referred to as Stribeck friction.

regime we will consider Coulomb, Viscous, and Stribeck models.

Stiction

Static friction consists of, at zero velocity, a resistance to motion which builds to a peak, then the object being acted upon begins to move and enters kinetic friction regimes. The force at which motion begins is called the breakaway force.

Coulomb friction

In Coulomb friction models, the friction force equation is very similar to that of the static friction equation, with the relation being

$$F = F_C \mathrm{sgn}(v), \tag{6.1}$$

$$F_C = \mu_C N, \tag{6.2}$$

where F_C is the Coulomb friction force, μ_C is the Coulomb friction coefficient, and N is the force normal to the surface upon which the object is sliding.

Viscous friction

Viscous friction theory stems from 19th century work in hydrodynamics theory, where friction is due to the viscosity of lubricants. A typical basic relation used to describe Viscous friction is

$$F = F_v v. \tag{6.3}$$

Or a model which fits better to experimental data can be obtained by adding some nonlinearity to the velocity dependence,

$$F = F_v |v|^{\delta_v} \mathrm{sgn}(v). \tag{6.4}$$

Where δ_v is a parameter which is dependent on the geometry of the system of interest.

Combined friction model with Stribeck friction

Stribeck observed that overall friction decrease with velocity is not a discontinuous function, but follows some nonlinear curve. We will create a combined friction model as in [73].

$$F = \begin{cases} F(v) & \text{if } v \neq 0 \\ F_e & \text{if } v = 0 \text{ and } |F_e| < F_S \\ F_S \text{sgn}(F_e) & \text{otherwise} \end{cases} \tag{6.5}$$

Many parameterizations of $F(v)$ are possible, one common one is

$$F(v) = F_C + (F_S - F_C)e^{-|v/v_S|^{\delta_S}} + F_v v \tag{6.6}$$

Where v_S is called the Stribeck velocity.

Application to a motor

The model parameters above can be found for a particular motor using system identification procedures. First a basic motor model is presented, and parameters determined from frequency response experiments, then a friction model implemented and parameters determined such that the model simulates the motor behavior sufficiently well.

Finally, a PD controller plus friction compensation scheme is developed and the model reference control design is prototyped in Matlab Simulink with Realtime Workshop, and results demonstrate mitigation of negative effects of friction. The nominal motor model is controlled by a tuned PD controller as well, and performance of each system is compared.

Derivation

A straightforward derivation of the equations of motion for a motor leads to (6.13), which can then be used as the nominal model of the motor, with the friction

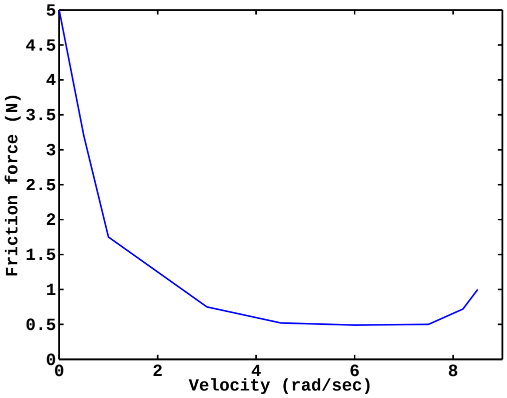


Figure 6.12: Friction of the motor characterized over frequency. The low frequency components are attributed to the Stribeck friction region, middle to Coulomb friction, and the high frequencies to the Viscous friction regimes. This plot, which represents an average over many trials, can be used to estimate parameters for the friction model.

estimation model running in parallel with the control, injecting friction compensation values into the basic control output.

Results

The motor model without friction compensation achieves, for a series of sine wave tests, an average absolute error of no less than 0.05 Rad, whereas with friction compensation in place, the error drops to 0.02 Rad, and the nonlinearities due to friction at low velocities are well approximated (these errors do not measure effectively with average error but are significant in the local regions, as they also cause phase differences between the model and actual motor).

The friction compensated closed loop system performs the best of the sys-

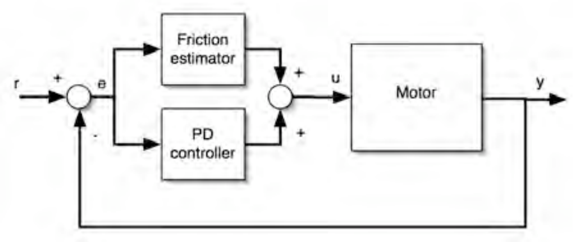


Figure 6.13: Compensation scheme block diagram.

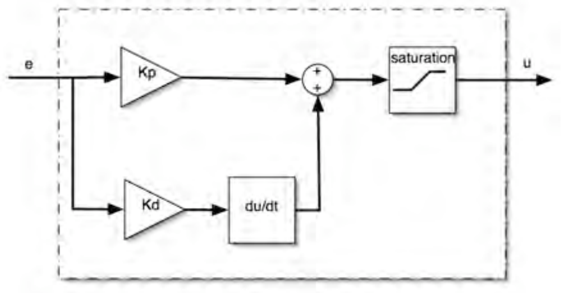


Figure 6.14: PD controller block diagram.

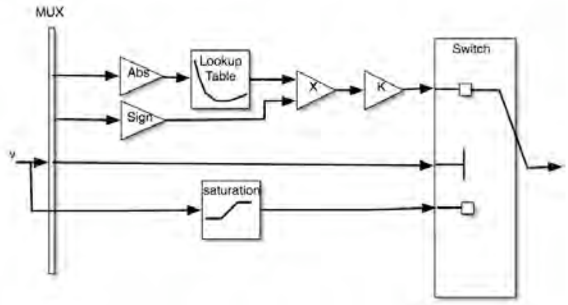


Figure 6.15: Friction estimator block diagram.

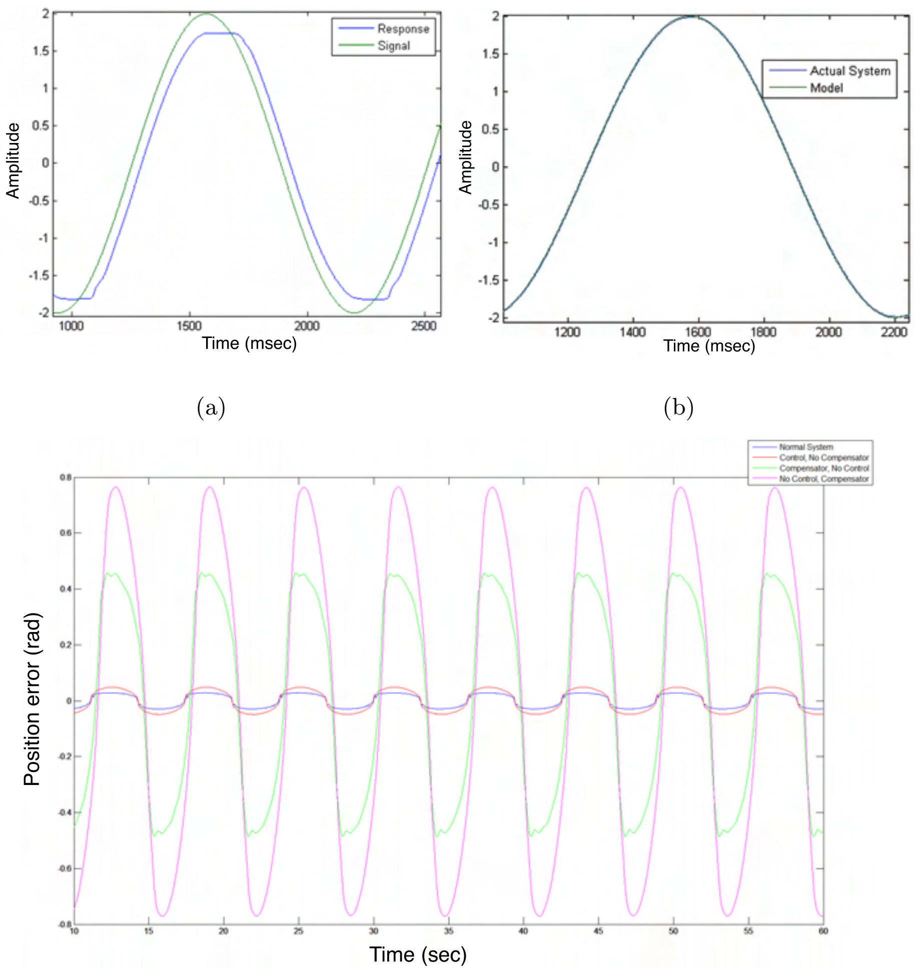
tems compared, with the smallest tracking error, as shown in (6.16(c)). Generally this scheme is quite effective, and very simple to implement in actual systems. When one is faced with significant friction to overcome, this technique of modeling the friction or nonlinearity can be a useful member of the 'grab-bag' of strategies to consider.

6.3.9 Strength/mass

The load capability of each joint is characterized by static and dynamic forces. Each joint was designed to withstand 50N of static load in all major axes, while removing as much mass as possible. Similar to the human skeletal structure, mass was removed in most of the inner spaces, leaving a thin truss where required. This resulted in a weight reduction of approximately an order of magnitude.

Various candidate materials were considered during the design phase. These ranged from many forms of plastic, resin, aluminum, steel, wood, and others. Given the type of design, and issues of manufacturability (discussed in the next sub-section), 6061 aluminum was ultimately selected.

An alternative way to design the robot would have been to use 3D printing methods such as FDM (fused-deposition modeling), however this method was less precise and resulted in components with lower strength-weight ratios. In addition, time was a factor, since components would take approximately one week from order to arrival. Some components, such as the tip of the finger/leg were fabricated using the FDM process. These components were not critical regarding precision, strength, and smoothness of surfaces (for the cable drive).



(a)

(b)

(c)

Figure 6.16: (a) Friction effects are clearly evident in this image, most notably a lag due to flat spots at low speeds, where friction becomes dominant. (b) Friction has been compensated for in this plot (this is still an open loop system, subject to the same signal as in part a). The difference between the actual system and model are insignificant, and friction has been mitigated significantly. (c) The tracking error (for a sine wave) of various configurations of the system is shown. The configurations are varied from no friction compensation and no control to friction compensation and control ('nominal system'). Note that the performance of the compensated, controlled system is the best.

6.3.10 Design for Manufacturability/Assembly

Manufacturability

All components were designed to be CNC mill or lathe-produced. G-code to control the CNC machines was generated in Mastercam Mill or Lathe v9.1, and finally processed within the proprietary software for each machine used. Early prototypes were produced on a HAAS VF-1 CNC Mill, and with manual Lathes. Later prototypes were produced on a FlashcutCNC Mill 2000 and Lathe 4400. Some components were rapid prototyped on a Lasercamm 2405 120W CO2 laser cutter.

Assembly

As a design becomes more complex, it becomes more and more imperative to consider assembly. In order to make a robot servicable and usable, it is important to balance simple assembly (as simple as possible) with number of parts and likelihood of disassembly needs. Assembling a cable drive is timeconsuming since it requires precise cable adjustments and lengths, and due to minimal size, non-minimal dexterity is required on the part of the assembling technician. Therefore, requirement one is: *each component should be designed so that not all joints must be disassembled if one joint must be.* Requirement two follows: *Electrical connection changes such as motor and sensor cabling should be capable of disconnection without complete disassembly of a leg.*

6.3.11 Cable drive transmission

In order to interact with the world in a way which simulates by its nature biological systems from the perspective of control, a system which can not only act in the world but the world can act and effect changes in the system was needed.

The key issue is *backdrivability.* A system is backdrivable if the system can not only apply energy to the external world by changing its state, but the external world can apply energy the system and alter its state. This definition leaves out the fact that backdrivability is really a matter of degree. Most belt and gear transmissions can be back-driven, but friction and backlash bring about unpleasant nonlinearities which interfere with ideal models. Biological systems tend to be highly backdrivable and as such the internal control systems within a human probably neglect things such as backlash and friction. Thus, if attempting to mimic the characteristics of the structure of biological control systems and intrinsic behavior as a result, it is appropriate to mitigate the characteristics of friction and backlash in the robot design.

The basic components of a low parts count cable drive consist of a pinion shaft, a spur shaft, high flexibility low stretch cable, a means of attaching the cable at both ends, a way of tensioning the cable, and a way of constraining the pinion and spur wheels or surfaces such that they can only rotate, and resist the force due to the required tension in the cable. A spring is not necessary as the cable itself acts as a stiff spring. Spring components in this case at the mounting point for the cable end would essentially decrease the reliability of the drive since forces acting in tension on the spring would tend to act as a point of deflection in the system, causing slip of the cable drive. Eliminating the spring components means finding an ultra-low stretch cable material. In this case a stainless steel cable specifically designed for ultra-low stretch miniature applications was found.

Walking is defined in this case as the process by which a cable wrapped around a shaft, not overlapping, in a pull-pull arrangement with friction, must invariably traverse the axis of the shaft. The cable will, at least, walk by one cable diameter per rotation of the shaft. We state 'at least' because there are other vari-

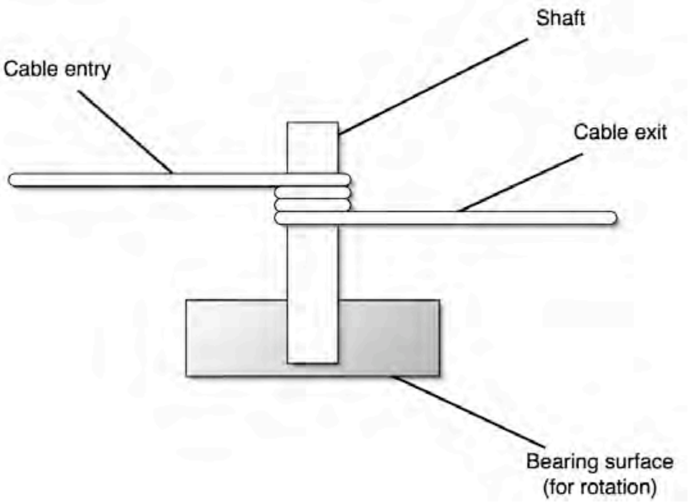


Figure 6.17: Basic cable wrap schematically

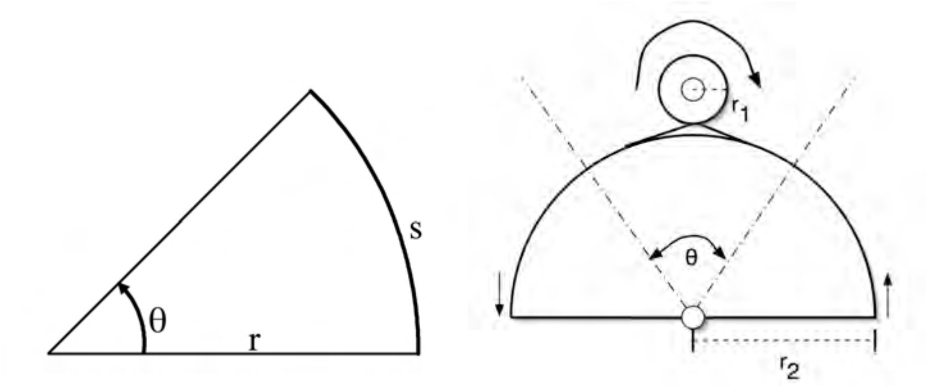
ables which affect the amplitude of walking. First we will demonstrate the simplest case, then we will discuss other factors, their significance, and methods of addressing each.

In the simplest case, let us show the minimum walking quantity. First, begin with a representation of a cable wrapped around a shaft. Assume no slip, and neglect friction of the cable with itself. Also, assume that initially, the cable entry and exit points are normal to the shaft face.

It is standard that a ratio for mechanical advantage is given by

$$\begin{aligned} n2\pi r_1 &= 2\pi r_2 \\ n &= \frac{r_2}{r_1}. \end{aligned} \tag{6.7}$$

In this case we want to travel some distance of the cable by rotating the pinion shaft. In doing so, there will be some minimal transversal walking. The total walking vs. distance traversed is related. To derive that relationship let us recall that from simple



(a)

(b)

Figure 6.18: (a) Arc length, and (b) rotations of the small pinion shaft cause rotation of spur shaft, or linear motion along the cable direction.

geometry the length of an arc is related to the radius of the arc and the angle of sweep

$$s = r\theta \tag{6.8}$$

Then we have two arcs, one that is a complete circle, representing the wrapped shaft, and the other the length of the cable to traverse. The relationship between shaft rotations and length of cable traversed is

$$\begin{aligned} \alpha s_1 &= s_2 \qquad (6.9) \\ \alpha &= \frac{s_2}{s_1} \\ \alpha &= \frac{r_2\theta}{2\pi r_1} \end{aligned}$$

Now if the cable walks one cable diameter per rotation of the pinion shaft, we can relate the axial distance walked by

$$\begin{aligned} \delta x_w &= d_c * \alpha \qquad (6.10) \\ \delta x_w &= d_c * \frac{r_2 \theta}{2\pi r_1} \end{aligned}$$

This relation does hold as an approximate estimate, and proves invaluable for determining the thickness required for the shaft length and thickness of the surface over which the cable must lay.

6.3.12 Computation of cable strength requirements

The cable strength is dependent on several factors, static and dynamic. The dynamic load capability of each cable is dependent on cable thickness, number of strands and strand organization, as well as material. A simple relation was derived and used with a factor of safety of five (due to impulsive forces) in order to determine minimum cable strength. This derivation will not be included here due to space considerations.

6.3.13 Final leg design

The final leg design meets the specifications set forth in the above sections. It is light, easy to fabricate, strong, durable, adaptable to locomotion and manipulation, modular, (and it happens to be aesthetically appealing) etc.

Figures 6.19 and 6.20 show that there were several design revisions from concept to a version 5 prototype, depicted in Figure 6.21 and 6.22.

The final joint design minimizes friction effects (a joint allowed to free-swing will continue to do so for significantly more than one minute before coming to rest).

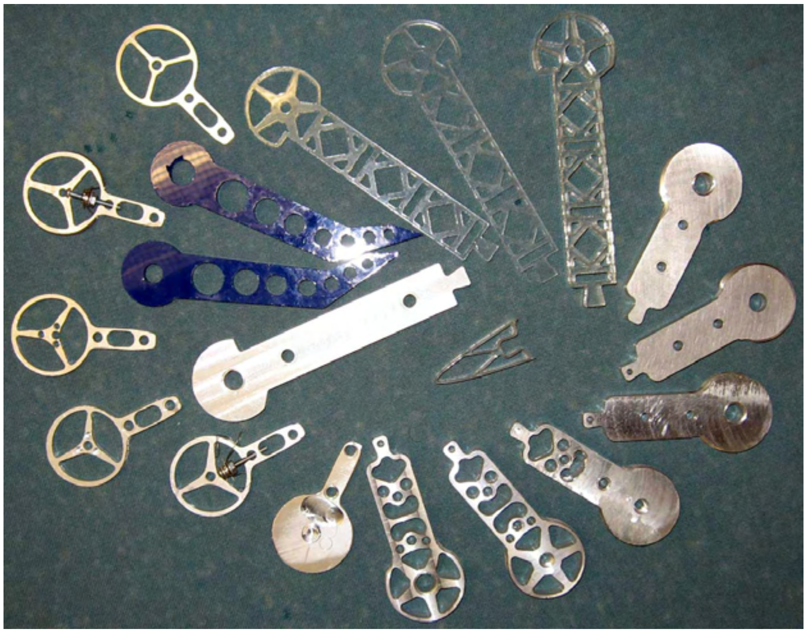


Figure 6.19: An example of one component's evolution from initial prototypes to final manufactured 6061 Aluminum component.

Unfortunately there is some friction introduced by the cable drive system and BLDC motor, when the bearings of the motor are under load. However this friction is still small compared to the forces involved, and the slightest force at the tip of the robot immediately causes motion of the motors.

The joints themselves are very stiff in all directions but the degree of freedom (which is very smooth and passive), with no perceptible play in undesirable directions.

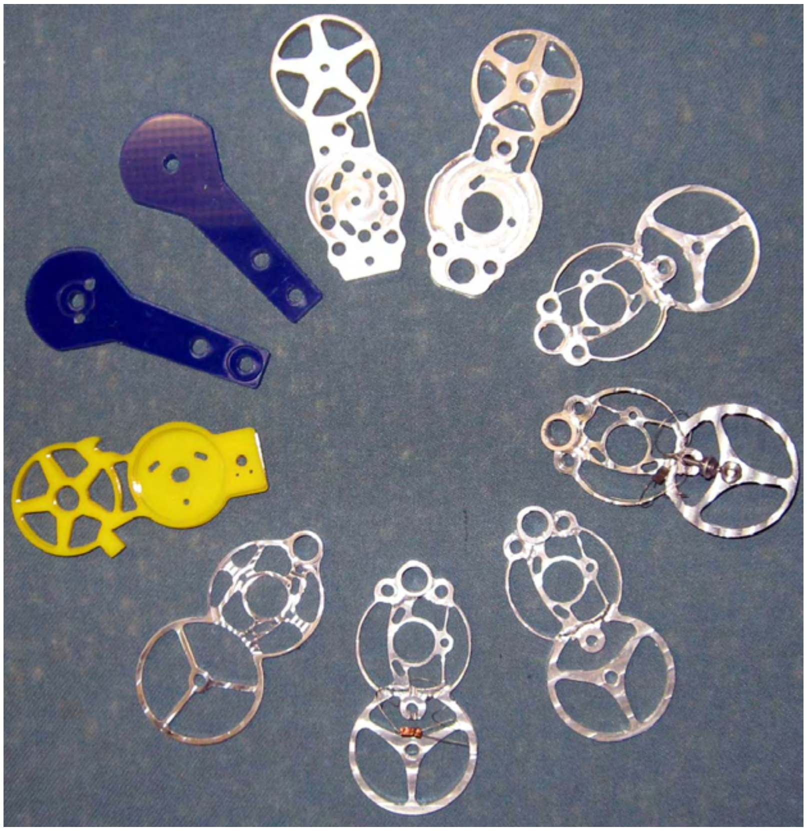


Figure 6.20: A second example of one component's evolution from initial prototypes to final manufactured 6061 Aluminum component.

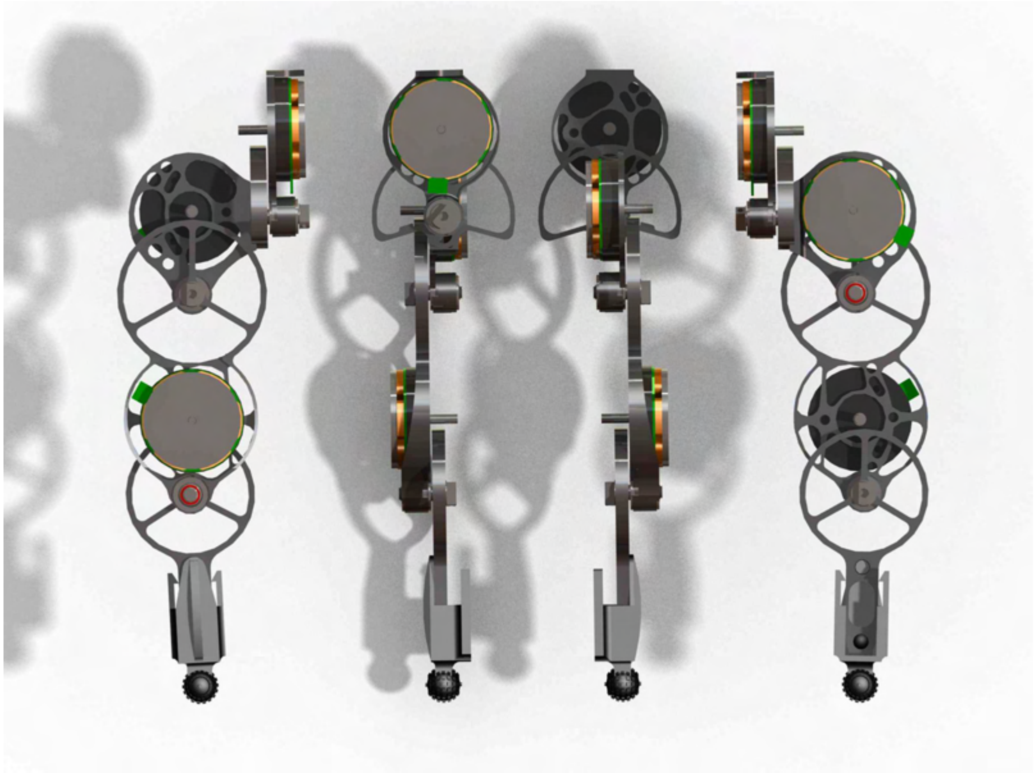


Figure 6.21: Left, top, bottom, and right projections of the modular robot leg design.

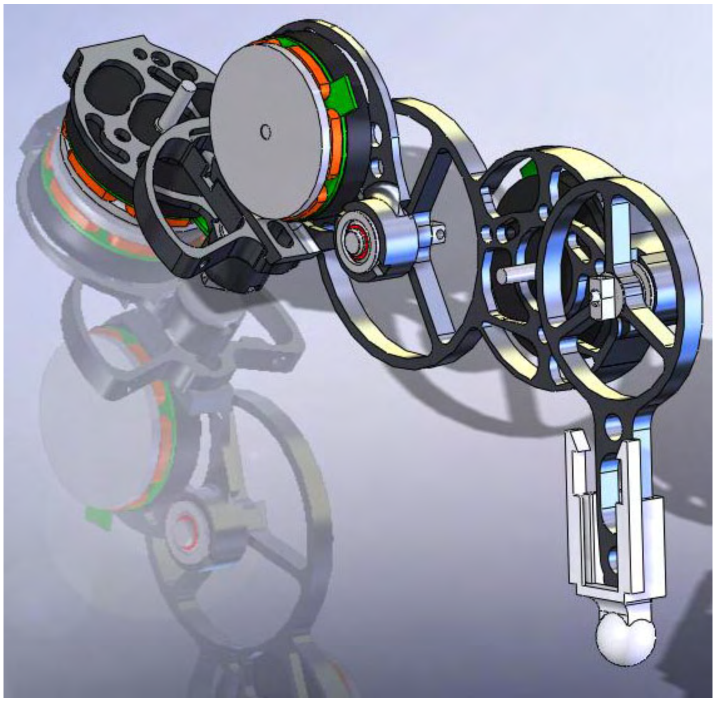


Figure 6.22: Rendering from an isographic projection of version 5 of the ModBot design.

6.3.14 Dynamic modeling

Equations of motion of a BLDC motor

The equations of motion for a brushless motor are identical to that of a DC motor. Therefore, the standard derivation begins with a current or voltage equation and a torque balance equation[69],

$$L\frac{di}{dt} = -Ri - K_b\dot{\theta} + v_a \tag{6.11}$$

$$J\ddot{\theta} = K_T i - B\dot{\theta} - T_L - T_f \tag{6.12}$$

and leads to, assuming a small inductance so the second order term drops away,

$$\frac{\Omega_m(s)}{V_a(s)} = \frac{K_m}{\tau s + 1}, \tag{6.13}$$

where K_m, the motor constant, is given by

$$K_m = \frac{K_T}{R_a B_m + K_T K_b}. \tag{6.14}$$

where the variables are defined as:

- B_m - Viscous damping in motor $(N \cdot m \cdot s/rad)$
- i - Current passing through the armature (A)
- J - Polar moment of inertia of armature $(N \cdot m \cdot s^2/rad)$
- K_b - Motor back emf constant $(V \cdot s/rad)$
- K_T - Motor torque constant $(N \cdot m/A)$
- L - Armature inductance (H)
- R_a - Armature resistance (Ohms)

- s - Laplace operator
- T_f - Coulomb friction torque in motor ($N \cdot m$)
- T_L - Externally applied load torque ($N \cdot m$)
- v_a - Voltage applied to armature (V)
- $V_a(s)$ - Laplace transform of armature voltage $v_a(t)$
- θ - Angular position of motor shaft (rad)
- τ_m - Mechanical time constant of motor (s)
- Ω_m - Angular velocity of motor (rad/s)
- $\Omega_m(s)$ - Laplace transform of angular velocity

6.3.15 Commutation methods for BLDC motors (brushless dc motors)

Brushless DC motors (referred to as BLDC motors) cannot be controlled in the same simple manner as DC motors (via applying a direct current through the motor coils). There are several advantages to the use of BLDC motors over brushed motors, however, including higher efficiency, lower friction, minimal torque ripple (if field-oriented or sinusoidal commutation is used), no wear except for the ball bearings (no periodic brush inspection requirements) and higher torque/weight ratio[110][39]. Indeed, with the inexpensive nature of microprocessors and increasing ease of use, brushless motors have been gaining steady favor for the past decade.

Block commutation

The simplest method of commutating a BLDC motor is 'trapezoidal' or 'block' commutation[110]. This consists of a few steps, beginning with measuring

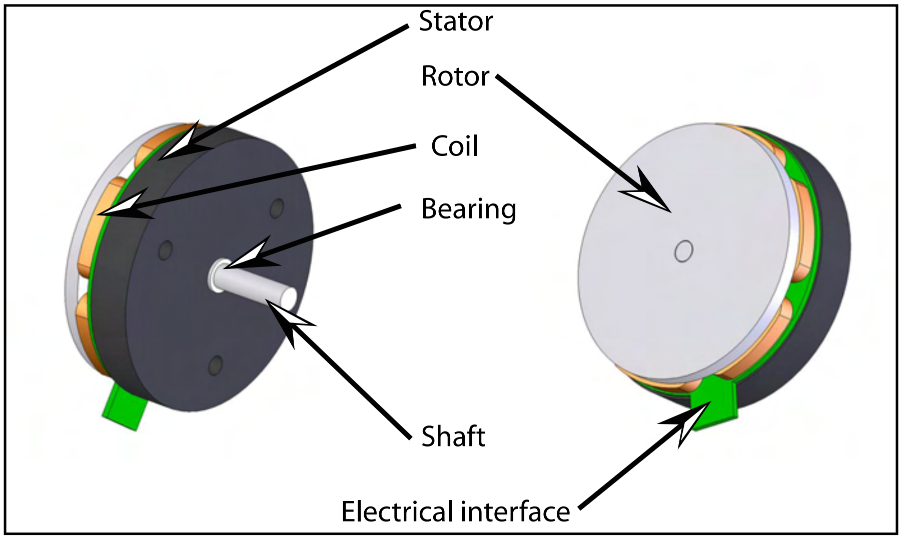


Figure 6.23: Basic components of a BLDC motor.

rotor position via Hall sensors, optical or magnetic encoders, or back EMF generated by each coil. The circuit design to switch the appropriate power levels for a three-phase BLDC motor is to be described in a forthcoming section. Essentially a rotor-position-dependent code is present in the Hall sensor data (typically Hall sensor outputs are filtered to be a binary signal). In stages, a pair of coils are connected in such a way as to allow current to flow between the two coils, forming a field which draws the rotor to rotate to align opposing fields. As the rotor rotates, the Hall sensor 'code' adjusts, and the current flow is adjusted to be through appropriate pairs of coils to continuously draw the rotor toward the moving field. This switching of the field flow leads to a torque ripple effect which is dependent on several motor parameters in terms of peak-to-peak wave amplitude.

Initial tests were performed using the Maxon EC32 flat motor with Hall sensors.The motor was connected to a computer running Matlab Realtime Workshop through a custom designed and built FET-based amplifier circuit. The motor is a

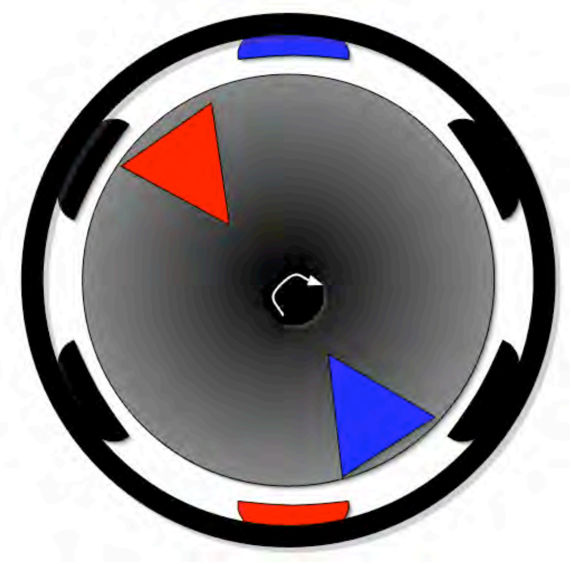


Figure 6.24: Simplified representation of a BLDC motor with torque created by applying a current through two opposing pairs of coils.

three-phase brushless motor with four pole-pairs, which means that two FETs were employed per motor coil for a total of six FETs. The motor commutation sequence as described previously, given a particular three-bit Hall sensor binary code, the coils were actuated in the appropriate combination of either connection to the positive, the negative, or no connection which will allow the current to flow in either direction, one direction, or no direction through a coil.

The circuit command signals were sent through a standard PC parallel port which can operate in realtime mode in Realtime Workshop. Realtime Workshop contains a block for the standard pc parallel port which was used. The algorithm, which is displayed in Figures 6.27-6.35, was executed at 20kHz. This allowed the motor coil current to be controlled through PWM. (PWM frequency must be at least double the maximum frequency of a motor for a moderately efficient low ripple constant cur-

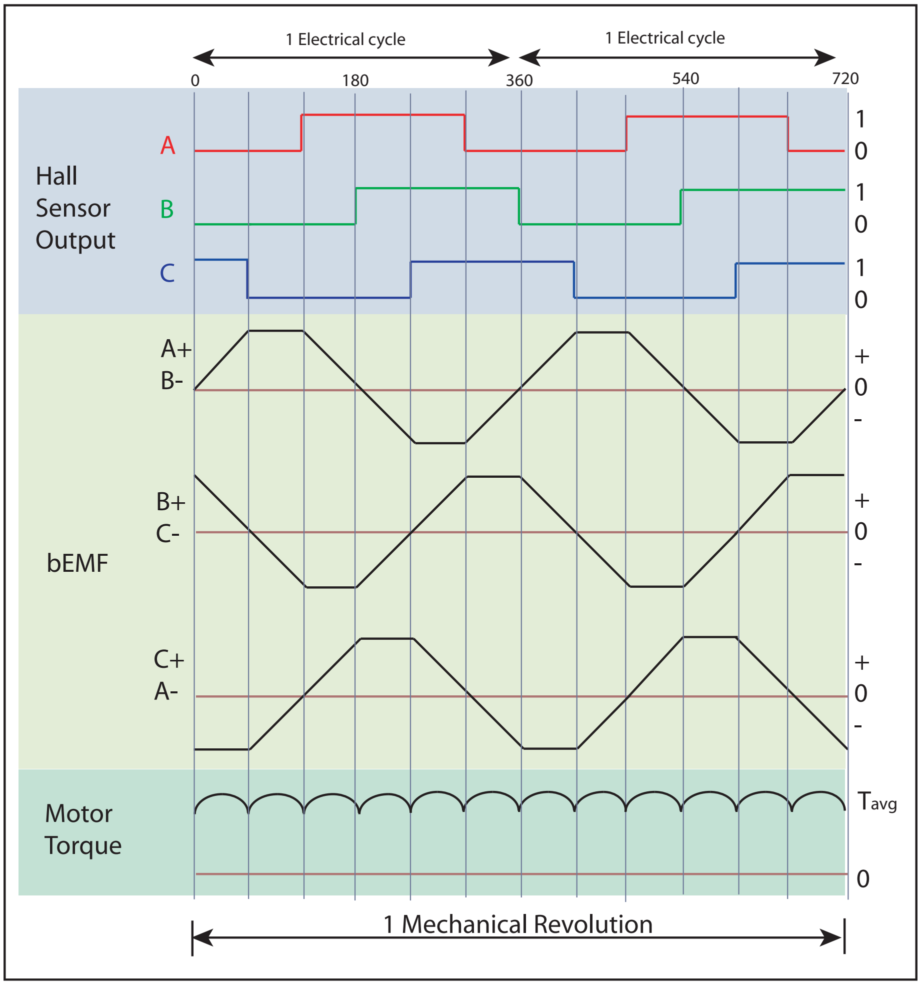


Figure 6.25: This denotes block commutation over one mechanical revolution for a typical two-phase BLDC motor with trapezoidal back-emf (adapted from AN885 technical note document from microchip.com). The torque ripple due to commutation is evident, and is undesirable for fine bio-mimetic control experiments.

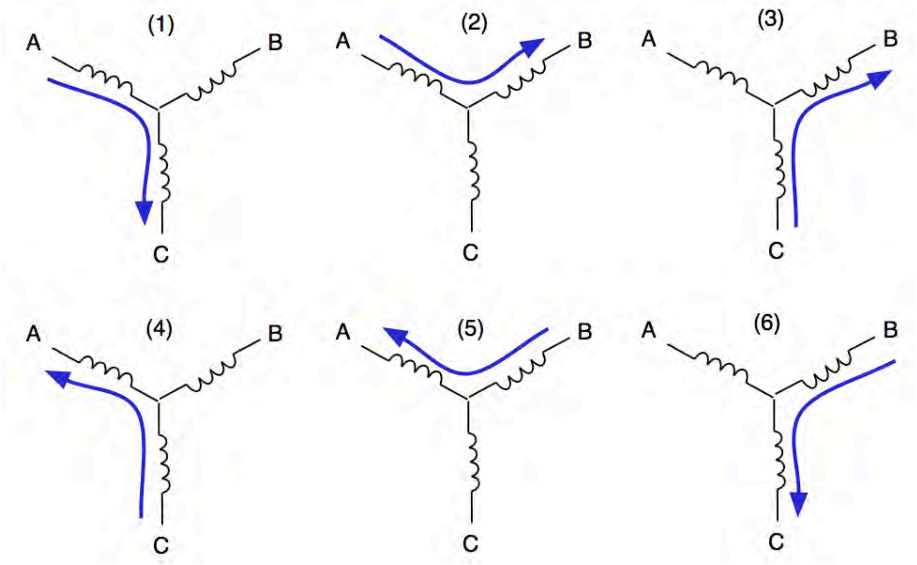


Figure 6.26: Block commutation sequence for a three-phase BLDC motor. Note the numerical sequence is the order, and that the third coil is left floating in each sequence.

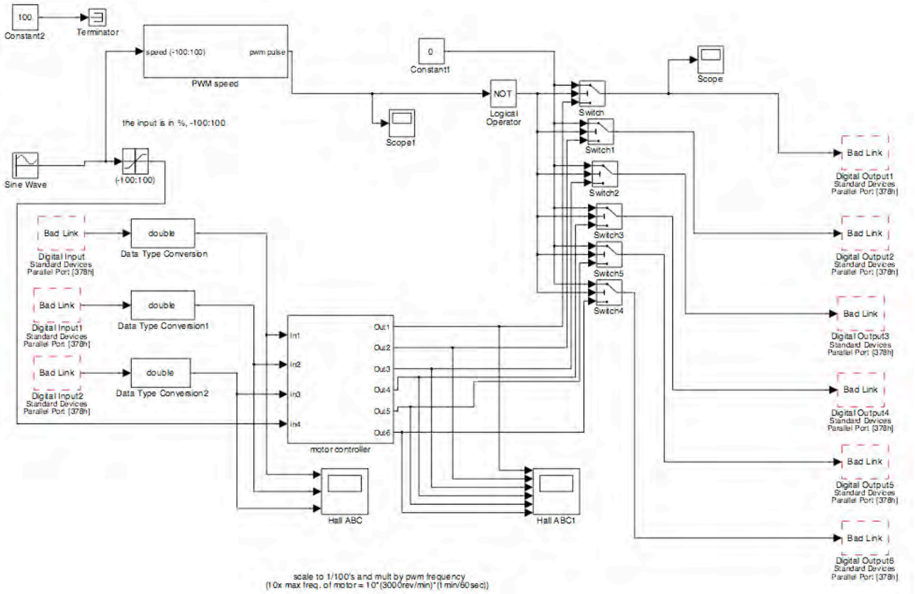


Figure 6.27: The overall view of the brushless motor block commutation model implemented in Matlab Simulink's Realtime Workshop.

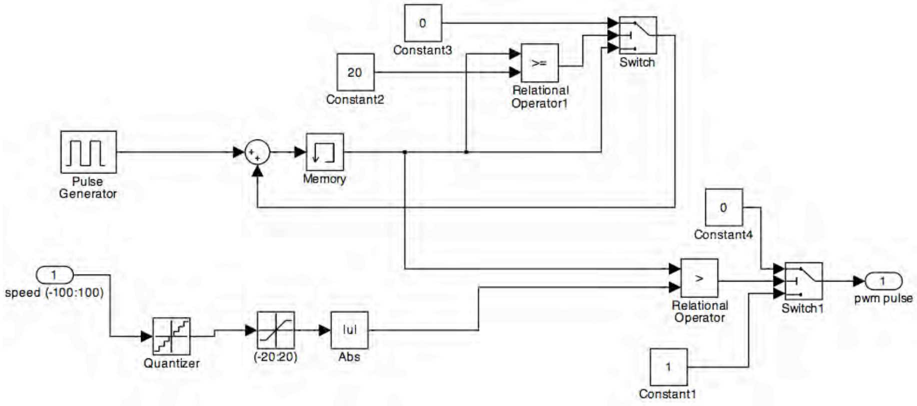


Figure 6.28: PWM generation scheme which works by repeating a cycle of pulses at a rate of 1kHz, with a number of pulses on versus off determining the PWM duty cycle (each pulse 20kHz fequency).

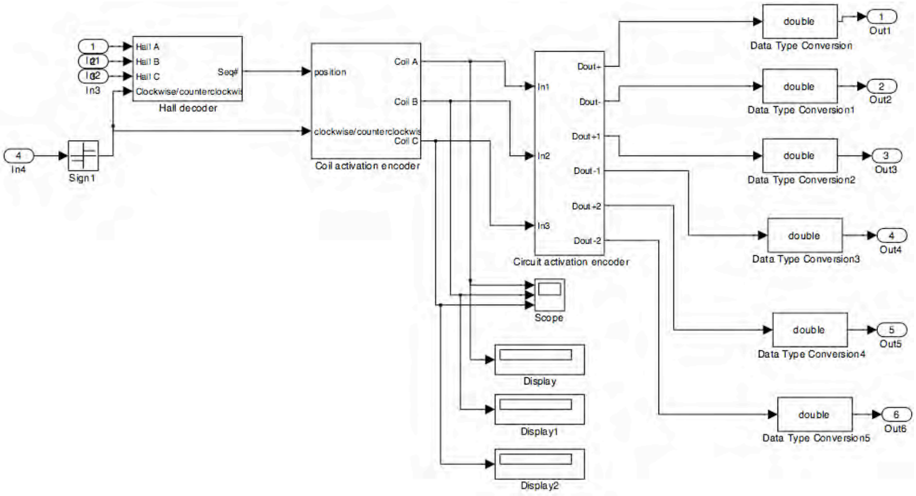


Figure 6.29: This depicts the overall Hall sensor decoding scheme. The three Hall sensors, corresponding to a 3-phase brushless motor's rotor electrical phase are used to decode a sector that the rotor is in. The coil activation encoder sends command to the circuit controller to activate a particular coil. The circuit activation encoder then flips appropriate logic bits to the FET transistors which create a positive, negative, or floating voltage in the appropriate coils.

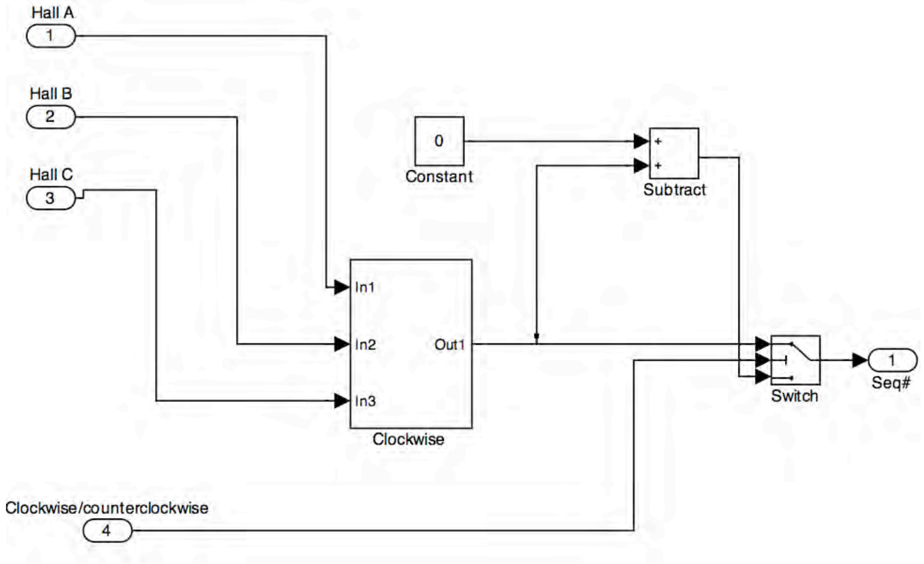


Figure 6.30: This figure depicts the Hall decoder, which has either a positive or negative direction component input.

rent). The experiment consists of tracking a sine wave, constant velocity set points, and step response inputs. These experiments were performed with the motor free running and with the motor coupled to a prototype robot leg. The amplitude of torque ripple in the Maxon motors with block commutation is unacceptably large.

The reason that this is unacceptably large is that the intention of the robot leg is to mimic biological systems (through torque control) and torque ripple (due to the actuator rather than the command) is non-present in muscle actuated systems.

This method of commutating BLDC motors is therefore more desirable in situations in which motor speed will tend to be higher and more constant because the torque ripple is velocity dependent due to the inertia and the time constant of the motor. In the case dealing with locomotion and object manipulation using the Modbot, RPMs tend to be low or 0, directions change frequently, and the motor oc-

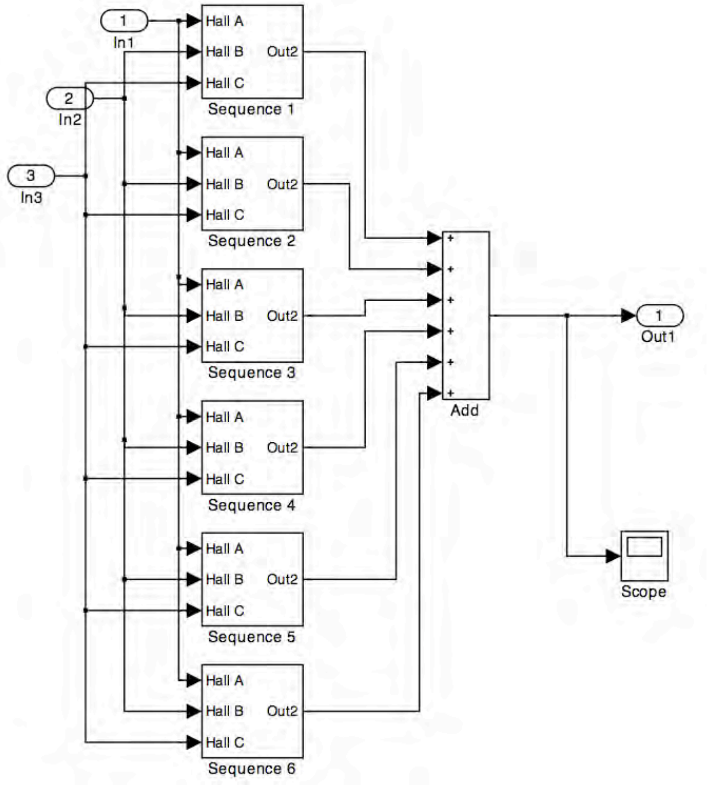


Figure 6.31: This is a detail of the Hall sensor decoding scheme. The appropriate value for each possible sector the rotor could be in is compared to the actual inputs as logical bit operations (which are very efficient computationally), and outputs the number of the sector

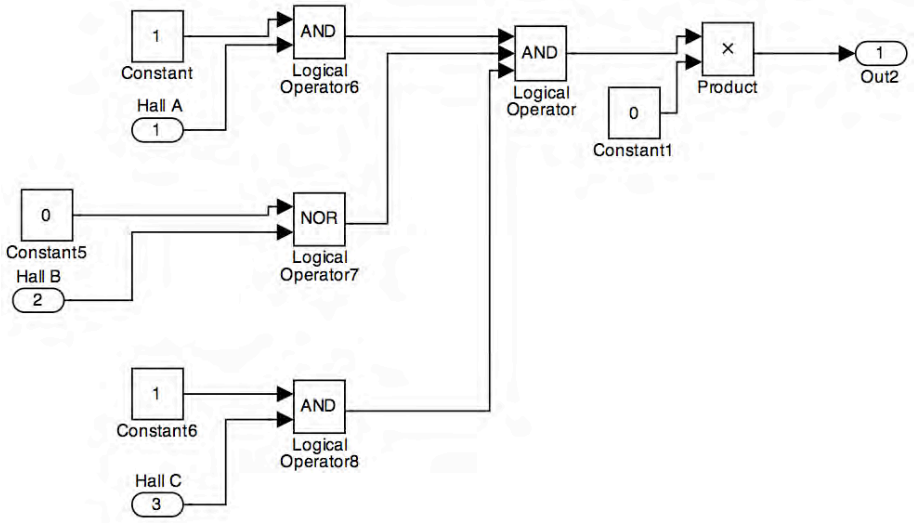


Figure 6.32: Detail of one of the six sector decoders. Only one will output a value per rotor position. Each one has a unique ID which is passed to the coil activation encoder.

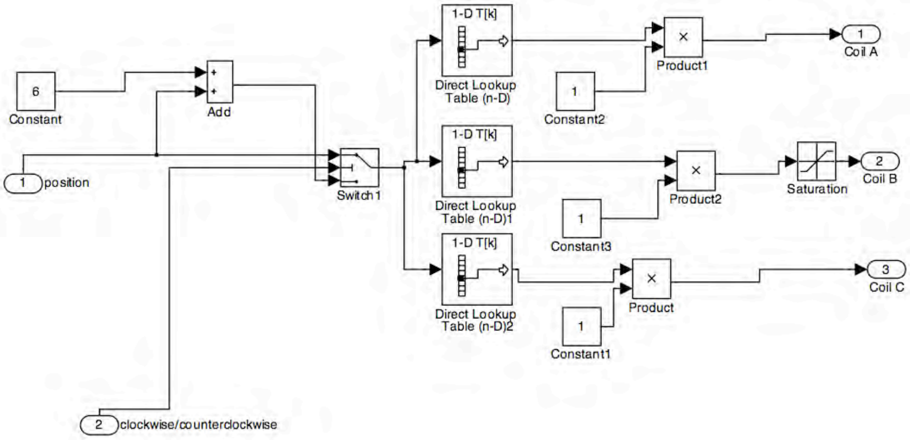


Figure 6.33: Lookup tables store the appropriate coil value per particular combinations of inputs.

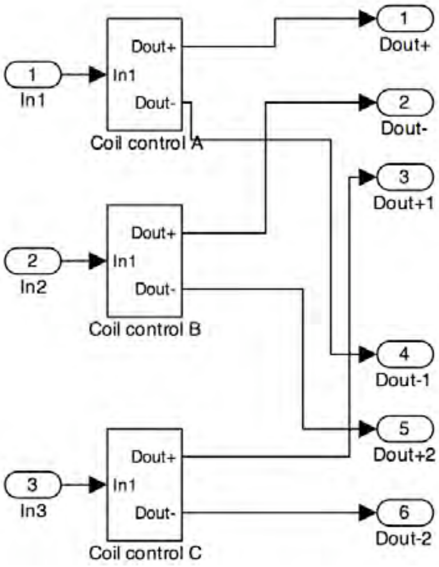


Figure 6.34: The circuit control logic determines, given a desired coil value for each coil, what pins to pull high or low in the triple h-bridge configuration circuitry.

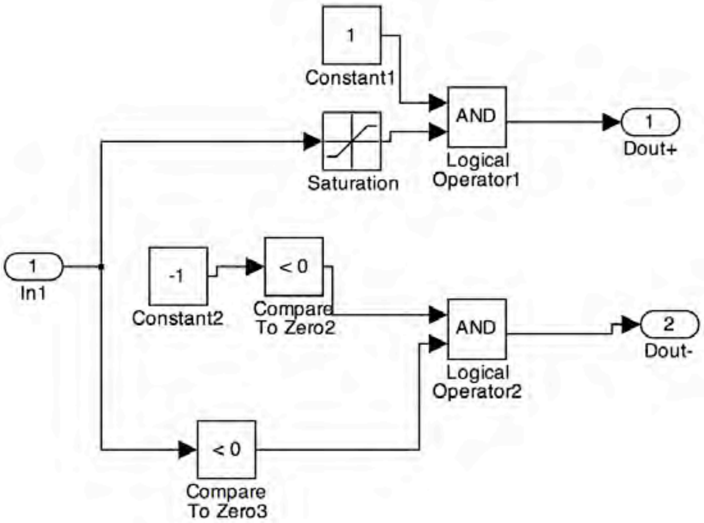


Figure 6.35: This depicts one of the circuit control logic blocks. Again, for computational efficiency bitwise logical comparisons are used.

casionally will be stalled. A commutation method which creates a smooth transition between stator pole locations must be employed. Two methods of note are sinusoidal commutation and field oriented control. Sinusoidal commutation is appropriate for low speeds, but inappropriate for high speeds due to computational complexity, back EMF altering the shape of the sine wave, etc. However, for the present application, speeds are low. Thus, sinusoidal commutation is selected.

The compact design of the robot leg does not allow direct external rotor position measurements, but position measurements are necessary to properly commutate a BLDC motor sinusoidally. The Hall sensor is passed through some circuity which filters the Hall sensor signal which transforms it into a square wave which only provides a very rough estimate of rotor position at approximately 25 counts per revolution. (The Maxon motor has four pole-pairs – north south pole magnets embedded in the motor.)

Upon experimentation on the Maxon motor, the magnetic field due to the rotor pole-pairs was not availabe for external measurement due to the closed rotor case design. Another motor, a Portescap NuvoDisk motor, does not have built in Hall sensors but the pole-pair construction does make the rotor magnetic field available for external measurement with Hall sensors. The Hall sensor signal is not interfered with by the field generated in the coils because they are oriented to be orthogonal to the field generated in the coils. By measuring the Hall sensor signal directly, a high resolution rotor position can be extracted from that data. This allows sinusoidal or any other form of commutation desired, as well as providing a position measurement of the robot leg joint associated with that particular motor.

6.3.16 Sinusoidal commutation, position estimation, and control using analog Hall Effect sensors

There are several considerations during implementation of a system which estimates not only the position of the Hall sensors relative to each other and overall motor coil phase, but also position, velocity, and number of quadrants crossed. These issues are to be addressed in the following sections.

Low pass filter

A passive low pass RC circuit was realized in order to attenuate signal frequencies above approximately 150kHz. The frequency response of the filter falls off as a first order transfer function, shown in Figure (6.36). The relationship between frequency and values of capacitance and resistance are given by

$$f_c = \frac{1}{2\pi RC}. \tag{6.15}$$

With the selected cutoff frequency, the RC constant which was used to select the values is given by

$$RC = 2\pi f_c = 2 * \pi * (1e6) = 6.28e6 \tag{6.16}$$

A 0.01μF ceramic capacitor and a 100 Ω 1% resistor were used to form the circuit, as in Figure (6.37).

Effective resolution

The effective resolution of the rotor angle estimate is found by computing the maximum peak-peak voltage (V_{pp}) of the Hall sensors, then the analog to digital converter (ADC)[7] resolution (given Vss-Vdd voltage scaling), then dividing the V_{pp}

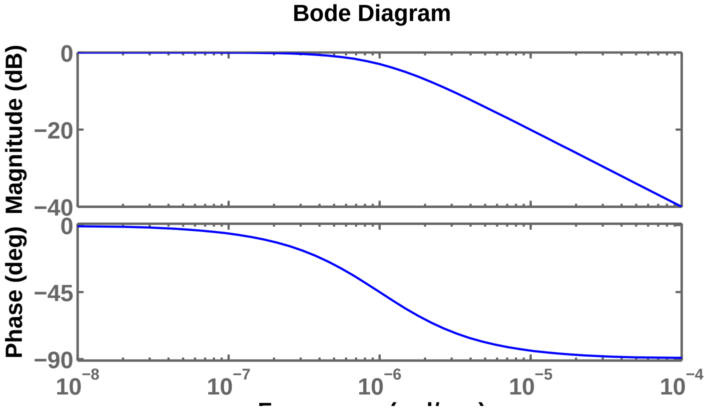


Figure 6.36: Frequency response of low pass filter for Hall sensors.

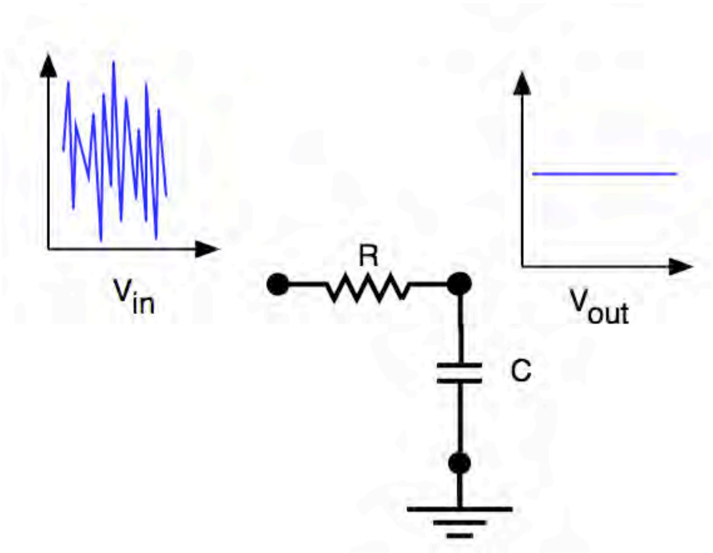


Figure 6.37: RC circuit, with $f_c = \frac{1}{2\pi RC}$, resistance in Ohms, and capacitance in Farads. The signal containing high frequency components enters the circuit, and leaves with high frequency components attentuated, leaving the lower frequency components intact.

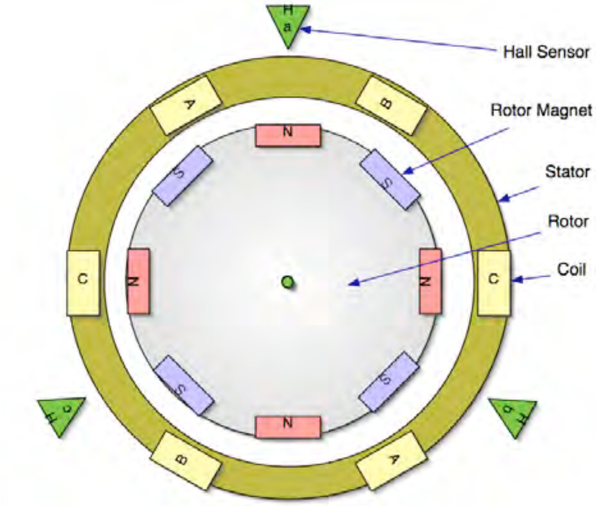


Figure 6.38: A simplified representation of a brushless DC motor with three phases and four pole pairs.

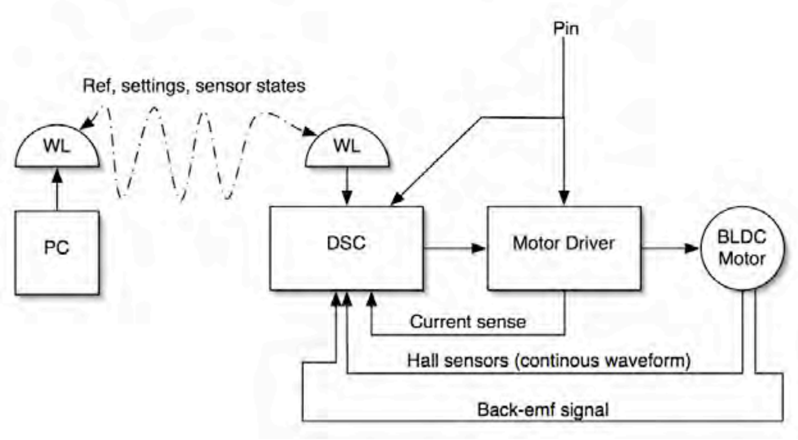


Figure 6.39: A block diagram representation of the motor control system, consisting of the PC providing, via bluetooth wireless connection, command reference, parameters, and receiving position and sensor data, the digital signal controller (DSC), motor driver circuitry, and BLDC motor.

by the resolution (below calcs given at 10 bit resolution, 12 bits are possible using this dsPIC[67], but affect its ability to sample simultaneously):

$$\begin{aligned} V_{pp} &= 0.400V \\ V_{adc-res} &= 3.3V/1024 \\ \text{counts/revolution} &= V_{pp}/V_{adc-res} * 4 \\ &= 1984 \\ \text{resolution}_{\text{degrees}} &= 360/1984 = 0.18^o \end{aligned} \tag{6.17}$$

12 bit resolution adc leads to 4932 counts per revolution, or resolution of 0.073^o.

The joint angle estimate resolution is, assuming no slip of the cable drive (which is probably not a perfect assumption - and will be removed by a direct measure of the joint angle in the future), the resolution of the rotor position divided by the gear ratio, which is approximately 20:1. Therefore the effective joint angle estimate resolution is $9e - 3^o$, which results in a worst case endpoint resolution of 0.9mm. This is quite sufficient for sensorimotor tasks, though a direct measurement of the joint angle will improve this resolution markedly, depending on the encoder used.

Relative angle estimation by quadrant determination

The Nuvo disk BLDC motors are constructed with three phases and four pole-pairs (4 sets of N-S magnets embedded in the rotor). This is key for determining the ratio of electrical cycles to mechanical revolutions. Considering one Hall sensor which outputs a voltage proportional to magnetic flux field density at a particular orientation and position, it is clear that one pole-pair (one north pole magnet at one rotor angle, and a south pole magnet positioned π radians apart along the outer radius of the rotor) will result in the measurement of a single sinusoid which repeats after one mechanical revolution. Thus, to commutate the coils, one would need to repeat a particular coil current pattern only once. Thus there is one electrical cycle per mechanical revolution.

The number of electrical cycles correspond to the number of pole-pairs of the rotor. Since we have four pole-pairs in this motor, there are four electrical cycles per mechanical revolution. This means that, since we have three Hall sensors mounted, we can uniquely determine position of the rotor within a $\pi/2$ radian segment. All that must be done to obtain a relative measure of angle which can exceed in either direction the $\frac{\pi}{2}$ radians electrical cycle is to build in software a zero crossing detector which can determine a crossing from one measure extreme to another when passing from one quadrant to another. This is done most computationally efficiently by measuring using an oscilloscope (or using a particular normalized ADC value determined onboard the embedded system, or computing with matlab) the three Hall signals during a mechanical revolution and estimating the phase difference of the sinusoids (which, since the sensors are mounted $\pi/3$ radians apart, should be close to $n\pi/3$, n the sensor number beginning with zero). Then the characteristics that determine which quadrant the rotor is in can be used for a switch statement or boolean comparison, then an integer sine and arcsine lookup table are stored in memory and used along with trigonometric relationships to determine the rotor angle. At this point the values to be injected back into the coils for commutation can be phase shifted in a controlled manner.

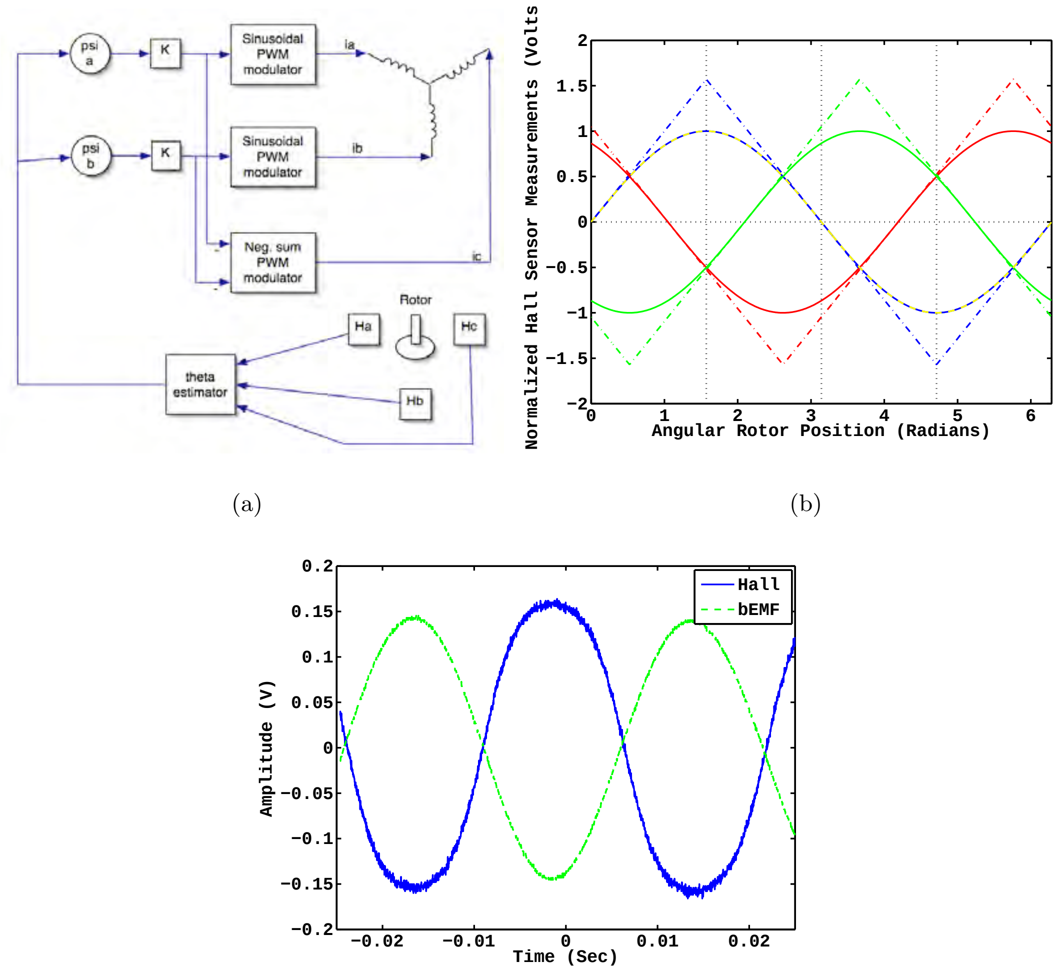
Sinusoidal commutation

As discussed previously, there are several methods for commutating a BLDC motor. These include block ('trapezoidal') commutation, sinusoidal commutation, and field-oriented control ('space vector modulation'). Block commutation has been previously discussed and rejected for this implementation, while sinusoidal commutation will be discussed presently and applied. Field-oriented control (FOC) is, in a sense, a special case or method of sinusoidal commutation. The main advantage of FOC is that it is useful for high and low speeds if one has available a fast DSP which can compute trigonometric transformations rapidly. In the present application low speeds and a lack of the fast floating point DSP's availability will be experienced and so sinusoidal control's increased simplicity is desirable.

Method of sinusoidal commutation

The maximum torque upon the rotor can be exerted when one of the poles is positioned directly between a pair of coils which are creating a field of opposing magnetic orientation. Considering the coordinate system of the rotor, the force due to the field generated in the coil acting upon the rotor can be decomposed into a tangential component and an orthogonal component which creates a torque balanced by the motor bearings as opposed to causing a rotation. Ideally one would create a torque only in the direction to cause acceleration of the rotor, as force acting upon the bearings is wasted energy. However some orthogonal forces are inevitable.

If the fields of the coil and permanent magnets are aligned, the rotor will be stationary. If they are misaligned slightly, a torque will act to rotate the rotor until they are aligned. If the misalignment is perpetuated by rotating the coil-based field ahead of the rotor's rotation, a continuous, cog-free rotation of the rotor occurs.



(a)

(b)

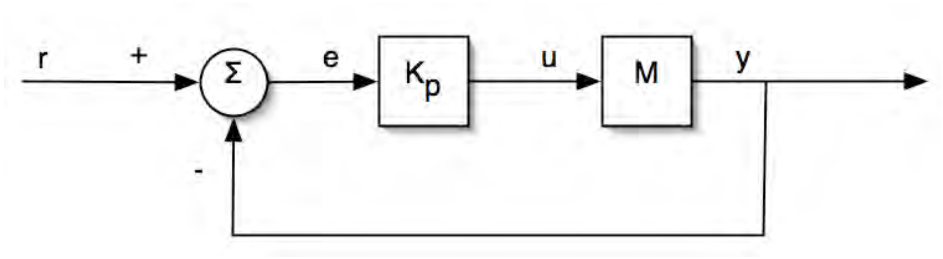
(c)

Figure 6.40: (a) Motor controller main blocks for open loop current control. (b) Hall sensor signals (solid) showing the phase difference between coil ($\pi/3$) A, B, and C, and the arcsine of each signal showing the periodic nature of the direct angular estimate. This emphasizes the need to recover the sector (using multiple sensors makes this possible) within the $\pi/2$ radian quadrant where absolute position within that quadrant is known.(c) Hall sensor signal scaled to comparable levels with a single coil's back emf during a constant angular velocity. Note that the Hall signal is inverted relative to the backemf, but in phase.

The torque generated will be determined by the amplitude of the sinusoids (which, in a digital system, can be realized via PWM), as well as the degree of phase shift of the active fields.

The fields are a sinusoidal function of rotor angular position, with each coil phase-shifted (since each coil in a three-phase motor is physically phase shifted by sixty degrees) and then phase shifted relative to rotor position. This rotor position phase shift can be constant or dynamic - a function of rotor speed.

Since friction and electronics such as the intrinsic freewheeling diodes contribute nonlinear effects to the motor dynamics, a local proportional feedback controller is implemented to account for these effects - taking in a particular current command, and outputting a PWM command. The actual coil current is measured with the use of high precision current sensing resistors. This measure is used as a feedback signal to drive the system to desired levels and prevent excess current flow (Figure 6.41(a)).



(a)

Figure 6.41: (a) Local feedback control helps cancel nonlinearities due to friction, and nonlinear electrical characteristics in the motor.

The controller can be a simple proportional control or a PI controller. Essentially ANY reasonable feedback will obey the axiom that feedback control tends to linearize nonlinearities in a system (depending on the type of nonlinearity). Even if a simple proportional control is used, performance can improve significantly. The gain is set using a standard classical control method - find the gain at which the system becomes unstable and then reduce the gain by an arbitrary factor such as one half. In practice this is very effective. A good treatment of related nonlinear phenomena is given in[27], and an example of improvement with an interesting application is given in [34].

The cycle frequency for the PWM is computed by considering characteristics of the physics of the motor used.

PWM is based on the fact that the motor, due to its inertia and LR (Inductance and resistance) characteristics, acts as a low pass filter. Thus a repeating square wave input current of varying frequencies will be filtered to a DC current between the maximum voltage and zero volts. The proportion of "on" time versus "off" time of the PWM wave determines the average motor current. The period of one of the PWM wave cycles determines the smoothness of the DC average current, as a function also of motor characteristics (inductance and resistance).

The less smooth the current waveform in the motor, the greater the power loss. Energy is lost during the switching time of the transistors, the resistance in the windings of the motor, and wires in the system.

To quantify the losses due to lack of smoothness of the current waveform we perform a simple example. If the current of a perfect DC signal in the motor is I, the power dissipation due to the motor, component, and wire resistances is given by

$$P = I^2R \tag{6.18}$$

However a 50:50 square wave (we assume a perfect square wave of current moving through the motor for simplicity of calculations) with a maximum current of $2I$ and minimum of zero would produce the relation

$$P = (2I)^2R/2 + (0)^2/2 = 2I^2R \tag{6.19}$$

This is double the energy loss. Thus it is desirable to achieve as smooth a DC current waveform in the motor as possible.

The quantity of RF (radio frequency) interference increases with higher switching frequencies, and in some cases this must be attenuated by one or more of several methods which will not be covered here since signals are on the order of kHz, and RF interference is more of an issue at MHz bandwidths.

In order to choose the switching frequency, one can simply choose from a standard frequency range, such as a few kHz, or compute the required switching frequency in order to achieve a desired percent of fluctuation of the current waveform as a function of motor parameters.

It can easily be shown from the equation derived from an LR circuit that the above equation is given by

$$i = Ie^{-t/\tau} = Ie^{-tR/L} \tag{6.20}$$

We desire that at time $t = T/2$ versus time $t = 0$

$$i_1 = (1 - \frac{P}{100}) \cdot i_0 \tag{6.21}$$

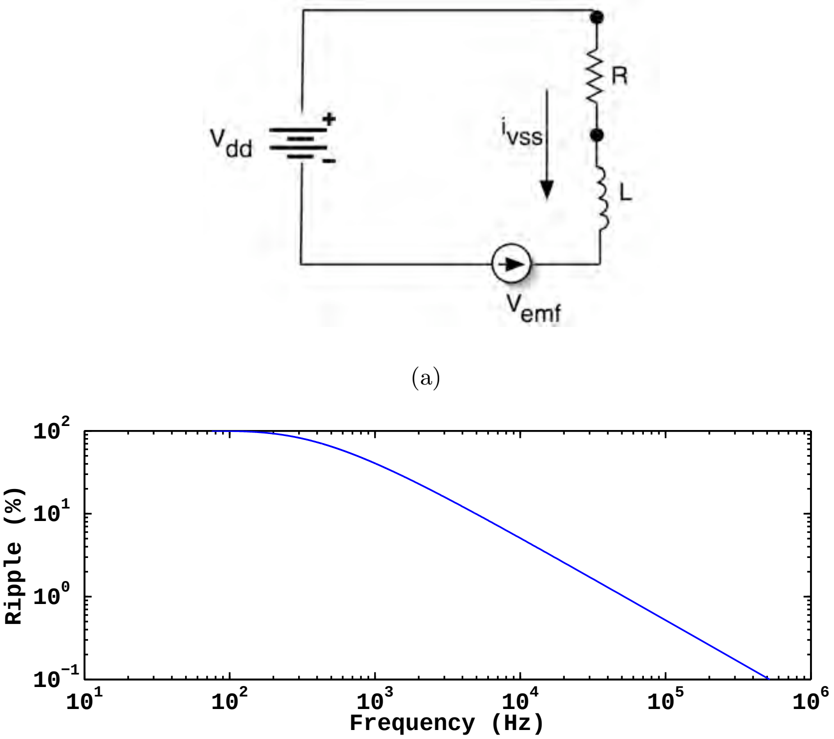
substituting this equation into the LR circuit equation and solving for T yields

$$T = \frac{-2L}{R}\ln(1 - \frac{P}{100}) \tag{6.22}$$

and knowing $f = 1/T$ we arrive at

$$f = \frac{-R}{2L \cdot \ln(1 - P/100)} \tag{6.23}$$

or, defining ripple percent $A_r = (1 - P/100)$, This requires knowing or measuring motor parameters of inductance (L) and resistance (R). The inductance and resistance of the Portescap BLDC motors are given to be $0.36mH$ and 3.75Ω. Thus, a ripple percentage (for a constant desired average current) vs. pwm frequency plot becomes an inverse logarithmic relation shown in Figure (6.42).



(a)

(b)

Figure 6.42: Power dissipation percent vs. PWM cycle frequency for R=3.75Ω and L=$0.36mH$. Required PWM frequency can be determined by this plot and the required power dissipation.

Thus for this motor, a ripple amplitude of 1% would be possible with a PWM frequency of 51.8kHz. Since the power amplification circuitry employed has an absolute maximum PWM frequency of 50kHz, the minimum ripple amplitude possible at this rate is 3.5%, which is still acceptable. However, it must be noted that increasing PWM frequency reduces effective resolution of the PWM duty cycle. The relation [67] is given by

$$\text{Resolution} = \frac{\log(2T_{PWM}/T_{CY})}{\log(2)}. \tag{6.24}$$

Here 'resolution' is in bits. Thus the effective PWM resolution at a frequency of 10kHz is roughly 4000, while at 40kHz reduces to 1000 (the clock is at 8MHz, but using the onboard PLL, the code can be executed at up to 40MIPS, or equivalent to an 80MHz clock). The PWM frequency can be adjusted by altering one or both parameters of the motor included in this equation - namely the resistance and inductance. Here we achieve satisfactory performance with this PWM frequency, and so adjusting motor parameters is left to future research.

Parameter estimation

There are several parameters associated with controlling the motor for maximum torque generation. These include individual Hall sensor position, phase shift, maximum and minimum for each sensor (which can drift or slip over time due to sensor drift and movement). Since the main parameters have simple state-dependent nonlinear dynamics, they can be estimated by an extended Kalman filter with 'fast' noise processes driving them to a zero estimation error.

Extended Kalman Filter to estimate unknown parameters

We need to estimate ϕ_i, α_i, and β_i for all i, knowing only the ideal and initial conditions. We also want to use these estimates to estimate θ as well. α_i^0 and β_i^0 are computed after power-up during an initialization routine. Then all the parameters must be gradually adjusted over time (ie the Hall sensors may shift with temperature which affects magnitude and phase). The future estimates are propagated by an extended Kalman filter. The dynamics are related in the following way between angle, phase and Hall sensor measures (where n is the number of pole-pairs in the BLDC motor's rotor, ϕ is the phase of the three sensors, ψ_i is the individual phase for the specified sensor relative to the first, $v(t)$ is noise with $\eta(0, \Omega_v)$ and r is the hall measurement, for a given sensor, i):

$$r_i(t) = \frac{2(\sin[n\theta(t) - \psi_i - \phi] - \alpha_i)}{\beta_i - \alpha_i} - 1 + v(t) \tag{6.25}$$

The following is a standard EKF form with continuous dynamics and observation model[32].

6.3.17 Discrete EKF with continuous dynamics and observation model

Given a nonlinear dynamical system with state space $x(t)$ and additive Gaussian noise $w(t)$, and nonlinear dynamics $f[x(t), t]$, the system model is given by

$$\dot{x}(t) = f[x(t), t] + w(t), \tag{6.26}$$

Where the properties of the noise are (with the covariance of the noise defined by $Q(t)$),

$$w(t) \sim N[0, Q(t)]. \tag{6.27}$$

The measurement model, with observation function $h(\cdot)$ and measurement sample k defined by z_k,

$$z_k = h_k[x(t)] + v_k, k = 1, 2, ..., \tag{6.28}$$

with noise properties (and noise covariance at sample k R_k)

$$v_k \sim N[0, R_k]. \tag{6.29}$$

The initial conditions are given by

$$x(0) \sim N[\hat{x}_0, P_0], \tag{6.30}$$

and

$$E[w(t)v_k^T] = 0, \forall k, \forall t. \tag{6.31}$$

The state estimate ($\hat{x}(t)$) is propagated by the

$$\dot{\hat{x}}(t) = f[\hat{x}(t), t], \tag{6.32}$$

and the error covariance $P(t)$ by

$$\dot{P}(t) = F[\hat{x}(t), t]P(t) + P(t)F^T[\hat{x}(t), t] + Q(t). \tag{6.33}$$

Then the estimate pre-measurement $(-)$ is updated to the post-measurement $(+)$ estimate by mixing the measurement with the estimate of the Kalman gain (K_k)

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k[z_k - h_k(\hat{x}_k(-))], \tag{6.34}$$

and the error covariance is updated according to

$$P_k(+) = [I - K_k H_k(\hat{x}_k(-)]P_k(-), \tag{6.35}$$

with the gain matrix computed by

$$K_k = P_k(-)H_k^T(\hat{x}_k(-))[H_k(\hat{x}_k(-))P_k(-)H_k^T(\hat{x}_k(-) + R_k]^{-1}. \tag{6.36}$$

The matrices $F(\cdot)$ and $H(\cdot)$ are linearized system and observation models, respectively according to

$$F[\hat{x}(t), t] = \frac{\partial f[x(t), t]}{\partial x(t)}\Big|_{x(t)=\hat{x}(t)}, \tag{6.37}$$

and

$$H[\hat{x}_k(-)] = \frac{\partial h_k[x(t_k)]}{\partial x(t_k)}\Big|_{x(t_k)=\hat{x}_k(-)}. \tag{6.38}$$

Implementation

Application of the previously defined EKF begins with definitions of the system state space. In order to control the BLDC motor in the face of uncertainties we need to include the following states:

- q - number of $\pi/2$ quadrants traversed since the last reset of the estimator.
- $\hat{\theta}$ - The rotor angle estimate in mechanical revolution space.
- $\dot{\hat{\theta}}$ - The rotor angular velocity estimate in mechanical revolution space.
- $\hat{\phi}$ - The mean of the estimate of all of the Hall sensor phase angles.
- $\hat{\psi}_i$ - The mean of the estimate of the individual Hall sensor placements relative to the others.
- σ_ϕ - The covariance of the estimate of phase angles.
- σ_ψ - The covariance of the estimate of the individual phase angles of the sensors relative to each other.

- τ - Motor command state (can be implemented as position, velocity, or torque depending on requirements).
- α_i - The minimum Hall sensor measurement for sensor i.
- β_i - The maximum Hall sensor measurement for sensor i.
- r - The normalized Hall sensor measurement values.
- ζ - The motor reference (can be implemented as position, velocity, or torque depending on requirements).

The state dynamics are augmented with the Kalman filter dynamics as in 3. The state is then:

$$x = \left[\begin{array}{cccccccccccccc} q & \hat{\theta} & \dot{\hat{\theta}} & \sigma_\theta & \sigma_{\dot{\theta}} & \hat{\phi} & \hat{\psi}_i & \sigma_\phi & \sigma_{\psi_i} & \tau & \alpha_i & \beta_i & r & \zeta \end{array} \right]^T \tag{6.39}$$

To create the overall system model, we consider first the individual components of the model (let us define $\tilde{\theta}$ as the indirect observation of position based on Hall sensor signals) :

$$\begin{aligned} r_i(t) &= 2\frac{\text{hall}_i(t) - \alpha_i}{\beta_i - \alpha_i} - 1 \qquad (6.40) \\ \tilde{\theta}(t) &= \Big[q\pi/2 + \frac{1}{3n}\Big([\text{asin}(r_0) - \phi] \\ &\quad + [\text{asin}(r_1) - \phi - \psi_1] \\ &\quad + [\text{asin}(r_2) - \phi - \psi_2]\Big)\Big] \\ q(t) &= q + \text{floor}\Big(\frac{1.8|\dot{\hat{\theta}}|}{\pi}\Big)\text{sign}(\dot{\hat{\theta}}) \end{aligned}$$

The $\tilde{\theta}$ equation can be considered to be an average between sensor measures. A simpler (computationally) implementation is

$$\tilde{\theta}(t) = q\pi/2 + \frac{1}{n}[\text{asin}(r_0) - \hat{\phi}], \tag{6.41}$$

which can then be averaged over several samples. If a perfect measure of θ were available (this is the angle within a $\pi/2$ quadrant we are interested in for the following), and there were no noise in the Hall sensor measurements, the relative sensor placements could be computed by

$$\begin{aligned} \theta &= \frac{1}{n}\text{asin}[r_0] - \phi \\ \psi_1 &= \frac{1}{n}\text{asin}[r_1] - \phi - \theta \\ \psi_2 &= \frac{1}{n}\text{asin}[r_2] - \phi - \theta. \end{aligned} \tag{6.42}$$

Though we need to estimate the measure of the previous three variables, the above relationships are useful in computing the estimates.

The state estimate propagation and error covariance propagation are combined into one matrix to form

$$\dot{\hat{x}}(t) = \begin{bmatrix} \text{floor}(\frac{1.8|\dot{\hat{\theta}}|}{\pi})\text{sign}(\dot{\hat{\theta}}) \\ \dot{\hat{\theta}} \\ -J^{-1}(K_e + \mu_f)\dot{\hat{\theta}} \\ 2\dot{\hat{\theta}}\sigma_\theta + \Omega_\theta \\ -2J^{-1}(K_e + \mu_f)\dot{\hat{\theta}}\sigma_{\dot{\theta}} + \Omega_{\dot{\theta}} \\ 0 \\ 0 \\ \Omega_\phi \\ \Omega_{\psi_i} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.43}$$

The Kalman gains are computed (at timestep k before the sample update $'(-)'$ but omitted for notational clarity) as

$$\begin{aligned}
K_{\theta_k} &= \sigma_{\theta_k}\mathrm{ncos}(n\hat{\theta}_k - \hat{\phi}_k)[\mathrm{ncos}^2(n\hat{\theta}_k - \hat{\phi}_k)\sigma_{\theta_k} + \Omega_{y_\theta}]^{-1} && (6.44)\\
K_{\dot{\theta}_k} &= n\sigma_{\theta_k}[\cos(n\hat{\theta}_k - \hat{\phi}_k) - \cos(n\hat{\theta}_{k-1} - \hat{\phi}_{k-1})]\\
&\quad * \left(\sigma_{\dot{\theta}_k} n[\cos(n\hat{\theta}_k - \hat{\phi}_k) - \cos(n\hat{\theta}_{k-1} - \hat{\phi}_{k-1})]^2 + \Omega_{y_{\dot{\theta}}}\right)^{-1}\\
K_{\sigma_{\theta_k}} &= K_{\theta_k}\\
K_{\sigma_{\dot{\theta}_k}} &= K_{\dot{\theta}_k}\\
K_{\phi_k} &= -\sigma_{\phi_k}\cos(n\hat{\theta}_k - \hat{\phi}_k)\left(\sigma_{\phi_k}\cos^2(n\hat{\theta}_k - \hat{\phi}_k) + \Omega_\phi\right)^{-1}\\
K_{\psi_{i_k}} &= -\sigma_{\psi_{i_k}}\cos(n\hat{\theta}_k - \hat{\phi}_k - \hat{\psi}_{i_k})\left(\sigma_{\psi_{i_k}}\cos^2(n\hat{\theta}_k - \hat{\phi}_k - \hat{\psi}_{i_k}) + \Omega_{\psi_{i_k}}\right)^{-1}\\
K_{\sigma_{\phi_k}} &= K_{\phi_k}\\
K_{\sigma_{\phi_{i_k}}} &= K_{\psi_{i_k}}
\end{aligned}$$

Now we define a composite Kalman gain, $K \in \Re^{n_x \times n_x}$, which will be a diagonal matrix,

$$K_k = \begin{bmatrix} 0 & & & & & & & & & \\ & K_{\theta_k} & & & & & & & & \\ & & K_{\dot{\theta}_k} & & & & 0 & & & \\ & & & K_{\sigma_{\theta_k}} & & & & & & \\ & & & & K_{\sigma_{\dot{\theta}_k}} & & & & & \\ & & & & & K_{\phi_k} & & & & \\ & & & & & & K_{\psi_{i_k}} & & & \\ & & 0 & & & & & K_{\sigma_{\phi_k}} & & \\ & & & & & & & & K_{\sigma_{\phi_{i_k}}} & \\ & & & & & & & & & 0 \end{bmatrix}. \qquad (6.45)$$

We define, for space considerations, the vector $d\omega \in \Re^{n_x \times 1}$ to represent the errors between observation and predictions (for the portions related to state measurement update), and the error covariance update magnitude (for the portions related to error covariance update), where all variables are pre-update (except for the $k - 1$

variables), but the $'(-)'$ is omitted for notational clarity,

$$\begin{aligned}
d\omega_{\theta} &= r_0 - \sin(n\hat{\theta}_k - \hat{\phi}_k) \\
d\omega_{\dot{\theta}} &= (r_k - r_{k-1}) - (\sin(n\hat{\theta}_k - \hat{\phi}_k) - \sin(n\hat{\theta}_{k-1} - \hat{\phi}_{k-1})) \\
d\omega_{\sigma_{\theta_k}} &= -n\sigma_{\theta_k}\cos(n\hat{\theta}_k - \hat{\phi}_k) \\
d\omega_{\sigma_{\dot{\theta}_k}} &= -n\sigma_{\theta_k}[\cos(n\hat{\theta}_k - \hat{\phi}_k) - \cos(n\hat{\theta}_{k-1} - \hat{\phi}_{k-1})] \\
d\omega_{\phi_k} &= r_0 - \sin(n\hat{\theta}_k - \hat{\phi}_k) \\
d\omega_{\psi_{i_k}} &= (r_0 - r_1) - \Big(\sin(n\hat{\theta}_k - \hat{\phi}_k) - \sin(n\hat{\theta}_k - \hat{\phi}_k - \hat{\psi}_{i_k})\Big) \\
d\omega_{\sigma_{\phi_k}} &= \sigma_{\phi_k}\cos(n\hat{\theta}_k - \hat{\phi}_k) \\
d\omega_{\sigma_{\psi_{i_k}}} &= \sigma_{\psi_{i_k}}\cos(n\hat{\theta}_k - \hat{\phi}_k - \hat{\psi}_{i_k})
\end{aligned} \tag{6.46}$$

The state estimate update and error covariance update are combined into one set of calculations in the following way,

$$\begin{bmatrix} 0 \\ \hat{\theta}_k(+) \\ \dot{\hat{\theta}}_k(+) \\ \sigma_{\theta_k}(+) \\ \sigma_{\dot{\theta}_k}(+) \\ \hat{\phi}_k(+) \\ \hat{\psi}_{i_k}(+) \\ \sigma_{\phi_k}(+) \\ \sigma_{\psi_{i_k}}(+) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{\theta}_k(-) \\ \dot{\hat{\theta}}_k(-) \\ \sigma_{\theta_k}(-) \\ \sigma_{\dot{\theta}_k}(-) \\ \hat{\phi}_k(-) \\ \hat{\psi}_{i_k}(-) \\ \sigma_{\phi_k}(-) \\ \sigma_{\psi_{i_k}}(-) \\ 0 \end{bmatrix} + K_k \begin{bmatrix} 0 \\ d\omega_{\theta_k} \\ d\omega_{\dot{\theta}_k} \\ d\omega_{\sigma_{\theta_k}} \\ d\omega_{\sigma_{\dot{\theta}_k}} \\ d\omega_{\phi_k} \\ d\omega_{\psi_{i_k}} \\ d\omega_{\sigma_{\phi_k}} \\ d\omega_{\sigma_{\psi_{i_k}}} \\ 0 \end{bmatrix}. \tag{6.47}$$

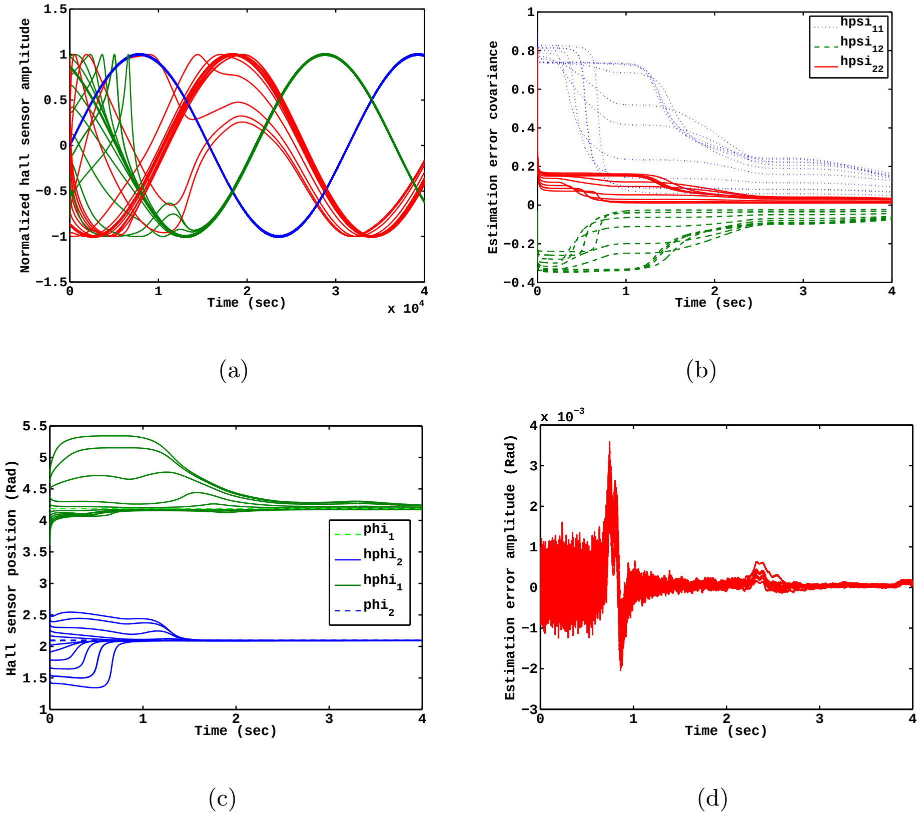
The variable phase ϕ can be omitted from this part of the estimation and held constant, tuned during a startup procedure, or allowed to drift only slowly, as it is mainly a torque maximizing variable. Thus if all the other parameters converge to the correct value, and ϕ is slightly non-optimized, the motor will function. It merely will not function at peak torque capability. If the value moves outside of about $\pi/2$ Radians, the torque generated in the BLDC motor will swap sign, and excessive current flow and coil temperature increases will occur. Clearly this is undesirable, so in application, numerical checks and resets are coded into the motor driver algorithm to prevent such problems.

Results

The performance of the estimation in simulation is shown in Figure 6.43 for multiple initial velocities. Since the parameters should drift slowly, a period of time is required initially for convergence, after which the values change little. This can be achieved by a simple startup pause time where the motor parameters wait for convergence. Current estimates can be saved for the next power-up of the motor driver in order to facilitate more rapid initialization.

As can be observed, clearly the estimator is capable of determining unknown placement parameters, maximum and minimum Hall sensor values, angular position, and angular velocity. The values of the parameters converge in approximately two seconds. As with all EKF-based estimators, convergence rate is determined by filter parameters such as the covariance of the noise, however the approximate nature of the extended kalman filter can be sensitive to filter parameter adjustments in a stability sense.

As can be seen in Figure 6.43(a), from any one Hall sensor, all three measures can be reconstructed using the mean of the EKF parameter estimates effectively.



(a)

(b)

(c)

(d)

Figure 6.43: (a) Shows the actual vs. estimated Hall sensor measurements for several random initializations of the parameter estimates. The estimates quickly converge to the correct value and track well during this constant velocity test. (b) Shows the convergence of the error covariance values, again for various initializations. (c) Shows the convergence of the mean estimates for ϕ for several initial values of the parameters. (d) Shows the Estimation error amplitude for the position of the BLDC motor's rotor.

6.4 Control of a modular robot with 3N degrees of freedom using decentralized PID combined with FAS controllers

6.4.1 Introduction

Control of a robot leg or finger consists of several important issues. Many of these are only defined and partially addressed in this dissertation, as the focus has been development of the underlying formalism required for the next steps of controlling the robot in more than just a trajectory sense. We begin by developing the kinematic equations describing the ModBot manipulators, then move to a derivation from first principles of the dynamic equations in 3D. Finally results are presented for our robot system based on simulation and real data for experiments involving set point and trajectory tracking, compliance, and passivity (actual as opposed to actively created passivity).

Future work will include validating models with open loop system identification techniques. Experiments will be performed to validate the applicability of the FASC to learning problems in the context of this robot. The robot will have to discover kinematic or dynamic parameters in order to achieve a task with changing goals. This FASC has already been validated for learning problems in Chapters 3 and 4. Once validated, the robot system will be used for manipulation and locomotion experiments.

6.4.2 Kinematics

Kinematics is defined as the science of the geometrical description of the motion of objects by their position and orientation, and the time derivatives thereof (velocity and acceleration).

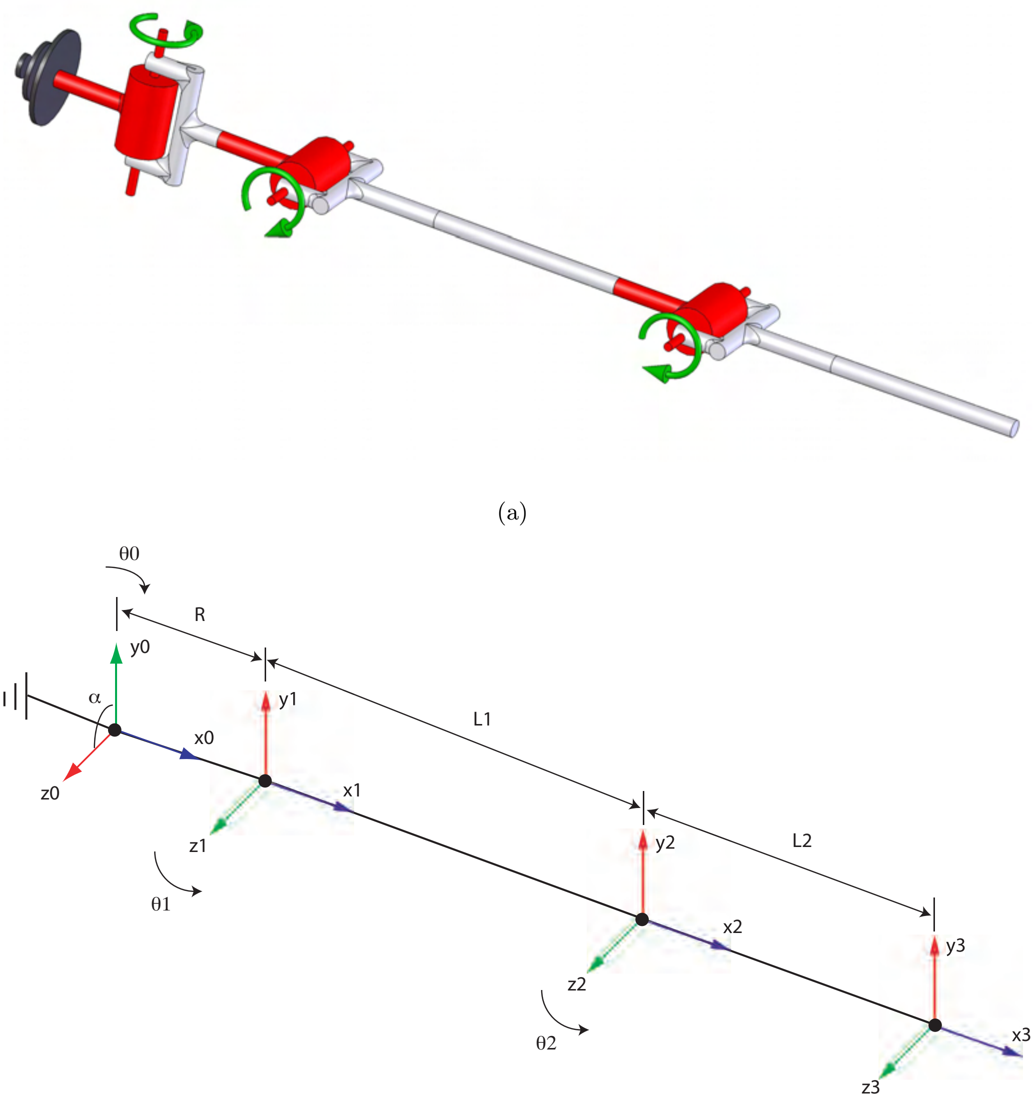
This method relies heavily upon rigid body motion, and is very common in engineering and robotics. It essentially consists of finding a transformation between each link's local reference coordinate system, and a global coordinate system. These transformations can be linked together to form a chain, and thus to describe a local coordinate point (the tip of the robot finger, for example) in the global coordinate system. Therefore, when one specifies that the robot should point to a target, strike a soccer ball, or move to a particular place, the control algorithm will know where it is and where it needs to be when it reaches its goal.

The *forward kinematics* problem in robotics is defined as the problem of determining the position of the end-effector in global coordinates (B_0) given the joint variables (such as joint angles in a rotational joint).

The *inverse kinematics* problem in robotics is essentially the opposite. Given the position of the end-effector in global coordinates (B_0), determine the joint variables required to make the end-effector reach that point. It should be intuitive that the inverse problem is not generally unique, and some algorithms use iterative methods to converge upon one set of possible joint variables to solve the inverse problem. However, sufficient care must be taken that the computed joint variables are within the constraints that the robot design imposes.

Structure-varing kinematic chains

Though the coordinate system can be represented in many ways, some simpler than others, the key issue here is that, during manipulation and locomotion we have so-called structure varying kinematic chains[reference Katsu Yamani - Simulating and generating motions of human figures]. For example, when a hand is coupled to an object, the kinematic structure of the chain linkage changes, which can lead to



(a)

(b)

Figure 6.44: (a)Kinematic representation of the robot leg. The robot is a 3R manipulator. The green arrow symbols represent rotation axes, and the black symbol represents a ground constraint. (b) Coordinate frames of a ModBot leg. Note that the base frame (0) is rotated an angle $\alpha = 90^0$ with regards to frame 1. The third frame is the endpoint, thus there is no θ_3 since there is no rotational degree of freedom at the endpoint.

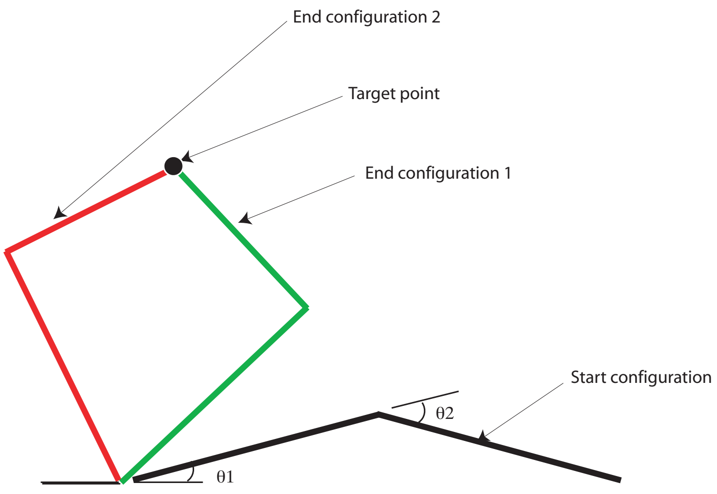


Figure 6.45: Given a target, there may be multiple configurations, and thus paths to attain that goal. Here we see two possible final configurations to reach a target point - green is one configuration, and red is another.

computationally expensive algorithms switching between control approaches. Additionally, collisions and contacts are still an open area of research, since those lead to high frequency and impulsive forces, as well as indeterminate constraint forces. This will be a focus for future research with the ModBot. Presently, the initial control of the robot is achieved using the following method based on the kinematics of an open chain. This will also demonstrate that the ModBot can be controlled using classical or modern models of sensorimotor control and learning.

Open chains

The typical industrial robot arm can be controlled to move in a particular way by the following sequence of computations, and we begin by controlling the ModBot in the following fashion:

1. Compute the current position by measuring the current joint angles, then use a generalized transformation to compute the end effector position (forward kinematics).

2. The desired end point is given, and to compute the required joint angles (which may not be unique), a simple function minimization is employed in order to solve the inverse kinematics problem (such as Newton-Rhapson iteration).

3. Now that the final joint variable values are known, we can compute either by linear interpolation or optimal trajectory the desired trajectory of the robot arm.

4. From the desired trajectory of the arm a desired position, velocity, and acceleration can be found whatever the present state is.

5. Using a model for the dynamics of the system, the required torques can be computed and injected into the system to move it to the desired location. This is where there is freedom in the design. We will consider (1)computed torque control for each joint based on a nominal model, and (2)applying our FAS controllers to coordinate the entire robot to achieve an optimal objective. In this second case it is possible to include a learning component.

Determining the forward position kinematic (FPK) transformations

The standard generalized transformation for a rigid link from link i to link i-1 in DH-coordinates is

$$ {}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\cos(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.48} $$

Thus the transformation from the local coordinate system of any particular link to the base coordinate system can be found by multiplying the local point coordinate (such as an end effector position) by as many of these particular transformations as needed to transform into the base coordinates.

DH-notation follows a simple set of rules for determining the four required parameters to define a link[44]:

1. The z_i axis is in the direction of the joint axis
2. The x-axis is defined to be parallel to the common normal between links (or $x_i = z_{i-1} \times z_i$
3. The y-axis is determined from the x and z axis, as it is chosen to be a right handed coordinate system.

Then each joint and link pair can be described by a coordinate transformation from the previous to next coordinate system by

$$ {}^{i-1}T_i = R(z_{i-1}, \theta_i) \cdot T(z_{i-1}, d_i) \cdot T(x_i, a_i) \cdot R(x_i, \alpha_i) \tag{6.49} $$

where each matrix is defined in the following way:

$$R(z_{i-1}, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.50}$$

$$T(z_{i-1}, d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.51}$$

$$T(z_i, a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.52}$$

$$R(x_i, \alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.53}$$

Which then form (6.48) when substituted into (6.49). Therefore the transformation from link-local coordinates to base coordinates is

$$^{0}T_i = \prod_{k=1}^{i} ({}^{k-1}T_k). \tag{6.54}$$

It is convenient to represent the parameters for a robot arm in a so-called 'DH-table.' The DH-table for an individual modbot leg is

Table 6.1: A table of Devanit-Hartenberg parameters defining the modbot 3 link leg.

Frame-num	d	a	α	θ
1	r	l_a	$\pi/2$	θ_1
2	0	l_b	0	θ_2
3	0	l_c	0	θ_3

Problems with rotations - singularities

There are two properties of rotations which can cause serious numerical problems:

1. Rotations are non-commutative
2. Spatial rotations do not allow a mapping in three dimensional Euclidean space which is topologically smooth

The first issue is resolved by maintaining the order of rotations during calculations.

The second issue means that, using one set of three numbers, it is impossible to represent every kind of rotation smoothly. There is at least one orientation where the coordinates are singular (at least two of the numbers are nonunique or undefined)[44]. Consider the system of lattitude and longitude which describes one's location on the earth's surface. At the two poles one may make small changes to their position and bring about massive changes in longitude. This is due to the fact that the coordinate system is a plane wrapped about a sphere - singularity is unavoidable using this method. One solution to this problem is to use a coordinate system based on four numbers (such as quaternion notation). The other is to simply be aware of singular points and avoid them. Engineers typically prefer to work with minimal numbers and so use a three number system (such as a rotation matrix based on sines and cosines).

6.4.3 Inverse position kinematics (IPK)

The inverse position kinematics problem is in general a more difficult problem than the forward position kinematics problem because there may be no solution, many solutions, or one solution. It is possible in some cases of robot structures that a closed form solution to the inverse kinematics problem exists. One must use iterative techniques when a unique solution does not exist or is difficult to determine due to computational complexity (high degree of freedom robots). There are many iterative techniques to solve inverse kinematics problems. In general, these techniques start with an initial sensible guess for the goal joint variables, such as zero, and then attempt to iteratively converge upon one set of the correct values. One such common technique is the Newton-Rhapson technique for minimizing a nonlinear function $T(\theta) = 0$ for θ.

The algorithm implemented for solving this particular IPK problem can be written in the following algorithmic form.

Algorithm : IPK estimation by Newton-Rhapson

1. $i = 0$.

2. *Make an initial guess for the joint variables,* θ^0.

3. *Compute the residual,* $\delta T(\theta^i) = J(\theta^i)\delta\theta^i$.

Check whether $||\delta T|| < \epsilon$ *(or the tolerance selected is* $< \epsilon$*). If it is, terminate the algorithm - the angles are sufficiently close to the answer.*

4. *Compute* $\theta^{i+1} = \theta^i + J^{-1}(\theta^i)\delta T(\theta^i)$.

5. $i = i + 1$*, go to (3).*

Figure 6.46(a)-6.46(c) depicts the inverse kinematics capabilities of this algorithm. Three sets of plots are shown where the robot begins each trial in the same posture, and a random target is generated. The algorithm computes one possible set of joint angles to solve the problem and then the command trajectory is computed by interpolation. Smooth velocity profiles could be computed with cubic spline interpolation with endpoint constraints. Here the demonstration is mainly for the inverse kinematics problem, so velocity profiles will ultimately be computed in an online fashion when the controller is implemented.

6.4.4 Dynamics

In order to generate the appropriate torques to cause the ModBot to follow the desired trajectory, or reach the set point and remain close to it, a good model of the dynamics of the system must be developed. There will be inevitable model errors (differences between the model and true system), and therefore errors must be corrected in feedback to improve open-loop performance.

We will begin by deriving the equations of motion in three dimensions for a serial n-link manipulator using transformation matrices, starting from the Lagrangean. The model will be controlled in simulation using proportional derivative (PD) feedback control, and compared with performance of the FAS controller.

The Lagrangean is defined as the difference between potential (PE) and kinetic energies (KE)[44][6]:

$$\mathcal{L} = KE - PE \tag{6.55}$$

We determine the dynamics by applying the Lagrange equation

$$\frac{d}{dt}\Big(\frac{\partial L}{\partial \dot{q}_i}\Big) - \frac{\partial L}{\partial q_i} = Q_i, i = 1, 2, ...n \tag{6.56}$$

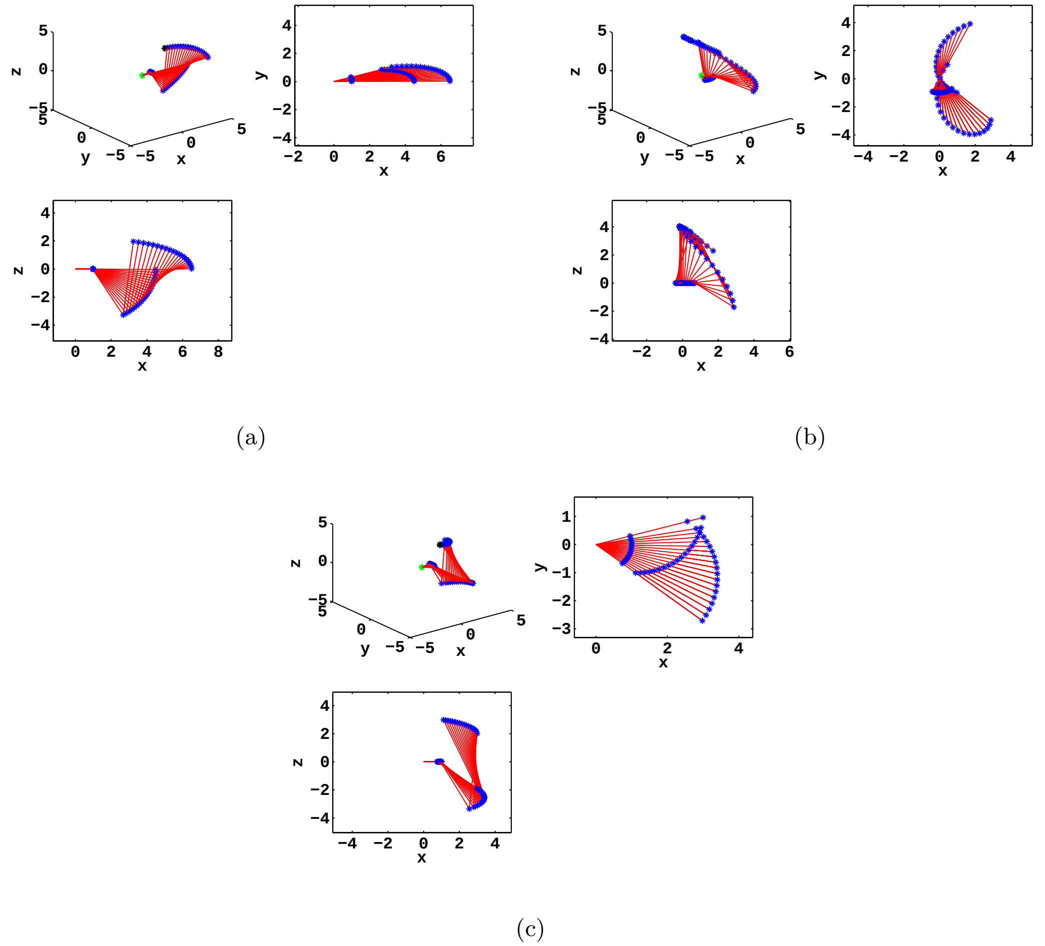


Figure 6.46: Demonstration of the Newton-Rhapson algorithm computing the target joint angles, then the robot interpolates between start and target points to create a trajectory. Additional perspectives are provided for clarity.

where q_i are the coordinates of expression of the energies driven by Q_i, the generalized nonpotential force.

The Kinetic energy of link i is written as

$$KE_i = \frac{1}{2}({}^o v_i^T) m_i ({}^o v_i) + \frac{1}{2}({}^o \omega_i^T)({}^o I_i)({}^o \omega_i) \tag{6.57}$$

where m_i is the link mass, ${}^i I_i$ is the moment of inertia matrix of the link in its from B_i, ${}^o v_i$ is the global velocity of the center of mass of the link, and ${}^o \omega_i$ is the link global angular velocity.

We express the translational and rotational velocity vectors of link i using the Jacobian of the link[44],

$$\dot{X}_i = \begin{bmatrix} {}^o v_i \\ {}^o \omega_i \end{bmatrix} = \begin{bmatrix} J_{D_i} \\ J_{R_i} \end{bmatrix} \dot{q} = J_i \dot{q}. \tag{6.58}$$

Given that I_i is the inertia matrix of link i about its center of mass, and is expressed in the base frame as

$$({}^o I_i) = ({}^o R_i)({}^i I_i)({}^o R_i^T), \tag{6.59}$$

The KE for the entire ModBot (without contributions due to the rotor inertia of the motors),

$$\begin{aligned} KE &= \sum_{i=1}^{n} KE_i \qquad (6.60) \\ &= \frac{1}{2}\sum_{i=1}^{n}\Big(({}^{o}v_i)^T m_i({}^{o}v_i) + ({}_{o}\omega_i^T)({}^{o}I_i)({}_{o}\omega_i)\Big) \\ &= \frac{1}{2}\sum_{i=1}^{n}\Big((J_{D_i}\dot{q}_i)^T m_i(J_{D_i}\dot{q}_i) + (J_{R_i}\dot{q}_i)^T({}^{o}I_i)(J_{R_i}\dot{q}_i)\Big) \\ &= \frac{1}{2}\dot{q}_i^T\left(\sum_{i=1}^{n}\Big(J_{D_i}^T m_i J_{D_i} + J_{R_i}^T({}^{o}I_i)J_{R_i}\Big)\right)\dot{q}_i. \end{aligned}$$

Define the inertia manipulator matrix D as

$$D = \sum_{i=1}^{n}\Big(J_{D_i}^T m_i J_{D_i} + J_{R_i}^T({}^{o}I_i)J_{R_i}\Big). \qquad (6.61)$$

We can rewrite the inertia in a more compact form as

$$KE = \frac{1}{2}\dot{q}_i^T D\dot{q}_i. \qquad (6.62)$$

The PE is defined as the sum of the potential energies of all links, with $({}^{o}g)$ the gravitational acceleration vector relative to the base frame,

$$\begin{aligned} V &= \sum_{i=1}^{n} V_i \qquad (6.63) \\ &= -\sum_{i=1}^{n} m_i({}^{o}g_i^T)({}^{o}r_i). \end{aligned}$$

The Lagrangean of the manipulator is

$$\begin{aligned}\mathcal{L} &= KE - PE \\ &= \frac{1}{2}\dot{q}_i^T D\dot{q}_i + \sum_{i=1}^{n} m_i({}^o g_i^T)({}^o r_i) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}\dot{q}_i\dot{q}_j + \sum_{i=1}^{n} m_i({}^o g_i^T)({}^o r_i)\end{aligned} \tag{6.64}$$

From that Lagrangean,

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial q_i} &= \frac{1}{2}\frac{\partial}{\partial q_i}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}\dot{q}_i\dot{q}_j\right) + \sum_{j=1}^{n} m_j({}^o g^T)\frac{\partial({}^o r_j)}{\partial q_i} \\ &= \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial D_{jk}}{\partial q_i}\dot{q}_j\dot{q}_k + \sum_{j=1}^{n} m_j({}^o g^T)J_{D_j}^{(i)}\end{aligned} \tag{6.65}$$

$$\frac{\partial\mathcal{L}}{\partial\dot{q}_i} = \sum_{j=1}^{n} D_{ij}\dot{q}_j \tag{6.66}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{q}_i} &= \sum_{j=1}^{n} D_{ij}\ddot{q}_j + \sum_{j=1}^{n}\frac{dD_{ij}}{dt}\dot{q}_j \\ &= \sum_{j=1}^{n} D_{ij}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial D_{ij}}{\partial q_k}\dot{q}_k\dot{q}_j.\end{aligned} \tag{6.67}$$

Here the generalized force, including an external force applied on the end-effector $F_e = \begin{bmatrix} -F_{en}^T & -M_{en}^T \end{bmatrix}^T$, and the force at actuator i, Q_i,

$$Q_i = Q_i + J^T F_e. \tag{6.68}$$

Define

$$H_{ijk} = \sum_{j=1}^{n} \sum_{k=1}^{n} \Big(\frac{\partial D_{ij}}{\partial q_k} - \frac{1}{2} \frac{D_{jk}}{\partial q_i} \Big) \tag{6.69}$$

and

$$G_i = \sum_{j=1}^{n} m_j g^T J_{D_j}^{(i)}. \tag{6.70}$$

We can write the equations of motion for the ModBot as

$$\sum_{j=1}^{n} D_{ij}(q)\ddot{q}_j + H_{ijm}\dot{q}_k\dot{q}_m + G_i = Q_i \tag{6.71}$$

which can be written, for clarity, in the following form,

$$D(q)\ddot{q} = H(q, \dot{q}) + G(q) = Q. \tag{6.72}$$

These are the equations of motion for a general n-link serial manipulator. The first term represents the inertial forces, the second represents the coriolis and centrifugal forces, and the third represents gravitational forces. These forces are balanced by the forces applied to the manipulator by both joint forces due to actuators and externally applied forces.

In the case of the ModBot, the forces at the joints are torques applied by actuators, thus the generalized force is given by $Q_i = \tau_i + J^T F_e$. If we work with joint angles as the coordinate system of interest (since this more directly relates to controlling joint torques), the dynamic equations of motion become

$$D(\theta)\ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta) = \tau + J^T F_e. \tag{6.73}$$

6.4.5 Naive feedback control for position control and testing

It is important to mention naive control, even if it has theoretical and practical limitations. Most industrial control is performed with some form of P, PI, PD, or PID control for two simple reasons - 1) it works well for position control, and 2) it is very easy to implement[76]. No model of the system dynamics are required, only the following equation with three parameters is required (for each actuator), and a measurement of actual positions, velocities, accelerations (or estimates of some of those quantities given measures of others) so that an error can be computed. This is referred to (as a result) as *naive* feedback control. It is useful in our case for testing the ModBot system by performing position control tasks. Once the mechanisms and electronics are tested, more useful control algorithms regarding sensorimotor learning and control will be applied. The equation for a PID controller is defined (for $j = 1, ...n$ and $\tilde{q} = q_j - q_{d_j}$, q_{d_j} is the desired position, and the k's are positive constant 'gains') by

$$\tau_j = -k_{jD}\dot{\tilde{\theta}}_j - k_{jP}\tilde{q}_j - k_{jI}\int_{s=0}^{t}\tilde{q}_j ds. \tag{6.74}$$

The first term is the derivative term, the second is the proportional term, and the third is the integral term. Combinations of some of these three terms can also be used for naive feedback control. Additionally, the integral term should have some method of integral anti-windup to prevent this term from diverging. PID (and its alternative formulations) control has been studied extensively and there is a wide literature supporting these methods.

6.4.6 Basic model-based control design with local feedback control for trajectory tracking performance

The above control scheme can be improved by including the dynamical model as part of the control - essentially combining an open loop component (given our model and where we are, what torque would we need to ideally inject to move the system along a prescribed trajectory), and a closed loop component (given the error between desired and actual trajectory states, inject a correction to the open loop control to make behavior more ideal). The error is given by

$$e = \theta - \theta_d, \tag{6.75}$$

$$\dot{e} = \dot{\theta} - \dot{\theta}_d. \tag{6.76}$$

Define the control law τ, by combining the controller computed with open loop dynamics, τ_d with a PD controller,

$$\tau_{ol} = D(\theta_d)\ddot{\theta}_d + H(\theta_d, \dot{\theta}_d) + G(\theta_d), \tag{6.77}$$

$$\tau = \tau_{ol} + k_D\dot{e} + k_P e. \tag{6.78}$$

In the case that the model captures precisely the exact dynamics of the system (which is generally not the case in reality due to friction, model errors, actuator saturations, noise, and un-modeled high frequency resonance modes), the acceleration is directly controllable, and the system could be controlled perfectly to follow a desired trajectory.

6.4.7 FAS controller for a modular robot with three links and three degrees of freedom

In the case of uncertainties which are not necessarily exposed by attempting to perform the control actions, this scheme would clearly fail. One can consider robust control where parametric uncertainties are taken into account when designing the controller[42][112]. However, in cases where some of the task parameters are unobservable, and may be large and time varying, a control design which must actively explore to expose the uncertain parameters is needed (especially if off-line estimation is not reasonable, such as in biological systems), while attempting to meet a control objective.

Therefore we will in the future consider the active exploration/exploitation FAS controller designed in chapter 3.

6.5 Robot performance

In this section the characteristics of the robot are compared against its design specifications.

6.5.1 Passivity

Passivity will be defined as a characteristic of a system which allows the world to 'push back' on the system. The level of passivity is measured by the amount of force required to move the system by an external application of energy. System properties which affect passivity are inertia and friction of the structure and joints, as well as the backdrivability of the actuation system.

For example, consider a standard CNC mill used in industrial fabrication. When the mill is not powered, and assuming the brakes are not applied, it would be very difficult to move one of the machine's degrees of freedom by applying a force at the endpoint of the milling bit. That is because of the friction in the drive system and backlash - the amount of force required to counteract friction in the drive train would have to be very large, and backlash in the gears may make it impossible to backdrive the system without damaging the gears. This is an example of low passivity by low backdrivability.

Consider again the CNC mill of the previous example. Let the reader imagine that the drive system has been replaced by an advanced frictionless no-backlash cable drive of some sort. The system will still have low passivity because this CNC machine, in order to be stiff and dampen high frequency oscillations which would result in poorly produced parts, is intentionally designed to be extremely massive and possess high inertia. Now it is possible to move the endpoint by applying external loads at the endpoint, but the high inertia inherent in the system would cause, again, large forces in order to bring about accelerations of the system.

Finally, the reader is invited to consider his or her fingers. Use one hand to push on one of your fingers while relaxing the finger to which you are applying the load. You will find that, depending on your level of control over your body, your finger poses little resistance to the application of loads. It moves in a compliant manner. Biological actuation systems tend to be coupled in complicated ways, but even though the large muscles in your forearm are coupled with your finger, you experience little inertia, friction, or backlash when your finger is passively moved around. This is yet another example of a system which has been engineered by nature with such intricate perfection in final design that human-engineered approximations cannot, at this point in time, hope to match (in this case in terms of passive dynamics). However, engineered systems possessing passive dynamics do not need to be 'better than' nature's systems, as an engineered system possesses its own design. Engineered systems need only possess required levels of passivity for the application (e.g. - in the case of a CNC mill passivity is undesirable and should be minimized, but in the case of a robot finger used to perform learning and to sensitively touch and expore objects, passivity is desirable and should be maximized).

The passive dynamics of the robot developed in this work are quite clear from Figures 6.47-6.50. An external disturbance acting on the robot is directly felt at the actuators (BLDC motors). This also means that a robot can make use of passivity for exploration- exploitation task purposes in simple ways. For example, if one wished to have a robot hand that could manipulate an unknown object while learning about its mass and structural properties, the ModBot could provide a basis

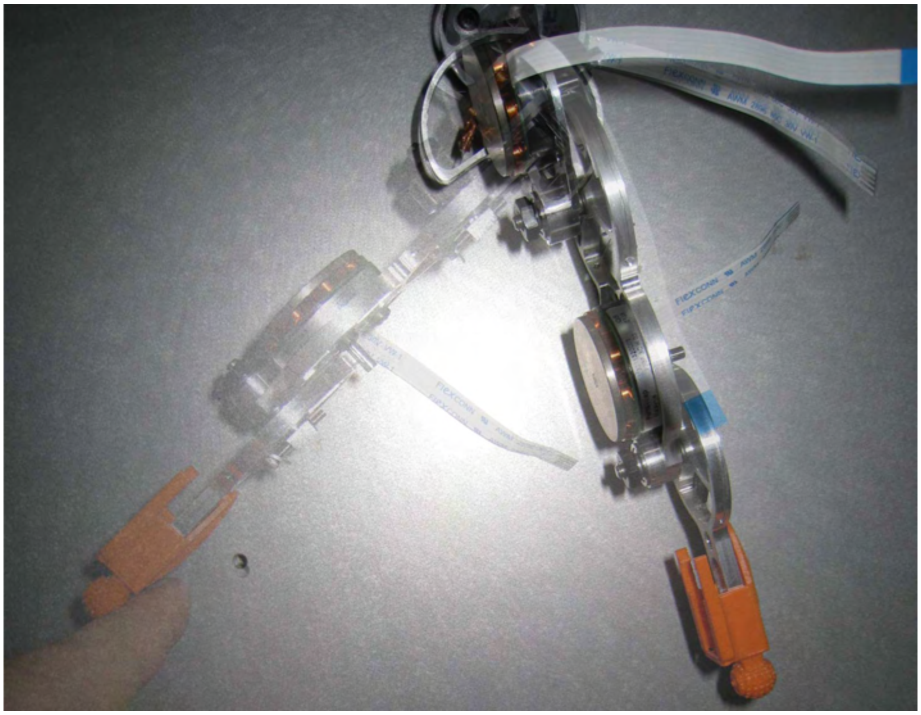


Figure 6.47: The robot's backdrivability allows the external world to affect the state of the robot; a must for manipulation tasks, and helpful for bio-mimetically realistic locomotion tasks.

for each finger. This type of device would also work well for compliant tasks where a task reference must be tracked but with some flexibility, such as pushing a peg into a hole with the hole location and peg shape initially unknown.

6.5.2 Basic controlled movement

The control design implemented in the robot is very basic for these tests - decentralized proportional control. These tests consist of trajectory tracking and set-point step impulse response. The angular position measurement is shown for a

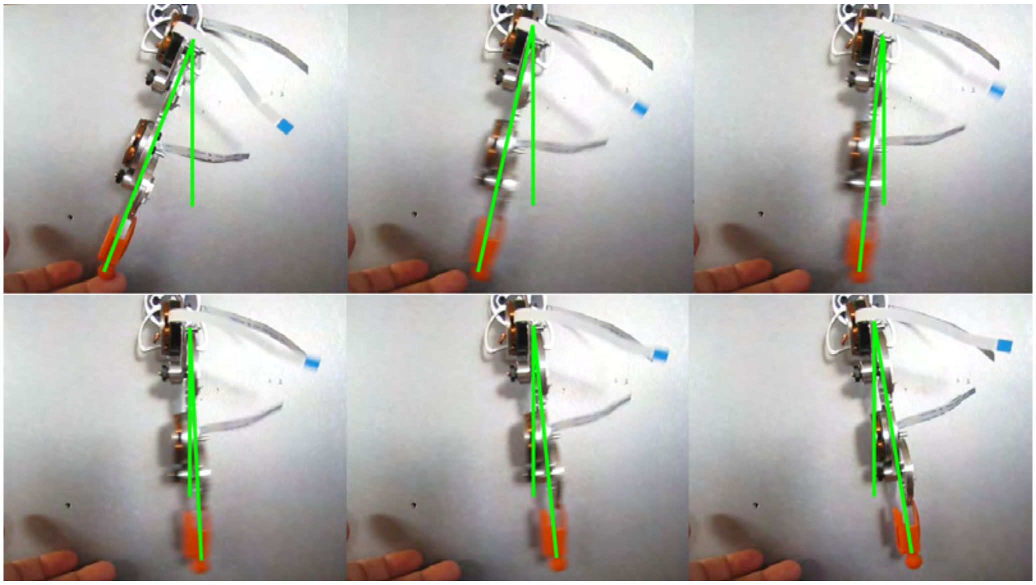


Figure 6.48: In this swing test the system can be seen to start from a raised angle in θ_0, then when released it swings to an opposing angle. A robot without passive dynamics would not be able to achieve this in a simple way.

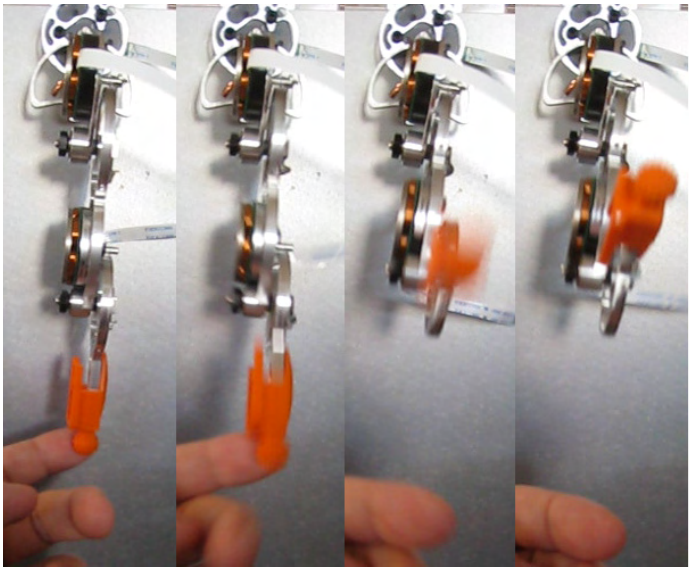


Figure 6.49: The passive dynamics of the robot are again displayed but this time along the first and second axes by a disturbance bringing about a large joint rotation in θ_1 and θ_2.

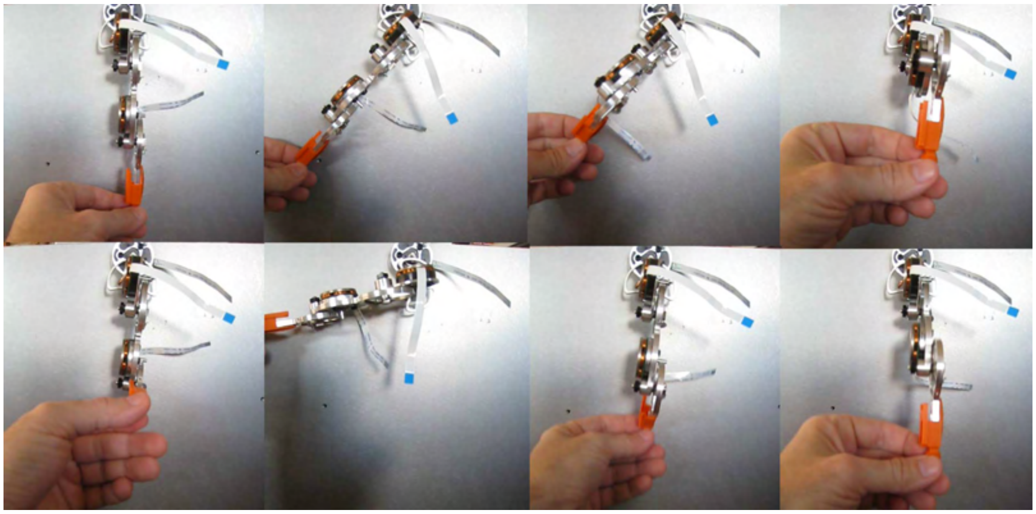
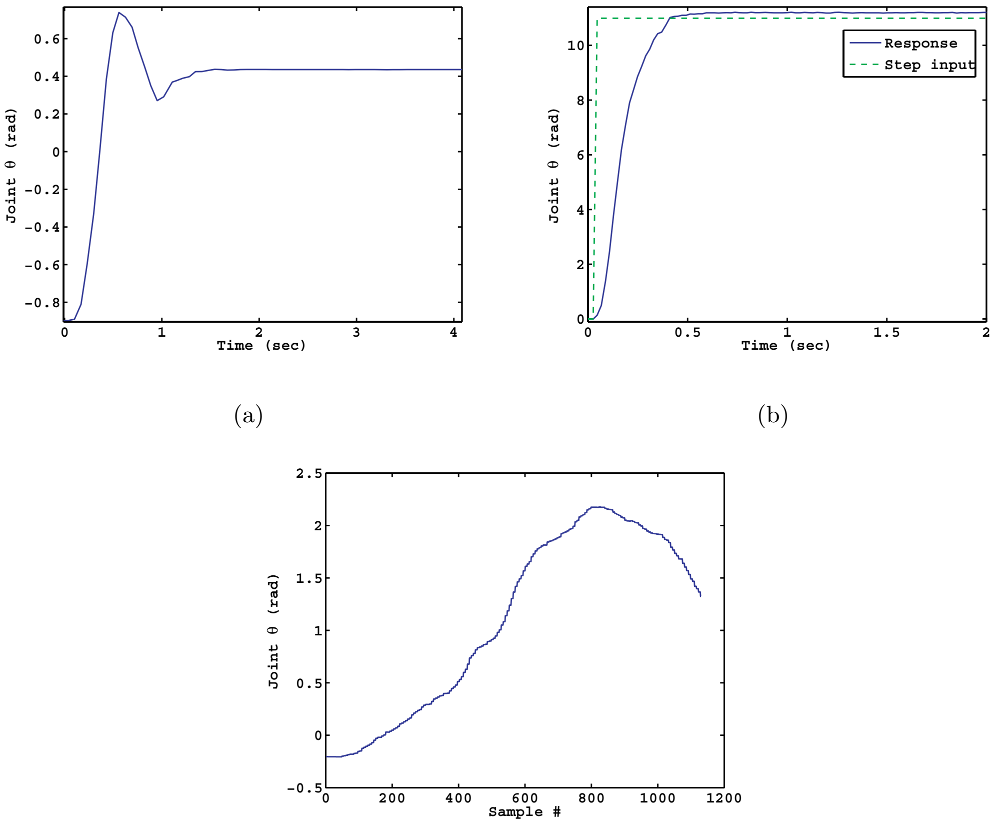


Figure 6.50: Here the robot is passively moved through a subspace of the workspace by a small external load (the human fingers pushing it gently).

random trajectory input from a human-controlled reference graphical user interface (GUI) program. The robot's passivity is discussed in the previous section, while here we demonstrate that the robot is compliant. Figure 6.51 shows that the robot, when under command to get to a particular target, can be backdriven by an external disturbance. In addition, it is worth noting that the controller maintains stability.

Acknowledgment

The material from this chapter is, in part, submitted for publication in the *IEEE Conference of Robotics, Science, and Systems, 2009*, by Simpkins, Alex, Kelley, Mike, and Todorov, Emanuel. The dissertation author is the primary researcher and author in this work. The second co-author listed in this publication directed and supervised the research which forms the basis for this chapter and Mr. Kelley was



(a)

(b)

(c)

Figure 6.51: (a) Step response with highly compliant (low gain) P controller. (b) Step response with moderately high gain (less compliant) PD controller. Note the steady state error - for precision tracking a PID may be more appropriate. (c) Trajectory of θ_3 when pushed externally by a human experimenter. The estimator is capable of reconstructing the position of the leg from Hall sensor measurements of the motor with a gearing ratio of approximately 16:1. Note the subtleties of the operator's movement captured by the measurement. This leg can also be used as part of haptic experiments and for virtual reality interaction, since the human can act on the robot, and the robot can act back on the human.

a research associate who participated in part of the development of the work in this publication.

Chapter 7

Additional results and future work

7.1 Introduction

In order to measure hand manipulation tasks, it is important that subjects have complete unhindered natural movements. At the same time, it is important to measure all joint angles (or at least the position of contacts with an object being manipulated). This could be done by optical means but there are many issues with occlusions. In other words, it may not be possible to have enough cameras to optically track every finger's end point on a surface being manipulated. Another important point is that one would ideally want to measure applied forces since grip force is such an important aspect of handling and manipulating objects in the world. Optical motion tracking lacks the capability for force measurement. A more simple approach is to instrument an object, and combine that information with information from a few optical tracking cameras. Such an instrumented object is not available commercially, and so the author designed and constructed one, using a novel approach for finger contact force and location measurement. Ultimately other versions will be developed with higher resolution, and the integration of 'painted' MEMS resistive matrices, allowing a more direct measure of contact position. However, at present such technology is not available, though the author is in communication with laboratories developing such materials science.

Here a device design methodology is presented using conventional FSR sensors to instrument objects and make them touch and force sensitive. Preliminary results validate the method for extracting position and force simultaneously from a surface which has been touched.

7.2 The wireless manipulation module: preliminary version

The wireless sensor device incorporates an accelerometer to directly measure acceleration, and force sensing resistors[41] in a unique configuration which allows an experimenter to measure the locations of each finger, as well as grip force and orientation of the object.

The configuration which allows one to measure finger location is based on a simple concept - for static force balance, an input force is balanced by output forces. Therefore, if a point force is applied to a beam, then the sum of the resultant forces balancing the force is the total applied force, and the proportion denotes position along one axis. By using three sensors, one can recover 2D position of applied force. By making the sensor configurations small enough, a very high resolution plus individual finger positions can be recovered over a surface. This data, along with the accelerometer measurements is read from a microprocessor (dsPIC33FJ256MC710) which streams the data over a bluetooth serial stream at 976kBps. The project is powered by a small light lithium polymer battery, and has all electronics integrated onboard.

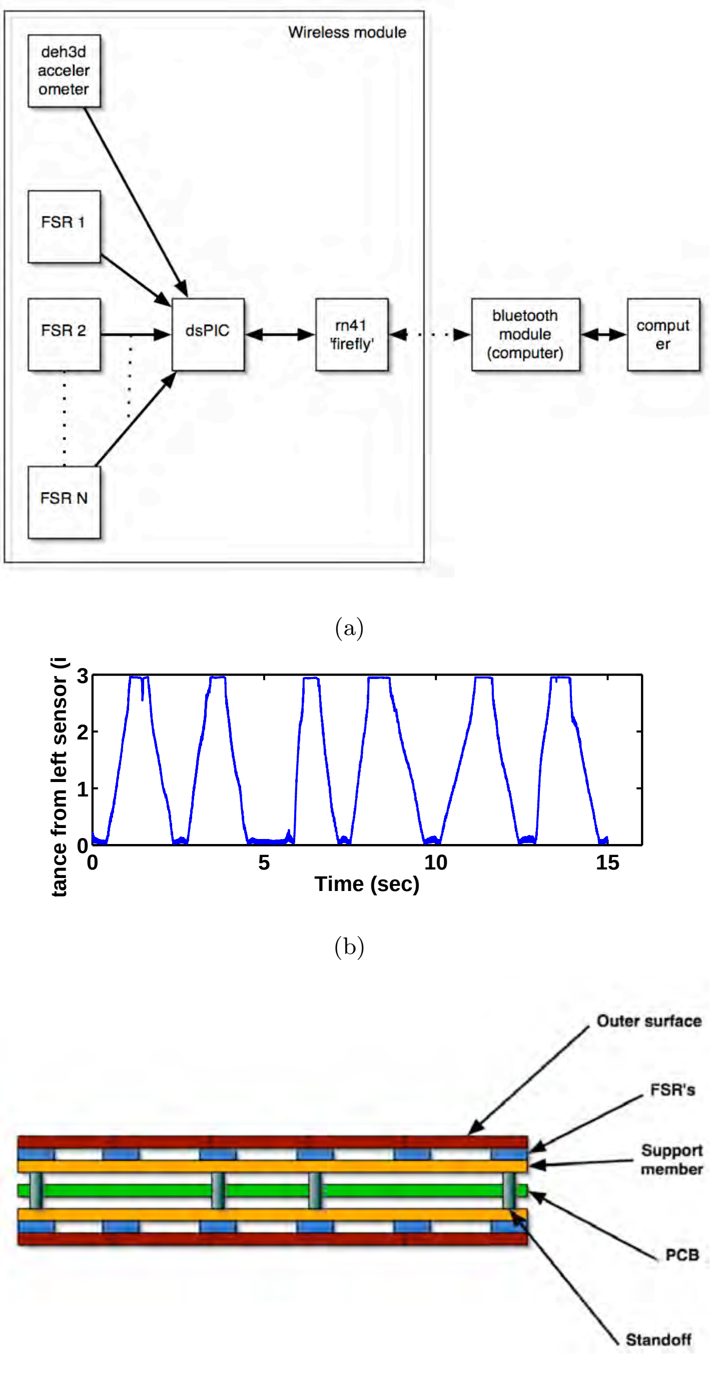
Though the device is completed, only preliminary studies for testing the device have been performed, so future work will use the device for object manipulation studies. The design concept is the key contribution here, being a unique method of measuring the spatial location and amplitude of an applied force.

7.3 FSR's explained

An FSR (Force Sensing Resistor) is a polymer thick film (PTF) device which behaves in such a way that the resistance across their terminals decreases with increasing applied forces. In general, FSR's respond according to an inverse power law characteristic which is roughly $1/R$. The circuit to measure an FSR can be configured to yield maximal information over a particular range of forces via a simple voltage divider into an analog to digital converter built into the dsPIC. A more complex operational amplifier configuration can also be used if desired. This may provide more precise control and proportions than a simple voltage divider circuit may allow. Resistance may vary depending on temperature, pressure, etc of the resistor. However, such variables may be compensated for with an active circuit. For initial tests, a basic voltage divider is sufficient to determine characteristic curves of the FSR's, repeatability, precision, and other desired behaviors.

7.3.1 Position and force extraction

Theorem 7.1 *The total applied force (F) and position of the applied force on an object can be determined by considering the combination of the total resultant forces (R_r^i, where r is the force type, and i is the unique force) and the proportions of resultants to each other, given the correct number of sensors measuring the resultant forces (depending on the shape, a minimum of two sensors can determine position in one axis, and three sensors for two axes).*

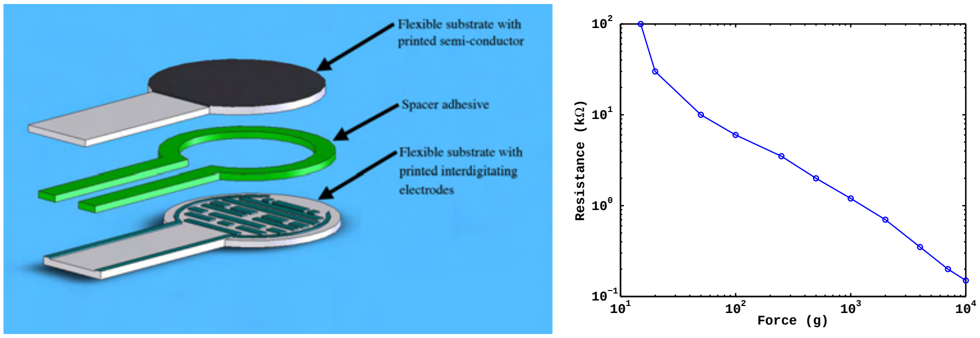


(a)

(b)

(c)

Figure 7.1: Basic block diagram of wireless sensor module informational components.


Flexible substrate with printed semi-conductor
Spacer adhesive
Flexible substrate with printed interdigitating electrodes
10^2
10^1
10^0
10^{-1}
Resistance (kΩ)
10^1
10^2
10^3
10^4
Force (g)

(a) (b)

Figure 7.2: (a) Basic component structure of an FSR. (b) Average force-resistance relationship for a typical FSR.

Proof 7.1 *Consider figures (7.3(a)) and (7.3(b)). The total applied force to an object which is in static equilibrium is found by Newtonian mechanics as*

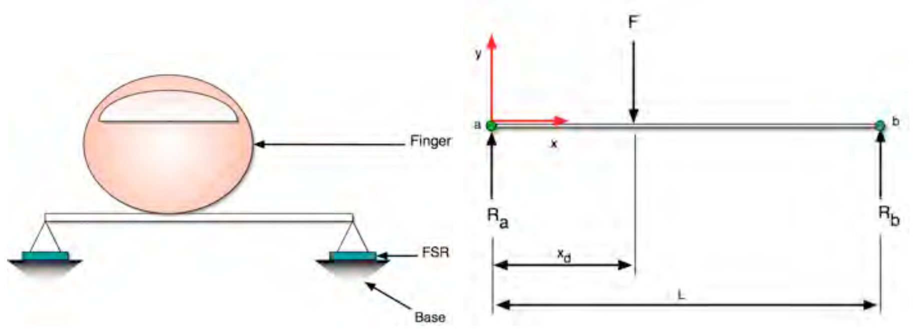
$$F = \sum_i R_i \tag{7.1}$$

which in the case of one degree of freedom and $i = 2$ *is*

$$F = R_a + R_b. \tag{7.2}$$

This can be extended into two dimensions by increasing i *as needed (to three or four). If the shape of the surface to determine 2-d position is square, then, since the FSR's only measure compression, it is necessary that* $i = 4$. *For a triangular geometry,* $i = 3$ *is sufficient to recover force and position information. Position in the x-direction relative to one extremal point at one sensor side, a, is found in the following way:*

$$x = \frac{\sum_y R_{x_a,y}}{\sum_x \sum_y R_{x,y}} L = \frac{R_{x_a}}{R_{x_a} + R_{x_b}} L \tag{7.3}$$



(a) (b)

Figure 7.3: Basic block diagram of wireless sensor module informational components.

where $R_{x,y}$ represents each sensor in x and y, and L is the distance between sensors in the x-axis. The y-direction location is found in the equivalent manner, in the y-axis.

In order to recover grip force versus acceleration forces, by measuring acceleration in each axis in addition to FSR force data, one can subtract appropriately to extract only grip force. Additionally, to avoid 'occlusions' due to multiple fingers contacting one force plate, the distance L must be slightly smaller than a finger width. This results in a reasonable number of analog input channels required to instrument a device of hand manipulation scale.

Preliminary results with the low dimensional prototype demonstrate the capability of this system to relay data wirelessly, measure and estimate force and position of force application.

Chapter 8

Conclusions and recommendations

8.1 Contributions of this dissertation

This dissertation has emphasized the need for, and presented approaches to modeling sensorimotor control and learning in biological systems by augmenting stochastic optimal control with optimal estimation theory. Optimal control theory has been shown in recent years to provide a powerful framework for explaining biological behavior. By studying these complex systems, control theorists and neuroscientists are not only understanding sensorimotor learning, but are also expanding the field of control theory as well.

The main approach to combining optimal control and estimation in this dissertation is to augment the state space with the estimator dynamics, thus producing a higher dimensional but fully observable system. This allows solutions (as well as problem formulations) in cases where uncertainty may play a role in the choice of actions - *explore to learn more or act to achieve a goal (or some mixture of the two)?* Though the resulting control problem is challenging to solve, it has been explained in this dissertation that there are methods which mitigate the curse of dimensionality and can compute solutions even for nonlinear stochastic problems. One method described in this work is function approximation. The resulting controller can be shown to make choices which approximate the choices a human might make in the same task.

The main contributions and results of this dissertation can be summarized as the following:

- Formulation of a framework for sensorimotor learning in biological systems which can be applied in a broad context. This formulation is based on a combination of stochastic optimal control and Bayesian inference.

- Solution methods are developed in this dissertation which demonstrate the ability to solve the high dimensional nonlinear stochastic problems resulting from the above point.

- A new definition for, and evidence supporting visuo-motor hand synergies is presented in this work. This is a complete redefinition of the notion of synergy and clarifies several conflicting results in the field of sensorimotor control and learning.

- New methods based on CCA, PCA, FA, and trajectory measures are formulated and presented along with results for human hand posture matching. These methods are useful for similar exploratory studies in other fields as well, such as engineering analysis.

- Robotic systems and techniques for developing further robotic systems which possess characteristics of backdrivability, passive dynamics, high efficiency, modularity, and minimal mass are elucidated in this dissertation.

- A new method of sensing position with a high degree of accuracy is presented, as well as a method of estimating sensor placement error and correcting for these errors based on optimal estimation.

- Patterns of variability in human hand coordinated control indicate that the noise in behavior is fairly consistent across tasks.

- Variability patterns across human fingers is similar (excluding the thumb which has a unique function), indicating each finger is controlled with the same basic algorithm - the brain has no reason to encode the information differently for different fingers.

- The exploratory studies of human hand postures demonstrate that dimensionality of the control of the human hand is far higher than previously found in any study. The result also supports the minimal intervention principle proposed in [102], that dimensionality depends on the difficulty of the task.

8.2 Recommendations for future work

- The function approximation-based global solutions work well, but another approach which may be appropriate in terms of modeling biological system approaches to sensorimotor control problems is to combine local and global methods.

- Experiments comparing the exploration/exploitation mathematical framework with human subjects will be expanded to include a pool of experts at the particular task, such that there is a lack of uncertainty when it comes to performing the task itself. This may expose the search strategies of human subjects when optimization processes have already taken place to maximize task performance.

- The posture matching project has exposed several interesting synergistic patterns in movement and postural hand control. These results can be applied to artificial systems in simulation and on the robots which we now have developed in order to create coordinated control of high dimensional systems such as a full degree of freedom hand, effectively reducing dimensionality of the state space to control.

- The modular robot is a tool which has been developed in this work, and there are many experiments which can be performed with this robot. These experiments relate to locomotion and manipulation - some are exclusive, and others will combine the two as needed (similar to the way a spider will use its legs for both manipulation and to sense the path ahead, or a human might use his or her fingers to explore a wall and attempt to flip a light switch). The initial control approaches will implement the FAS scheme in hardware, possibly via a reduced order controller. These experiments will also be performed in simulation. Simulation and actual performance will be compared.

Appendix A

Derivation of the stochastic Hamilton Jacobi Bellman (HJB) equation for finite time problems

Consider the minimization of a value function in order to find the optimal control for a stochastic system. Recall that the value function is related to a cost function by the following

$$V(t) = \begin{cases} J_{max} - J & t < t_f \\ V_{min} = \phi[x(t_f), t_f] & t = t_f \end{cases} \tag{A.1}$$

So the maximum value occurs at the minimum cost, which is quite intuitive. Now consider some time interval, $[t_1, t_f]$, $t_0 \leq t_1 \leq t_f$, with the control

$$u(t) = u^*(t) \tag{A.2}$$

in $[t_1, t_f]$ unbounded, where the $(\cdot)^*$ denotes the optimal control. The stochastic system dynamics are given by

$$\dot{x}^*(t) = f\big[x^*(t), u^*(t), t\big] + L(t)w(t) \tag{A.3}$$

The optimal value function is of the form

$$V = \mathscr{E}\Big\{\phi[x(t_f), t_f] + \int_{t_0}^{t_f} \ell[x(t), u(t), t]dt\Big\}. \tag{A.4}$$

The total derivative of the value function with respect to time gives us the extremal points of the function.

$$\frac{dV^*}{dt} = -\mathscr{E}\Big\{\ell[x(t), u(t), t]\Big\} \tag{A.5}$$

Or, taking a series expansion which retains second order terms[94], we can express the incremental change in V^* by

$$\begin{aligned} \frac{dV^*}{dt}\Delta t &= \mathscr{E}\Big\{\frac{\partial V^*}{\partial t}\Delta t + \frac{\partial V^*}{\partial x}\Delta t + \frac{1}{2}\Big[\dot{x}^T\frac{\partial^2 V^*}{\partial x^2}\dot{x}\Big]\Delta t^2 + ...\Big\} \\ &= \mathscr{E}\Big\{V_t^*\Delta t + V_x^*(f + Lw)\Delta t + \frac{1}{2}(f + Lw)^T V_{xx}^*(f + Lw)\Delta t^2\Big\} \end{aligned} \tag{A.6}$$

$$\begin{aligned} \mathscr{E}[w(t)] &= 0 \\ \mathscr{E}[g(x)] &= g(x) \end{aligned} \tag{A.7}$$

where $g(x)$ is any of the functions of x. Then, dividing by Δt, (A.6) becomes

$$\begin{aligned} \frac{dV^*}{dt} &= V_t^* + V_x^* f + \frac{1}{2}Tr\Big\{\mathscr{E}[(f + Lw)^T V_{xx}^*(f + Lw)]\Delta t\Big\} \\ &= V_t^* + V_x^* f + \frac{1}{2}Tr\Big\{\mathscr{E}[V_{xx}^*(f + Lw)(f + Lw)^T]\Delta t\Big\} \end{aligned} \tag{A.8}$$

Note f and w are uncorrelated.

$$\begin{aligned} \frac{dV^*}{dt} &= V_t^* + V_x^* f + \lim_{\Delta t \to 0}\frac{1}{2}Tr\Big\{V_{xx}^*[\mathscr{E}(ff^T)\Delta t + L\mathscr{E}(ww^T)\Delta t]\Big\} \\ &= V_t^* + V_x^* f + \frac{1}{2}Tr\Big\{V_{xx}^* LWL^T)\Big\} \end{aligned} \tag{A.9}$$

The stochastic principle of optimality is found by combination of equations (A.9) and (A.5)

$$V_t^* = -\min_u \left\{ \ell[x(t), u(t), t] + V_x^* f[x^*(t), u(t), t] + \frac{1}{2} Tr[V_{xx}^* L(t) W(t) L(t)] \right\} \quad \text{(A.10)}$$

Appendix B

Principal Component Analysis derivation

Introduction

Often, when dealing with multi-dimensional data, one is often interested in knowing which dimensions or combinations of dimensions account for the variability in the data[82][100][63][17]. The problem then is to find a normalized linear combination of variables with maximum variance[3]. (Note in this appendix $(\cdot)'$ represents complex conjugate transpose)

Theorem B.1 *Let the j-component random vector X be mean-centered ($X = X_{raw} - \mu$, $\mathscr{E}[X] = 0$), $\mathscr{E}[XX']$ be the covariance of X. Then there exists an orthogonal linear transformation U such that $\mathscr{E}UU' = \Lambda$, given by*

$$U = \beta' X \tag{B.1}$$

and

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_j \end{bmatrix}, \tag{B.2}$$

with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_j \geq 0$ *the roots of* $|\Sigma - \lambda I| = 0$*. The kth column of* β*,* β_k*, satisfies* $(\Sigma - \lambda_k I)\beta_k = 0$*. The kth principal component of X,* $U_k = \beta_k' X$ *has maximum variance of all normalized linear combinations uncorrelated with* $U_1, ..., U_{k-1}$*.* β_k *is the vector of principal component loadings, and* λ *denotes the variance accounted for by the kth principal component. [3][62]*

Proof B.1 *Let* $X \in \Re^{ixj}$ *be the random vector of* j *variables (if* X *is not mean-centered, then* $X = X_{non\_mc} - \mu$*). Let the covariance matrix of* X *be defined by*

$$\Sigma = \mathscr{E}\big[XX'\big] \tag{B.3}$$

Let β *be a* j *sized column vector of coefficients normalized by*

$$\beta'\beta = 1 \tag{B.4}$$

The variance of $\beta' X$ *is*

$$\mathscr{E}\big[\beta' X\big]^2 = \mathscr{E}\big[\beta' XX'\beta\big] = \beta'\Sigma\beta \tag{B.5}$$

Now we add the constraint (B.4) using Lagrange multipliers:

$$\psi = \beta'\Sigma\beta + \lambda(\beta'\beta - 1). \tag{B.6}$$

In order to maximize ψ*, the partial derivative is taken with respect to* β*, and set equal to zero*

$$\frac{\partial\psi}{\partial\beta} = 2\Sigma\beta - 2\lambda\beta = 0 \tag{B.7}$$

$$\Big(\Sigma - \lambda I\Big)\beta = 0 \tag{B.8}$$

To satisfy (B.4), $\Sigma - \lambda I$ must be singular, i.e. the determinant

$$|\Sigma - \lambda I| = 0. \tag{B.9}$$

(B.9) is a j-th degree polynomial with j roots λ, $\lambda_1 \geq \lambda_2 \geq \lambda_3 ... \geq \lambda j$.

Each of the roots represents the variance of $\beta' X$ for a particular β, as is shown by

$$\beta'\Big(\Sigma - \lambda I\Big)\beta = 0. \tag{B.10}$$

$$\beta'\Sigma\beta = \lambda\beta'\beta = \lambda. \tag{B.11}$$

Thus the maximum variance is given by β_1 as λ_1. Thus $U_1 = \beta_1' X$ is a normalized linear combination with maximum variance. Now we will determine another normalized linear combination $\beta' X$ which is uncorrelated with U_1, i.e.:

$$\mathscr{E}\beta' X U_1 = \mathscr{E}\beta' X X' \beta_1 = \beta'\Sigma\beta_1 = \lambda_1 \beta'\beta_1 = 0 \tag{B.12}$$

We assume $\Sigma \neq 0$.

Now we want to maximize

$$\psi_2 = \beta'\Sigma\beta - \lambda(\beta'\beta - 1) - 2\nu_1\beta'\Sigma\beta_1. \tag{B.13}$$

We follow a similar procedure to previously with the added constraint

$$\frac{\partial\psi_2}{\partial\beta} = 2\Sigma\beta - 2\lambda\beta - 2\nu_1\Sigma\beta_1 = 0. \tag{B.14}$$

$$0 = 2\beta_1'\Sigma\beta - 2\lambda\beta_1'\Sigma\beta_1 = -2\nu_1\lambda_1, \tag{B.15}$$

which leads to $\nu_1 = 0$ The procedure is as follows: at the (r+1)st iteration, find β such that

$$0 = \mathscr{E}\beta' X U_i = \mathscr{E}\beta' X X' \beta^{(i)} = \beta'\Sigma\beta^{(i)} = \lambda_{(i)}\beta'\beta^{(i)}, \tag{B.16}$$

$$i = 1, ...r.$$

Generally to maximize

$$\phi_{r+1} = \beta'\Sigma\beta - \lambda(\beta'\beta - 1) - 2\sum_{i=1}^{r} \nu_i \beta'\Sigma\beta^{(i)}, \tag{B.17}$$

with λ *and* $\nu_1 ... \nu_r$ *Lagrange multipliers, set the following equal to zero,*

$$\frac{\partial \phi_{r+1}}{\partial \beta} = 2\Sigma\beta - 2\lambda\beta - 2\sum_{i=1}^{r} \nu_i \Sigma\beta^{(i)}. \tag{B.18}$$

Additionally, multiplying the above on the left side by $\beta^{(j)'}$*, we have*

$$0 = 2\beta^{(j)'}\Sigma\beta - 2\lambda\beta^{(j)'}\beta - 2\nu_j\beta^{(j)'}\Sigma\beta^{(j)}. \tag{B.19}$$

As Anderson points out, if $\lambda_{(j)} \neq 0$, $-2\nu_j\lambda_{(j)} = 0$ *and* $\nu_j = 0$*. If* $\lambda_{(j)} = 0$*, then* $\Sigma\beta^{(j)} = \lambda_{(j)}\beta^{(j)} = 0$ *and the jth term in the sum in (B.19) vanishes. Therefore,* β *must satisfy (B.8) and* λ *must satisfy (B.9). It can be shown that there are exactly p components that can be found. Then we can write a matrix representation of the main equations as*

$$\Lambda = \begin{bmatrix} \lambda_{(1)} & 0 & \cdots & 0 \\ 0 & \lambda_{(2)} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_{(p)} \end{bmatrix}, \tag{B.20}$$

$$\Sigma\beta = \beta\Lambda, \tag{B.21}$$

$$\beta'\beta = I, \tag{B.22}$$

which leads from the previous two equations to

$$\beta'\Sigma\beta = \Lambda. \tag{B.23}$$

Considering

$$\begin{aligned} |\Sigma - \lambda I| &= |\beta'| \cdots |\Sigma - \lambda I| \cdots |\beta|, \qquad \text{(B.24)} \\ &= |\beta' \Sigma \beta - \lambda \beta' \beta| = |\Lambda - \lambda I|, \\ &= \prod (\lambda_{(i)} - \lambda), \end{aligned}$$

it is clear that the roots of (B.24) are the diagonal elements of Λ, *and the theorem is proven.*

Bibliography

[1] S Adamovich, M Berkinblit, W Hening, J Sage, and H Poizner. The interaction of visual and proprioceptive inputs in pointing to actual and remembered targets in parkinson's disease. *Neuroscience*, 104:1027–1041, 2001.

[2] B. Anderson and J. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[3] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley and Sons, 2003.

[4] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.

[5] G. H. Arutyunyan, V. S. Gurfinkel, and M. L. Mirskii. Investigation of aiming at a target. *Biophysics*, 13:536–538, 1968.

[6] H. Asada and J.-J. E. Slotine. *Robot Analysis and Control*. John Wiley and Sons, New York, NY, 1986.

[7] S. Ball. *Analog interfacing to embedded microprocessors: Real world design*. Butterworth-Heinemann, Woburn, MA, 2001.

[8] A. Bandura. *Media effects: Advances in theory and research*, chapter Social cognitive theory of mass communication, pages 61–90. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, 1967.

[9] Y. Bar-Shalom. Stochastic dymamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195, 1981.

[10] N. Bernstein. *The coordination and regulation of movements*. Pergamon, London, 1967.

[11] D. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, Bellmont, MA, 2nd edition, 2001.

[12] D. Bertsekas and S. Schreve. *Stochastic Optimal Control: The Discrete Time Case.* Athena Scientific, 1995.

[13] S. Bittanti and G. Picci, editors. *Identification, adaptation, and learning: The science of learning models from data*, volume 153 of *F: Computer and Systems Sciences.* Springer-Verlag, New York, 1994.

[14] V. Cheung, A. d'Avella, M. Tresch, and E. Bizzi. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *Journal of Neuroscience*, 25(27):6419–6434, July 2005.

[15] J. Clifford. *Edward L. Thorndike sane positivist.* Wesleyan University Press, Middleton, 1984.

[16] L. Collatz. *The Numerical Treatment of Differential Equations.* Springer-Verlag, 1966.

[17] A. d'Avella, A. Portone, L. Fernandez, and F. Control of fast-reaching movements by muscle synergy combinations. *The Journal of Neuroscience*, 26(30):7791–7810, July 2006.

[18] Force Dimension. delta.6 haptic device, force feedback interface. Technical report, Force Dimension, PSE-C, CH-1015 Lausanne, Switzerland, January 2009.

[19] O. Donchin, J.T. Francis, R. Shadmehr, and F. Mussa-Ivaldi. Quantifying generalization from trial-by-trial behavior of adaptive systems that learn with basis functions: Theory and experiments in human motor control. *Journal of Neuroscience*, 23(27):9032–9045, 2003.

[20] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12:219–245, 2000.

[21] R. Durrett. *Stochastic calculus: A practical introduction.* Probability and stochastics series. CRC Press, Inc., Boca Raton, FL, 1996.

[22] A. Edsinger-Gonzales. *Robot Manipulation in Human Environments.* PhD thesis, Massachussets Institute of Technology, 2007.

[23] A. Edsinger-Gonzales and J. Weber. Domo : A force sensing humanoid robot for manipulation research. *International Journal of Humanoid Robotics*, 2004.

[24] A. Feldbaum. Dual control theory i-iv. *Automation and Remote Control*, 21:874-880, 21:1033-1039, 22:1-12, 22:109-121, 1960-61.

[25] A. Feldman. Once more on the equilibrium-point hypothesis (lamda-model) for motor control. *Journal of Motor Behavior*, 18(4):723–744, 1995.

[26] J. Ferziger. *Numerical Methods for Engineering Application.* John Wiley and Sons, New York, NY, 2nd edition edition, 1998.

[27] J. Fiene and G. Niemeyer. Switching motor control: An integrated amplifier design for improved velocity estimation and feedback. *In proceedings of the IEEE International Conference on Robotics and Automation*, 5:4504–4509, April 2004.

[28] N.M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. *IEEE Proc - Control Theory Appl.*, 147(1), January 2000.

[29] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.

[30] G. Franklin, D. Powell, and A. Emami-Naeini. *Feedback control of dynamic systems.* Addison Wesley, New York, 3rd edition, 1995.

[31] Y. C. Fung. *Biomechanics.* Springer-Verlag, 2nd edition, 1993.

[32] A. Gelb, editor. *Applied Optimal Estimation.* The MIT Press, 1984.

[33] H. Gomi and M. Kawato. Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Nature*, 272:117–120, 1996.

[34] G. Graham. Practical design and stabilizing control of a dynamic balancing and flipping robot. Master's thesis, University of California, San Diego, La Jolla, CA, 2007.

[35] C. Harris and D. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 294:780–784, 1998.

[36] W. Herzog. *Skeletal muscle mechanics: From mechanisms to function.* John Wiley and Sons, 2000.

[37] E. Hilgard and G. Bower. *Theories of Learning.* Prentice Hall, Inc., Englewood Cliffs, NJ, 1975.

[38] B. Hoff and M. Arbib. Models of trajectory formation and temporal interaction of reach and grasp. *Journal of Motor Behavior*, 25(3):175–192, 1993.

[39] A. Hughes. *Electric motors and drives: Fundamentals, types and applications.* Elsevier North-Holland, Inc., New York, 3rd edition, 2006.

[40] A. F. Huxley. *Biophysics and Biophysical Chemistry*, chapter Muscle structure and theories of contraction in Progress. Pergamon Press, London, 1957.

[41] Inc. Interlink Electronics. Force sensing resistor, 2006-2008.

[42] P. Ioanno and J. Sun. *Robust Adaptive Control.* Prentice Hall, 1996.

[43] V. Ivanchenko and R. A. Jacobs. A developmental approach aids motor learning. *Neural Computation*, 15:2051–2065, 2003.

[44] R. Jazar. *Theory of Applied Robotics: Kinematics, Dynamics, and Control.* Springer-Verlag, Riverdale, NY, 2007.

[45] M. Jeannerod. *The Neural and Behavioral Organization of Goal Directed Movements.* Oxford Science Publications, 1988.

[46] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[47] E. Kandel, J. Schwartz, and T. Jessell. *Principles of Neuroscience.* McGraw-Hill, Inc., New York, 3rd edition, 1991.

[48] J. Keller and H. McKean, editors. *Stochastic differential equations*, volume VI of *SIAM-AMS Proceedings.* American Mathematical Society, Providence, Rhode Island, 1973.

[49] K. P. Kording and D. M. Wolpert. Bayesian integration in sensorimotor learning,. *Nature*, 427:244–247, 2004.

[50] K. P. Kording and D. M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427:244–247, 2004.

[51] K. P. Kording and D. M. Wolpert. The loss function of sensorimotor learning. *Proceedings of the National Academy of Sciences*, 101:9839–42, 2004.

[52] K. P. Kording and D. M. Wolpert. Bayesian decision theory in sensorimotor control. *Trends in Cognitive Science*, 10(7):319–326, 2006.

[53] H. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer-Verlag, New York, New York, 1992.

[54] F. Lacquaniti, M. Corrozzo, and N. Borghese. Time-varying mechanical behavior of multijointed arm in man. *Journal of Neurophysiology*, 69(5):1443–1464, 1993.

[55] M. L. Latash. On the evolution of the notion of synergy. In G. Gantchev and et al, editors, *Motor Control, Today and Tomorrow*, pages 181–196. Sofia, 1999.

[56] W. Li. *Optimal Control for Biological Movement Systems.* PhD thesis, University of California, San Diego, La Jolla, CA, 2006.

[57] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear bio-logical movement systems. *Proc. of the 1st International Conference on Informatics in Control, Automation and Robotics*, 1:222–229, August 2004.

[58] W. Li, E. Todorov, and R. E. Skelton. Estimation and control of systems with multiplicative noise with linear matrix inequalities. *In proceedings of the American Control Conference*, pages 1811–1816, June 2005.

[59] D. Liu. *Computational and Psychophysical Studies of Goal-Directed Arm Movements.* PhD thesis, University of California, San Diego, La Jolla, CA, 2008.

[60] D. Liu and E. Todorov. Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. *Journal of Neuroscience*, 27(35):9354–9368, 2007.

[61] L. Ljung. *System Identification: Theory for the User.* Prentice Hall, PTR, Upper Saddle River, NJ, 2nd edition edition, 1999.

[62] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, 1992.

[63] T. Martin, S. Norris, B. Greger, and T. Thach. Dynamic coordination of body parts during prism adaptation. *Journal of Neuroscience*, 88:1685–1694, 2002.

[64] D. Martinez, R. Bond, and M. Vai, editors. *High performance embedded computing handbook: A systems perspective.* CRC Press, Inc., Boca Raton, FL, 2008.

[65] P. Marwedel. *Embedded system design.* Kluwer Academic Publishers, Norwell, MA, 2003.

[66] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.

[67] Microchip. *dsPIC33FJXXXMCX06/X08/X10 Motor Control Family Data Sheet*. Microchip, inc., 2007.

[68] P. Morasso and F. A. Mussa-Ivaldi. Trajectory formation and handwriting: a computational model. *Biological Cybernetics*, 45:131–142, 1982.

[69] C. Nachtigal. *Instrumentation and control: Fundamentals and applications*. John Wiley and Sons, 1990.

[70] Y. Nakamura. *Advanced robotics: Redundancy and optimization*. Addison Wesley, New York, 1991.

[71] K. M. Newell and D. E. Vaillancourt. Dimensional change in motor learning. *Human Movement Science*, 20:695–715, 2001.

[72] B. Nigg and W. Herzog. *Biomechanics*. John Wiley and Sons, New York, 1994.

[73] H. Olsson, K. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky. Friction models and friction compensation. *European Journal of Control*, 4(3):176–195, 1998.

[74] D. Paluska, M. Mataric, and R. Ambrose. *Biomimetic Robot Control in "Biologically Inspired Intelligent Robots"*, volume PM122. SPIE Press, May 2003.

[75] A. Pasupathy and E. Miller. Different time courses of learning-related activity in the prefrontal cortex and striatum. *Nature*, 433:873–876, 2005.

[76] C. L. Phillips and H. T. Nagle. *Digital control system: Analysis and design*. Prentice Hall, Inc., Englewood Cliffs, NJ, 2nd edition, 1990.

[77] C. Pozrikidis. *Numerical Computation in Science and Engineering*. Oxford University Press, 1998.

[78] Z. Pylyshyn. Computational models and empirical constraints. *Behavioral and brain sciences*, 1:93–128, 1980.

[79] M. Raibert. *Legged robots that balance*. The MIT Press, Cambridge, Massachusetts, 1986.

[80] E. Rivin. *Mechanical design of robots*. McGraw-Hill, Inc., New York, NY, 1987.

[81] T. Saaty and J. Alexander. *Thinking with models: Mathematical models in the physical, biological, and social sciences.* Pergamon Press, Inc, Elmsford, New York, 1st edition, 1981.

[82] M. Santello, M. Flanders, and J. Soechting. Patterns of hand motion during grasping and the influence of sensory guidance. *The Journal of Neuroscience*, 22(4):1425–1435, February 2002.

[83] M. Santello, M. Flanders, and J.F. Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, (18):10105–15, 1998.

[84] M. Santello and J.F. Soechting. Gradual molding of the hand to object contours. *Journal of Neurophysiology*, 79:1307–1320, 1998.

[85] L. Schettino, S. Adamovich, W. Hening, E. Tunik, J. Sage, and H. Poizner. Hand preshaping in parkinson's disease: effects of visual feedback and medication state. *Experimental Brain Research*, 2005.

[86] S.H. Scott. Optimal feedback control and the neural basis of volitional motor control. *Nat Rev Neurosci*, 5:532–546, 2004.

[87] R. Shadmehr and F. Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14:3204–3224, 1994.

[88] J. Si, A. Barto, W. Powell, and D. Wunsch, editors. *Handbook of Learning and Approximate Dynamic Programming.* IEEE Press on Computational Intelligence, 2004.

[89] A. Simpkins and E. Todorov. Optimal tradeoff between exploration and exploitation. American Control Conference, IEEE Computer Society, 2008.

[90] B. F. Skinner. *The Behavior of Organisms.* D. Appleton-Century Co., 1938.

[91] B. F. Skinner. Are theories of learning necessary? *Psychological Review*, 57:193–216, 1950.

[92] H. W. Sorenson, editor. *Kalman Filtering: Theory and Application.* IEEE Press, 1985.

[93] R. Stein and W. Hunt. *Electric power system components: Transformers and rotating machines.* Van Nostrand Reinhold Company, New York, 1979.

[94] R. Stengel. *Stochastic Optimal Control: Theory and Application.* John Wiley and Sons, 1986.

[95] D. Stroock. *Markov processes from K. Ito's perspective.* Number 155 in Annals of mathematical studies. Princeton University Press, Princeton, New Jersey, 2003.

[96] SensAble Technologies. Specifications for the phantom® desktoptm haptic device. Technical report, SensAble Tehnologies, 2009.

[97] E. Todorov. *Studies of Goal Directed Movements.* PhD thesis, Massachussets Institute of Technology, 1998.

[98] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7:907–915, 2004.

[99] E. Todorov and Z. Ghahramani. Unsupervised learning of sensory-motor primitives. IEEE, In Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2003.

[100] E. Todorov and Z. Ghahramani. Analysis of synergies underlying complex hand manipulation. San Francisco, CA, USA, September 1-5, 2004 2004. IEEE, Proceedings of the 26th Annual International Conference of the IEEE EMBS.

[101] E. Todorov and M. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.

[102] E. Todorov and M. Jordan. A minimal intervention principle for coordinated movement. *Advances in Neural Information Processing Systems*, 15:27–34, 2003.

[103] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. *In proceedings of the American Control Conference*, 2005.

[104] E. Torres and D. Zipser. Reaching to grasp with a multi-jointed arm. i. computational model. *Journal of Neurophysiology*, 88(5):2355–2367, 2002.

[105] J. E. Traister. *Complete handbook of electric motor controls.* Prentice Hall, Englewood Cliffs, NJ, 1986.

[106] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement: Minimum torque-change model. *Biological Cybernetics*, 61:89–101, 1989.

[107] T. Vilis and J. Hore. Effects of changes in mechanical state of limb on cerebellar intention tremor. *Journal of Neurophysiology*, 40(5):1214–1224, 1977.

[108] D. J. White. *Markov decision processes.* John Wiley and Sons, New York, 1993.

[109] B. Wittenmark. Adaptive dual control methods: An overview. In *5th IFAC symposium on Adaptive Systems in Control and Signal Processing*, pages 67–73, 1995.

[110] P. Yedamale. An885 : Brushless dc (bldc) motor fundamentals. Technical report, Microchip inc., 2003.

[111] G. I. Zahalak. A comparison of the mechanical behavior of the cat soleus muscle with a distribution-moment model. *Journal of Biomechanical Engineering*, 108:131–140, 1986.

[112] K. Zhou and J. Doyle. *Essentials of Robust Control.* Prentice Hall, Upper Saddle River, NJ, 1998.