UNIVERSITY OF CALIFORNIA, SAN DIEGO

Optimal Control for Biological Movement Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Aerospace Engineering)

by

Weiwei Li

Committee in charge:

Professor Emanuel Todorov, Chair
Professor Robert E. Skelton, Co-Chair
Professor Robert R. Bitmead
Professor Miroslav Krstic
Professor Jochen Triesch

2006

The dissertation of Weiwei Li is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2006

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to first thank my advisors, Professor Emanuel Todorov and Professor Robert E. Skelton, for giving me the chance to do this work, for the encouragement, and for the guidance. Especially I would like to express my deepest gratitude to Professor Todorov for the many helpful technical discussions and for his many useful suggestions and advices on my thesis work. I am also very thankful to Professor Skelton, for teaching me mathematical rigor, and for broadening my view and understanding of system and control theory. Many thanks to Professor Robert R. Bitmead, Professor Miroslav Krstic and Professor Jochen Triesch for taking the time to participate as committee members. Above all I am also indebted to Professor Philip E. Gill for his guidance and support.

Many thanks go to my coworkers as well as my many good friends, in particular, Waileung Chan, Faming Li, Jean-Paul Pinaud, Jun Yan, Jie Zeng at Dynamic Systems and Control Group; Dan Liu, Xiuchuan Pan, Tom Sullivan, Nigel Singh and Dongsung Huh at Natural Computation Group. My experiences at UCSD would not have been as enjoyable and rewarding without them.

Last, but not least, I want to express sincere gratitude to my husband, Xiuchuan Pan, for his patience, support and valued assistance during the course of my graduate study. Thanks to my family for their love and continual support.

The text of Chapter 4, in part, was originally published in Proceedings of the 1^{st} International Conference on Informatics in Control, Automation & Robotics. The thesis author was the primary researcher and author in these works and the co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

The text of Chapter 5, in part, has been submitted for publication in Automatica. The thesis author was the primary researcher and author in these works and the co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

CURRICULUM VITAE

Education

1994	B.S. in Electrical Engineering Xidian University, China
1997	M.S. in Electrical Engineering (Flight Dynamics & Controls) Beijing Institute of Control Engineering, China
2006	Ph.D. in Engineering Sciences (Aerospace Engineering) University of California, San Diego, USA

Publications

W. Li and E. Todorov, "Iterative Linearization Methods for Approximately Optimal Control and Estimation Design of Nonlinear stochastic Systems," submitted to *Automatica*, 2006.

W. Li, E. Todorov and X. Pan, "Hierarchical Feedback and Learning for Multi-joint Arm Movement Control," in *Proc. of the 27th IEEE Conference on Engineering in Medicine and Biology Society*, Sep. 2005.

W. Li, E. Todorov and R. E. Skelton, "Estimation and Control of Systems with Multiplicative Noise via Linear Matrix Inequalities," in *Proc. of American Control Conference*, June 2005, pp. 1811–1816.

E. Todorov, W. Li and X. Pan, "From Task Parameters to Motor Synergies: A Hierarchical Framework for Approximately-optimal Control of Redundant Manipulator," *Journal of Robotic Systems* 22(11), 691–710, 2005.

W. Li, R. E. Skelton and E. Todorov, "State Estimation with Finite Signal-to-Noise Models via Linear Matrix Inequalities," submitted to *Journal of Dynamic Systems, Measurement and Control*, 2005.

E. Todorov and W. Li, "A Generalized Iterative LQG Method for Locally-optimal Feedback Control of Constrained Nonlinear Stochastic System," in *Proc. of American Control Conference*, June 2005, pp. 300–306.

W. Li and E. Todorov, "Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems," in *Proc. of the 1st International Conference on Informatics in Control, Automation & Robotics*, Vol. 1, Aug. 2004, pp. 222–229.

W. Li, E. Todorov and X. Pan, "Hierarchical Optimal Control of Redundant Biomechanical Systems," in *Proc. of the 26th IEEE Conference on Engineering in Medicine and Biology Society*, Sep. 2004, pp. 4618–4621.

W. Li and Robert E. Skelton, "State Estimation with Finite Signal-to-Noise Models," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Dec. 2003, pp. 5378–5383.

E. Todorov and W. Li, "Optimal Control Methods Suitable for Biomechanical Systems," in *Proc. of the 25th IEEE Conference on Engineering in Medicine and Biology Society*, Sep. 2003, pp. 1758–1761.

ABSTRACT OF THE DISSERTATION

Optimal Control for Biological Movement Systems

by

Weiwei Li

Doctor of Philosophy in Engineering Sciences (Aerospace Engineering)

University of California, San Diego, 2006

Professor Emanuel Todorov, Chair

Professor Robert E. Skelton, Co-Chair

Optimal control model of biological movement provides a natural starting point for describing observed everyday behavior, and is so far the most successful model in motor control. However, in their present form, such models have a serious limitation—they rely on Linear-Quadratic-Gaussian formalism, while in reality biomechanical systems are strongly non-linear, the disturbances are control-dependent, muscle activations are non-negative, and performance criteria are rarely quadratic. In order to handle such complex problems, we develop an iterative Linear-Quadratic-Gaussian method for locally-optimal control and estimation of nonlinear stochastic systems subject to control constraints. The new method constructs an affine feedback control law, obtained by minimizing a novel quadratic approximation to the optimal cost-to-go function. It also constructs a modified extended Kalman filter corresponding to the control law. The control law and filter are iteratively improved until convergence. Finally, the performance of the algorithm is illustrated in the context of reaching movements on a realistic human arm model.

The second focus of this thesis is on the integration of optimality principles with hierarchical control scheme. We present a general approach to designing feedback control

hierarchies for redundant biomechanical systems, that approximate the (non-hierarchical) optimal control law but have much lower computational demands. This hierarchy has two levels of feedback control: the high level is designed as an optimal feedback controller operating on a simplified plant; the low level is responsible for transforming the dynamics of the true plant into the desired virtual dynamics. The new method will be useful not only for modelling neural control of movement, but also for designing Functional Electric Stimulation systems that have to achieve task goals by activating muscles in real time.

Another contribution of this thesis is to present an estimation design method for multiplicative noise system using linear matrix inequalities (LMIs) approach. Sufficient conditions for the existence of the state estimator are provided; these conditions are expressed in terms of LMIs and the parametrization of all admissible solutions is provided. We show that a mild additional constraint for scaling will make the problem convex.

Chapter 1

Introduction

Movements like reaching for a glass of water, pressing the keyboard with the finger, or hitting a tennis ball are generally accomplished without much difficulty. However, controlling even the simplest movement such as picking up a glass of water, the brain must take into account an enormous amount of information processing, including the starting position and velocity of the arm, the force required in the fingers to hold the glass, and the knowledge of target position, in order to signal the stimulation of muscle and certain configuration of joints for the appropriate movement. During the past two decades, many researchers have been making their efforts to study geometric properties of movement trajectories, sequence of muscle activations, etc. in the field of motor control. However, understanding how the brain integrates sensory information and generates the motor command to produce coherent motor behavior remains a challenging problem scientifically.

From an engineering perspective, the biological movement system can be considered as a system whose inputs are the motor commands emanating from the controller within the central nervous system (CNS), whose outputs are the sensory feedback signals that is a function of the state variables. For example, considering a human arm model, the motor command could represent the torques generated around the joints, the state variables could be the joint angles and angular velocities. Recently, computational models are being used to simulate simple movement tasks and compare the outcomes with real

human behaviors, thereby testing ideas of how brain achieves all kinds of sophisticated motor control. Consequently, theories of motor function assume that movement is controlled in the best way — i.e. optimally. Transforming this idea into quantitative models requires a good understanding of optimal control theory. But the results are well worth the effort: such optimality models not only provide accurate accounts of many empirical phenomena, but also elucidate fundamental questions in the field. While traditional emphasis has been on optimizing desired movement trajectories, recent work has focused on optimal feedback control mechanisms which can accomplish complex behavior goals. This approach provides an elegant theoretical framework for trying to explain why the system behaves as it does, and to specify the control laws that generate the observed behavior online.

Understanding the control of movement is not just a theoretical study. There are a lot of people who suffer from paralysis around the world. They cannot make even the most basic normal arm movement. Restoring motor function to impaired individuals is very important for improving the quality of their life. This could be achieved through functional electric stimulation (FES) of paralyzed muscles, or various robotic prostheses. Recent technological advances such as implantable muscle stimulators hold great promise for the restoration of motor function. These sophisticated devices will only become useful if we can find appropriate methods to control them. Therefore, this thesis research makes it possible to provide suitable control algorithms for those prosthetic devices.

1.1 Organization of Thesis

The goal of this thesis research is to develop a new theoretical framework for the sensory-motor control and to advance our understanding how the brain controls the movement. The main theme is to study and design some novel approaches for this redundant movement system, and to increase the comprehension how the control commands for generating the movements are implemented in the central nervous system and how they are acquired via the optimal control strategy. Ultimately, the ideas gained from these studies will be useful to provide a general framework for constructing computa-

tional models of biological movement, which could explain why the system behaves as it does, and to specify the control laws most suitable for generating the observed behavior.

Chapter 2 contains the review of the literature. Some widely known background material is elaborated here. Included reviews are in the area of computational models of the motor control, properties of biological movement systems and some approaches for the control of movement.

Chapter 3 describes a mathematical model for the 2-link human arm including 10 state dimensions and 6 muscle actuators. We then elaborate several cost functions relevant to different tasks that we are interested in. These cost functions combine accuracy and energetic efficiency, which can predict not only average behavior movement, but also the task-specific sensorimotor contingencies that the central nervous system(CNS) uses to make intelligent adjustments. Studying the optimal control problem for this complex biomechanical system will be the main topic in the remaining chapters of this thesis research.

Chapter 4 develops an iterative Linear Quadratic Regulator (ILQR) method, and compares its performance with ordinary differential equation methods, conjugate gradient descent method and differential dynamic programming algorithm on the challenging biological movement control problem. This method uses iterative linearization of the nonlinear system around a nominal trajectory, and computes a locally optimal feedback control law via a modified LQR technique. This control law is then applied to the linearized system, and the result is used to improve the nominal trajectory incrementally.

Chapter 5 presents an iterative Linear-Quadratic-Gaussian method for locally-optimal control and estimation of nonlinear stochastic systems subject to control constraints. This method extends the efficient and well-developed LQG framework to the domain of nonlinear control. Briefly, the iteration starts with some control law that is applied to the nonlinear system — obtaining an average trajectory and control sequence. We then linearize the dynamics and quadratize the cost in the vicinity of that state-control trajectory, apply dynamic programming locally, and use the result to improve the initial control law. The algorithm has quadratic convergence, similar to a Newton's method.

The main contribution is that the new method constructs an affine feedback control law, obtained by minimizing a novel quadratic approximation to the optimal cost-to-go function. It also constructs a modified extended Kalman filter corresponding to the control law. We illustrate the application of this extended LQG methodology in the context of reaching movements, and study the properties of the new algorithm through extensive numerical simulations.

Chapter 6 outlines a general approach to design feedback control hierarchies for redundant biomechanical systems. The major goal here is to explore a new method for modelling the neural control of movement, which is accomplished by cooperating the multiple levels of sensorimotor system. Our hierarchy has two levels of feedback control: the high level is designed as an optimal feedback controller operating on a simplified virtual plant; the low level is responsible for transforming the dynamics of the true plant into the desired virtual dynamics. This approach is then applied to the reaching task using two detailed models of the human arm developed in chapter 3.

Chapter 7 begins by using linear matrix inequalities (LMIs) approach to design the state feedback controller for linear discrete time systems with multiplicative noise. Another part of this chapter focuses on the state estimator design. The sufficient conditions for the existence of the state estimator are expressed in terms of LMIs, and the parametrization of all admissible solutions is also addressed. Finally the performance of this estimator is examined by applying to a simple hand movement system.

Chapter 8 concludes the thesis with some final remarks and proposes some future research directions.

Chapter 2

Background and Significance

2.1 Optimality Principles in Motor Control

Producing even the simplest movement involves an enormous amount of information processing. When we move our hand to a target location, there are infinitely many possible paths and velocity profiles that the multi-joint arm could take, and each trajectory can be achieved by multiple combinations of joint angles. Furthermore, since we have many more muscles than joints, each arm configuration can be generated by an infinite variety of muscle activation patterns. Biomechanical redundancy makes the motor system very flexible, but at the same time requires a very well designed controller that can choose intelligently among the many possible alternatives. Optimal control theory provides a principled approach to this problem — it postulates that the movement patterns being chosen are the ones optimal for the task at hand.

2.1.1 Optimal Performance Criteria for Movement Planning

Optimality principles form the basis of many physical sciences. In the field of motor control, optimality principles not only yield accurate descriptions of observed phenomena, but are well justified *a priori*. This is because the sensorimotor system is the product of optimization processes (i.e. evolution, development, learning, adaptation)

which continuously improve behavioral performance. Therefore, optimality provides an elegant framework for trying to explain why the system behaves as it does, and to specify the control laws that generate the observed behavior. When dealing with this optimal control problem, a specific cost function has to be chosen in order to evaluate the performance of the system under control, and the objective is to minimize the value of this cost function.

Based on the observation that point-to-point movements are smooth in Cartesian space, Hogan [42] proposed the minimum jerk model in which the cost function depends on the squared first derivative of Cartesian hand acceleration or 'jerk'. Such kind of cost function depends on the kinematics of the motion, and the variables of interest include the positions (e.g. joint angles or hand Cartesian coordinates), angular velocities, accelerations and higher derivatives. Let $x(t)$, $y(t)$ denote the Cartesian coordinates of the hand position at time t, the minimum jerk model is based on the following cost function:

$$J = \frac{1}{2}\int_0^T \left(\left(\frac{d^3x}{dt^3}\right)^2 + \left(\frac{d^3y}{dt^3}\right)^2 \right) dt, \tag{2.1}$$

where T is the duration of the movement. By assuming the movement to start and end with zero velocity and acceleration, the optimal trajectory is of the following:

$$\begin{aligned} x(t) &= x_0 + (x_T - x_0)\left(10\left(\frac{t}{T}\right)^3 - 15\left(\frac{t}{T}\right)^4 + 6\left(\frac{t}{T}\right)^5\right) \\ y(t) &= y_0 + (y_T - y_0)\left(10\left(\frac{t}{T}\right)^3 - 15\left(\frac{t}{T}\right)^4 + 6\left(\frac{t}{T}\right)^5\right) \end{aligned} \tag{2.2}$$

where (x_0, y_0) and (x_T, y_T) are the initial and final positions at time $t = 0$ and $t = T$ respectively. From (2.2), one can see that the trajectory predicted by using this criterion is a straight-line Cartesian hand path with bell-shaped velocity profile, which is consistent with the empirical data for rapid movement made without the accuracy constraint. Since $x(t)$ and $y(t)$ depend only on the initial and final positions of the hand and movement time T, the optimal trajectory is determined only by the kinematics of the movement and is independent of the dynamics of musculoskeletal system which generates the motion.

By studying large range movements and the observation of significant curvature (which is unable to be explained by the minimum jerk model) in the paths of the motion,

Uno et al. [125] proposed an alternative model — the minimum torque-change model — in which the trajectories are selected to optimize a function that penalizes the rate of change of torque. Such kind of cost function depends on the dynamics of the arm, and the variables of interest contain torque change, muscle tension and muscle command. Let $d\tau_i/dt$ denote the rate of change of torque at the i_{th} joint, the minimum torque-change model is based on the following cost function:

$$J = \frac{1}{2}\int_0^T \sum_{i=1}^{n} \left(\frac{d\tau_i}{dt}\right)^2 dt, \tag{2.3}$$

Because the minimum torque change criterion depends on the arm dynamics, it can reproduce the gradually curved trajectory which cannot be explained by the minimum jerk model.

Although the minimum jerk and torque-change models predict many aspects of trajectories which are consistent with empirical data, and inspire a wealth of experimental and theoretical work, they still have several features which can't make them satisfying as models of movement. These models only consider the smooth trajectories, but don't propose any advantage for smoothness of movement. Moreover, there has been no principled explanation why the central nervous system should have evolved to optimize torque-change or jerk; and how the central nervous system could estimate such complex quantities and integrate them over the duration of a trajectory is also unknown.

In an attempt to solve for these problems, Harris and Wolpert [38] in 1998 presented a minimum-variance theory of eye and arm movement based on the assumption that the neural control signals are corrupted by noise whose variance increased with the size of the control signal. They proposed that in the presence of such signal-dependent noise, moving as rapidly as possible requires large control signals, which would carry more noise and cause the trajectories to deviate from the desired path. These deviations will be accumulated over the duration of the movement, and finally would increase the variability in the final position. Accuracy could be improved by having low amplitude control signals, but the movement will be accordingly slow. Thus, signal-dependent noise inherently imposes the speed-accuracy trade-off which is in agreement with the well-known Fitt's Law. By minimizing the variance of the final eye or arm position for a specified

movement duration, the minimum-variance solution can capture the important features of natural saccadic eye movements or goal-directed arm movements. The encouraging advantage of this approach is that the cost function they chose, such as the variance of the final position or the consequences of this accuracy, are directly available to the central nervous system and the optimal trajectory could be learned from the repeated movements.

So far a number of hypothetical cost function for biological movement have been examined. How to choose an associated cost function which is more pertinent in predicting the human movement is still one of the challenging issues in biological motor control.

2.1.2 Open-loop and Closed-loop Control

The majority of existing optimality models in motor control have been formulated in open-loop (feed-forward), and plan a desired trajectory while ignoring the online feedback. Actually open-loop control is used by motor system to control posture and movement. For example, catching a ball is a visually trigged open-loop response. The key principle of open-loop control is that it depends on the ability of the central nervous system to predict the consequence of sensory events. The most popular approach has been to optimize the sequence of muscle activation or limb state [30, 38, 125]. The widespread use of optimization methods in open-loop trajectory planning creates the impression that the optimal control necessarily predicts highly stereotyped trajectories. These are based on the assumption that there is a separation between the trajectory planning and trajectory execution for the completion of complex task.

However, the most remarkable property of biological movements (in comparison with synthetic ones) is that they can accomplish complex high-level goals in the presence of large internal fluctuations, noise, delays, and unpredictable changes in the environment. For such kind of system, open-loop approaches can only yield suboptimal performance. Optimal performance can be only achieved through an elaborate feedback control scheme [73, 41, 60, 67, 116], that predict not only average behavior but also the task-specific sensorimotor contingencies used to generate intelligent adjustments online. Such ad-

justments enable biological systems to "solve a control problem repeatedly rather than repeat its solution, " [8] and thus afford remarkable levels of performance.

Indeed, focus has recently shifted towards stochastic optimal feedback control models (Fig. 2.1). This approach has already clarified a number of long-standing issues related to the control of redundant biomechanical systems. Some of the most surprising results show that, the optimal feedback controllers for such systems obey a *minimal intervention* principle [116, 117]. By correcting only task-relevant errors, the model minimizes the potential effects of the noise. An optimal feedback controller has nothing to gain from correcting task-irrelevant deviations, because it only concerns the task performance and by definition such deviations do not interfere with the performance. Moreover, generating a corrective control signal can be detrimental, because, in the motor system, the noise is known to be signal-dependent and therefore could increase, and the cost being minimized most likely includes a control-dependent effort penalty which could also increase. If this minimum intervention principle holds, and the noise perturbs the system in all directions, the interplay of noise and control processes will result in larger variability in task-irrelevant directions. At the same time, if certain deviations are not corrected, it implies that certain dimensions of control space are not being used. This model suggests that rather than specifying a desired trajectory, the motor system use the optimal feedback control to deal with the deviations that interfere with the goal of the task.

However, feedback control in biological movement systems is subject to potential difficulties with stability, because there exist a significant amount of delay in the sensory feedback. In robotics, almost all practical applications depend on feedback controls. This is because feedback delays in artificial system can be made very small, hence, sampling and control frequencies can be quite high. But, in the biological movement control, such delays are very large compared with the movement duration of fast movement (e.g.150ms). For example, the delay of visual feedback on arm movements ranges from 100-150ms, and relatively fast spinal feedback loop still requires 30-50ms time delay. These delays can cause the system unstable when we try to make the rapid movement under the feedback control [80]. Hence fast or smooth movement cannot be executed

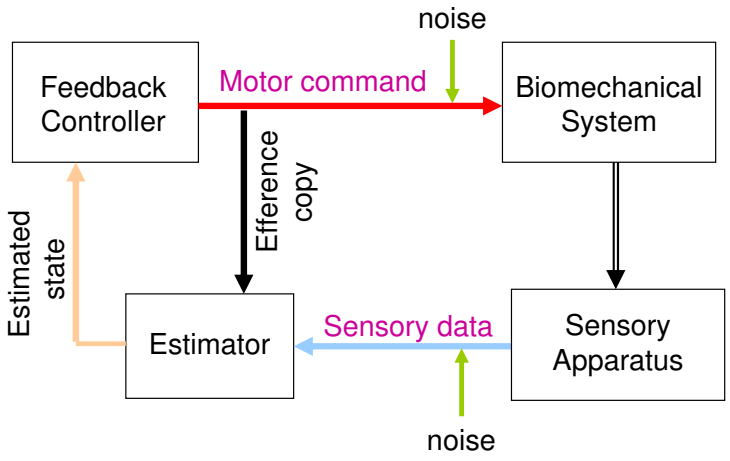


Figure 2.1. Sensorimotor Closed-loop Control System.

depending only on the feedback control.

Kalman filter plays a significant role in the history of statistical estimation theory and is the greatest discovery in control and systems theory of the twentieth century. It has enabled many things that could not have been done without it. It provides immediate application for the control of complex dynamic systems. To control a dynamic system, you must first know what it is doing. However, it is not possible to measure every variable that you want to control, and the Kalman filter provides a means for estimating those information from noisy measurement. Back to the biological motor systems, in order to control our movements, the central nervous system needs to know the state variables that it wants to control. With the combination of sensory feedback information and forward model, [131, 132] claimed that the Kalman filter can be used in motor control to compensate for the sensorimotor delays and to reduce the uncertainty in the state estimate which results from noise inherent in both the sensory and motor signals. This model has been supported by empirical studies investigating the estimation of hand position [130] , posture [60] and head orientation.

2.1.3 Motor Learning

Humans demonstrate a remarkable ability to generate accurate and appropriate motor behavior under many different and often uncertain environmental conditions. A lot of researchers have recently shown increasing interests in understanding how we learn to control our movements and predict the consequences of our actions under a predictably or unpredictably varying environment.

Shadmehr [103] investigated reaching movements in an altered mechanical environment. This environment is a force field implemented by a robot manipulandum whose end-effector is held by the subject while making the reaching movement. The experiment results show that, with practice, the hand trajectories in the force field converged to a path observed during unperturbed condition. They found evidence for the existence of a desired trajectory for moving the hand along this desired path. They also indicated that, during adaptation to a force field which significantly changes the dynamics of a reaching movement, the motor controller achieved this desired performance through the updating of the internal model which predicts the forces acting on the hand as it performs the tasks.

There are three main computational approaches for motor learning: supervised, unsupervised and reinforcement learning. In supervised learning, it requires a desired output corresponding to each input. The error between the desired output and the actual output is computed and is used to adjust the parameters inside the learning system. This process is mathematically formulated as an optimization problem in which the cost function is one-half the squared error. The learning algorithm adjusts the parameters of the system so as to minimize this cost function. While, for unsupervised learning, the environment provides inputs but gives neither desired output nor any measurement of reward. The main problem of the unsupervised learning is that there is no guarantee whether the learning rules will be useful for the decision making and control or not.

Reinforcement learning differs from supervised learning by requiring significantly less information to be available to the learner. Rather than requiring a performance error or a desired target in the control space, reinforcement learning algorithms require only an

evaluation of total future reward. The only objective of reinforcement learning is to maximize the total reward it receives in the long run. The reward function defines whether the system is performing well or not; it does not provide any information about how to correct an error. Sutton and Barto [109] have developed a variety of reinforcement learning algorithms and explained its relationship with optimal control theory. The problem of this model is that it has only been applied to low-dimensional control problems.

Most recent study by Konrad P. Körding and Daniel M. Wolpert [57] demonstrated that the central nervous system employs probabilistic models during sensorimotor learning. If the brain works in the Bayesian way, it would optimally use the prior knowledge. To use a bayesian strategy, the brain would need to represent the prior distribution and the level of uncertainty in the sensory feedback. The experiments show that subjects internally represent both the statistical distribution of the task and their sensory uncertainty, combining them in a manner consistent with a performance-optimizing bayesian process.

2.1.4 Hierarchical Control

Sensory-motor control occurs simultaneously on many levels [8, 73]. The pioneering work of Sherrington has emphasized that biological control hierarchies are composed of a large number of parallel feedback loops, understanding how the multiple levels of the sensorimotor system cooperate to produce integrated movement has been a central theme in neuroscience. The most thorough investigation of the levels of human motor control was undertaken by Bernstein [8] more than 50 years ago, by taking into account evolutionary, anatomical, and a wide range of behavior evidence. The insight of Berstein's hierarchy involves four levels: posture and muscle tone, muscle synergies, dealing with three-dimensional space, and organizing complex actions that pursue some abstract goals. Computational modeling that aims to capture the essence of feedback control hierarchies is still in its infancy [73, 68, 69, 119, 124]. In robotics, the idea of hierarchical control [14, 54, 92] was popularized by Brooks [14], who constructed low-level control circuits driving a mobile robot. These circuits were then coordinated by

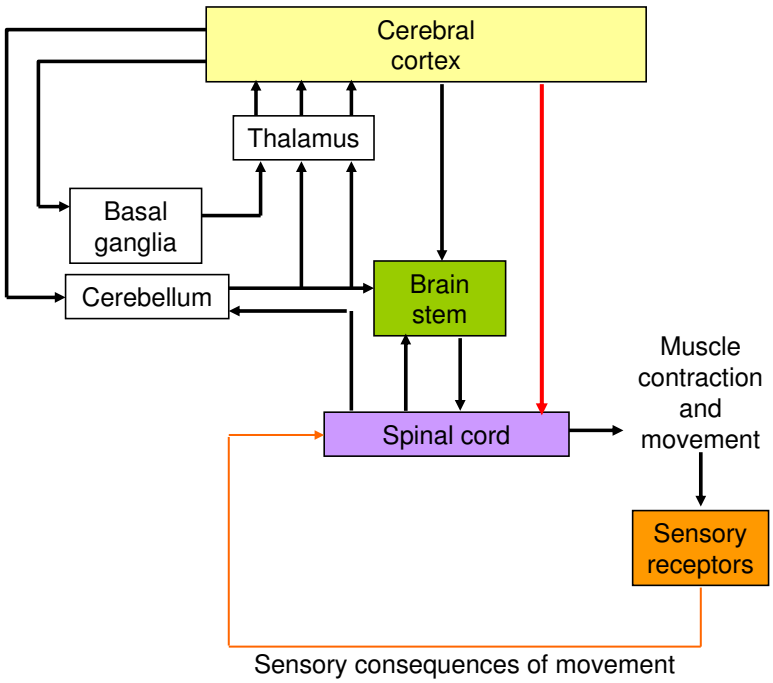


Figure 2.2. Sensorimotor Hierarchical Control Structure. There are three levels in this hierarchy: spinal cord, brain stem and the cortical motor area. They are organized hierarchically and in parallel. The motor area of cerebral cortex can influence spinal cord both directly and indirectly through brain stem descending system. All these three levels receive sensory inputs and are also under the influence of two sub-cortical system: basal ganglia and cerebellum. Both basal ganglia and cerebellum act on cerebral cortex through thalamus.

a high-level controller that achieved the reasonable locomotion. This work was an existence proof of the usage of hierarchical control, but it doesn't point to general design principles.

By studying optimal control for redundant system, Todorov and Jordan [116, 117] proposed that the optimal feedback control laws for typical motor tasks obey a "minimal intervention" principle: deviations from the average trajectory are only corrected when they interfere with the task goals. The hierarchical structure of biological system (Fig. 2.2) and this minimal intervention principle give us a key insight that, most of the human movements involves at least two levels of feedback control acting simultane-

ously: a low level loop which provides various automatisms and corrections that help the leading level; a high level loop which monitors progress and exploits many different ways of performing goal-directed movement control. Since the low level presents the intrinsic muscle properties, it has to transform the resulting commands into appropriate muscle activations. Actually, by combining the optimality principle and hierarchical control strategy, it is possible for us to develop a general method for feedback control of redundant biomechanical systems, and discover a new framework for sensorimotor control.

2.2 Properties of Biological Movement Systems

2.2.1 High dimensionality

Biomechanical systems have a number of characteristics that distinguish them from the synthetic systems commonly studied in control theory. It is very important to understand those distinguishing characteristics in order to study the control of natural movement and design control architectures to achieve complex behaviors.

One of most remarkable properties of biomechanical systems is that the state spaces have unusually high dimensionality. For example, consider a 2-link, 6-muscle arm model which is often studied in motor control, the state space includes 2 joint angles and 2 joint velocities — since we are dealing with a second-order system. A realistic state description should also include 6 muscle activations, because muscles act as low-pass filters of neural activity, with non-negligible time constant. A similar count for a complete model of the arm (excluding the hand) yields 20 dynamic states, and 50 muscle states. Such state spaces cannot be discretized, which rules out all methods relying on discretization.

2.2.2 Uncertainties and Noises

Noise is a very important concept in modelling biological sensorimotor processing. Back in 1954, Fitts [29] showed the possibility that rapid aimed movements are affected by stochastic noise in neuromotor channels: the variability of motor errors increases

with the magnitude of the movement. However, it only interpreted the effects of noise in terms of mathematical information theory, and didn't develop a detailed quantitative account of the mechanisms responsible for random movement variability.

The substantial variability of biological movements indicates that the sensorimotor system operates in the presence of large (mostly internal) disturbances. Noise exists both in the sensory information and the motor commands. For example, noise in the sensory input will result in uncertainty in the position at which the object is perceived, and noise in the motor commands describes the uncertainty of actual force produced by the muscle [79], which will lead to movement inaccuracy and variability.

Noise in the motor commands is signal-dependent — with standard deviation increasing linearly with the magnitude of the motor command signal (control signal). This is a very important assumption, which is consistent with the observation captured by the empirical Fitt's law, and supported by the empirical finding [79, 100]. Optimal control of such systems should obviously take this phenomenon into account, because an appropriately chosen control signal can actually decrease the noise. Recently, this model is widely used in studying biological movement [38, 116, 117, 118, 120, 121].

2.3 Relevant Approaches in Optimal Control Theory

Many theories in the physical sciences are expressed in terms of optimality principles, which often provide the most parsimonious description of the laws governing a system's behavior. Optimality is also playing an increasingly important role in the field of motor control. This is partly motivated by the parsimony and empirical success of optimal control models of biological movement. Perhaps more importantly, such models are appealing because all the processes that give rise to a specific motor system under investigation are in a sense optimization processes, that over time cause the system to perform better and better. It is therefore natural to use the limit of optimal performance as the starting point for theoretical investigations of motor control. Such investigations can only be productive, however, if we have efficient numerical methods that are suitable for controlling biological movement systems. Here we present several of the most popular

methods for solving optimal control problems. It will be very useful, in later sections, for comparisons between our approach and currently used methods.

All the numerical algorithms provided in the subsequent sections are aimed at solving the following problem. Consider a dynamical system with state variable $x \in \mathbb{R}^m$ and control signal $u \in \mathbb{R}^n$

$$\dot{x}(t) = f(x, u, t), \qquad x(0) = x_0, \tag{2.4}$$

and the performance criterion to be minimized is expressed as

$$J_0 = \Phi\big(x(T), T\big) + \int_0^T L\big(x(t), u(t), t\big)dt, \tag{2.5}$$

where [0, T] is the time interval of interest. The symbol $\Phi\big(x(T), T\big)$ represents a final time penalty and the integrand $L\big(x(t), u(t), t\big)$ dictates the nature of the optimizing solution. The goal of optimal control problem is to find an admissible control $u^*(t)$ that drives the system to follow a trajectory $x^*(t)$ such that the cost function (2.5) is minimized. Here we are concerned with comparing the existing algorithms only for deterministic control problems, but we will include the stochastic case and develop our new algorithm in the future chapters.

2.3.1 Linear Matrix Inequalities (LMIs)

A wide variety of problems arising in system and control theory can be reduced to a convex and quasi-convex optimization problems involving matrix inequality. The history of LMI goes back more than 100 years ago when Lyapunov showed that the system $\dot{x}(t) = Ax(t)$ is stable if and only if there exists a positive definite matrix P such that $A^TP + PA < 0$. The requirement $P > 0, A^TP + PA < 0$ is called Lyapunov inequality. The important breakthrough came in the early 1960's when Yakubovich introduced what we now call the positive-real Lemma, which provided certain ways to solve LMIs by graphical methods. J.C Willems in 1971 transformed the quadratic optimal control problem into the LMI

$$\begin{bmatrix} A^TP + PA + Q & PB + C^T \\ B^TP + C & R \end{bmatrix} \geq 0 \tag{2.6}$$

and pointed out that it can be solved by studying the symmetric solutions of the Algebraic Riccati Equation

$$A^T P + PA - (PB + C^T)R^{-1}(B^T P + C) + Q = 0 \tag{2.7}$$

The major advance was the observation that the LMIs that arise in system and control theory can be formulated as convex optimization problems, and what the most important is that many LMIs for which no analytical solution is likely to be found can be solved numerically, such as using ellipsoid algorithm and interior-point methods.

2.3.2 Dynamic Programming

Dynamic programming was developed based on Bellman's Optimality Principle. Rather than considering a cost function (2.5), we consider the value function

$$V\big(x(\tau),\tau\big) = \Phi\big(x(T),T\big) + \int_{\tau}^{T} L\big(x(t),u(t),t\big)dt, \tag{2.8}$$

which tells us how much cost will accumulate if the system is initialized in state $x(\tau)$ at time τ, and controlled according to a certain control law until the end of the movement. By defining the Hamiltonian $\mathcal{H}$ as

$$\mathcal{H} = L\big(x(t),u(t),t\big) + \frac{\partial V^*}{\partial x}\big(x(t)\big)^T\ f\big(x(t),u(t),t\big) \tag{2.9}$$

The optimality condition is

$$\frac{\partial V^*}{\partial t}\big(x(t)\big) = -\min_u \mathcal{H}\Big(x(t),u(t),\frac{\partial V^*}{\partial x}\big(x(t)\big),t\Big), \tag{2.10}$$

$$V^*\big(x(T),T\big) = \Phi\big(x(T),T\big). \tag{2.11}$$

The partial differential equation (2.10) is known as the Hamilton-Jacobi-Bellman (HJB) equation. It provides the rule for solving optimal control problems using dynamic programming. Bellman's optimality principle refers to $V(t,x(t))$ and $u^*(t)$ at all possible states $x(t)$, and therefore leads to global methods that compute an optimal control law. Such methods typically represent the value function on a discrete grid, and use difference equations to compute the value function in a backward pass through time. On a

finite grid, it is guaranteed to converge to the globally optimal solution in finite time. But the problem is that one cannot discretize a high dimensional state space due to the exponential growth of the number of grid points with dimensionality (i.e."the curse of dimensionality").

2.3.3 Riccati Equations

Riccati equations were originally developed in order to theoretically deal with calculus of variations problems. They are easily derived from linear quadratic programming problems. Consider a linear system

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(0) = x_0, \tag{2.12}$$

and the quadratic performance criterion to be minimized is expressed as

$$J_0 = \frac{1}{2}x(T)^T Q_T\, x(T) + \frac{1}{2}\int_0^T \Big(x(t)^T Q x(t) + u(t)^T R u(t)\Big)dt, \tag{2.13}$$

To use the HJB equation, we first define the Hamiltonian $\mathcal{H}$ as

$$\mathcal{H} = \frac{1}{2}\Big(x(t)^T Q x(t) + u(t)^T R u(t)\Big) + \frac{\partial V^*}{\partial x}\big(x(t)\big)^T \Big(Ax(t) + Bu(t)\Big). \tag{2.14}$$

A necessary condition for u(t) to minimize $\mathcal{H}$ is that $\frac{\partial \mathcal{H}}{\partial u} = 0$, thus the optimal control u is

$$u^* = -R^{-1}B^T \frac{\partial V^*}{\partial x}\big(x(t)\big) \tag{2.15}$$

Assuming there is a symmetric matrix $S(t)$ such that $V^*(t) = \frac{1}{2}x(t)^T S(t) x(t)$ for all $t \le T$, substitute it and (2.15) into HJB equation (2.10), we obtain

$$0 = \frac{1}{2}x^T \dot{S} x + \frac{1}{2}x^T Q x - \frac{1}{2}SBR^{-1}B^T Sx + x^T SAx \tag{2.16}$$

Since the above equation must be hold for all $x(t)$, so

$$-\dot{S} = A^T S + SA - SBR^{-1}B^T S + Q, \qquad S(T) = Q_T. \tag{2.17}$$

This matrix differential equation is called the Riccati equation which could be solved by the backward integration. Although the HJB equation is very difficult to solve, it provides a straightforward solution for the linear quadratic-quadratic controller, and it

gives insight regarding the nature of optimal solutions. In recent years, Riccati equations are mainly used as a theoretical tool for the study of stability properties of feedback optimal regulator systems.

2.3.4 Gradient Methods

Gradient methods are characterized by iterative algorithms for improving estimates of the control history $u(t)$ so that the optimality conditions and the boundary conditions are satisfied. A particular appeal of this approach is that, the dynamic system equation is solved exactly on each iteration, with the control being adjusted from step to step to further reduce the cost. Take, for example, the general system (2.4) with the performance index (2.5), the optimality conditions are

$$\dot{\lambda}^T = -\frac{\partial \mathcal{H}}{\partial x}, \qquad \lambda^T(T) = \left.\frac{\partial \Phi\big(x(t),t\big)}{\partial x}\right|_{t=T}, \tag{2.18}$$

$$\frac{\partial \mathcal{H}}{\partial u} = 0. \tag{2.19}$$

where the Hamiltonian is $\mathcal{H} = L\big(x(t),u(t),t\big) + \lambda^T\big[f(x,u,t)\big]$, and λ are lagrange multipliers.

The goal of gradient methods is to use the gradient information to obtain u_{k+1} such that u_k converges to the optimal control solution as $k \to \infty$. General first-order gradient methods that operate on the control variable typically assume the form:

$$u_{k+1} = u_k + \varepsilon_k \nabla_u \mathcal{H}(u_k) \tag{2.20}$$

where ε_k is a step size chosen by an appropriate line search, which is critical for the speed of convergence to the minimizing control; and $\nabla_u \mathcal{H}(u_k)$ is the gradient of the Hamiltonian at u_k, which defines the magnitude and direction of the local slope of the function.

First-order gradient methods usually show great improvements in the first few iterations but have slow convergence near the optimal solution. While the second-order gradient methods have fast convergence characteristics as the optimum is approached.

The conjugate gradient method combines the advantages of both methods while eliminating their disadvantages. It uses conjugate directions instead of the local gradient for going downhill. For example, a sequence of directions $p_1, p_2, \cdots$ is generated which is conjugate with respect to $\nabla_{uu}\mathcal{H}$, that is,

$$p_i^T \nabla_{uu}\mathcal{H} p_j = 0, \qquad i \neq j \tag{2.21}$$

The conjugate gradient method is one of the most popular and effective methods for solving symmetric positive definite systems. It proceeds by generating vector sequences of iterates, residuals corresponding to the iterates, and search directions used in updating the iterates and residuals. In this method, each new residual is orthogonal to all the previous residuals and search directions; and each new search direction is constructed (from the residual) to be conjugate to all the previous residuals and search directions.

Chapter 3

Modelling of Arm Dynamics and Behavior Movements

The major objective of this thesis is to explore efficient computational methods for controlling complex biomechanical systems. While, the development and testing the new algorithms require detailed biomechanical models. In this chapter, we first formulate the kinematics and dynamics of a 2-link arm and then add realistic muscle actuators to it. We then build a much more complicated musculo-skeletal model of the human arm including 24 muscles which can potentially control 5 degrees of freedom. In addition, we discuss a small family of motor tasks which will allow us to test those new algorithms developed in the following chapters. Here we consider tasks such as moving the hand to a specified target location, or possible passing through some intermediate targets — called via points, or avoiding the obstacle during the movement. This family of tasks will allow us to address significant features of human arm movement: the hand path tends to be straight but slightly curved, the velocity profile of the hand trajectory is smooth and bell-shaped; and speed-accuracy trade-offs. Finally we provide some numerical analysis of human movements based on existing optimality models and explain those characteristics of human movements described in section 3.3.

3.1 Planar Model of the Human Arm

3.1.1 The dynamics of a 2-link arm

Consider an arm model with 2 joints (shoulder and elbow), moving in the horizontal plane (Fig. 3.1A). The inverse dynamics is

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) + \mathcal{B}\dot{\theta} = \tau, \tag{3.1}$$

where $\theta \in \mathbb{R}^2$ is the joint angle vector (shoulder: θ_1, elbow: θ_2), $\mathcal{M}(\theta) \in \mathbb{R}^{2\times 2}$ is a positive definite symmetric inertia matrix, $\mathcal{C}(\theta, \dot{\theta}) \in \mathbb{R}^2$ is a vector centripetal and Coriolis forces, $\mathcal{B} \in \mathbb{R}^{2\times 2}$ is the joint friction matrix, and $\tau \in \mathbb{R}^2$ is the joint torque. Here we consider direct torque control where τ is the control signal. In (3.1), the expressions of the different variables and parameters are given by

$$\theta = \begin{pmatrix} \theta_1 & \theta_2 \end{pmatrix}^T, \qquad \tau = \begin{pmatrix} \tau_1 & \tau_2 \end{pmatrix}^T, \tag{3.2}$$

$$\mathcal{M} = \begin{pmatrix} d_1 + 2d_2 cos\theta_2 & d_3 + d_2 cos\theta_2 \\ d_3 + d_2 cos\theta_2 & d_3 \end{pmatrix}, \tag{3.3}$$

$$\mathcal{C} = \begin{pmatrix} -\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{\theta_1}^2 \end{pmatrix} d_2 sin\theta_2, \qquad \mathcal{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \tag{3.4}$$

$$d_1 = I_1 + I_2 + m_2 l_1^2, \qquad d_2 = m_2 l_1 s_2, \qquad d_3 = I_2, \tag{3.5}$$

where $b_{11} = b_{22} = 0.05, b_{12} = b_{21} = 0.025$, m_i is the mass, l_i is the length of link i, s_i is the distance from the joint center to the center of the mass for link i, and I_i is the moment of inertia. See Table 3.1 for detail.

Based on equations (3.1)-(3.5), we can compute the forward dynamics

$$\ddot{\theta} = \mathcal{M}(\theta)^{-1}\Big(\tau - \mathcal{C}(\theta, \dot{\theta}) - \mathcal{B}\dot{\theta}\Big), \tag{3.6}$$

and write the 2-link human arm system into a state space form with the state variable $x \in \mathbb{R}^4$, control input $u \in \mathbb{R}^2$ as

$$\underline{\textbf{MODEL 1}}: \qquad \dot{x} = F(x) + G(x)u, \tag{3.7}$$
$$x = \Big(\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2\Big)^T, \qquad u = \tau = (\tau_1\ \tau_2)^T,$$

Table 3.1. Parameters of 2-link arm

Symbol	Value	Unit
m_1	1.4	kg
m_2	1.1	kg
l_1	0.3	m
l_2	0.33	m
s_1	0.11	m
s_2	0.16	m
I_1	0.025	kg m^2
I_2	0.045	kg m^2

In (3.7), the vector functions $F(x), G(x)$ are given by

$$F(x) = \Big(F_1(x) \;\; F_2(x) \;\; F_3(x) \;\; F_4(x)\Big)^T, \tag{3.8}$$

where

$$F_1(x) = x_3, \qquad F_2(x) = x_4, \tag{3.9}$$

$$\begin{aligned} F_3(x) = \frac{1}{d(x_2)}\Bigg\{ & - d_2 d_3 (x_3 + x_4)^2 sinx_2 - d_2^2 x_3^2 sinx_2 cosx_2 - d_2(b_{21}x_3 + b_{22}x_4)cosx_2 \\ & + (d_3 b_{11} - d_3 b_{21})x_3 + (d_3 b_{12} - d_3 b_{22})x_4 \Bigg\}, \end{aligned} \tag{3.10}$$

$$\begin{aligned} F_4(x) = \frac{1}{d(x_2)}\Bigg\{ & d_2 d_3 x_4 (2x_3 + x_4) sinx_2 + d_1 d_2 x_3^2 sinx_2 + d_2^2 (x_3 + x_4)^2 sinx_2 cosx_2 \\ & + d_2\Big[(2b_{21} - b_{11})x_3 + (2b_{22} - b_{12})x_4\Big]cosx_2 \\ & + (d_1 b_{21} - d_3 b_{11})x_3 + (d_1 b_{22} - d_3 b_{12})x_4 \Bigg\}, \end{aligned} \tag{3.11}$$

$$G(x) = \frac{1}{d(x_2)} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ d_3 & -(d_3 + d_2 cosx_2) \\ -(d_3 + d_2 cosx_2) & d_1 + 2d_2 cosx_2 \end{pmatrix}. \tag{3.12}$$

note that $d(x_2) = d_1 d_3 - d_3^2 - (d_2 cosx_2)^2$ is the determinant of inertia matrix $\mathcal{M}$.

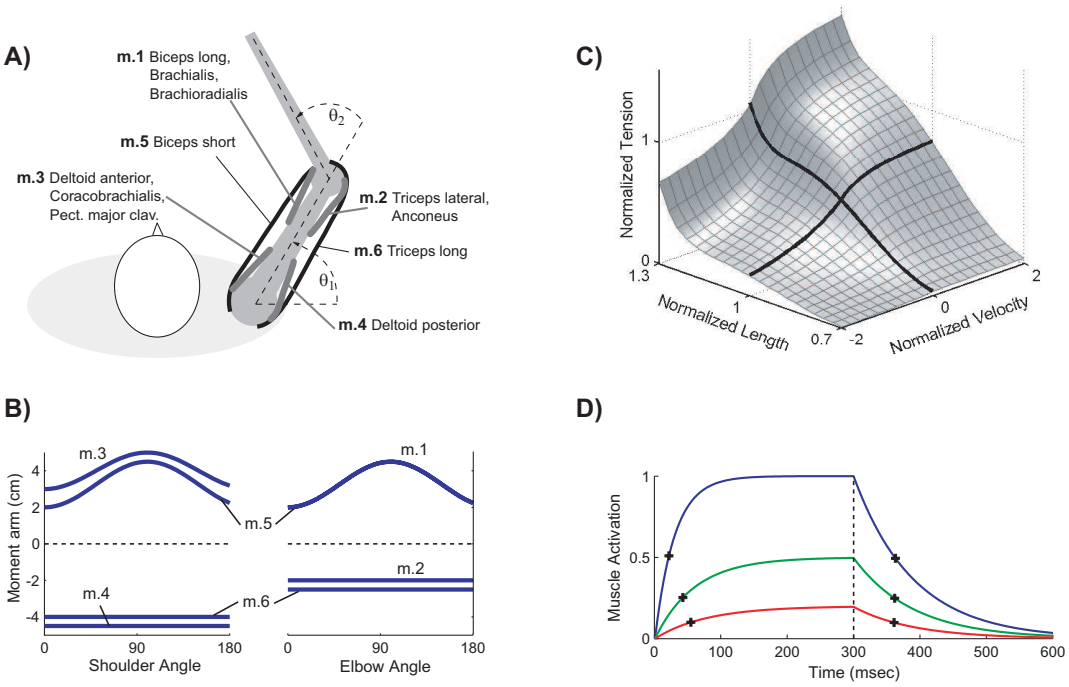


Figure 3.1. (A) 2-link 6-muscle arm; (B) Joint torques; (C) Length-velocity-tension curve; (D) Muscle activation dynamics.

3.1.2 A model of muscle actuators

There are a large number of muscles that act on the arm in the horizontal plane. But since we have only 2 degrees of freedom (Fig. 3.1A), these muscles can be organized into 6 actuator groups: elbow flexors (1), elbow extensors (2), shoulder flexors (3), shoulder extensors (4), biarticulate flexors (5), and biarticulate extensors (6). The joint torques produced by a muscle are a function of its moment arms (Fig. 3.1B), length-velocity-tension curve (Fig. 3.1C), and activation dynamics (Fig. 3.1D), which is given by

$$\tau = M(\theta)\, T\big(a, l(\theta), v(\theta, \dot{\theta})\big). \tag{3.13}$$

The moment arm $M(\theta) \in \mathbb{R}^{2\times 6}$ is defined as the perpendicular distance from the muscle's line of action to the joint's center of rotation. Moment arms are roughly constant for extensor muscles, but vary considerably with joint angle for flexor muscles. For each flexor group, this variation is modelled with a function of the form $a + b\, cos(c\, \theta)$, where the constants have been adjusted to match experimental data. This function provides

Table 3.2. The tension $T(a,l,v)$ produced by a muscle

$$T\left(a,l,v\right)=A\left(a,l\right)\left(F_{\mathrm{L}}\left(l\right)F_{\mathrm{V}}\left(l,v\right)+F_{\mathrm{P}}\left(l\right)\right)$$

$$A\left(a,l\right)=1-\exp\left(-\left(\frac{a}{0.56\ N_{\mathrm{f}}\left(l\right)}\right)^{N_{\mathrm{f}}(l)}\right)$$

$$N_{\mathrm{f}}\left(l\right)=2.11+4.16\left(\frac{1}{l}-1\right)$$

$$F_{\mathrm{L}}\left(l\right)=\exp\left(-\left|\frac{l^{1.93}-1}{1.03}\right|^{1.87}\right)$$

$$F_{\mathrm{V}}\left(l,v\right)=\begin{cases}\dfrac{-5.72-v}{-5.72+\left(1.38+2.09\ l\right)\ v},\ v\leq 0\\ \dfrac{0.62-\left(-3.12+4.21\ l-2.67\ l^{2}\right)\ v}{0.62+v},\end{cases}$$

$$F_{\mathrm{P}}\left(l\right)=-0.02\exp\left(13.8-18.7\ l\right)$$

a good fit to data — not surprising, since moment arm variations are due to geometric factors related to the cosine of the joint angle. It can also be integrated analytically, which is convenient since all muscle lengths need to be computed at each point in time.

The tension produced by a muscle obviously depends on the muscle activation a, but also varies substantially with the length l and velocity v of that muscle. Fig. 3.1C, based on the publicly available Virtual Muscle model [15], illustrates that dependence for maximal activation. We will denote this function with $T(a,l,v)$.

Mammalian muscles are known to have remarkable scaling properties, meaning that they are all similar after proper normalization: length l is expressed in units of L_0, where L_0 is the length at which maximal isometric force is generated, and velocity v is expressed in units of L_0/sec. The unitless tension in Fig. 3.1C is scaled by $31.8N$ per square centimeter of physiological cross-sectional area (PCSA) to yield physical tension T. The PCSA parameters used in the model are the sums of the corresponding parameters for all muscles in each group (1: $18cm^2$; 2: $14cm^2$; 3: $22cm^2$; 4: $12cm^2$; 5: $5cm^2$; 6: $10cm^2$). Muscle length (and velocity) are converted into normalized units of L_0 using information about the operating range of each muscle group (1: $0.6-1.1$; 2: $0.8-1.25$; 3: $0.7-1.2$; 4: $0.7-1.1$; 5: $0.6-1.1$; 6: $0.85-1.2$).

Muscle activation $a_i (i = 1, \cdots, 6)$ is not equal to instantaneous neural input u_i, but is generated by passing u_i through a filter that describes calcium dynamics. This is reasonably well modelled with a first order nonlinear filter of the form

$$\dot{a_i} = \Big((1 + \sigma_u \varepsilon) u_i - a_i \Big) \Big/ t(u_i, a_i), \tag{3.14}$$

where

$$t(u_i, a_i) = \begin{cases} t_{deact} + u_i (t_{act} - t_{deact}), & u_i > a_i, \\ t_{deact}, & \text{otherwise.} \end{cases}$$

The input-dependent activation dynamics $t_{act} = 30 msec$ is faster than the constant deactivation dynamics $t_{deact} = 66 msec$. Fig. 3.1D illustrates the response of this filter to step inputs that last $300 msec$. Note that the half-rise times are input-dependent, while the half-fall times are constant (crosses in Fig. 3.1D). The neural inputs u are disturbed by multiplicative noise, whose standard deviation is 20% of its magnitude — which means $\sigma_u = 0.2$ in (3.14), while ε is a zero-mean Gaussian white noise with unity covariance.

We notice that adding muscles to the dynamical system results in 6 new state variables. Combining the forward dynamics (3.6) and muscle actuator model (3.14), we could write the system into a state space form with the state $x \in \mathbb{R}^{10}$ and control $u \in \mathbb{R}^6$ as follows

$$\underline{\textbf{MODEL 2}}: \qquad \begin{aligned} &\dot{x} = \mathcal{F}(x) + \mathcal{G}(x)(1 + \sigma_u \varepsilon) u, \\ &x = (\theta_1 \ \theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2 \ a_1 \ \cdots \ a_6)^T, \qquad u = (u_1 \ \cdots \ u_6)^T. \end{aligned} \tag{3.15}$$

3.2 3D Musculo-Skeletal Model of the Human Arm

A large number of research and studies has reported on developing biomechanical models of human upper limb, including three chained mechanisms — the shoulder girdle, the elbow and the wrist. The proper description and simulation of the musculoskeletal structure of the upper limb is necessary to predict realistic human movement. Engin et al. in 1987 presented a shoulder model with quantitative descriptions of the individual joint sinus cones [28]. Högfors et al. in 1987 applied the optimization techniques to

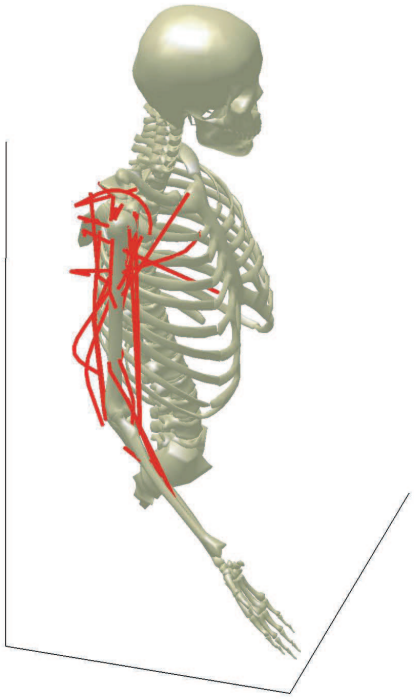


Figure 3.2. Muscle paths of 5-DOF human arm

predict muscle forces as functions of the static arm position and the external loads, which provided an improved description of the shoulder model [43]. A dynamic shoulder model was proposed by van der Helm et al. by means of the finite element method [40], where the bones were modelled as rigid segments connected by ball and socket joints; all muscles and ligaments were modelled as straight or curved lines of action between their connections on the bones. Based on the high resolution images obtained from Visible Human Project (VHP) dataset, in 2001, Garner and Pandy [34] developed a complete biomechanical model of the arm, which included the three-dimensional movements of all the bones of the upper limb. The model enclosed seven anatomical joints from the shoulder girdle down to the wrist, and used thirteen degree-of-freedom to describe the relative position and orientation of seven upper-extremity bones.

To simplify the problem, we approximate the arm segments (upper arm, forearm and palm) by rigid cylinders. Our model (Fig. 3.2) uses five degree-of-freedom to describe the relative movements of the segments: the shoulder is modelled as a three degree-of-freedom joint (only the glenohumeral joint was taken into account here), with flexion-

Table 3.3. Musculoskeletal model of the 5DOF upper limb

	shoulder θ_1	shoulder θ_2	shoulder θ_3	elbow θ_4	elbow θ_5
pectoralis major clavicular	√	√	√		
pectoralis major sternal	√	√	√		
pectoralis major ribs	√	√	√		
latissimus dorsi thoracic	√	√	√		
latissimus dorsi lumbar	√	√	√		
latissimus dorsi iliac	√	√	√		
deltoid clavicular	√	√	√		
deltoid acromial	√	√	√		
deltoid scapular	√	√	√		
supraspinatus	√	√	√		
infraspinatus	√	√	√		
subscapularis	√	√	√		
teres minor	√	√	√		
teres major	√	√	√		
coracobrachialis	√	√	√		
triceps brachii long	√	√	√	√	
triceps brachii medial				√	
triceps brachii lateral				√	
biceps brachii short	√	√	√	√	√
biceps brachii long				√	√
brachialis				√	
brachioradialis				√	√
supinator					√
pronator teres				√	√

extension, abduction-adduction and external-internal rotation; the elbow is modelled as a two degree-of-freedom joint (humeroulnar joint and radioulnar joint), with flexion-extension and pronation-supination movements. The inverse dynamics of 5-DOF arm model is

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) + \mathcal{B}\dot{\theta} = \tau, \tag{3.16}$$

where $\theta \in \mathbb{R}^5$ is the joint angle vector (shoulder: θ_1, θ_2, θ_3; elbow: θ_4, θ_5), $\mathcal{M}(\theta) \in \mathbb{R}^{5\times5}$ is a positive definite symmetric inertia matrix, $\mathcal{C}(\theta, \dot{\theta}) \in \mathbb{R}^5$ is a vector centripetal and Coriolis forces, $\mathcal{B} \in \mathbb{R}^{5\times5}$ is the joint friction matrix, and $\tau \in \mathbb{R}^5$ is the joint torque.

This 5-DOF arm model contains 24 muscles (Table 3.3 & Fig. 3.2), and most of

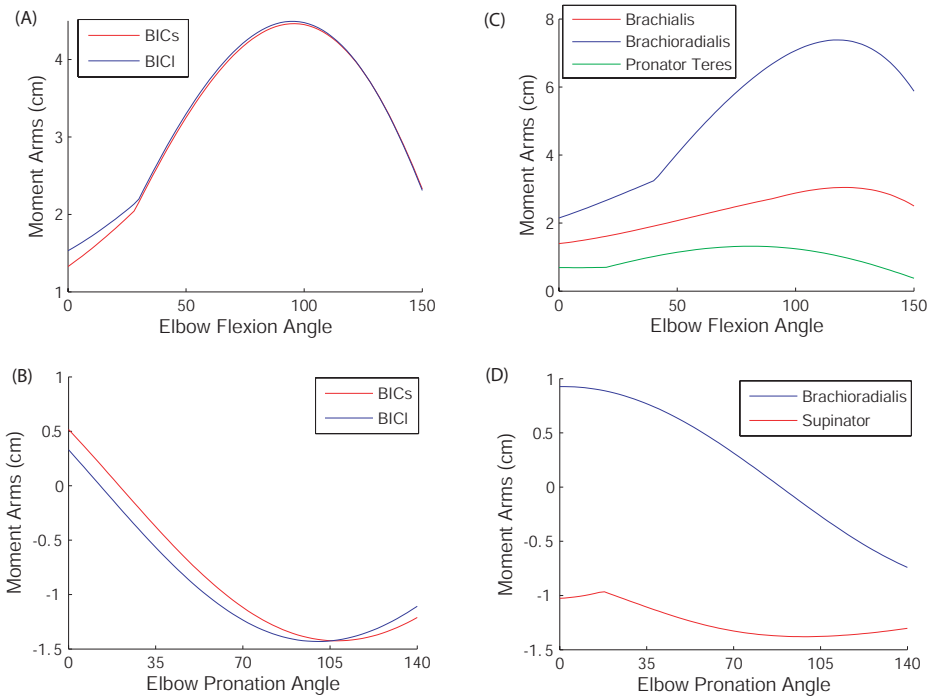


Figure 3.3. Elbow flexion moment arms for (A) biceps brachii, (B) brachialis, brachioradialis and pronator teres ; and forearm pronation moment arms for (C) biceps brachii, (D) brachioradialis and supinator.

muscles in the upper limb potentially control many degrees of freedom, i.e., triceps brachii long spans glenohumeral and humeroulnar joint $(\theta_1, \cdots, \theta_4)$, which together accounts for 4 degrees of freedom in the model. Muscle activation dynamics a_i $(i = 1, \cdots, 24)$ is modelled as a nonlinear low-pass filter (Eq. 3.14) as described in section 3.1.2. The geometry of muscles that wrap around a skeleton in 3-dimensional is rather complex. The Muscle Wrapping Toolbox developed in our lab can compute musculoskeletal geometry based on the obstacle-set method [33]. The toolbox computes the shortest muscle paths that satisfy all obstacle constraints, which then yields the posture-specific moment arms (i.e., Fig. 3.3) and muscle lengths needed for dynamic simulation. While the dynamic simulation of serial-link upper limb is implemented using Matlab Robotics Toolbox.

3.3 Representation of Different Behavior Movements with Cost Functions

Most existing optimal control models [30, 38, 42, 60, 72, 84, 115, 125] in arm movements, eye movements and some other body movements predict average movement trajectories or muscle activity, by optimizing a variety of cost functions. For example, smoothness optimization introduced in the minimum-jerk model [30, 42] successfully predicted the average trajectories in arm movement, where it accounts for the straight paths and bell-shaped speed profiles of reaching movements. Ideally, the performance criterion assumed in an optimal control model should correspond to what the sensorimotor system is trying to achieve. But how could this be quantified? One can simply optimize whatever subjects were asked to optimize, under the constraint that each muscle can produce limited force in order to satisfy the energy consumption requirement. However, in many cases, the cost function that is relevant to the sensorimotor function may not directly correspond to our intuitive understanding of "the task".

Computational modelling here aims to construct the feedback control strategy that yields the best possible performance when motor noise as well as sensory uncertainty and delays are taken into account. In the following section, we illustrate several cost functions relevant to different tasks that we are interested in. These cost functions will take into account accuracy and energetic efficiency.

3.3.1 Reaching task

All tasks considered here are spatial. Spatial accuracy is defined in terms of hand position and sometimes its derivatives. The task we study first is a reaching task, where the arm has to start at some initial position and move to a target in a specified time interval. It also has to stop at the target, and do all that with minimal energy consumption. There are good reasons to believe that such costs are indeed relevant to the neural control of movement [116]. For the 2-link arm model shown in Fig. 3.1A, the cost function for reaching task will depend on the vector of joint angles $[\theta_1\ \theta_2]^T$ only through

the forward kinematic function $e(\theta)$, where e and $\dot{e}$ are the 2D vectors of Cartesian hand coordinates computed as the following

$$e(\theta) = \begin{pmatrix} l_1\ cos\theta_1 + l_2\ cos(\theta_1 + \theta_2) \\ l_1\ sin\theta_1 + l_2\ sin(\theta_1 + \theta_2) \end{pmatrix}, \tag{3.17}$$

$$\dot{e}\left(\theta, \dot{\theta}\right) = \Gamma \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \tag{3.18}$$

$$\Gamma = \frac{\partial e(\theta)}{\partial \theta} = \begin{pmatrix} -l_1\ sin\theta_1 - l_2\ sin(\theta_1 + \theta_2) & -l_2\ sin(\theta_1 + \theta_2) \\ l_1\ cos\theta_1 + l_2\ cos(\theta_1 + \theta_2) & l_2\ cos(\theta_1 + \theta_2) \end{pmatrix}, \tag{3.19}$$

where l_i is the length of link i. Then the cost function is defined as

$$J_1 = w_1 \left\| e(\theta(T)) - e^* \right\|^2 + w_2 \left\| \dot{e}\left(\theta(T), \theta(\dot{T})\right) \right\|^2 + \frac{1}{2} \int_0^T r\|u\|^2\, dt, \tag{3.20}$$

where e^* is the desired target position defined in Cartesian hand coordinates, the second term $\left\| \dot{e}\left(\theta(T), \theta(\dot{T})\right) \right\|^2$ enforces stopping at the target. The two constants w_1, w_2 specify the importance of positional versus velocity accuracy, and r is the weight of the energy term.

3.3.2 Reaching with obstacle avoidance

The second task is to implement the reaching and to avoid an obstacle during the movement. The obstacle is defined as a circle with a certain radius $r_{obstacle}$, and has fixed location. The arm starts from rest at some initial position, and has to reach the specified target and avoid the obstacle during the reaching, with minimal control energy. The cost function is

$$J_2 = w_1 \left\| e(\theta(T)) - e^* \right\|^2 + w_2 \left\| \dot{e}\left(\theta(T), \theta(\dot{T})\right) \right\|^2 + \frac{1}{2} \int_0^T r\|u\|^2\, dt + q_{obstable}, \tag{3.21}$$

$$q_{obstable} = \begin{cases} k1 \int_0^T \left(\ell(\theta(t)) - r_{obstacle}\right)^{-2} dt & when\ l > r \\ \infty & otherwise \end{cases}$$

where the target e^* is defined in Cartesian hand coordinates; $\ell = \sqrt{\|e(\theta(t)) - e_{obstacle}\|^2}$ is the distance between the hand position and the center of the obstacle center, where $e_{obstacle}$ is the Cartesian coordinates of the center of the obstacle; w_1, w_2, r and k_1 are the weighting coefficients.

3.3.3 Hitting task

The task of hitting will be modeled by specifying a target position e and a desired velocity $\dot{e}$ at the end of the movement. The cost function now becomes

$$J_3 = w_1 \left\| e(\theta(T)) - e^* \right\|^2 + w_2 \left\| \dot{e}\left(\theta(T), \dot{\theta(T)}\right) - \dot{e}^* \right\|^2 + \frac{1}{2}\int_0^T r\|u\|^2\, dt, \tag{3.22}$$

where e^* is the desired target position defined in Cartesian hand coordinates, $\dot{e}^*$ is the desired velocity at the end of movement, r is the weighting coefficient of energy term; the two constants w_1, w_2 specify the importance of positional versus velocity accuracy. Note that the final velocity is non-zero, and so the hand cannot stay at the target.

3.3.4 Via-point task

The task of passing through via-points, with positions $e_1^* \cdots e_{n_v}^*$ and passage times $t_1^* \cdots t_{n_v}^*$, can be encoded with the cost

$$J_4 = w_1 \left\| e(\theta(t_i^*)) - e_i^* \right\|^2 + \frac{1}{2}\int_0^T r\|u\|^2\, dt, \tag{3.23}$$

If in addition we want the hand to stop at the final target, or to hit the final target with specified velocity, the corresponding cost can be added.

3.4 Relevant Features of Human Movement

Considering unconstrained point-to-point reaching movement in the horizontal plane, humans tend to generate roughly straight hand trajectories with single-peaked, bell-shaped speed profiles. Several models have been proposed to explain these characteristics [30, 84, 125, 126]. Slightly curved hand path does occur, depending on the area of the workspace in which the movement is performed. When subjects were instructed to move while avoiding an obstacle or through an intermediate point, the hand path appeared to be curved and the speed profile had often several peaks. In this case, the minima between two adjacent speed peaks correspond to distinct curvature peaks in the curved path.

In moving the arm from one target to another, one usually has to make some compromise between the duration and the precision of the movement. This speed-accuracy trade-off phenomenon, known as Fitts' law (3.24), states that the faster the movement, the less accurately will reach the target [29]:

$$T = a + b\left(log_2 \left(\frac{2A}{W} \right) \right) \tag{3.24}$$

In (3.24), T is the movement time, a and b are regression coefficients, A is the distance of movement from starting point to target center, and W is the target width corresponding to "accuracy". The quantity $log_2 \left(\frac{2A}{W} \right)$ is called the index of difficulty (ID) which describes the difficulty of the motor tasks. Fitts' law has been applied to hundreds of subsequent studies in human-computer interaction(HCI) and graphical user interfaces. Its first HCI application was a major factor leading to the mouse's commercial introduction. Often, a relatively fast inaccurate movement will be followed by short corrective movement in order to bring the limb back to the target. This trade-off has also been extensively studied for a variety of goal-directed reaching movements in the literature.

Another important aspect of human movement is an inverse non-linear relationship between the velocity and the curvature of its trajectory during the movement, widely referred to as the two-thirds power law [49]. This relationship is formalized by the following:

$$\omega = c\, \kappa^{\beta} \tag{3.25}$$

Equivalently,

$$v = c\, r^{1-\beta} \tag{3.26}$$

In (3.25), ω is the angular velocity of the hand, κ is the instantaneous curvature of the path, c is a proportionality constant, and the accepted value of the exponent $\beta = \frac{2}{3}$. In (3.26), v is the tangential velocity, r is radius of curvature $(1/\kappa)$. It has been shown to exist not only in hand movement, but also in eye-motion, locomotion and was even demonstrated in motion perception and prediction. People argue that the two-thirds power law can be derived from smoothness or smoothness inducing mechanisms or to the result of noise inherent in the motor system.

A majority of computational models that seek to capture these features are based on the assumption that they arise from the optimization of some criteria by the sensorimotor system. In the following section, we will provide some general analysis of human arm movements and explain those characteristics described so far based on existing optimization models: minimum control energy, minimum torque-change (Eq. 2.3) and the well-known minimum jerk models (Eq. 2.1).

3.5 Movement Analysis with Existing Optimality Models

Figure 3.4 shows hand paths and velocity profiles for ten unconstrained horizontal movements produced by the minimum torque-change model with 2-link 6-Muscle Arm. Compared with the results of minimum jerk model—whose hand trajectory are strictly straight [42] and lack the typical curvature of human movements, the trajectories of minimum torque-change model are closer to those produced by humans than that of the purely kinematic minimum jerk model: they are not completely straight but are instead slightly curved as shown in Fig. 3.4(a); the associated speed profiles (Fig. 3.4(b)) were single-peaked and bell-shaped. For the minimum control effort model, the simulation trajectories and velocity profiles shown in Fig. 3.5 are the same as those of the minimum torque-change model. These predicted trajectories were consistent with the experimental data reported in [82].

Figures 3.6 and 3.7 describe results from simulation of free movements between two targets passing through a via-point P1 or P2. Both the minimum torque-change model and the minimum energy model predict gently curved hand paths with double-peaked speed profiles. In this group of movements, we consider two subcases, with identical starting point (shown as circle) and the target (shown as star), but with mirror-image via-points (see Fig. 3.6(a) and Fig. 3.7(a) shown as cross). That is, the two via-points P1 and P2 are located symmetrically with respect to the line connecting the common starting point and the target. Both models generate two different shapes of trajectories corresponding to the two subcases; for the movement passing through the via-point P1, a convex curved path is formed; in contrast, for the movement passing through the via-

point P2, a concave path is formed. In particular, the convex path and the concave path are not symmetric with respect to the line connecting the starting point and the target. The associated speed profile has two peaks as illustrated in Fig. 3.6(b) and Fig. 3.7(b), which are consistent with experimental data shown in [84, 125].

Furthermore, we notice that changing the location of the via-point results in either a single-peak or double-peak hand speed profile (Fig. 3.8 and Fig. 3.9). Here the trajectories are generated by moving the hand between two targets and passing through via-points P0, P1 and P2 respectively within 250ms. In both the models, if the via-point P0 is located near to the line connecting the initial and the final targets, the hand path is roughly straight, and the hand speed profile is single peaked (shown as dotted line); on the other hand, if the via-point P1 or P2 is located further away from the line connecting two targets, highly curved movements are produced and the hand speed profiles are double-peaked (shown as solid and diamond curves).

Finally, we illustrate another kinematic feature related to the depth of velocity valley and the height of the curvature peak. It is easy to see that both the hand paths and velocity profiles are identical for the minimum torque-change model (Fig. 3.10) and the minimum energy model (Fig. 3.11). Here three movements with the same 500ms duration are simulated, the only difference is the timing to pass through the via-point P1: one is 150ms (dashed curve), and the other two are 250ms (solid curve) and 320ms (diamond curve) respectively. We notice that the resulting three trajectories are clearly different (Fig. 3.10(a)). Since the via-point P1 is located symmetrically between the starting point and target, when the passing time is 150ms, the first part of the movement (from starting point to via-point P1) has to be faster than the other two cases; hence the corresponding speed profile has the highest peak (yellow dashed curve), while the hand path is less curved. After passing through the via-point, there is plenty of time left (350ms) to reach the target, therefore this slower movement results in much more curved hand path compared with the other two cases. And we can conclude that, for the highly curved movements with double-peaked speed profiles, the speed valley is temporally associated with the curvature peak. The larger the curvature of hand path, the deeper the valley in the double-peaked speed profile (Fig. 3.10(b)).

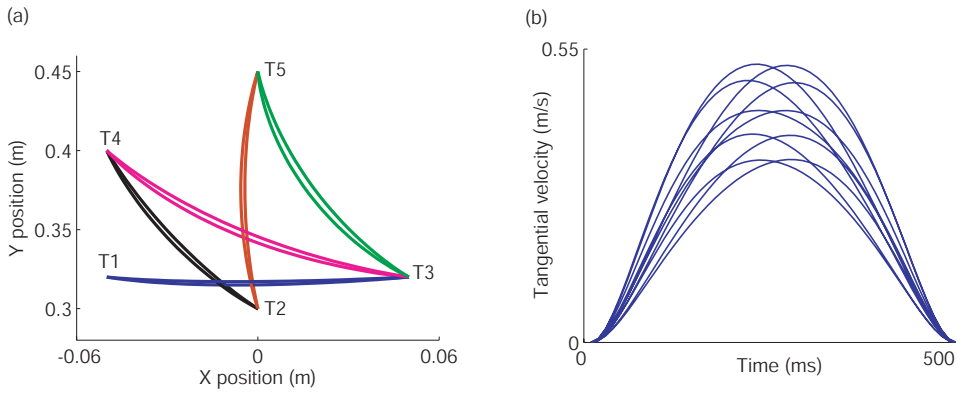


Figure 3.4. Typical trajectories generated by the minimum torque-change model for free movements between two targets: (a) Hand paths (T1 ↔ T3, T4 ↔ T2, T4 ↔ T3, T5 ↔ T2, T5 ↔ T3. (b) Corresponding hand tangential velocity profiles along the paths.

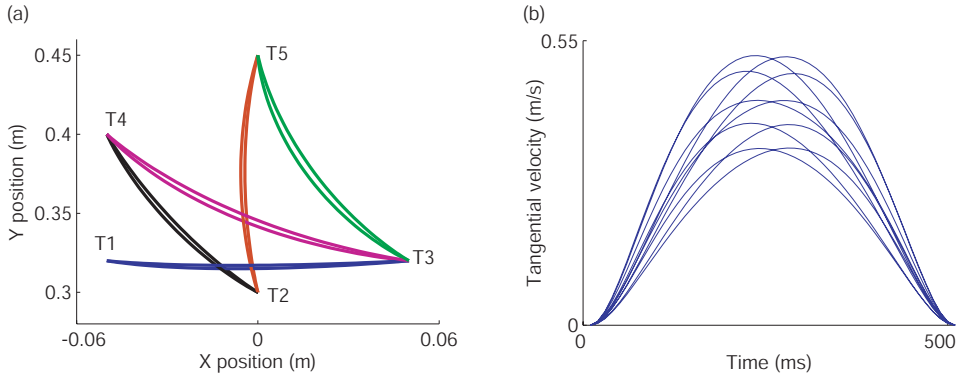


Figure 3.5. Typical trajectories generated by the minimum energy model for free movements between two targets: (a) Hand paths and (b) Tangential velocity profiles.

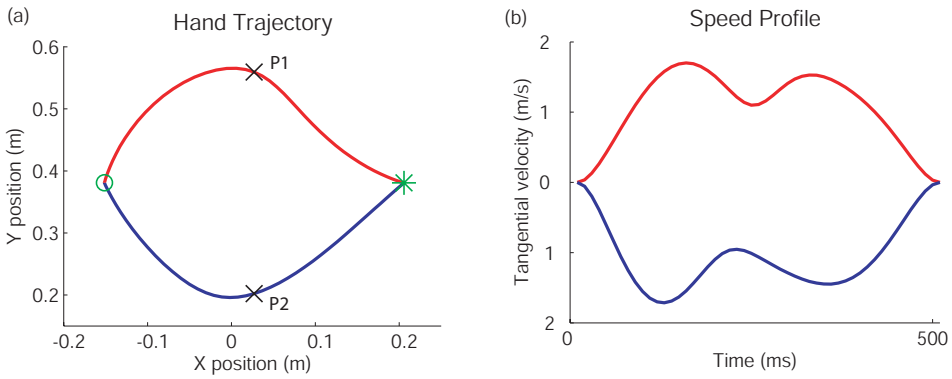


Figure 3.6. Typical trajectories generated by the minimum torque-change model for free movements passing through a via-point P1 or P2 within 250ms: (a) Hand paths (Circle → P1 → Star and Circle → P2 → Star). (b) Corresponding hand tangential velocity profiles.

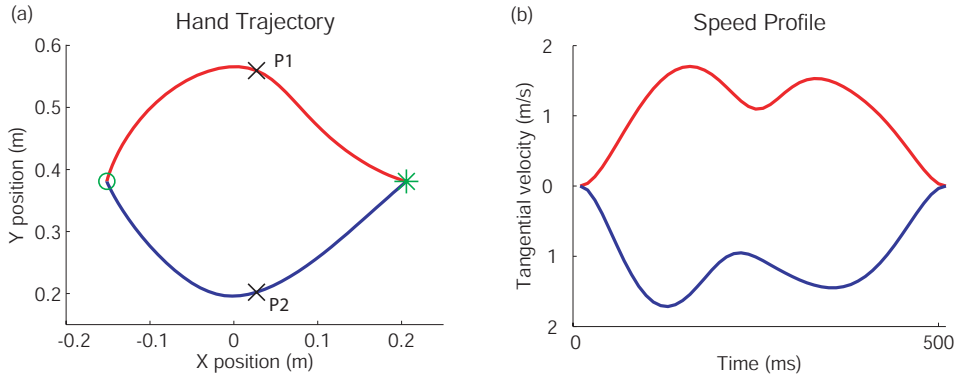


Figure 3.7. Typical trajectories generated by the minimum energy model for free movements passing through a via-point P1 or P2 within 250ms: (a) Hand paths (Circle → P1 → Star and Circle → P2 → Star). (b) Corresponding hand tangential velocity profiles.

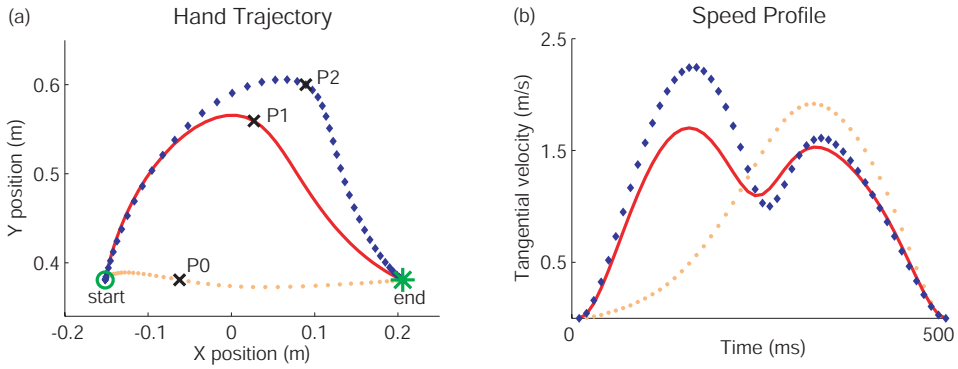


Figure 3.8. Typical trajectories generated by the minimum torque-change model for free movements passing through via-points P0, P1 and P2 respectively within 250ms: (a) Hand paths. (b) Corresponding hand tangential velocity profiles.

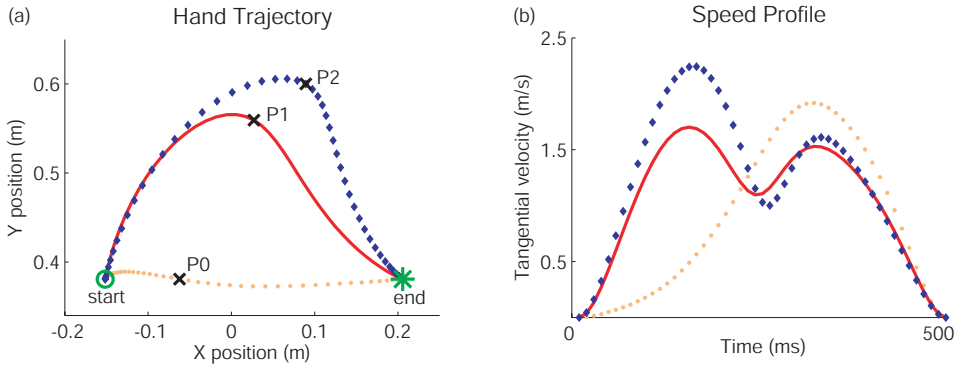


Figure 3.9. Typical trajectories generated by the minimum energy model for free movements passing through via-points P0, P1 and P2 respectively within 250ms: (a) Hand paths. (b) Corresponding hand tangential velocity profiles.

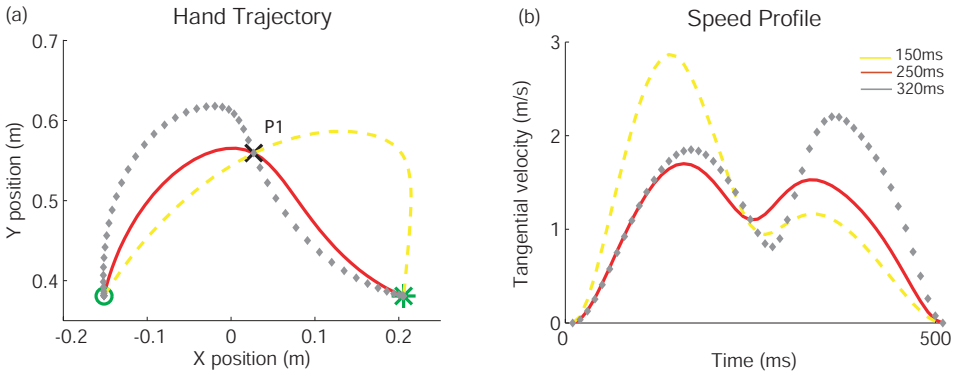


Figure 3.10. Typical trajectories generated by the minimum torque-change model for free movements passing through a via-point P1 within different timing: (a) Hand paths (Solid line: passing through P1 before 250ms; Dashed line: passing through P1 before 150ms; Diamond line: passing through P1 before 320ms). (b) Corresponding hand tangential velocity profiles.

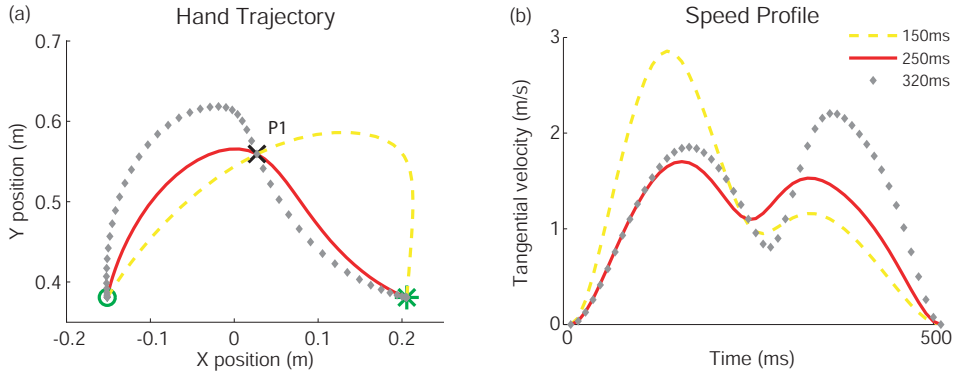


Figure 3.11. Typical trajectories generated by the minimum energy model for free movements passing through a via-point P1 within different timing: (a) Hand paths (Solid line: passing through P1 before 250ms; Dashed line: passing through P1 before 150ms; Diamond line: passing through P1 before 320ms). (b) Corresponding hand tangential velocity profiles.

Chapter 4

Iterative Linear Quadratic Design for Arm Movement

4.1 Overview

The majority of present optimality models in motor control have a serious limitation — they rely on the Linear-Quadratic-Gaussian formalism, while in reality biomechanical systems are strongly non-linear. In this chapter we restrict our attention to developing an iterative linear quadratic regulator method for optimal feedback control of nonlinear biomechanical system. This method uses iterative linearization of the nonlinear system around a nominal trajectory, and computes a locally optimal feedback control law via a modified LQR technique. The control law is then applied to the linearized system, and the result is used to improve the nominal trajectory incrementally. We then apply the new algorithm on a realistic 2-Link 6-Muscle biomechanical model of the human arm, as well as two simpler dynamical systems: 2-Link torque controlled arm and inverted pendulum. Comparisons with three existing methods (Ordinary Differential Equations, Conjugate Gradient Descent and Differential Dynamic Programming) demonstrate that this method converges substantially faster and finds slightly better solutions.

4.2 ILQR Approach to Nonlinear Deterministic Systems

Consider a discrete time nonlinear dynamical system with state variable $x_k \in \mathbb{R}^{n_x}$ and control $u_k \in \mathbb{R}^{n_u}$

$$x_{k+1} = f(x_k, u_k). \tag{4.1}$$

The cost function is written in the quadratic form

$$J_0 = \frac{1}{2}(x_N - x^*)^T Q_f (x_N - x^*) + \frac{1}{2}\sum_{k=0}^{N-1}\Big(x_k^T Q x_k + u_k^T R u_k\Big), \tag{4.2}$$

where x_N describes the final state (each movement lasts N steps), x^* is the given target. The state cost-weighting matrices Q and Q_f are symmetric positive semi-definite, the control cost-weighting matrix R is positive definite. All these matrices are assumed to have proper dimensions. Note that when the true cost is not quadratic, we can still use a quadratic approximation to it around a nominal trajectory.

Our algorithm is iterative. Each iteration starts with a nominal control sequence u_k, and a corresponding nominal trajectory x_k obtained by applying u_k to the dynamical system in open loop. When good initialization is not available, one can use $u_k = 0$. The iteration produces an improved sequence u_k, by linearizing the system dynamics around u_k, x_k and solving a modified LQR problem. The process is repeated until convergence. Let the deviations from the nominal u_k, x_k be $\delta u_k, \delta x_k$. The linearization is

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k, \tag{4.3}$$

where $A_k = D_x f(x_k, u_k), B_k = D_u f(x_k, u_k)$. D_x denotes the Jacobian of $f(\cdot)$ with respect to x, D_u denotes the Jacobian of $f(\cdot)$ with respect to u, and the Jacobians are evaluated along x_k and u_k.

Based on the linearized model (4.3), we can solve the following LQR problem with the cost function

$$\begin{aligned} J &= \frac{1}{2}(x_N + \delta x_N - x^*)^T Q_f\,(x_N + \delta x_N - x^*) \\ &\quad + \frac{1}{2}\sum_{k=0}^{N-1}\Big\{(x_k + \delta x_k)^T Q\,(x_k + \delta x_k) + (u_k + \delta u_k)^T R\,(u_k + \delta u_k)\Big\}. \end{aligned} \tag{4.4}$$

We begin with the Hamiltonian function

$$\begin{aligned} H_k &= \frac{1}{2}(x_k + \delta x_k)^T Q\ (x_k + \delta x_k) + \frac{1}{2}(u_k + \delta u_k)^T R\ (u_k + \delta u_k) \\ &\quad + \delta\lambda_{k+1}^T (A_k \delta x_k + B_k \delta u_k), \end{aligned} \tag{4.5}$$

where $\delta\lambda_{k+1}$ is Lagrange multiplier.

The optimal control improvement δu_k is given by solving the state equation (4.3), the costate equation

$$\delta\lambda_k = A_k^T \delta\lambda_{k+1} + Q(\delta x_k + x_k), \tag{4.6}$$

and the stationary condition which can be obtained by setting the derivative of Hamiltonian function with respect to δu_k to zero

$$0 = R(u_k + \delta u_k) + B_k^T \delta\lambda_{k+1} \tag{4.7}$$

with the boundary condition

$$\delta\lambda_N = Q_f(x_N + \delta x_N - x^*). \tag{4.8}$$

Solving for (4.7) yields

$$\delta u_k = -R^{-1} B_k^T \delta\lambda_{k+1} - u_k. \tag{4.9}$$

Hence, substituting (4.9) into (4.3) and combining it with (4.6), the resulting Hamiltonian system is

$$\begin{pmatrix} \delta x_{k+1} \\ \delta\lambda_k \end{pmatrix} = \begin{pmatrix} A_k & -B_k R^{-1} B_k^T \\ Q & A_k^T \end{pmatrix} \begin{pmatrix} \delta x_k \\ \delta\lambda_{k+1} \end{pmatrix} + \begin{pmatrix} -B_k u_k \\ Q x_k \end{pmatrix}. \tag{4.10}$$

It is clear that the Hamiltonian system is not homogeneous, but is driven by a forcing term dependent on the current trajectory x_k and u_k. Because of the forcing term, it is not possible to express the optimal control law in linear state feedback form (as in the classic LQR case). However, we can express δu_k as a combination of a linear state feedback plus additional terms, which depend on the forcing function.

Based on the boundary condition (4.8), we assume

$$\delta\lambda_k = S_k \delta x_k + v_k \tag{4.11}$$

for some unknown sequences S_k and v_k. Substituting the above assumption into the state and costate equation, and applying the matrix inversion lemma yields the optimal controller.

Theorem 1 *Given the system $x_{k+1} = f(x_k, u_k)$ and its linearization around the nominal trajectory $\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k$ with the performance index given in (4.4), the optimal controller is given by*

$$\begin{aligned}
\delta u_k &= -K\delta x_k - K_v v_{k+1} - K_u u_k, && (4.12)\\
K &= (B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1} A_k, && (4.13)\\
K_v &= (B_k^T S_{k+1} B_k + R)^{-1} B_k^T, && (4.14)\\
K_u &= (B_k^T S_{k+1} B_k + R)^{-1} R, && (4.15)\\
S_k &= A_k^T S_{k+1}(A_k - B_k K) + Q, && (4.16)\\
v_k &= (A_k - B_k K)^T v_{k+1} - K^T R u_k + Q x_k && (4.17)
\end{aligned}$$

with boundary conditions

$$S_N = Q_f, \qquad v_N = Q_f(x_N - x^*). \tag{4.18}$$

Proof 1 *In order to find the equations (4.12)-(4.17), use (4.11) in the state equation (4.3) to yield*

$$\delta x_{k+1} = (I + B_k R^{-1} B_k^T S_{k+1})^{-1}(A_k \delta x_k - B_k R^{-1} B_k^T v_{k+1} - B_k u_k). \tag{4.19}$$

Substituting (4.11) and the above equation into the costate equation (4.6) gives

$$\begin{aligned}
S_k \delta x_k + v_k =& Q\delta x_k + A_k^T S_{k+1}(I + B_k R^{-1} B_k^T S_{k+1})^{-1} \\
& (A_k \delta x_k - B_k R^{-1} B_k^T v_{k+1} - B_k u_k) + A_k^T v_{k+1} + Q x_k.
\end{aligned}$$

By applying the matrix inversion lemma [1] *to the above equation, we obtain*

$$S_k = A_k^T S_{k+1}\Big[I - B_k(B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1}\Big] A_k + Q,$$

[1] $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$.

and

$$\begin{aligned} v_k =& A_k^T v_{k+1} - A_k^T S_{k+1}\Big[I - B_k(B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1}\Big] B_k R^{-1} B_k^T v_{k+1} \\ & - A_k^T S_{k+1}\Big[I - B_k(B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1}\Big] B_k u_k + Q x_k. \end{aligned}$$

By using $(R + B_k^T S_{k+1} B_k)^{-1} = R^{-1} - (R + B_k^T S_{k+1} B_k)^{-1} B_k^T S_{k+1} B_k R^{-1}$, *the second term in* v_k *becomes*

$$-A_k^T S_{k+1} B_k (R + B_k^T S_{k+1} B_k)^{-1} B_k^T v_{k+1},$$

while the third term in v_k *can also be written as*

$$-A_k^T S_{k+1} B_k (R + B_k^T S_{k+1} B_k)^{-1} R u_k.$$

Therefore, with the definition of K *in (4.13), the above* S_k, v_k *can be written into the forms as given in (4.16) and (4.17).*

Furthermore, substituting (4.11) and (4.19) into (4.9) yields

$$\begin{aligned} \delta u_k =& -(R + B_k^T S_{k+1} B_k)^{-1} B_k^T S_{k+1} A_k \delta x_k - (R + B_k^T S_{k+1} B_k)^{-1} B_k^T v_{k+1} \\ & - (R + B_k^T S_{k+1} B_k)^{-1} R u_k. \end{aligned}$$

By the definition of K, K_v *and* K_u *in (4.13)-(4.15), we can rewrite above* δu_k *as the form in (4.12).* ■

With the boundary condition S_N given as the final state weighting matrix in the cost function (4.4), we can solve for an entire sequence of S_k by the backward recursion (4.16). It is interesting to note that the control law δu_k consists of three terms: a term linear in δx_k whose gain is dependent on the solution to the Riccati equation; a second term dependent on an auxiliary sequence v_k which is derived from the auxiliary difference equation (4.17); and a third term dependent on the nominal control u_k whose gain also relied on the Riccati equation solution. Once the modified LQR problem is solved, an improved nominal control sequence can be found: $u_k^* = u_k + \delta u_k$.

4.3 Application to Nonlinear System

4.3.1 Optimal control problems to be studied

We have implemented a 2-DOF torque controlled model and a 2-DOF 6-muscle model of the human arm in the horizontal plane at chapter 3. Here we also include an inverted pendulum problem often used for numerical comparisons.

2-link Torque-controlled and 2-link Muscle-controlled arm

Using the standard equation of motion for a 2-link arm given in (3.6), the dynamics of this arm model can be rewritten here in a state space form

$$\dot{x} = F(x) + G(x)u,$$

For the torque controlled arm model (**MODEL 1**), the state and control are given by

$$x = (\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2)^T, \qquad u = (\tau_1\ \tau_2)^T.$$

while for the 2-link 6-muscle arm model (**MODEL 2**), because muscles have activation states, the state vector is 10-dimensional and the control vector is 6-dimensional

$$x = (\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2\ a_1 \cdots a_6)^T, \qquad u = (u_1 \cdots u_6)^T.$$

The task we study is a reaching task, where the arm has to start at some initial position and move to a target in a specified time interval. It also has to stop at the target, and do all that with minimal energy consumption. The cost function is redefined as

$$J_0 = \frac{1}{2}\big(\theta(T) - \theta^*\big)^T\big(\theta(T) - \theta^*\big) + \frac{1}{2}\int_0^T ru^T u\, dt, \tag{4.20}$$

where $r = 0.0001$ and θ^* is the desired target posture. In the definition of the cost function, the first term means that the joint angle is going to the target θ^* which represents the reaching movement; the second term illustrates the energy efficiency.

Inverted Pendulum

Consider a simple pendulum where m denotes the mass of the bob, l denotes the length of the rod, θ describes the angle subtended by the vertical axis and the rod, and

μ is the friction coefficient. For this example, we assume that $m = 1kg$, $l = 1m, g = 9.8m/s^2, \mu = 0.01$. The state space equation of the pendulum is

$$\dot{x}_1 = x_2, \tag{4.21}$$

$$\dot{x}_2 = \frac{g}{l} sin x_1 - \frac{\mu}{ml^2} x_2 + \frac{1}{ml^2} u, \tag{4.22}$$

where the state variables are $x_1 = \theta$, $x_2 = \dot{\theta}$. The goal is to make the pendulum swing up. The control objective is to find the control $u(t)$, $0 < t < T$ and minimize the performance index

$$J_0 = \frac{1}{2}\big(x_1(T)^2 + x_2(T)^2\big) + \frac{1}{2}\int_0^T ru^2 dt, \tag{4.23}$$

where $r = 1e - 5$.

4.3.2 Optimal Trajectories

Here we illustrate the optimal trajectories found by iterating equations (4.12)-(4.18) on each of the three control problems. Fig. 4.1A and Fig. 4.1B show the optimal trajectory of the arm joint angles θ_1 (shoulder angle) and θ_2 (elbow angle). We find that the shoulder angle and the elbow angle arrive to the desired posture $\theta_1 = 1rad, \theta_2 = 1.5rad$ respectively. Fig. 4.1C shows a set of optimal trajectories in the phase space, for a pendulum being driven from different starting points to the goal point. For example, S2 describes a starting point where the pendulum is hanging straight down; trajectory 2 shows that the pendulum swing directly up to the goal.

For the 2 link muscle-controlled arm model, Fig. 4.2 illustrates how the current joint space trajectory converges with the number of iterations. Although the iteration starts from poor initial control — which produces an arbitrary movement (it 0), it has the rapid improvement and the ILQR method can find a good control in about 5 iterations.

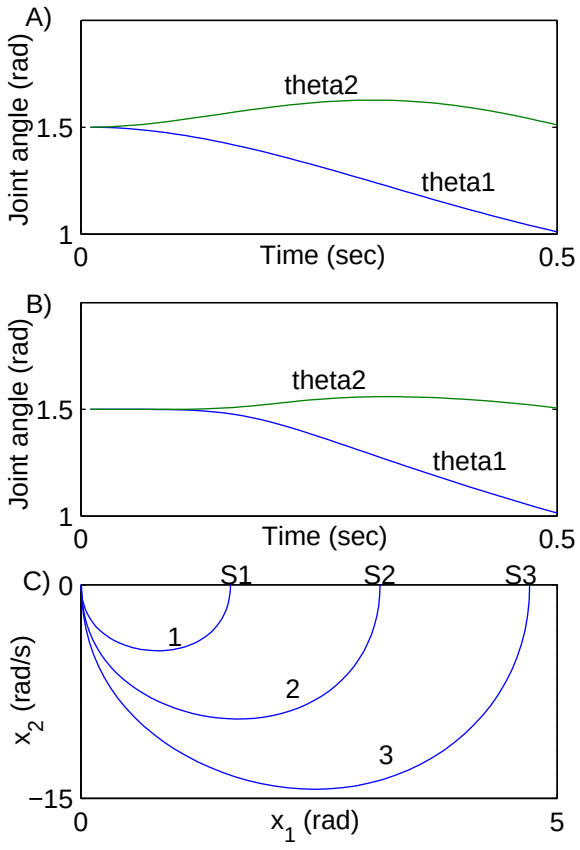


Figure 4.1. Optimal trajectories. (A) 2 Link torque-controlled arm; (B) 2 Link muscle-controlled arm; (C) Inverted pendulum.

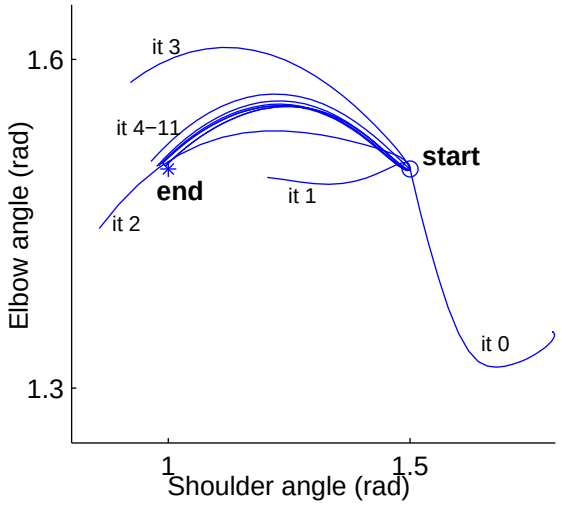


Figure 4.2. Trajectories of 2-link 6-muscle arm for different iterations

4.4 Comparison with Existing Local Algorithms

Existing algorithms for nonlinear optimal control can be classified in two groups, based respectively on Bellman's Optimality Principle and Pontryagin's Maximum Principle [16, 63]. The former yields globally optimal solutions, but involves a partial differential equation (Hamilton-Jacobi-Bellman equation) which is only solvable for low-dimensional systems. While various forms of function approximation have been explored, presently there is no known cure for the curse of dimensionality. Since the biological control problems we are interested in tend to have very high dimensionality (the 10 dim arm model is a relatively simple one), we do not believe that global methods will be applicable to such problems in the near future.

Therefore we have chosen to pursue local trajectory-based methods related to the Maximum Principle. These methods iteratively improve their estimate of the extremal trajectory. We compare: (1) ODE solves the system of state-costate ordinary differential equations resulting from the Maximum Principle, using the BVP4C boundary value problem solver in Matlab; (2) CGD is a gradient descent method, which uses the Maximum Principle to compute the gradient of the total cost with respect to the nominal control sequence, and then calls an optimized conjugate gradient descent routine; (3) differential dynamic programming (DDP) [45] performs dynamic programming in the neighborhood of the nominal trajectory, using second order approximations.

All algorithms were implemented in Matlab, and used the same dynamic simulation. Table 4.1 compares the CPU time and number of iterations for all algorithms on all three problems. Note that the time per iteration varies substantially (and in the case of ODE the number of iterations is not even defined) so the appropriate measure of efficiency is the CPU time. On all problems studied, the new ILQR method converged faster than the three existing methods, and found a better solution. The speed advantage is most notable in the complex arm model, where ILQR outperformed the nearest competitor by more than a factor of 10.

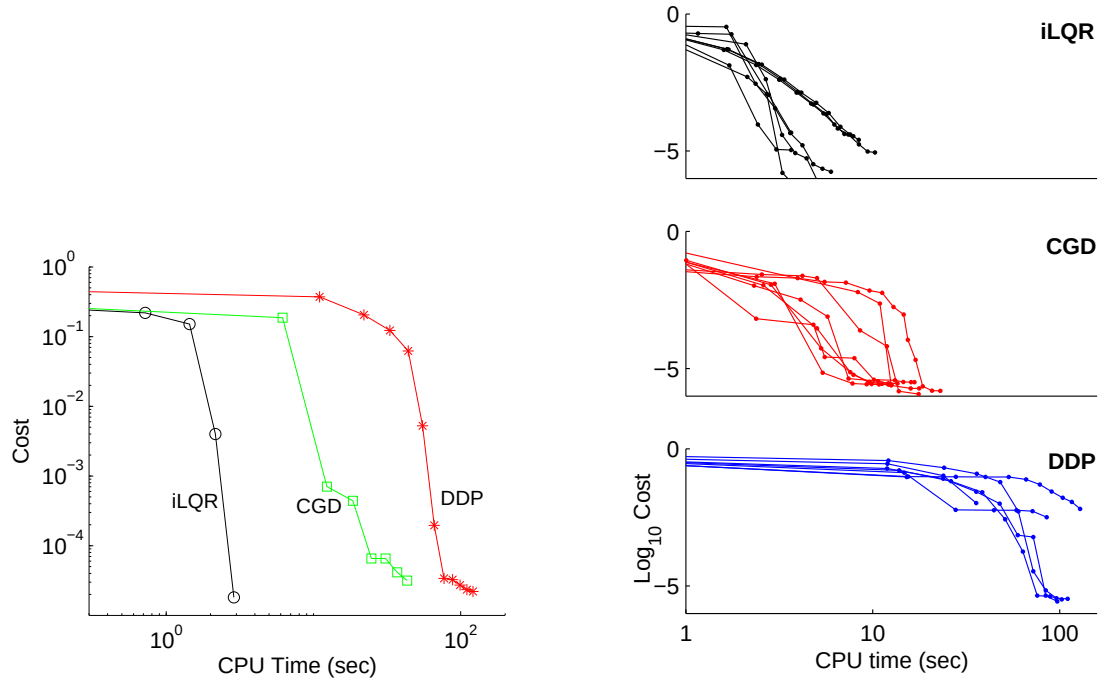
Figure 4.3 illustrates how the cost of the nominal trajectory decreases with the number of iterations for 8 different initial conditions. Compared with CGD and DDP

Table 4.1. Comparison of four methods

	Method	Time (sec)	Iteration
Torque control arm	ODE	11.20	N/A
	CGD	8.86	17
	DDP	2.65	15
	ILQR	**0.41**	**6**
Muscle control arm	ODE	>400	N/A
	CGD	91.14	14
	DDP	181.39	15
	ILQR	**8.31**	**8**
Inverted pendulum	ODE	6.33	N/A
	CGD	4.95	9
	DDP	1.61	20
	ILQR	**0.26**	**5**

method, the new ILQR method converged faster than the other methods, and found a better solution. Also we have found that the amount of computation per iteration for ILQR method is much less than the other methods. This is because gradient descent requires a line search (without which it works poorly); the implementation of DDP uses a second-order approximation to the system dynamics and Levenberg-Marquardt algorithm to compute the inverse matrix – both of which take a significant amount of time to compute. But we also need Levenberg-Marquardt method to achieve stable iterations.

Trajectory-based algorithms related to Pontryagin's Maximum Principle in general find locally-optimal solutions, and complex control problems may exhibit many local minima. It is useful to address the presence of local minima. Fig. 4.4 shows how the cloud of 100 randomly initialized trajectories gradually converge for the muscle-controlled arm model by using the ILQR method. The black curves describe the top 50% results where the shoulder angle and elbow angle arrive to their desired posture respectively, while the light curves show the bottom 50% results. There are local minima, but half the time the algorithm converges to the global minimum. Therefore, it can be used with a small number of restarts.



(a) (b)

Figure 4.3. (a) Cost vs. Iterations for 2-link 6-muscle Model. (b) Comparison of Cost vs. Iterations for 2-link 6-muscle Model based on 8 different initial conditions.

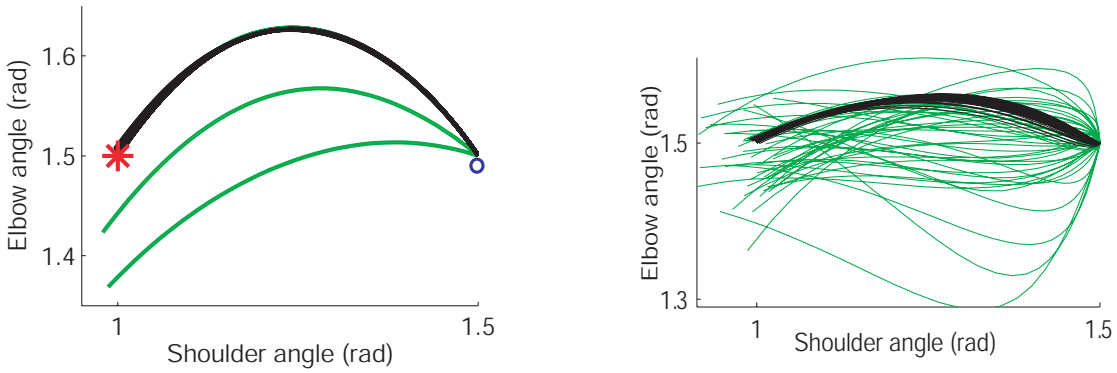


Figure 4.4. Trajectories of 2-link torque-controlled model (left) and 2-link 6-muscle model (right) for different initial conditions (Black color describes the top 50% results, light color describes the bottom 50% results)

4.5 Summary

Here we developed a new Iterative Linear Quadratic Regulator (ILQR) algorithm for optimal feedback control of nonlinear dynamical systems. We illustrated its application to a realistic 2-link, 6-muscle arm model, as well as simpler control problems. The simulation results suggest that the new method is more effective compared to the three most efficient methods that we are aware of.

While the control problems studied here are deterministic, the variability of biological movements indicates the presence of large disturbances in the motor system. It is very important to take these disturbances into account when studying biological control. In particular, it is known that the motor noise is control-dependent, with standard deviation proportional to the mean of the control signal. Such noise has been modelled in the LQG setting before [117]. Since the present ILQR algorithm is an extension to the LQG setting, it should be possible to treat nonlinear systems with control-dependent noise using similar methods. And we will do that in the next chapter.

Furthermore, the present formulation assumes that all of the state variables are available, which is not the case in the real world. Therefore, the method developed in the future should be extendable to situations that we can design the filter to provide state estimate for the optimal feedback control purpose. However, the difficulty is that the separation principle of estimation and control is violated in the presence of signal-dependent noise (which is one of characteristics in our biological movement models). The related work in [120] only showed the result for the linear dynamical system. In the next chapter, we will discuss iterative optimal control and estimation design for nonlinear stochastic systems, which will extend the ILQR setting as much as possible, and adapt it to the online control and estimation problems that nervous system faces.

This chapter, in part, was originally published in Proceedings of the 1^{st} International Conference on Informatics in Control, Automation & Robotics. The thesis author was the primary researcher and author in these works and the co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

Chapter 5

Iterative Stochastic Optimal Control and Estimation Design for Human Arm Movement

5.1 Motivation

Solving complex optimal control and estimation problems with stochastic uncertainties—that do not fit in the well-developed Linear-Quadratic-Gaussian (LQG) formalism—remains a challenge in many fields of science and engineering. Most existing numerical methods has followed two different approaches. Global methods based on the Hamilton-Jacobi-Bellman (HJB) equations and the powerful idea of dynamic programming can yield globally-optimal feedback control laws for general stochastic systems. While, such methods involve discretization of the state and control spaces — which makes them inapplicable to high-dimensional problems, due to the curse of dimensionality. Given the large number of variables needed to capture the state of the biomechanical system (e.g. joint angles, joint velocities, muscle activations), we believe that global methods are not well suited for Motor Control problems. Local methods based on Pontryagins Maximum principle avoid the curse of dimensionality, by solving a set of ordinary differential equations (ODEs) — via shooting, relaxation, collocation, or gradient descent. But the

resulting locally-optimal control laws are open-loop, and stochastic dynamics cannot be taken into account.

By taking the advantage of both local and global methods, Jacobson and Mayne [45] introduced Differential Dynamic Programming (DDP) — a successive approximation technique for solving nonlinear dynamic optimization problems. This method finds a local minimizing solution by applying the principle of optimality locally, in the immediate neighborhood of the current nominal trajectory. The improvement does not rely on the calculus of variations, but is based on dynamic programming. DDP is known to have second-order convergence and numerically appears to be more efficient [64] than efficient implementations of Newton's method [91]. We recently applied iterative Linear-Quadratic Regulator design (ILQR) to study the biological movement, it turns out to be significantly more efficient than DDP: by a factor of 10 on reasonably complex control problems [67]. This ILQR method uses iterative linearization of the nonlinear dynamics around the current trajectory, and improves that trajectory via modified Riccati equations. It yields feedback control law — which is a major advantage compared to open-loop methods. However, this method is still deterministic. Another shortcoming is that, unlike open-loop methods, it cannot deal with control constraints and non-quadratic cost functions. Our goal here is to remove these limitations.

While the new algorithm should be applicable to a range of problems, our specific motivation for developing it is the modelling of biological movement. Such modelling has proven extremely useful in the study of how the brain controls movement [115]. Yet, progress has been impeded by the lack of efficient methods that can handle realistic biomechanical control problems. The most remarkable characteristics of such problems are: high-dimensional nonlinear dynamics; control constraints (e.g. non-negative muscle activations); multiplicative noise, with standard deviation proportional to the magnitude of control signals or state variables; complex performance criteria, that are rarely quadratic in the state variables.

Before deriving our new iterative Linear-Quadratic-Gaussian (ILQG) method, we give a more detailed overview of what is new here:

1. Most dynamic programming methods use quadratic approximations to the optimal cost-to-go function. All such methods are "blind" to additive noise. However, in many problems of interest the noise is control-dependent, and such noise can easily be captured by quadratic approximations as we show below. Our new ILQG method incorporates control-dependent and state dependent noise — which turns out to have an effect similar to an energy cost. In practical situations, the state of the plant is only available through noisy measurement. When the state of the plant is fully observable, optimal LQG-like solutions can be computed efficiently as shown by several authors [6, 36, 78, 129, 96]. Such methodology has also been used to model reaching movements [41]. Most relevant to the study of sensorimotor control, however, is the partially-observable case, our goal here is to address that problem.

2. Quadratic approximation methods are presently restricted to unconstrained problems. Generally speaking, constraints make the optimal cost-to-go function non-quadratic. But since we are approximating that function anyway, we might as well take into account the effects of control constraints to the extent possible. Our new ILQG method does that — by modifying the feedback gain matrix whenever an element of the open-loop control sequence lies on the constraint boundary.

3. Quadratic approximation methods are based on Riccati equations: define a quadratic optimization problem that the optimal controls satisfy at time step t, solve it analytically, and obtain a formula for the optimal cost-to-go function at time step $t-1$. Optimizing a quadratic is only possible when the Hessian is positive-definite. This is of course true in the classic LQG setting, but when LQG methods are used to approximate general nonlinear dynamics with non-quadratic costs, the Hessian can (and in practice does) have zero and even negative eigenvalues. The traditional remedy is to "fix" the Hessian, using a Levenberg-Marquardt method, or an adaptive shift scheme, or simply replace it with the identity matrix (which yields the steepest descent method). The problem is that after fixing the Hessian, the optimization at time step t is no longer performed exactly — contrary to what the Riccati equations assume. Instead of making this invalid assumption, our new method takes the fixed Hessian into account, and constructs a cost-to-go approximation consistent with the resulting control law. This is

done by modified Riccati-like equations.

Of course, many techniques are already available for solving non-linear optimal control problems. You could find a detailed description of early techniques through [16]. Relevant iterative method has also been described by Luus [75], who used the dynamic programming in an iterative fashion. This chapter presents an iterative Linear-Quadratic-Gaussian method for locally optimal control and estimation of nonlinear stochastic dynamical systems subject to control constraints. The main contribution of the new method derived in the current research is that it constructs an affine feedback control law, obtained by minimizing a novel quadratic approximation to the optimal cost-to-go function. It also constructs a modified extended Kalman filter corresponding to the control law. The key important thing is that the two results together provide an iterative coordinate-descent algorithm, which is guaranteed to converge to a filter and a control law optimal with respect to each other.

The organization of this chapter is as follows. Section 5.2 formulates the original optimal problem we want to solve. In section 5.3 and 5.4 we present an LQG approximation to the original optimal control problem and compute an approximately-optimal control law under the assumption that state estimates are already obtained by unbiased linear filter. Section 5.5 designs the optimal filter corresponding to the given control law. The control law and filter are iteratively improved until convergence. Finally, section 5.6 illustrates the application of this extended LQG methodology in the context of reaching movements and obstacle avoidance, and explores numerically the convergence properties of the algorithm on a complex biomechanical control problem involving a stochastic model of the human arm.

5.2 Problem Formulation

Consider the nonlinear dynamical system described by the stochastic differential equations

$$d\mathbf{x^p} = f(\mathbf{x^p}, \mathbf{u^p})dt + \mathcal{F}(\mathbf{x^p}, \mathbf{u^p})d\omega, \tag{5.1}$$

along with the output equation

$$d\mathbf{y^p} = g(\mathbf{x^p}, \mathbf{u^p})dt + \mathcal{G}(\mathbf{x^p}, \mathbf{u^p})\, dv, \tag{5.2}$$

where state variable $\mathbf{x^p} \in \mathbb{R}^{n_x}$, control input $\mathbf{u^p} \in \mathbb{R}^{n_u}$, measurement output $\mathbf{y^p} \in \mathbb{R}^{n_y}$, and standard Brownian motion noise $\omega \in \mathbb{R}^{n_\omega}$, $v \in \mathbb{R}^{n_v}$ are independent of each other.

Let $\ell(t, \mathbf{x^p}, \mathbf{u}^p)$ be an instantaneous cost rate, $h(\mathbf{x^p}(T))$ be terminal cost incurred at the end of the process, T a specified final time. Define the cost-to-go function $v^\pi(t, \mathbf{x^p})$ as the total cost expected to accumulate if the system is initialized in state $\mathbf{x^p}$ at time t, and controlled until time T according to the control law π

$$v^\pi(t, \mathbf{x^p}) \triangleq E\left[h(\mathbf{x^p}(T)) + \int_t^T \ell(\tau, \mathbf{x^p}(\tau), \pi(\tau, \mathbf{x^p}(\tau)))\ d\tau\right] \tag{5.3}$$

The expectation is taken over the instantiations of the stochastic process ω. The admissible control signals may be constrained: $\mathbf{u^p} \in \mathcal{U}$.

The objective of optimal control is to find the control law $\pi^*(t, \mathbf{x^p})$ that minimizes $v^\pi(0, \mathbf{x^p}(0))$. Note that the globally-optimal control law does not depend on a specific initial state. However, finding this control law in complex problems is unlikely. Instead, we seek locally-optimal control laws: we will present an LQG approximation to our original optimal control problem and compute an approximately-optimal control law. The present formulation in this chapter assumes that the state of system is measurable through delayed and noisy sensors, therefore we will also design an optimal filter in order to extract the accurate state information from noisy measurement data.

5.3 Local LQG Approximation

5.3.1 Linearization

In this section the locally-optimal control law is computed using the method of dynamic programming. Time is discretized as $k = 1, \cdots, N$, with time step $\Delta t = T/(N-1)$. Our derived algorithm is iterative. Each iteration starts with a nominal control sequence $\bar{\mathbf{u}}^p$, and a corresponding nominal trajectory $\bar{\mathbf{x}}^p$ obtained by applying $\bar{\mathbf{u}}^p$ to the deterministic system $\dot{\mathbf{x}}^p = f(\mathbf{x}^p, \mathbf{u}^p)$ with $\bar{\mathbf{x}}^p(0) = \mathbf{x}_0^p$. This can be done by Euler integration $\bar{\mathbf{x}}_{k+1}^p = \bar{\mathbf{x}}_k^p + \Delta t\ f(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$.

By linearizing the system dynamics and quadratizing the cost functions around $(\bar{\mathbf{x}}^p, \bar{\mathbf{u}}^p)$, we obtain a discrete-time linear dynamical system with quadratic cost. Importantly the linearized dynamics no longer describe the state and control variables, but instead they describe the state and control deviations $\mathbf{x}_k = \mathbf{x}_k^p - \bar{\mathbf{x}}_k^p$, $\mathbf{u}_k = \mathbf{u}_k^p - \bar{\mathbf{u}}_k^p$, and $\mathbf{y}_k = \mathbf{y}_k^p - \bar{\mathbf{y}}_k^p$. Written in terms of these deviations — state variable $\mathbf{x}_k \in \mathbb{R}^{n_x}$, control input $\mathbf{u}_k \in \mathbb{R}^{n_u}$, measurement output $\mathbf{y}_k \in \mathbb{R}^{n_y}$, the modified LQG approximation to our original optimal control problem becomes

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathcal{C}_k(\mathbf{x}_k, \mathbf{u}_k)\xi_k, \tag{5.4}$$

$$\mathbf{y}_k = F_k \mathbf{x}_k + E_k \mathbf{u}_k + \mathcal{D}_k(\mathbf{x}_k, \mathbf{u}_k)\eta_k, \tag{5.5}$$

$$\text{cost}_k = q_k + \mathbf{x}_k^T \mathbf{q}_k + \frac{1}{2}\mathbf{x}_k^T Q_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{r}_k + \frac{1}{2}\mathbf{u}_k^T R_k \mathbf{u}_k + \mathbf{u}_k^T P_k \mathbf{x}_k, \tag{5.6}$$

where

$$A_k = I + \Delta t \frac{\partial f}{\partial \mathbf{x}^p}, \qquad B_k = \Delta t \frac{\partial f}{\partial \mathbf{u}^p}, \qquad F_k = \frac{\partial g}{\partial \mathbf{x}^p}, \qquad E_k = \frac{\partial g}{\partial \mathbf{u}^p}, \tag{5.7}$$

$$\mathcal{C}_k(\mathbf{x}_k, \mathbf{u}_k) \triangleq \left[\mathbf{c}_{1,k} + C_{1,k}^{\mathbf{x}} \mathbf{x}_k + C_{1,k}^{\mathbf{u}} \mathbf{u}_k,\ \cdots\ , \mathbf{c}_{n_\omega,k} + C_{n_\omega,k}^{\mathbf{x}} \mathbf{x}_k + C_{n_\omega,k}^{\mathbf{u}} \mathbf{u}_k\right], \tag{5.8}$$

$$\mathcal{D}_k(\mathbf{x}_k, \mathbf{u}_k) \triangleq \left[\mathbf{d}_{1,k} + D_{1,k}^{\mathbf{x}} \mathbf{x}_k + D_{1,k}^{\mathbf{u}} \mathbf{u}_k,\ \cdots\ , \mathbf{d}_{n_v,k} + D_{n_v,k}^{\mathbf{x}} \mathbf{x}_k + D_{n_v,k}^{\mathbf{u}} \mathbf{u}_k\right], \tag{5.9}$$

$$\mathbf{c}_{i,k} = \sqrt{\Delta t}\ \mathcal{F}^{[i]}, \qquad C_{i,k}^{\mathbf{x}} = \sqrt{\Delta t}\ \frac{\partial \mathcal{F}^{[i]}}{\partial \mathbf{x}^p}, \qquad C_{i,k}^{\mathbf{u}} = \sqrt{\Delta t}\ \frac{\partial \mathcal{F}^{[i]}}{\partial \mathbf{u}^p}, \tag{5.10}$$

$$\mathbf{d}_{i,k} = \frac{1}{\sqrt{\Delta t}}\ \mathcal{G}^{[i]}, \qquad D_{i,k}^{\mathbf{x}} = \frac{1}{\sqrt{\Delta t}}\ \frac{\partial \mathcal{G}^{[i]}}{\partial \mathbf{x}^p}, \qquad D_{i,k}^{\mathbf{u}} = \frac{1}{\sqrt{\Delta t}}\ \frac{\partial \mathcal{G}^{[i]}}{\partial \mathbf{u}^p}, \tag{5.11}$$

and

$$q_k = \Delta t\ \ell, \qquad \mathbf{q}_k = \Delta t\ \frac{\partial \ell}{\partial \mathbf{x}^p}, \qquad Q_k = \Delta t\ \frac{\partial^2 \ell}{\partial (\mathbf{x}^p)^2}, \tag{5.12}$$

$$\mathbf{r}_k = \Delta t\ \frac{\partial \ell}{\partial \mathbf{u}^p}, \qquad R_k = \Delta t\ \frac{\partial^2 \ell}{\partial (\mathbf{u}^p)^2}, \qquad P_k = \Delta t\ \frac{\partial^2 \ell}{\partial \mathbf{u}^p \partial \mathbf{x}^p}, \tag{5.13}$$

are computed at each $(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$.

The initial state has known mean $\hat{\mathbf{x}}_1$ and covariance Σ_1. All the matrices $A_k, B_k, F_k, E_k, \mathbf{c}_{i,k}, C_{i,k}^{\mathbf{x}}, C_{i,k}^{\mathbf{u}}, \mathbf{d}_{j,k}, D_{j,k}^{\mathbf{x}}, D_{j,k}^{\mathbf{u}}$ $(i = 1, \cdots, n_\omega, \texttt{and}\ j = 1, \cdots, n_v)$ are assumed to be given with the proper dimensions. The independent random variables $\xi_k \in \mathbb{R}^{n_\omega}$ and $\eta_k \in \mathbb{R}^{n_v}$ are zero-mean Gaussain white noises with covariances $\Omega^\xi = I$ and $\Omega^\eta = I$ respectively. Note that $\mathcal{F}^{[i]}$ and $\mathcal{G}^{[i]}$ denote the i^{th} column of matrix $\mathcal{F} \in \mathbb{R}^{n_x \times n_\omega}$ and $\mathcal{G} \in \mathbb{R}^{n_y \times n_v}$ respectively. At the final time step $k = N$, the cost is defined as $q_N + \mathbf{x}_N^T \mathbf{q}_N + \frac{1}{2}\mathbf{x}_N^T Q_N \mathbf{x}_N$, where $q_N = h, \mathbf{q}_N = \frac{\partial h}{\partial \mathbf{x}^p}$, and $Q_N = \frac{\partial^2 h}{\partial (\mathbf{x}^p)^2}$.

Here we are using a noise model which includes control-dependent, state-dependent and additive noises. This is sufficient to capture noise in the system — which is what we are mainly interested in. Considering the sensorimotor control, noise in the motor output increases with the magnitude of the control signal. Incorporating the state-dependent noise in analysis of sensorimotor control could allow more accurate modelling of feedback form sensory modalities and various experimental perturbations. In the study of estimation and control design for the system with control-dependent and state-dependent noises, the well-known Separation Principle of standard LQG design is violated. This complicates the problem substantially, and forces us to develop a new structure of recursive controller and estimator.

5.3.2 Computing the cost-to-go function (Partially observable case)

In practical situations, the state of the controlled plant are only available through noisy measurement. While the implementation of the optimal control law depends on the state of the system, we have to design an estimator in order to extract the correct information of the state. Here we are assuming that the approximately-optimal control law is allowed to be an affine function of $\hat{\mathbf{x}}_k$ — the unbiased estimate of state $\mathbf{x}_k$, and

the estimator has the form

$$\hat{\mathbf{x}}_{k+1} = A_k\hat{\mathbf{x}}_k + B_k\mathbf{u}_k + K_k(\mathbf{y}_k - F_k\hat{\mathbf{x}}_k - E_k\mathbf{u}_k), \tag{5.14}$$

where the filter gains K_k are non-adaptive, i.e., they are determined in advance and cannot change as a function of the specific controls and observations within a simulation run. Detailed derivation for computing the filter gain K_k will be presented in section 5.5.

The approximately-optimal control law for the LQG approximation will be shown to be affine, in the form

$$\mathbf{u}_k = \boldsymbol{\pi}_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k, \qquad k = 1, \cdots, N, \tag{5.15}$$

where l_k describes the open-loop control component (it arises because we are dealing with state and control deviations, and is needed to make the algorithm iterative), L_k is the feedback control gain. The control law we design is approximately-optimal because we may have control constraints and non-convex costs, and also because we use linear Gaussian approximations. Let the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ be the total cost expected to accumulate if the system (5.4) is initialized in state $\mathbf{x}_k$ at time step k, and controlled according to $\boldsymbol{\pi}_k$ for the remaining time steps.

Lemma 1 *Suppose the control law $\boldsymbol{\pi}_k$ for system (5.4)-(5.5) has already been designed for time steps $k+1, \cdots, N$. If the control law is affine in the form (5.15), then the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ has the form*

$$v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) = \frac{1}{2}\mathbf{x}_k^T S_k^{\mathbf{x}}\mathbf{x}_k + \frac{1}{2}\hat{\mathbf{x}}_k^T S_k^{\hat{\mathbf{x}}}\hat{\mathbf{x}}_k + \mathbf{x}_k^T S_k^{\mathbf{x}\hat{\mathbf{x}}}\hat{\mathbf{x}}_k + \mathbf{x}_k^T \mathbf{s}_k^{\mathbf{x}} + \hat{\mathbf{x}}_k^T \mathbf{s}_k^{\hat{\mathbf{x}}} + \mathsf{s}_k \tag{5.16}$$

where the parameters $S_k^{\mathbf{x}}, S_k^{\hat{\mathbf{x}}}, S_k^{\mathbf{x}\hat{\mathbf{x}}}, \mathbf{s}_k^{\mathbf{x}}, \mathbf{s}_k^{\hat{\mathbf{x}}}$, and s_k can be computed recursively backwards in time as

$$\begin{aligned} S_k^{\mathbf{x}} &= Q_k + A_k^T S_{k+1}^{\mathbf{x}} A_k + F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k F_k + 2A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} K_k F_k \\ &\quad + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}, \qquad S_N^{\mathbf{x}} = Q_N, \end{aligned} \tag{5.17}$$

$$S_k^{\hat{\mathbf{x}}} = (A_k - K_k F_k)^T S_{k+1}^{\hat{\mathbf{x}}}(A_k - K_k F_k) + L_k^T H L_k + L_k^T G^{\hat{\mathbf{x}}} + (G^{\hat{\mathbf{x}}})^T L_k, \; S_N^{\hat{\mathbf{x}}} = 0, \tag{5.18}$$

$$S_k^{\mathbf{x}\hat{\mathbf{x}}} = F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}(A_k - K_k F_k) + A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}(A_k - K_k F_k) + (G^{\mathbf{x}})^T L_k, \quad S_N^{\mathbf{x}\hat{\mathbf{x}}} = 0, \tag{5.19}$$

$$\mathbf{s}_k^{\mathbf{x}} = \mathbf{q}_k + A_k^T \mathbf{s}_{k+1}^{\mathbf{x}} + F_k^T K_k^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + (G^{\mathbf{x}})^T l_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k}$$
$$+ \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}, \qquad \mathbf{s}_N^{\mathbf{x}} = \mathbf{q}_N, \quad (5.20)$$

$$\mathbf{s}_k^{\hat{\mathbf{x}}} = (A_k - K_k F_k)^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + L_k^T H l_k + L_k^T \mathbf{g} + (G^{\hat{\mathbf{x}}})^T l_k, \qquad \mathbf{s}_N^{\hat{\mathbf{x}}} = 0, \quad (5.21)$$

$$s_k = q_k + s_{k+1} + l_k^T \mathbf{g} + \frac{1}{2} l_k^T H l_k + \frac{1}{2} \left(\sum_{i=1}^{n_\omega} \mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \mathbf{d}_{i,k}^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} \right),$$
$$s_N = q_N. \quad (5.22)$$

and

$$H \triangleq R_k + B_k^T (S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) B_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}}$$
$$+ \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}}, \quad (5.23)$$

$$\mathbf{g} \triangleq \mathbf{r}_k + B_k^T (\mathbf{s}_{k+1}^{\mathbf{x}} + \mathbf{s}_{k+1}^{\hat{\mathbf{x}}}) + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}, \quad (5.24)$$

$$G^{\mathbf{x}} \triangleq P_k + B_k^T (S_{k+1}^{\mathbf{x}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) A_k + B_k^T (S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) K_k F_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}}$$
$$+ \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}, \quad (5.25)$$

$$G^{\hat{\mathbf{x}}} \triangleq B_k^T (S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) (A_k - K_k F_k). \quad (5.26)$$

Proof. Consider the control law which has been designed for time steps $k+1, \cdots, N$, and at time step k is given by $\mathbf{u}_k = \boldsymbol{\pi}_k(\hat{\mathbf{x}}_k) = l_k + L_k \hat{\mathbf{x}}_k$ (note that in the later derivation we will use the shortcut $\boldsymbol{\pi}_k$ in place of the control signal $\boldsymbol{\pi}_k(\hat{\mathbf{x}}_k)$ that our control law generates). Let $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ be the corresponding cost-to-go function, then the Bellman equation is

$$v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) = \text{immediate cost} + E\left[v_{k+1}(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1}) | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right] \quad (5.27)$$

Based on the dynamics function (5.4) and (5.14), the conditional mean and covariance

of $\mathbf{x}_{k+1}$ and $\hat{\mathbf{x}}_{k+1}$ are

$$E\left[\mathbf{x}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] = A_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k, \tag{5.28}$$

$$E\left[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] = (A_k - K_kF_k)\hat{\mathbf{x}}_k + B_k\boldsymbol{\pi}_k + K_kF_k\mathbf{x}_k, \tag{5.29}$$

$$Cov\left[\mathbf{x}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] = \sum_{i=1}^{n_\omega}\left(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)\left(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)^T, \tag{5.30}$$

$$Cov\left[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] = K_k\sum_{i=1}^{n_v}\left(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)\left(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)^T K_k^T. \tag{5.31}$$

Since

$$\begin{aligned} E\left[\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] &= Cov\left[\mathbf{x}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] \\ &\quad + E\left[\mathbf{x}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right]\left(E\left[\mathbf{x}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right]\right)^T, \end{aligned} \tag{5.32}$$

$$\begin{aligned} E\left[\hat{\mathbf{x}}_{k+1}\hat{\mathbf{x}}_{k+1}^T|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] &= Cov\left[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] \\ &\quad + E\left[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right]\left(E\left[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right]\right)^T, \end{aligned} \tag{5.33}$$

$$E\left[\hat{\mathbf{x}}_{k+1}\mathbf{x}_{k+1}^T|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] = \left((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k\right)(A_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k)^T, \tag{5.34}$$

applying the formulation of cost-to-go function defined in (5.16), substituting (5.28)-(5.34) into the conditional expectation in Bellman equation, it yields

$$\begin{aligned} &E\left[v_{k+1}(\mathbf{x}_{k+1},\hat{\mathbf{x}}_{k+1})|\mathbf{x}_k,\hat{\mathbf{x}}_k,\boldsymbol{\pi}_k\right] \\ &= \frac{1}{2}trS_{k+1}^{\mathbf{x}}\left[\sum_{i=1}^{n_\omega}\left(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)\left(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)^T\right] \\ &+ \frac{1}{2}trS_{k+1}^{\mathbf{x}}\left[\sum_{i=1}^{n_\omega}(A_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k)(A_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k)^T\right] \\ &+ \frac{1}{2}trS_{k+1}^{\hat{\mathbf{x}}}\left[K_k\sum_{i=1}^{n_v}\left(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)\left(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\boldsymbol{\pi}_k\right)^T K_k^T\right] \\ &+ \frac{1}{2}trS_{k+1}^{\hat{\mathbf{x}}}\left[\left((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k\right)\left((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k\right)^T\right] \\ &+ trS_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\left[\left((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k\right)(A_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k)^T\right] \\ &+ (A_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k)^T\mathbf{s}_{k+1}^{\mathbf{x}} + \left((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\boldsymbol{\pi}_k\right)^T\mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + \mathsf{s}_{k+1} \end{aligned}$$

Using the fact that $tr(UV) = tr(VU)$ in the above equation, and substituting the immediate cost (5.6) and the above equation into (5.27), the resulting cost-to-go function becomes

$$
\begin{aligned}
& v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) \\
&= \frac{1}{2}\mathbf{x}_k^T \Bigg[Q_k + A_k^T S_{k+1}^{\mathbf{x}} A_k + F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k F_k + 2A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} K_k F_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} \\
&\qquad + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}} \Bigg] \mathbf{x}_k \\
&\quad + \frac{1}{2}\hat{\mathbf{x}}_k^T (A_k - K_k F_k)^T S_{k+1}^{\hat{\mathbf{x}}} (A_k - K_k F_k) \hat{\mathbf{x}}_k \\
&\quad + \frac{1}{2}\boldsymbol{\pi}_k^T \Bigg[R_k + B_k^T (S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) B_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} \\
&\qquad + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}} \Bigg] \boldsymbol{\pi}_k \\
&\quad + \mathbf{x}_k^T \left[F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} (A_k - K_k F_k) + A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} (A_k - K_k F_k) \right] \hat{\mathbf{x}}_k \\
&\quad + \boldsymbol{\pi}_k^T \Bigg[P_k + B_k^T (S_{k+1}^{\mathbf{x}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) A_k + B_k^T (S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) K_k F_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} \\
&\qquad + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}} \Bigg] \mathbf{x}_k \\
&\quad + \boldsymbol{\pi}_k^T B_k^T (S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}) (A_k - K_k F_k) \hat{\mathbf{x}}_k \\
&\quad + \mathbf{x}_k^T \left[\mathbf{q}_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} + A_k^T \mathbf{s}_{k+1}^{\mathbf{x}} + F_k^T K_k^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} \right] \\
&\quad + \hat{\mathbf{x}}_k^T (A_k - K_k F_k)^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} \\
&\quad + \boldsymbol{\pi}_k^T \left[\mathbf{r}_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} + B_k^T (\mathbf{s}_{k+1}^{\mathbf{x}} + \mathbf{s}_{k+1}^{\hat{\mathbf{x}}}) \right] \\
&\quad + q_k + \mathbf{s}_{k+1} + \frac{1}{2} \left(\sum_{i=1}^{n_\omega} \mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \mathbf{d}_{i,k}^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} \right)
\end{aligned} \tag{5.35}
$$

Substituting (5.23)-(5.26) into the above equation, the $\boldsymbol{\pi}_k$-dependent terms in (5.35) becomes

$$\frac{1}{2}\boldsymbol{\pi}_k^T H \boldsymbol{\pi}_k + \boldsymbol{\pi}_k^T (\mathbf{g} + G^{\mathbf{x}} \mathbf{x}_k + G^{\hat{\mathbf{x}}} \hat{\mathbf{x}}_k) \tag{5.36}$$

Since we assume that the control law has the general form given in (5.15), replacing $\boldsymbol{\pi}_k$

with $l_k + L_k\hat{\mathbf{x}}_k$, it yields

$$\begin{aligned}
&\frac{1}{2}\hat{\mathbf{x}}_k^T \left(L_k^T H L_k + L_k^T G^{\hat{\mathbf{x}}} + (G^{\hat{\mathbf{x}}})^T L_k\right) \hat{\mathbf{x}}_k + \mathbf{x}_k^T (G^{\mathbf{x}})^T L_k \hat{\mathbf{x}}_k \\
&+ \mathbf{x}_k^T (G^{\mathbf{x}})^T l_k + \hat{\mathbf{x}}_k^T \Big(L_k^T H l_k + L_k^T \mathbf{g} + (G^{\hat{\mathbf{x}}})^T l_k\Big) + l_k^T \mathbf{g} + \frac{1}{2} l_k^T H l_k
\end{aligned} \tag{5.37}$$

Now the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ becomes

$$\begin{aligned}
&v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) \\
&= \frac{1}{2}\mathbf{x}_k^T \Bigg[Q_k + A_k^T S_{k+1}^{\mathbf{x}} A_k + F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k F_k + 2A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} K_k F_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} \\
&\qquad + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}\Bigg] \mathbf{x}_k \\
&+ \frac{1}{2}\hat{\mathbf{x}}_k^T \left[(A_k - K_k F_k)^T S_{k+1}^{\hat{\mathbf{x}}} (A_k - K_k F_k) + L_k^T H L_k + L_k^T G^{\hat{\mathbf{x}}} + (G^{\hat{\mathbf{x}}})^T L_k\right] \hat{\mathbf{x}}_k \\
&+ \mathbf{x}_k^T \left[F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} (A_k - K_k F_k) + A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} (A_k - K_k F_k) + (G^{\mathbf{x}})^T L_k\right] \hat{\mathbf{x}}_k \\
&+ \mathbf{x}_k^T \Bigg[\mathbf{q}_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} (D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} + A_k^T \mathbf{s}_{k+1}^{\mathbf{x}} \\
&\qquad + F_k^T K_k^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + (G^{\mathbf{x}})^T l_k\Bigg] \\
&+ \hat{\mathbf{x}}_k^T \left[(A_k - K_k F_k)^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + L_k^T H l_k + L_k^T \mathbf{g} + (G^{\hat{\mathbf{x}}})^T l_k\right] \\
&+ q_k + \mathsf{s}_{k+1} + \frac{1}{2}\left(\sum_{i=1}^{n_\omega} \mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \mathbf{d}_{i,k}^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}\right) + l_k^T \mathbf{g} + \frac{1}{2} l_k^T H l_k
\end{aligned} \tag{5.38}$$

By applying the defined formulation of cost-to-go function given in (5.16), we could obtain (5.17)-(5.22) immediately which completes the proof.

5.3.3 Computing the cost-to-go function (Fully observable case)

Suppose the state of system (5.4) are available in the implementation of the optimal control design, Lemma 1 readily leads to the following corollary.

Corollary 1 *Suppose the control law $\boldsymbol{\pi}_k$ for system (5.4) has already been designed for time steps $k+1, \cdots, N$. If the control law is affine in the form $\mathbf{u}_k = l_k + L_k \mathbf{x}_k$, where*

$k = 1, \cdots, N$, then the cost-to-go function $v_k(\mathbf{x}_k)$ has the form

$$v_k(\mathbf{x}_k) = \frac{1}{2}\mathbf{x}_k^T S_k^{\mathbf{x}} \mathbf{x}_k + \mathbf{x}_k^T \mathbf{s}_k^{\mathbf{x}} + s_k \tag{5.39}$$

where the parameters $S_k^{\mathbf{x}}, \mathbf{s}_k^{\mathbf{x}}$, and s_k can be computed recursively backwards in time as

$$S_k^{\mathbf{x}} = Q_k + A_k^T S_{k+1}^{\mathbf{x}} A_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + L_k^T H L_k + L_k^T G + G^T L_k, \quad S_N^{\mathbf{x}} = Q_N, \tag{5.40}$$

$$\mathbf{s}_k^{\mathbf{x}} = \mathbf{q}_k + A_k^T \mathbf{s}_{k+1}^{\mathbf{x}} + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + L_k^T H l_k + L_k^T \mathbf{g} + G^T l_k, \qquad \mathbf{s}_N^{\mathbf{x}} = \mathbf{q}_N, \tag{5.41}$$

$$s_k = q_k + s_{k+1} + \frac{1}{2}\sum_{i=1}^{n_\omega} \mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \frac{1}{2} l_k^T H l_k + l_k^T \mathbf{g}, \qquad s_N = q_N, \tag{5.42}$$

and

$$H \triangleq R_k + B_k^T S_{k+1}^{\mathbf{x}} B_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}}, \tag{5.43}$$

$$\mathbf{g} \triangleq \mathbf{r}_k + B_k^T \mathbf{s}_{k+1}^{\mathbf{x}} + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k}, \tag{5.44}$$

$$G \triangleq P_k + B_k^T S_{k+1}^{\mathbf{x}} A_k + \sum_{i=1}^{n_\omega} (C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}}. \tag{5.45}$$

5.4 Optimal Controller Design

As we saw in (5.35), the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ depends on the control $u_k = \boldsymbol{\pi}_k(\hat{\mathbf{x}}_k)$ through the term

$$a(\mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k) = \frac{1}{2}\boldsymbol{\pi}_k^T H \boldsymbol{\pi}_k + \boldsymbol{\pi}_k^T (\mathbf{g} + G^{\mathbf{x}} \mathbf{x}_k + G^{\hat{\mathbf{x}}} \hat{\mathbf{x}}_k)$$

This expression is quadratic in $\boldsymbol{\pi}_k$ and can be minimized analytically, but the problem is that the minimum depends on $\mathbf{x}_k$ while $\boldsymbol{\pi}_k$ is only a function of $\hat{\mathbf{x}}_k$. To obtain the optimal control law at time step k, we have to take an expectation over $\mathbf{x}_k$ conditional on $\hat{\mathbf{x}}_k$, and find the function $\boldsymbol{\pi}_k$ that minimizes the resulting expression. Since $E[\mathbf{x}_k | \hat{\mathbf{x}}_k] = \hat{\mathbf{x}}_k$, we have

$$\alpha(\hat{\mathbf{x}}_k, \boldsymbol{\pi}_k) \triangleq E[a(\mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k) \mid \hat{\mathbf{x}}_k] = \frac{1}{2}\boldsymbol{\pi}_k^T H \boldsymbol{\pi}_k + \boldsymbol{\pi}_k^T (\mathbf{g} + G\hat{\mathbf{x}}_k) \tag{5.46}$$

where $G = G^{\mathbf{x}} + G^{\hat{\mathbf{x}}}$. Ideally we would choose $\boldsymbol{\pi}_k$ that minimizes $\alpha(\hat{\mathbf{x}}_k, \boldsymbol{\pi}_k)$ subject to whatever control constraints are present. However, this is not always possible within the family of affine control laws $\boldsymbol{\pi}_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$ that we are considering. Since the goal of the LQG stage is to approximate the optimal controller for the nonlinear system in the vicinity of $\bar{\mathbf{x}}_k^p$, we will give preference to those control laws that are optimal/feasible for small $\mathbf{x}_k$, even if that (unavoidably) makes them sub-optimal/infeasible for larger $\mathbf{x}_k$.

5.4.1 Second-order methods

If the symmetric matrix H in (5.46) is positive semi-definite, we can compute the unconstrained optimal control law

$$\boldsymbol{\pi}_k = -H^{-1}(\mathbf{g} + G\hat{\mathbf{x}}_k), \tag{5.47}$$

and deal with the control constraints as described below. But when H has negative eigenvalues, there exist $\boldsymbol{\pi}_k's$ that make a arbitrarily negative. Note that the cost-to-go function for the nonlinear problem is always non-negative, but we are using an approximation to the true cost, we may encounter situations where a does not have a minimum. In that case we use $\mathcal{H}$ to resemble H, because H still contains correct second-order information; and so the true cost-to-go decreases in the direction $-\mathcal{H}^{-1}(\mathbf{g} + G\hat{\mathbf{x}}_k)$ for any positive definite matrix $\mathcal{H}$.

One possibility is to set $\mathcal{H} = H + \big(\epsilon - \lambda_{min}(H)\big)I$ where $\lambda_{min}(H)$ is the minimum eigenvalue of H and $\epsilon > 0$. This is related to the Levenberg-Marquardt method, and has the potentially undesirable effect of increasing all eigenvalues of H and not just those that are negative. Another possibility is to compute the eigenvalue decomposition $[V, D] = eig(H)$, replace all elements of the diagonal matrix D that are smaller than ϵ with ϵ (obtaining a new diagonal matrix $\mathcal{D}$), and then set $\mathcal{H} = V\mathcal{D}V^T$. The eigenvalue decomposition is not a significant slowdown, because we have to perform a matrix inversion anyway and we can do so by $\mathcal{H}^{-1} = V\mathcal{D}^{-1}V^T$. It is not yet clear which of the two methods works better in practice. Note that we may also want to use $\mathcal{H}$ instead of H when the eigenvalues are positive but very small — because in that case H^{-1} can cause very large control signal that will push the original system outside the range of validity

of our LQG approximation.

Lemma 2 *The optimal control law is computed as*

$$\mathbf{u}_k = l_k + L_k \hat{\mathbf{x}}_k, \tag{5.48}$$

where

$$l_k = -\mathcal{H}^{-1}\mathbf{g}, \qquad L_k = -\mathcal{H}^{-1}G, \tag{5.49}$$

$$\mathcal{H} = \begin{cases} H + \big(\epsilon - \lambda_{min}(H)\big)I, & \epsilon > 0 \quad (\textit{Method 1}) \\ V\mathcal{D}V^T, & [V,D] = eig(H) \quad (\textit{Method 2}) \end{cases} \tag{5.50}$$

$$\begin{aligned} H \triangleq\ & R_k + B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})B_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} \\ & + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}}, \end{aligned} \tag{5.51}$$

$$\mathbf{g} \triangleq \mathbf{r}_k + B_k^T(\mathbf{s}_{k+1}^{\mathbf{x}} + \mathbf{s}_{k+1}^{\hat{\mathbf{x}}}) + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k} + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}, \tag{5.52}$$

$$\begin{aligned} G \triangleq\ & P_k + B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})A_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} \\ & + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}, \end{aligned} \tag{5.53}$$

where $S_{k+1}^{\mathbf{x}}, S_{k+1}^{\hat{\mathbf{x}}}, S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}, \mathbf{s}_{k+1}^{\mathbf{x}}, \mathbf{s}_{k+1}^{\hat{\mathbf{x}}}, s_{k+1}$ *can be obtained through (5.17)-(5.22) backwards in time.*

5.4.2 Constrained second-order methods

The problem here is to find the control law $\mathbf{u}_k = \boldsymbol{\pi}_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$ minimizing (5.46) subject to constraints $\mathbf{u}_k + \bar{\mathbf{u}}_k^p \in \mathcal{U}$, assuming that H has already been replaced with a positive definite $\mathcal{H}$ (see the above section). Given that $\mathbf{x}_k$ is unconstrained, the only general way to enforce the constraints $\mathcal{U}$ is to set $L_k = 0$. In practice we do not want to be that conservative, since we are looking for an approximation to the nonlinear problem that is valid around $\mathbf{x}_k = 0$. Either way we can ignore the $L_k\mathbf{x}_k$ term in the constraint satisfaction phase, and come back to the computation of L_k after the open-loop component l_k has been determined.

The unconstrained minimum of $\mathbf{u}_k^T\mathbf{g} + \frac{1}{2}\mathbf{u}_k^T\mathcal{H}\mathbf{u}_k$ is $\mathbf{u}_k^* = -\mathcal{H}^{-1}\mathbf{g}$. If it satisfies the constraint $\mathbf{u}_k^* + \bar{\mathbf{u}}_k^p \in \mathcal{U}$ we are done. Otherwise we have two options. The more efficient but less accurate method is to backtrack once, i.e. to find the maximal $\epsilon \in [0; 1]$ such that $\epsilon\mathbf{u}_k^* + \bar{\mathbf{u}}_k^p \in \mathcal{U}$. This is appropriate in the early phase of the iterative algorithm when the nominal trajectory $\bar{\mathbf{x}}_k^p$ is still far away from $\bar{\mathbf{x}}_k^{p^*}$; in that phase it makes more sense to quickly improve the control law rather than refine the solution to an LQG problem that is an inaccurate approximation to the original problem. But in the final phase of the iterative algorithm we want to obtain the best control law possible for the given LQG problem. In that phase we use quadratic programming. When the constraint set is specified by a collection of linear inequalities, and given that $\mathcal{H}$ is positive definite, the active set algorithm (which is a greedy quadratic programming method) can be used to quickly find the global constrained minimum.

Once the open-loop component l_k is determined, we have to compute the feedback gain matrix L_k. If $l_k + \bar{\mathbf{u}}_k^p$ is inside $\mathcal{U}$, small changes $L_k\mathbf{x}_k$ will not cause constraint violations and so we can use the optimal $L_k = -\mathcal{H}^{-1}G$. But if $l_k + \bar{\mathbf{u}}_k^p$ lies on the constraint boundary $\partial\mathcal{U}$, we have to modify L_k so that $L_k\mathbf{x}_k$ can only cause changes along the boundary. This is not only because we want to avoid constraint violations. The fact that $l_k + \bar{\mathbf{u}}_k^p$ is on $\partial\mathcal{U}$ means that the unconstrained minimum $\mathbf{u}_k^*$ is actually outside $\mathcal{U}$, and so a change $L_k\mathbf{x}_k$ orthogonal to the boundary $\partial\mathcal{U}$ cannot produce a better feasible control.

Modifying L_k is straightforward in the typical case when the range of each element of $\mathbf{u}_k$ is specified independently. In that case we simply set to zero the rows of $-\mathcal{H}^{-1}G$ corresponding to elements of $l_k + \bar{\mathbf{u}}_k^p$ that have reached their limits.

5.5 Optimal Estimator Design

It is well known that, for models with control-dependent and state-dependent noises, the optimal filter is very difficult to compute in practice. For this kind of models, the construction of suboptimal filters that approximate the optimal one becomes very important.

So far we computed the optimal control law for any fixed sequence of filter gains K_k. In order to preserve the optimality of the control law obtained in the previous section and attain an iterative algorithm with guaranteed convergence, we need to compute a fixed sequence of filter gains that are optimal for a given control law. Thus our objective here is the following: given the control law $\mathbf{u}_1, \cdots, \mathbf{u}_{N-1}$ (which is optimal for the previous filter $K_1, \ldots, K_{N-1}$), compute a new suboptimal filter evaluated by minimizing the magnitude of its estimation errors, in conjunction with the given control law. Once the iterative algorithm has converged and the control law has been designed, we could use an adaptive filter in place of the fixed-gain filter in run time.

Lemma 3 *With the definition of the unconditional means* $m_k^e \triangleq E[e_k]$, $m_k^{\hat{\mathbf{x}}} \triangleq E[\hat{\mathbf{x}}_k]$, *and the unconditional covariances* $\Sigma_k^e \triangleq E\left[e_k e_k^T\right]$, $\Sigma_k^{\hat{\mathbf{x}}} \triangleq E\left[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T\right]$, *and* $\Sigma_k^{\hat{\mathbf{x}}e} \triangleq E\left[\hat{\mathbf{x}}_k e_k^T\right]$, *the optimal filter gain for system (5.4)-(5.5) is computed as*

$$\hat{\mathbf{x}}_{k+1} = A_k\hat{\mathbf{x}}_k + B_k\boldsymbol{\pi}_k + K_k(\mathbf{y}_k - F_k\hat{\mathbf{x}}_k - E_k\boldsymbol{\pi}_k), \tag{5.54}$$

$$K_k = A_k\Sigma_k^e F_k^T\left(F_k\Sigma_k^e F_k^T + \mathcal{P}_k\right)^{-1}, \tag{5.55}$$

$$m_{k+1}^{\hat{\mathbf{x}}} = (A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e + B_kl_k, \qquad m_1^{\hat{\mathbf{x}}} = \hat{\mathbf{x}}_1, \tag{5.56}$$

$$m_{k+1}^e = (A_k - K_kF_k)m_k^e, \qquad m_1^e = 0, \tag{5.57}$$

$$\begin{aligned}\Sigma_{k+1}^{\hat{\mathbf{x}}} = {} & (A_k + B_kL_k)\Sigma_k^{\hat{\mathbf{x}}}(A_k + B_kL_k)^T + K_kF_k\Sigma_k^eA_k^T + (A_k + B_kL_k)\Sigma_k^{\hat{\mathbf{x}}e}F_k^TK_k^T \\ & + K_kF_k\Sigma_k^{e\hat{\mathbf{x}}}(A_k + B_kL_k)^T + \Big((A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e\Big)l_k^TB_k^T \\ & + B_kl_k\Big((A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e\Big)^T + B_kl_kl_k^TB_k^T, \qquad \Sigma_1^{\hat{\mathbf{x}}} = \hat{\mathbf{x}}_1\hat{\mathbf{x}}_1^T,\end{aligned} \tag{5.58}$$

$$\Sigma_{k+1}^e = (A_k - K_kF_k)\Sigma_k^eA_k^T + \mathcal{M}_k, \qquad \Sigma_1^e = \Sigma_1, \tag{5.59}$$

$$\Sigma_{k+1}^{\hat{\mathbf{x}}e} = (A_k + B_kL_k)\Sigma_k^{\hat{\mathbf{x}}e}(A_k - K_kF_k)^T + B_kl_k(m_k^e)^T(A_k - K_kF_k)^T, \quad \Sigma_1^{\hat{\mathbf{x}}e} = 0, \tag{5.60}$$

and

$$\begin{aligned}\mathcal{P}_k = \sum_{i=1}^{n_v}\Big[& \mathbf{d}_{i,k}(m_k^{\hat{\mathbf{x}}} + m_k^e)^T(D_{i,k}^{\mathbf{x}})^T + D_{i,k}^{\mathbf{x}}(m_k^{\hat{\mathbf{x}}} + m_k^e)\mathbf{d}_{i,k}^T + \mathbf{d}_{i,k}(l_k + L_km_k^{\hat{\mathbf{x}}})^T(D_{i,k}^{\mathbf{u}})^T \\ & + D_{i,k}^{\mathbf{u}}(l_k + L_km_k^{\hat{\mathbf{x}}})\mathbf{d}_{i,k}^T + \mathbf{d}_{i,k}\mathbf{d}_{i,k}^T + D_{i,k}^{\mathbf{x}}\Big((m_k^{\hat{\mathbf{x}}} + m_k^e)l_k^T + (\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{e\hat{\mathbf{x}}})L_k^T\Big)(D_{i,k}^{\mathbf{u}})^T \\ & + D_{i,k}^{\mathbf{u}}\Big(l_k(m_k^{\hat{\mathbf{x}}} + m_k^e)^T + L_k(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e})\Big)(D_{i,k}^{\mathbf{x}})^T + D_{i,k}^{\mathbf{x}}(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e)(D_{i,k}^{\mathbf{x}})^T \\ & + D_{i,k}^{\mathbf{u}}\Big(l_kl_k^T + l_k(m_k^{\hat{\mathbf{x}}})^TL_k^T + L_km_k^{\hat{\mathbf{x}}}l_k^T + L_k\Sigma_k^{\hat{\mathbf{x}}}L_k^T\Big)(D_{i,k}^{\mathbf{u}})^T\Big],\end{aligned} \tag{5.61}$$

$$\mathcal{M}_k = \sum_{i=1}^{n_\omega} \Big[\mathbf{c}_{i,k}(m_k^{\hat{\mathbf{x}}} + m_k^e)^T (C_{i,k}^{\mathbf{x}})^T + C_{i,k}^{\mathbf{x}}(m_k^{\hat{\mathbf{x}}} + m_k^e)\mathbf{c}_{i,k}^T + \mathbf{c}_{i,k}(l_k + L_k m_k^{\hat{\mathbf{x}}})^T (C_{i,k}^{\mathbf{u}})^T$$
$$+ C_{i,k}^{\mathbf{u}}(l_k + L_k m_k^{\hat{\mathbf{x}}})\mathbf{c}_{i,k}^T + \mathbf{c}_{i,k}\mathbf{c}_{i,k}^T + C_{i,k}^{\mathbf{x}}\Big((m_k^{\hat{\mathbf{x}}} + m_k^e) l_k^T + (\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{e\hat{\mathbf{x}}}) L_k^T\Big)(C_{i,k}^{\mathbf{u}})^T$$
$$+ C_{i,k}^{\mathbf{u}}\Big(l_k (m_k^{\hat{\mathbf{x}}} + m_k^e)^T + L_k(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e})\Big)(C_{i,k}^{\mathbf{x}})^T + C_{i,k}^{\mathbf{x}}(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e)(C_{i,k}^{\mathbf{x}})^T$$
$$+ C_{i,k}^{\mathbf{u}}\Big(l_k l_k^T + l_k (m_k^{\hat{\mathbf{x}}})^T L_k^T + L_k m_k^{\hat{\mathbf{x}}} l_k^T + L_k \Sigma_k^{\hat{\mathbf{x}}} L_k^T\Big)(C_{i,k}^{\mathbf{u}})^T \Big]. \tag{5.62}$$

Proof. Rewrite the system dynamics and state estimator as the following

$$\begin{aligned}
\mathbf{x}_{k+1} &= A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathcal{C}_k(\mathbf{x}_k, \mathbf{u}_k)\xi_k, \\
\mathbf{y}_k &= F_k \mathbf{x}_k + E_k \mathbf{u}_k + \mathcal{D}_k(\mathbf{x}_k, \mathbf{u}_k)\eta_k, \\
\hat{\mathbf{x}}_{k+1} &= A_k \hat{\mathbf{x}}_k + B_k \mathbf{u}_k + K_k(\mathbf{y}_k - F_k \hat{\mathbf{x}}_k - E_k \mathbf{u}_k),
\end{aligned}$$

where $\mathbf{u}_k = \boldsymbol{\pi}_k(\hat{\mathbf{x}}_k) = l_k + L_k \hat{\mathbf{x}}_k, (k = 1, \cdots, N-1)$, note that we will use the shortcut $\boldsymbol{\pi}_k$ in place of the control signal for the convenience, and K_k is the filter gain that minimizes the functional

$$J = E\left[e_{k+1}^T \mathcal{T} e_{k+1}\right], \qquad \mathcal{T} \geq 0, \tag{5.63}$$

where the estimation error $e_{k+1} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}$, and the estimation error dynamics is given by

$$e_{k+1} = (A_k - K_k F_k)e_k + \mathcal{C}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\xi_k - K_k \mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\eta_k. \tag{5.64}$$

Based on the estimation error dynamics (5.64), the conditional mean and covariance of e_{k+1} are

$$E\left[e_{k+1} | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right] = (A_k - K_k F_k)\, e_k, \tag{5.65}$$

$$Cov\left[e_{k+1} | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right] = \mathcal{C}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\, \mathcal{C}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)^T + K_k \mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\, \mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)^T K_k^T, \tag{5.66}$$

By applying the properties of conditional expectation, we obtain

$$E\left[e_{k+1} e_{k+1}^T | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right] = Cov\left[e_{k+1} | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right] + E\left[e_{k+1} | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right] \left(E\left[e_{k+1} | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right]\right)^T, \tag{5.67}$$

$$E\left[e_{k+1} e_{k+1}^T\right] = E\Big[E\left[e_{k+1} e_{k+1}^T | \mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right]\Big], \tag{5.68}$$

The terms in $E\left[e_{k+1}e_{k+1}^T|\mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\pi}_k\right]$ that dependent on K_k are

$$(A_k - K_k F_k)e_k e_k^T (A_k - K_k F_k)^T + K_k \mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)^T K_k^T.$$

With the definition of $\Sigma_k^e \triangleq E\left[e_k e_k^T\right]$ and

$$\mathcal{P}_k = E\left[\mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\, \mathcal{D}_k^T(\mathbf{x}_k, \boldsymbol{\pi}_k)\right], \tag{5.69}$$

the unconditional expectation of the K_k-dependent expression above becomes

$$(A_k - K_k F_k)\Sigma_k^e (A_k - K_k F_k)^T + K_k \mathcal{P}_k K_k^T,$$

Using the fact that $E\left[e_{k+1}^T \mathcal{T} e_{k+1}\right] = tr(E\left[e_{k+1}e_{k+1}^T\right]\mathcal{T})$, it follows that the K_k-dependent terms in J becomes

$$\beta(K_k) = tr\mathcal{T}(A_k - K_k F_k)\Sigma_k^e (A_k - K_k F_k)^T + tr\mathcal{T} K_k \mathcal{P}_k K_k^T, \tag{5.70}$$

the minimum of $\beta(K_k)$ is found by setting the derivative with respect to K_k to zero. Using the matrices identities $\frac{\partial}{\partial X} tr(XA) = A^T$ and $\frac{\partial}{\partial X} tr(AXBX^T) = A^T X B^T + AXB$, we obtain

$$\frac{\partial \beta(K_k)}{\partial K_k} = \mathcal{T}\Big(K_k\left(F_k \Sigma_k^e F_k^T + \mathcal{P}_k\right) - A_k \Sigma_k^e F_k^T\Big) = 0, \tag{5.71}$$

hence

$$K_k = A_k \Sigma_k^e F_k^T \left(F_k \Sigma_k^e F_k^T + \mathcal{P}_k\right)^{-1}. \tag{5.72}$$

To complete the proof of the lemma, we need to compute the unconditional covariance. By substituting the control law $\mathbf{u}_k = \boldsymbol{\pi}_k(\hat{\mathbf{x}}_k) = l_k + L_k \hat{\mathbf{x}}_k$, we can rewrite the state estimator as

$$\hat{\mathbf{x}}_{k+1} = (A_k + B_k L_k)\hat{\mathbf{x}}_k + B_k l_k + K_k F_k e_k + K_k \mathcal{D}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\eta_k, \tag{5.73}$$

With the definition of the unconditional means $m_k^e, m_k^{\hat{\mathbf{x}}}$, the unconditional covariances $\Sigma_k^{\hat{\mathbf{x}}}, \Sigma_k^{\hat{\mathbf{x}}e}$ and the additional definition

$$\mathcal{M}_k = E\left[\mathcal{C}_k(\mathbf{x}_k, \boldsymbol{\pi}_k)\, \mathcal{C}_k^T(\mathbf{x}_k, \boldsymbol{\pi}_k)\right], \tag{5.74}$$

given in lemma 3, using the fact that $\Sigma_k^{e\hat{\mathbf{x}}} = (\Sigma_k^{\hat{\mathbf{x}}e})^T$ and $\Sigma_k^{\mathbf{x}} = E\left[(\hat{\mathbf{x}}_k + e_k)(\hat{\mathbf{x}}_k + e_k)^T\right] = \Sigma_k^{\mathbf{x}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e$ and the equation (5.64) and (5.73), the updates for the unconditional means and covariances are

$$
\begin{aligned}
m_{k+1}^{\hat{\mathbf{x}}} &= (A_k + B_k L_k) m_k^{\hat{\mathbf{x}}} + K_k F_k m_k^e + B_k l_k, \\
m_{k+1}^e &= (A_k - K_k F_k) m_k^e, \\
\Sigma_{k+1}^{\hat{\mathbf{x}}} &= (A_k + B_k L_k) \Sigma_k^{\hat{\mathbf{x}}} (A_k + B_k L_k)^T + K_k F_k \Sigma_k^e F_k^T K_k^T + K_k \mathcal{P}_k K_k^T \\
&\quad + (A_k + B_k L_k) \Sigma_k^{\hat{\mathbf{x}}e} F_k^T K_k^T + K_k F_k \Sigma_k^{e\hat{\mathbf{x}}} (A_k + B_k L_k)^T \\
&\quad + \Big((A_k + B_k L_k) m_k^{\hat{\mathbf{x}}} + K_k F_k m_k^e\Big) l_k^T B_k^T \\
&\quad + B_k l_k \Big((A_k + B_k L_k) m_k^{\hat{\mathbf{x}}} + K_k F_k m_k^e\Big)^T + B_k l_k l_k^T B_k^T, \qquad (5.75) \\
\Sigma_{k+1}^e &= (A_k - K_k F_k) \Sigma_k^e (A_k - K_k F_k)^T + K_k \mathcal{P}_k K_k^T + \mathcal{M}_k, \qquad (5.76) \\
\Sigma_{k+1}^{\hat{\mathbf{x}}e} &= (A_k + B_k L_k) \Sigma_k^{\hat{\mathbf{x}}e} (A_k - K_k F_k)^T + B_k l_k (m_k^e)^T (A_k - K_k F_k)^T \\
&\quad + K_k F_k \Sigma_k^e (A_k - K_k F_k)^T - K_k \mathcal{P}_k K_k^T, \qquad (5.77)
\end{aligned}
$$

By substituting (5.72) into the above equations and combining the term $K_k F_k \Sigma_k^e F_k^T K_k^T + K_k \mathcal{P}_k K_k^T$, we can rewrite the update equations (5.75)-(5.77) into forms (5.58)-(5.60) which are exactly the same as what we obtain in lemma 3.

Furthermore, based on the definition of $\mathcal{P}_k$ and $\mathcal{M}_k$ given in (5.69) and (5.74), and all the definitions of the unconditional means and unconditional covariances, $\mathcal{P}_k$ and $\mathcal{M}_k$ can be computed using (5.61)-(5.62) which completes the proof. □

5.6 Application to 2-link Torque-controlled Arm Movements

We have thus far tested the algorithm on the reaching movements for a 2-link arm model as described in section 3.1.1. However, what the most important here is that we take into account the disturbance existed in the motor system — in particular, the motor noise is modelled as control-dependent, with standard deviation proportional to the mean of the control signal. Rewrite the dynamics of system as following

$$\dot{x} = F(x) + G(x)(1 + \sigma_u \varepsilon)u, \qquad (5.78)$$

where the state and control are given by $x = (\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2)^T,\ u = \tau = (\tau_1\ \tau_2)^T$. The control inputs u is disturbed by the multiplicative noise, whose standard deviation is 20% of its magnitude — which means $\sigma_u = 0.2$ in (5.78), while ε is a zero-mean Gaussian white noise with unity covariance.

The sensory feedback carries the information about position and velocity

$$y = (\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2)^T + \zeta, \tag{5.79}$$

where the sensory noise ζ has zero-mean Gaussian distribution with unity covariance.

5.6.1 Center-out reaching task

In order to demonstrate the effectiveness of our design, we applied ILQG method to the human arm model described above. Note that, in this model, we introduce a signal-dependent noise whose standard deviation is 20% of the control signal magnitude.

Here we use the center-out reaching task which is commonly studied in the Motor Control — the targets e^* (shown as stars in Fig. 5.1A) are arranged in a circle with 0.1m radius around the starting position. The arm starts from rest at $(\theta_1 = \pi/4, \theta_2 = \pi/2)$, and has to reach a specified target in $0.5s$, with minimal control energy. The cost function is defined as

$$J_1 = \left\| e(\theta(T)) - e^* \right\|^2 + 0.001 \left\| \dot{e}\Big(\theta(T), \theta(\dot{T})\Big) \right\|^2 + \frac{1}{2} \int_0^T 0.0001 \|u\|^2\, dt, \tag{5.80}$$

where $e(\theta)$ and $\dot{e}(\theta, \dot{\theta})$ is the forward kinematics transformation from joint coordinates to Cartesian hand coordinates.

Figure 5.1 shows average behavior for the fully observable case: hand paths in (A), tangential speed profiles in (B). We find out that the movement kinematics share many features with experimental data on human arm movements — nearly straight movement paths and bell-shaped speed profiles.

Now we look at the partial observable case where the states of system are obtained by the estimator. Although the state of the controlled plant are only available through noisy measurement, the movement trajectories shown in Fig. 5.2A illustrates that the hand

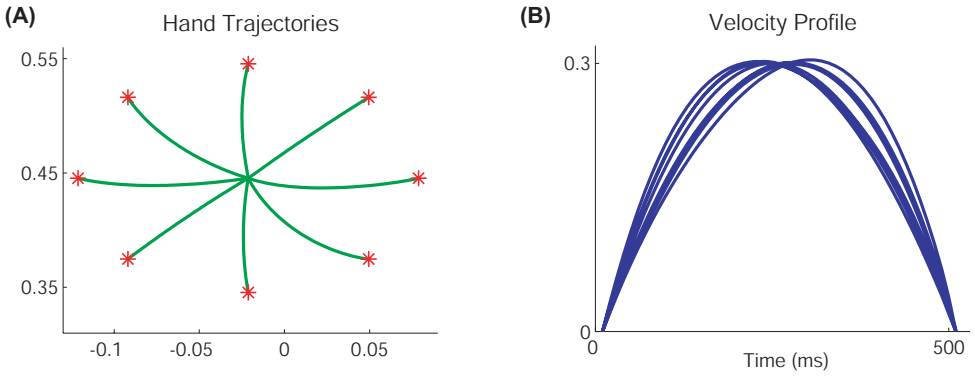


Figure 5.1. Fully observable case: average behavior of the ILQG controller for reaching movements, using a 2-link human arm model. (A) Hand paths for movement in 8 directions; (B) Speed profiles.

could still arrive to the desired target position accurately. Another encouraging result is that in terms of CPU time. Fig. 5.2B shows how the total cost decreased over iterations of the algorithm, for reaching in 8 different directions. On average, ILQG found a locally-optimal time-varying feedback control law in about 8 seconds (on a 2.8GHz Pentium 4 machine, in Matlab).

Trajectory-based algorithms related to Pontryagins Maximum Principle in general find locally-optimal solutions, and complex control problems may exhibit many local minima. Finally, we explored the issue of local minima for the arm control problem. We used 50 different initializations, for each of 8 movement directions. The final trajectories are given in Fig. 5.3, where Fig. 5.3A shows that, for the fully observable case, all the optimization runs converge to a solution very similar to the best solution we find for the corresponding target direction. Fig. 5.3B shows how the cloud of 50 randomly initialized trajectories gradually converge for the partial observable case by using ILQG method. There are local minima, but half the time the algorithm converges to the same result. Therefore, the derived algorithm is relatively very robust, and a small number of restarts of ILQG are sufficient to discover what appears to be the global minimum in a relatively complex control problem.

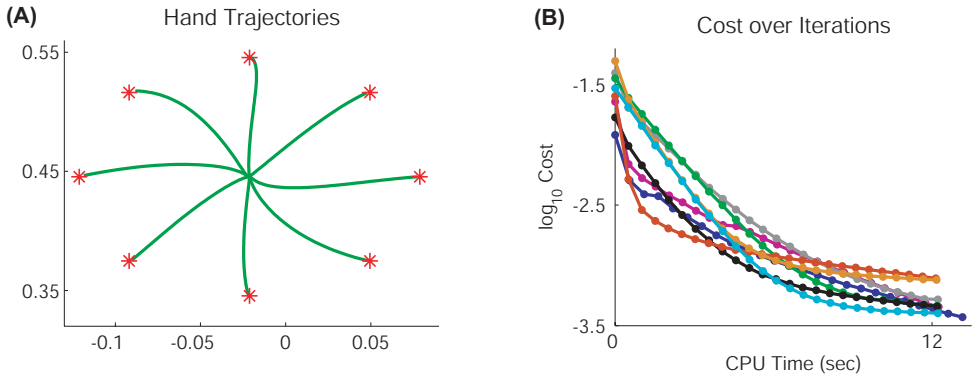


Figure 5.2. Partial observable case: average behavior of the ILQG controller and estimator for reaching movements, using a 2-link human arm model. (A) Hand paths for movement in 8 directions; (B) Cost over iterations.

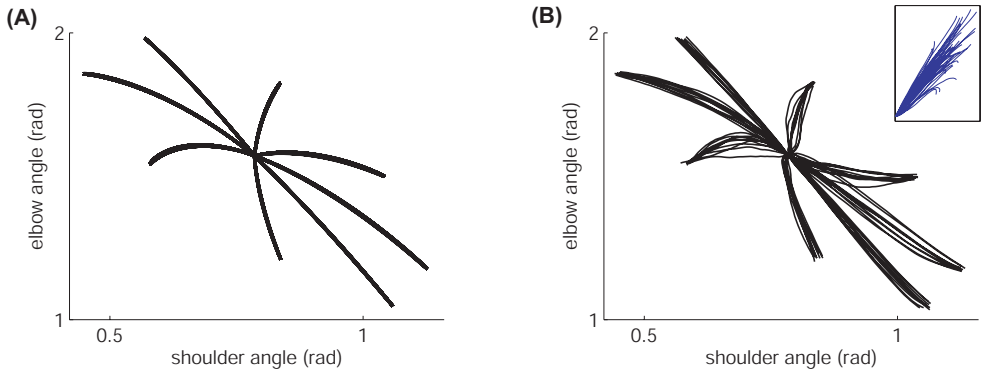


Figure 5.3. Hand paths for random 50 initial control laws (blue, inset) and optimized paths (black) to 8 targets obtained by using those initial conditions. (A) fully observable case; (B) partial observable case.

5.6.2 Reaching task with the obstacle avoidance

The second task is to implement the reaching task and to avoid the obstacle during the movement, while the obstacle is defined as a circle with a certain radius $r_{obstacle} = 0.02m$, and is arranged in the fixed position. The distance between the starting position and the target is about 0.15m. The arm starts from rest at $\theta_1 = \pi/4, \theta_2 = \pi/2$, and has to reach the specified target and avoid the obstacle during the reaching, with minimal

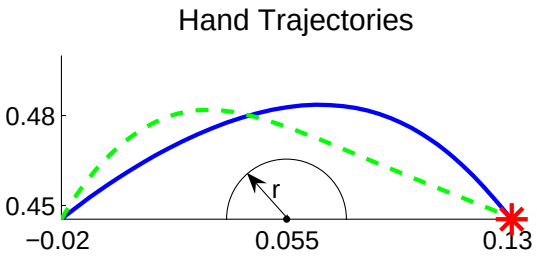


Figure 5.4. Average behavior of the ILQG controller and estimator for reaching movement with obstacle avoidance, using a 2-link human arm model. Blue curve: fully observable case; green dashed curve: partial observable case. Note that obstacle circle radius $r = 0.02m$.

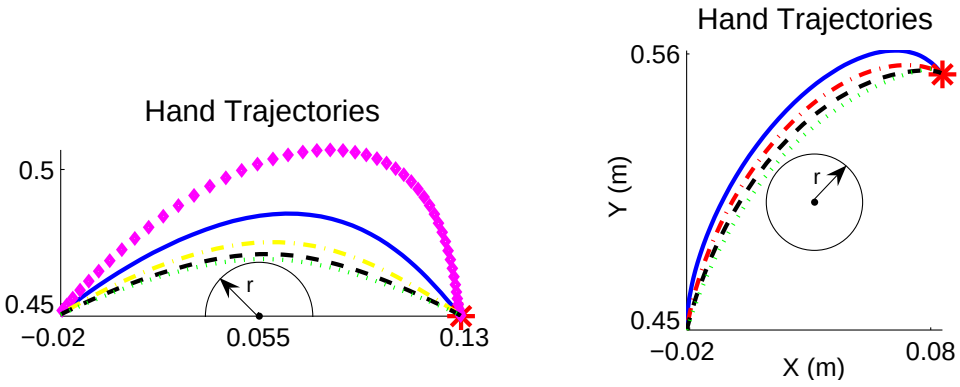
control energy. The cost function (3.21) is rewritten here as

$$J_2 = \left\| e(\theta(T)) - e^* \right\|^2 + 0.001 \left\| \dot{e}\Big(\theta(T), \dot{\theta}(T)\Big) \right\|^2 + \frac{1}{2}\int_0^T 0.0001\|u\|^2\, dt + q_{obstable},$$
$$q_{obstable} = \begin{cases} k1 \int_0^T \Big(\ell(\theta(t)) - r_{obstacle}\Big)^{-2} dt & when\ l > r \\ \infty & otherwise \end{cases} \tag{5.81}$$

where the target e^* is defined in end-point coordinates; the weighting coefficient $k_1 = 1e-8$, the obstacle radius $r_{obstacle}$ is equal to $0.02m$.

Figure 5.4 shows the movement trajectories and illustrates how the hand avoid the obstacle circle (marked as dark) and arrive to the desired target position (marked as red star) as close as possible. The blue curve is obtained under the condition that the state variable of the controlled plant is available; while the green dashed curve illustrates the hand movement trajectory tying together the optimal feedback control and estimation design. In the later case, the state of the controlled plant are approximately computed, because of the existence of estimation error, we found out that the green curve becomes more curved at the beginning of the movement compared with the result on fully observable case.

Here the optimal control problem is solved for minimizing a performance criterion. Fig. 5.5(a) shows the behavior of movement trajectories by changing the penalty weighting k_1 on the obstacle avoidance in the objective function (5.81). The bigger the weight



(a) contribution of obstacle cost rate

(b) contribution of movement duration

Figure 5.5. (a) Comparison of movement behavior by choosing different weighting coefficients k_1 on the obstacle cost rate (fully observable case). Magenta diamond: $k_1 = 1e-7$; Blue solid: $k_1 = 1e-8$; Yellow dashdot: $k_1 = 1e-9$; Black dashed: $k_1 = 1e-10$; Green dotted: $k_1 = 1e-11$. (b) Comparison of movement behavior by choosing different movement duration (fully observable case). Blue solid: $700msec$; Red dashdot: $500msec$; Black dashed: $350msec$; Green dotted: $200msec$. The obstacle $r = 0.02m$.

penalty k_1 is, the hand movement trajectory is further away from the obstacle, and Fig. 5.5(a) exactly explains this phenomena. Another interesting result in behavioral experiments shows that a longer movement duration can be predicted to cause a more curved trajectory. Our simulation result Fig. 5.5(b) quantitatively supported those previous studies with an enormous amount of data.

5.6.3 Comparison on a family of robotic manipulators

Our method was compared on a family of robotic manipulators (Fig. 5.6a), which were simulated with the Matlab Robotics Toolbox [20]. The manipulators had between 2 and 7 hinge joints and moved in the plane. The kinematics were scaled so that all links of a given manipulator were equal in length, and the end effector (filled circle) was 1 m away from the base in the shown configuration. The dynamics were also scaled, so that all links had the same mass and the sum of all link masses was 1 kg. Material density was kept constant, and set so that a 1 m cylindrical link with diameter 0.1 m had mass

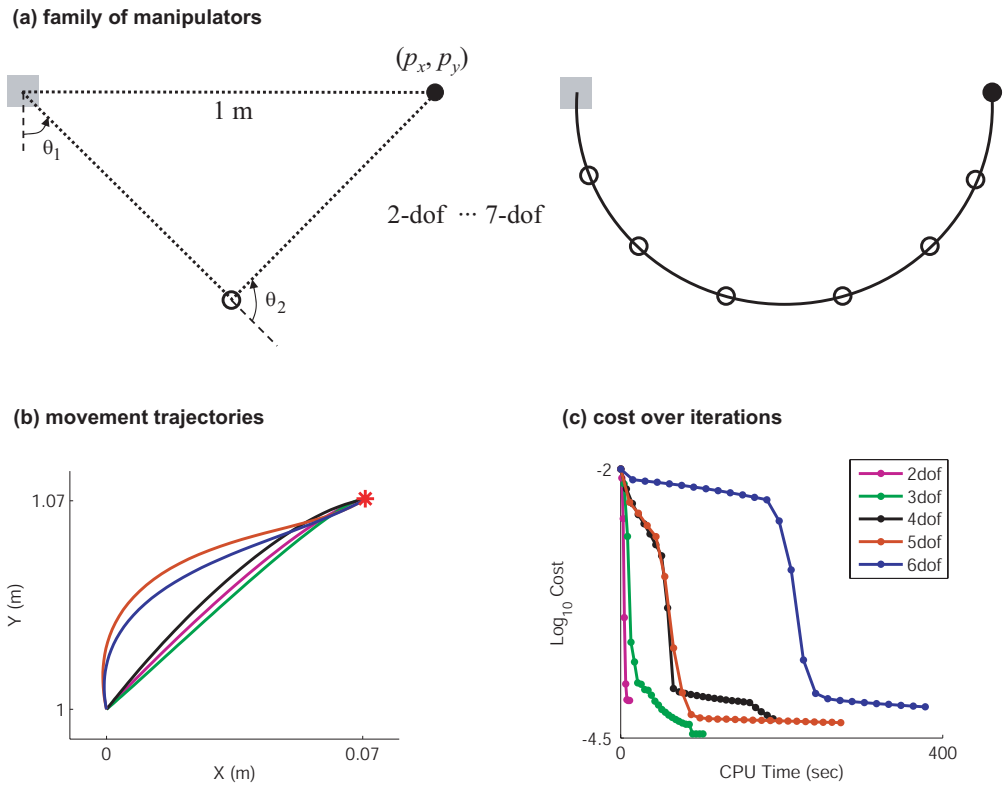


Figure 5.6. Comparison of control methodologies on a family of robotic manipulators.

1 kg. The link diameter for each manipulator was computed given the constraint on total mass and the fixed material density. Link inertia was then found assuming uniform density.

We defined an optimal control problem with the following total cost:

$$J_3 = \left\|p(T) - p^*\right\|^2 + 0.001\left\|\dot{p}(T)\right\|^2 + \frac{1}{2}\int_0^T 0.0001\|u\|^2\, dt, \tag{5.82}$$

where p^* is the target (in cartesian space) of a reaching movement executed with the manipulator, $N = 50$ is the number of time steps, step duration is $\triangle t = 0.01s$ (resulting in a $T = 0.5s$ movement). Target is 0.1m away from the starting position.

We compared two indices of performance for each manipulator. The first index is movement trajectories (Fig. 5.6b). In the non-redundant case (degree of freedom (DOF) = 2), the movement trajectory is almost straight, but as redundancy increases we see clear difference, the movement path is getting curved. The second index is the efficiency

of the derived algorithm. Fig. 5.6c illustrates how the cost of the nominal trajectory decreases with the number of iterations. Compared with the 2DOF manipulator, the more complicated manipulator will take longer time to converge and find the optimal solution.

5.7 Application to 2-link 6-muscle Arm Movements

Now we applied iLQG method to the 2 link 6 muscle arm (**MODEL 2**) given in (3.15). Note that this model is stochastic: we include signal-dependent noise in the control signal, with standard deviation equal to 20% of the control signal. Again we test the algorithm on the center-out reaching task — where the targets are arranged in a circle with 0.1 m radius around the starting position. The arm starts from the rest position at $(\theta_1 = \pi/4, \theta_2 = \pi/2)$, and has to reach a specified target in $350ms$.

Figure 5.7 shows average behavior for the fully observable case: hand paths in (A), tangential speed profiles in (B), and muscle activations in (C). We find out that both the movement kinematics and the muscle activations share many features with experimental data on human arm movements — but a detailed discussion of the relevance to Motor Control is beyond the scope of this thesis. Another encouraging result is that in terms of CPU time. On average, the algorithm can find a locally-optimal time-varying feedback control law in about 12 seconds (on a 2.8GHz Pentium 4 machine, in Matlab) on the arm reaching task in the absence of noise, and 16 seconds when multiplicative noise is included.

Figure 5.8 illustrates the robustness to noise: open-loop control in (A), closed-loop control in (B), and closed-loop control optimized for a deterministic system in (C). Closed-loop control is based on the time-varying feedback gain matrix L generated by the ILQG method, while open-loop control only uses the final u constructed by the algorithm. As the endpoint error ellipses show, the feedback control scheme substantially reduces the effects of the noise, and benefits from being optimized for the correct multiplicative noise model.

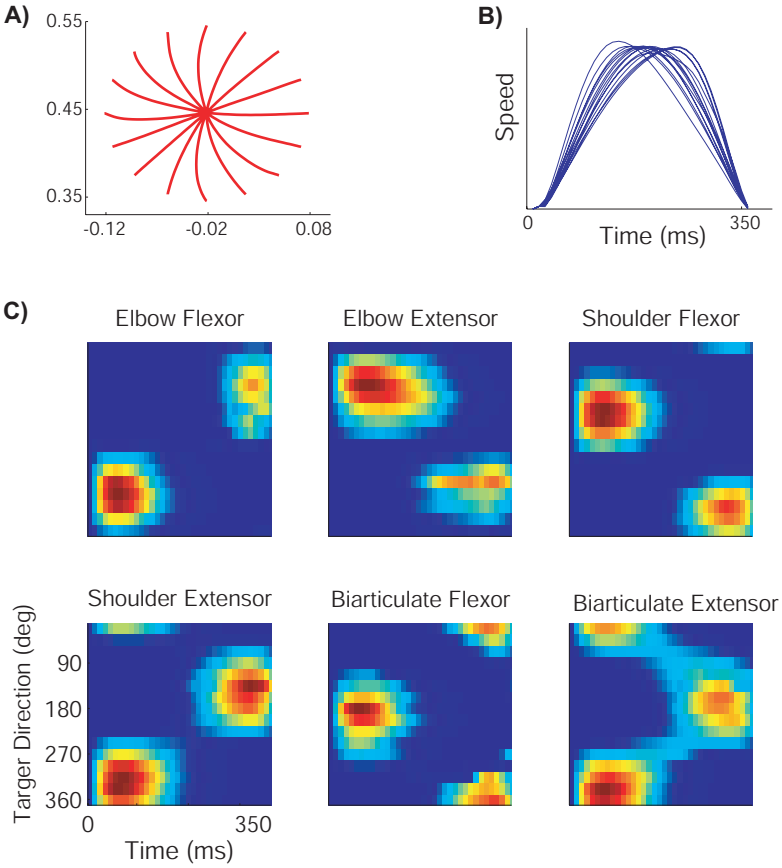


Figure 5.7. Fully observable case: average behavior of the ILQG controller for reaching movements, using a 2-link 6-muscle human arm model. (A) Hand paths for movement in 16 directions; (B) Speed profiles; (C) Muscle activations.

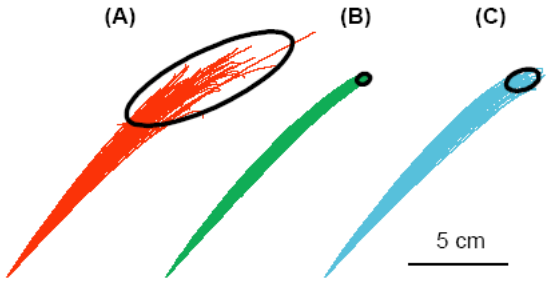


Figure 5.8. Effects of control-dependent noise on hand reaching trajectories, under different control laws. (A) open-loop control; (B) closed-loop control; (C) closed-loop controller optimized for deterministic system.

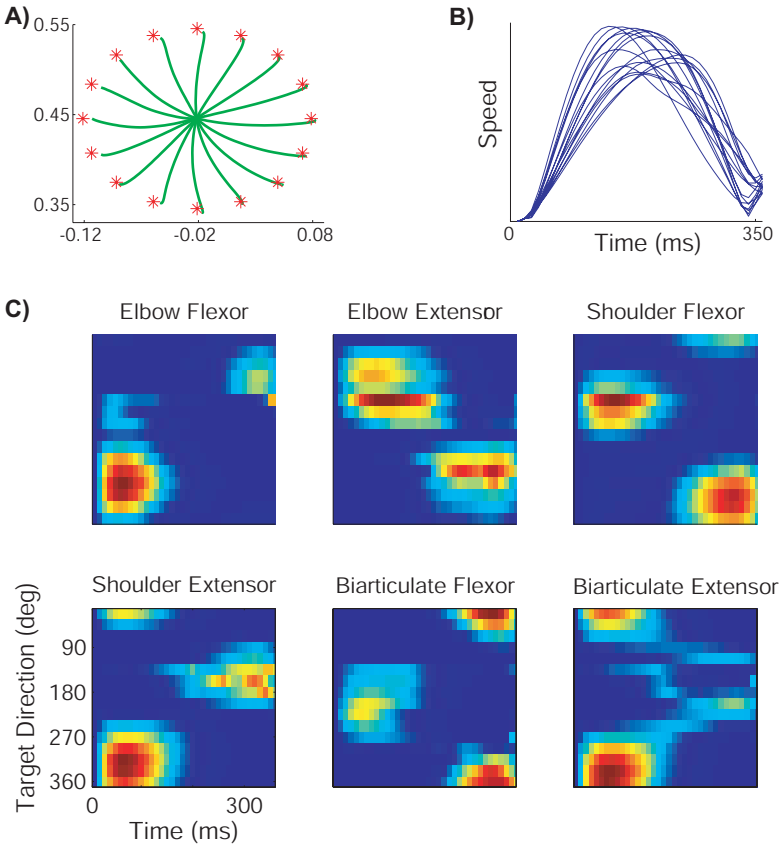


Figure 5.9. Partial observable case: average behavior of the ILQG controller and estimator for reaching movements, using a 2-link 6-muscle human arm model. (A) Hand paths for movement in 16 directions; (B) Speed profiles; (C) Muscle activations.

Now we look at the partial observable case where the state of the controlled plant are only available through noisy measurement. Using the optimal estimator designed in section 5.5, Fig. 5.9A shows that the hand could still reach the target position (shown as red stars in Fig. 5.9A) as accurately as possible. Fig. 5.9B shows tangential speed profiles, for reaching in 16 different directions, and they remain in bell-shaped. Fig. 5.9C shows the muscle activation patterns for elbow flexor, elbow extensor, shoulder flexor, shoulder extensor, biarticulate flexor and biarticulate extensor.

To test for the presence of multiple local minima on this arm control problem, we initialize the ILQG algorithm with 50 random control sequences, for each of 8 movement directions. Fig. 5.10 illustrate the final optimal movement trajectories. For the fully

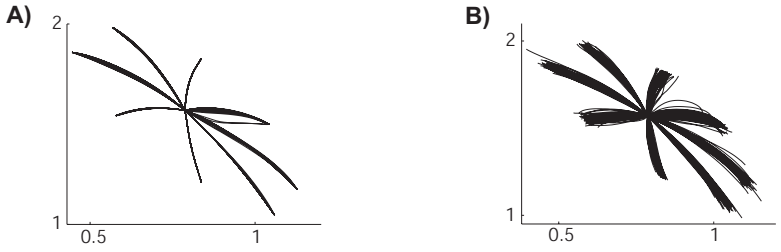


Figure 5.10. The optimized hand paths obtained by using 50 different initial conditions for each of 8 movement directions. (A) fully observable case; (B) partial observable case.

observable case, Fig. 5.10A shows that, Over 97% of the optimization runs (10 out of 400) converge to a solution very similar to the best solution we find for the corresponding target direction. Fig. 5.10B shows how the cloud of 50 randomly initialized trajectories gradually converge for the partial observable case.

5.8 Summary

Linear quadratic Gaussian (LQG) control is one of the most fundamental and widely used tools in modern engineering. However, the most problematic assumption of classical LQG control theory for stochastic linear systems is that the stochastic disturbances are additive and not control or state dependent. Motor noise in biological movement systems that we are interested in has been found to be multiplicative [38, 116], with standard deviation proportional to the magnitude of the control signal. And it is becoming increasingly clear that some extensively studied phenomena, such as trajectory smoothness, speed-accuracy trade-offs, task-dependent impedance, and structured variability, are closely related to the multiplicative nature of motor noise. For the real control system design, feedback is based on delayed and noisy sensors that may not measure all the state variables, it is therefore desirable to extend the algorithm to the partially observable case by combining it with an extended Kalman filter. This results in a coupled estimation-control problem, which is more complicated in the presence of multiplicative noise.

This chapter presents an iterative Linear-Quadratic-Gaussian method for locally-

optimal control and estimation of nonlinear stochastic dynamical systems subject to control constraints. The new method constructs an affine feedback control law obtained by minimizing a novel quadratic approximation to the optimal cost-to-go function. It also constructs a modified extended Kalman filter corresponding to the control law. The control law and filter are iteratively improved until convergence. Although our work is motivated by studying biological movement control, the present results could be of interest to a wider audience. The most important is that our approach yields a numerical algorithm with stable convergence achieved through backtracking line search; and convergence in the vicinity of a local minimum is quadratic. Finally, the application of the algorithm is illustrated in the context of reaching movements and obstacle avoidance on a complex biomechanical control problem involving a stochastic model of the human arm.

The text of this chapter, in part, has been submitted for publication in Automatica. The thesis author was the primary researcher and author in these works and the co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

Chapter 6

Hierarchical Optimal Control for Biological Movement Systems

6.1 Introduction

Neural control of movement is accomplished by a complex hierarchy of recurrently connected brain regions. Understanding how the multiple levels of the sensorimotor system cooperate to produce integrated action has been a central theme in Neuroscience since the pioneering work of Sherrington. Perhaps the most thorough investigation of the levels of human motor control was undertaken by Bernstein [8], who concluded that every motor skill involves at least two levels of feedback control: a leading level that monitors progress and steers the biomechanical plant towards the achievement of the task goals, and a background level that provides various automatisms and corrections that help the leading level. More recently, recordings in a variety of motor areas have revealed a progression of increasingly abstract (and increasingly hard to describe) neural representations. The development of the so-called hierarchical control has also been extended to large-scale control systems [46, 65, 105]. Through multilevel methodologies, a large-scale control system's complexity can be relaxed by solving a family of subproblems which are of smaller dimensions and are more easily handled. But despite this long standing interest and the wealth of relevant data and intuitive notions, quantita-

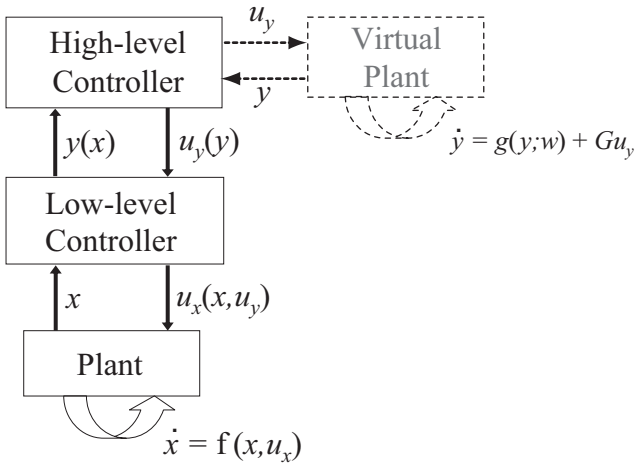


Figure 6.1. Illustration of the hierarchical control structure

tive theories of hierarchical sensorimotor control are surprisingly rare [73]. Indeed, the computational ideas that can provide a solid foundation for such theories have emerged only recently. The objective here is to synthesize these ideas into a general approach to hierarchical feedback control, and illustrate the approach with a specific model of arm movement control.

The structure of biological movements examined in this chapter involves two-level feedback control hierarchies (Fig. 6.1): a low-level controller is responsible for transforming high-level commands into appropriate muscle activations, and also for transforming the dynamics of the real plant into the virtual dynamics assumed by the high level. The high-level controller performs goal-directed movement control, but is partly isolated from the complexity of the physical plant. As shown in Fig. 6.1, the low-level controller receives information about the plant state x, as well as a high-level control signal u_y, and generates appropriate muscle activations u_x. It also produces a more abstract state representation y that is sent to the high-level controller. The high-level controller receives y and generates abstract control command u_y to coordinate the low-level unit.

Here we focus on the study of hierarchical optimal control for human arm movements. The optimal feedback controller on the high-level will be designed using an iterative Linear-Quadratic-Gaussian (iLQG) method derived in the previous chapter. The

simplified virtual model assumed on the high-level may not be a perfect model of the augmented plant, therefore, an identification process is also included which can continuously calibrate the expectations on the high level. The low-level has to activate muscles so as to accomplish desired effects in the task-relevant space. Since there are more muscles than task variables, the low-level faces a redundancy problem which will be solved using static optimization. The control of biological movement systems is a very complex problem, because such systems have nonlinear dynamics and high-dimensionality, preventing the use of many traditional methods for controller design. The goal of this chapter is to combine the optimality principle and hierarchical strategy in order to develop a new framework for the sensorimotor system, which hopefully could better reflect the real control problem that the brain faces.

This chapter is organized as follows. In section 6.2 we introduce the biological motivation of hierarchical control scheme and its relation to optimal feedback control. In section 6.3 the general hierarchical feedback control framework for the nonlinear system is formulated. Section 6.4 presents the numerical results for a realistic human arm model, in the context of reaching task.

6.2 Biological Motivation, and relation to Optimal Control

6.2.1 Minimal intervention principle

The recent theory of motor coordination [116, 117] has revealed a somewhat surprising fact about the optimal way to control redundant systems. They have shown that optimal feedback controllers obey a "minimal intervention" principle: rather than correcting all deviations from a desired trajectory, such controllers only act in the task-relevant subspace and leave task-irrelevant deviations uncorrected. Loosely speaking, this is done by extracting a small set of task-relevant features (through a set of sensory synergies), performing feedback control in that feature space, and mapping the resulting abstract controls into control signals for the real actuators (through a set of motor synergies). The difficulty with optimal feedback controllers for complex systems is that

they are extremely hard to design — perhaps even for the brain.

The minimal intervention principle suggests a natural approximation to optimal feedback control in redundant tasks, using a two-level feedback control hierarchy. The basic idea (see Fig. 6.1) is to design optimal feedback controllers that optimize the true cost function, but for a simpler virtual plant with state $y(x)$. This corresponds to Bernsteins leading level of control, where the brain monitors a small number of task parameters y instead of the full state x, and generates an abstract control signal u_y interpreted as desired change in y. The background level then has the job of transforming the dynamics of the actual plant into that of the virtual plant, i.e. computing the real control signal $u_x(x, u_y)$ such that the desired change in $y(x)$ is accomplished, with minimal control effort.

6.2.2 Structured motor variability

A signature of both optimal feedback control and hierarchical control schemes is selective control of task-relevant parameters, which implies increased variability in task-irrelevant parameters. This phenomenon has been observed in a wide range of behaviors, and has been quantified via the "uncontrolled manifold" method for comparing task-relevant and task-irrelevant variance [101]. It is also illustrated here with data from the following experiment. Six subjects were asked to move repeatedly between three targets attached to their body (one target on each leg, and one on the left upper arm). Subjects held a 25 cm wooden pointer in their right hand, and always used the same tip (TIP 1) to touch the center of the targets. We measured the position and orientation of the center of the pointer with a Polhemus Liberty sensor (240 Hz sampling rate), which allowed us to calculate the positions of both TIP 1 and TIP 2. The experimental setup is shown in Fig. 6.2(a). The goal of the experiment was to quantify the positional variance of the movement paths.

The data were analyzed as follows. The start and end times of each movement were found using a velocity threshold, applied when TIP 1 was within 10 cm of a target. Outlier movements, that failed to reach within 10 cm of their target, were eliminated.

Each movement was resampled in space, at equal intervals with length 2% of the total path length for that movement (this resampling is needed to suppress temporal misalignment). The mean and covariance for each sample point were computed, separately for each subject and movement target. Positional variance was defined as the trace of the covariance matrix. The means and covariances were then averaged over subjects, and plotted in Fig. 6.2(b) (± 1 standard deviation ellipsoids are shown at 25% intervals).

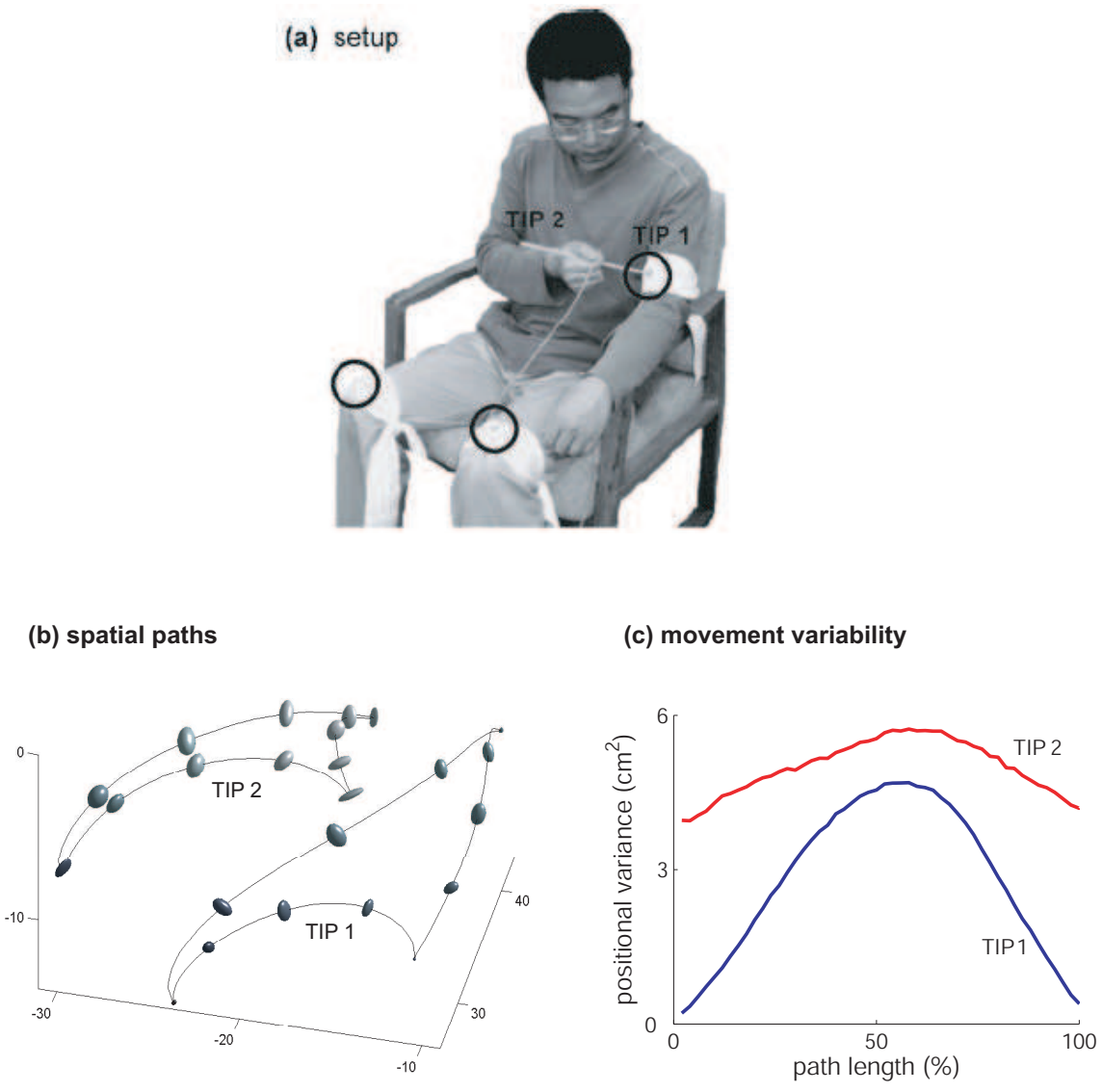


Figure 6.2. Experimental illustration of increased variability in redundant dimensions.

To facilitate comparison, we further averaged over the three targets and plotted the positional variances of the two tips of the pointer in Fig. 6.2(c).

The main result shown in Fig. 6.2(c) nicely illustrates the typical structure of motor variability. The position of the task-relevant TIP 1 is less variable than the task-irrelevant TIP 2. Furthermore, both variances are smaller in the task-relevant portions of the movement (when the pointer approaches the target) compared to the middle of the movement. The latter effect is much stronger for the task-relevant TIP 1.

These observations make it clear that the nervous system is not using the classic robotics approach — which is to plan a trajectory in joint space, and then execute that trajectory using some combination of computed torque and servo control. Since a joint-space trajectory plan contains no information regarding task relevance, the execution system would have no choice but to track faithfully all details of the plan — including the irrelevant ones. This would result in increased energy consumption, and increased errors due to control-dependent noise.

6.3 General Hierarchical Control Approach

Consider a continuous nonlinear dynamical system with state $x(t) \in \mathbb{R}^{n_x}$ and control signal $u_x(t) \in \mathbb{R}^{n_{u_x}}$, whose dynamics is

$$\dot{x} = \mathrm{a}\big(x(t)\big) + B\big(x(t)\big)u_x(t), \qquad x(0) = x_0. \tag{6.1}$$

Let $\ell(x(t), u(t), t) = q(x(t)) + r(x(t), u_x(t))$ be the instantaneous cost rate, $h(x(T))$ the final cost and T a specified final time, then the performance index is defined as

$$J = h(x(T)) + \int_0^T \ell(x(t), u_x(t), t)dt, \tag{6.2}$$

Our goal is to design controllers for the dynamical system (6.1) that minimize the above performance criterion. Here we focus on hierarchical control approach to solve for this general nonlinear optimal control problem. Fig. 6.1 shows that the decision making of this problem could be divided into two levels which are coordinated each other.

6.3.1 High-level dynamics model

Let $x \in \mathbb{R}^{n_x}$ be a state vector of a redundant biomechanical plant (representing joint angles, joint velocities, muscle activation states, and relevant aspects of the environment), and $q(x)$ be a cost function that quantifies the notion of a "task". Redundancy means that the function $q(x)$ depends on the state x only through a reduced set of (more abstract) variables $p(x)$, i.e. $q(x)$ can be written as $q(p(x))$. In the task of reaching, for example, performance depends only on the position of the hand relative to the target.

We propose the following principles for designing the task features and the desired high-level dynamics. First, it should be sufficiently simple, so that optimal control methods become feasible; second, it should be sufficiently close to the true dynamics, so that the transformer does not have to guess blindly (recall that the transformer does not know what the task is); third, it should contain $p(x)$ as state variables, so that the true cost $q(p(x))$ can be measured and optimized by the leading level.

We emphasize that setting the high-level state vector $y = p(x)$ will most likely violate the principle. Instead, we include in the virtual dynamical state y both $p(x)$ and its derivatives, so as to match the order of the differential equations governing the true dynamics. In the task of reaching, the virtual state will include hand position, velocity, and net muscle force (expressed in hand space); the virtual control signal will specify desired change in force.

The high-level state vector $y \in \mathbb{R}^{n_y},\ n_y \leq n_x$, is a static function of the plant state:

$$y = \varphi(x) \tag{6.3}$$

The function φ defines the mapping between the plant state x and the high-level state y, and it is supplied by the user. For example, if x is joint angle and angular velocity, y is end-effector position and velocity, the mapping function could be easily defined using the transformation between joint space and end-effector coordinates.

Now the question here is how to define high-level dynamics. Of course we can choose whatever high-level dynamics we desire. But what should we desire? One possibility is to choose a linear y dynamics. However we believe that the y dynamics should mimic

the x dynamics to the extent possible, so that the high-level controller can exploit the natural plant dynamics while operating on a lower-dimensional system. Differentiating (6.3) with respect to t and using (6.1), the actual dynamics of y becomes

$$\dot{y} = \Phi(x) \Big(\mathrm{a}(x) + B(x) u_x \Big) \tag{6.4}$$

where $\Phi(x) \triangleq \frac{\partial \varphi}{\partial x}$ is the Jacobian of the function φ.

Our design method seeks to create a desired high-level dynamics

$$\dot{y} = g(y) + G(y)\, u_y, \tag{6.5}$$

where u_y is the high-level control signal, g, G are assumed to be some known functions for now. Therefore, matching the passive dynamics in (6.4) and (6.5) yields

$$g\big(\varphi(x)\big) = \Phi(x)\mathrm{a}(x), \tag{6.6}$$

Matching the control-dependent terms in (6.4) and (6.5) yields

$$G\big(\varphi(x)\big)\, \tilde{u}_y(x, u_x) = \Phi(x) B(x) u_x, \tag{6.7}$$

where $\tilde{u}_y(x, u_x)$ corresponds to the desired high-level control signals, and should be chosen using physical intuition. For example, if u_x is joint torque and y is end-effector position and velocity, $\tilde{u}_y(x, u_x)$ could be either end-effector acceleration or force.

In addition to this standard technique for designing the high-level dynamics, the hierarchical framework here allows another novel method. In this method, the passive dynamics is no longer modeled explicitly on the high level, instead, it is obtained through the online estimation. Initially, we could use a linear model to describe the high-level dynamics followed by the learning of high-level dynamics.

6.3.2 High-level controller design

What we provide here is a way to generate control law u_y which affects x in such a way that the corresponding changes in y are as desired. Of course, optimization of the high-level controller will be facilitated if the cost function can be captured on the

high level. At the beginning, we have already assumed that y should contain enough information to compute the state-dependent cost $q(x)$. Therefore, given $y = \varphi(x)$, we could easily compute the high-level state-dependent cost term

$$q(\varphi(x)) \approx q(x). \tag{6.8}$$

However, the control cost $r(x, u_x)$ cannot be represented exactly on the high level, because it typically contains independent contributions from all components of u_x. This is the reason why the proposed hierarchical scheme is only an approximation to optimal control. But still we would like an approximation to the control energy term $r(x, u_x)$. This requires a function $\tilde{r}$ such that

$$\tilde{r}(\varphi(x), u_y) \approx r(x, u_x). \tag{6.9}$$

The objective of the high-level is to find the control u_y that minimizes the following performance criterion

$$J = h\left(\varphi\big(x(T)\big)\right) + \int_0^T \Big[q\big(\varphi(x(t))\big) + \tilde{r}(\varphi(x), u_y)\Big] dt. \tag{6.10}$$

The optimal feedback controller on the high level will be designed using an iterative Linear-Quadratic-Gaussian (iLQG) method that we have developed in the precious chapter. This method extend the efficient and well-developed LQG framework to the domain of nonlinear control. Briefly, the iteration starts with some control law that is applied to the nonlinear system — obtaining an average trajectory and control sequence. We then linearize the dynamics and quadratize the cost in the vicinity of that state-control trajectory, apply dynamic programming locally, and use the result to improve the initial control law. The algorithm has quadratic convergence, similar to a Newton's method.

6.3.3 Low-level controller design

In section 6.3.1, we define that the high-level state vector is a static function of the plant state (6.3), and obtain the actual and the desired high-level dynamics. The key of low-level controller design is the following: choose $u_x(x, u_y)$ so that the desired y dynamics (6.5) and the actual y dynamics (6.4) are identical; when multiple solutions

exist, use the control cost $r(x, u_x)$ to resolve redundancy. Setting "*desired* y = *actual* y" yields

$$\Phi(x)B(x)u_x = g(y) + G(y)u_y - \Phi(x)\mathrm{a}(x)$$

Hence the low-level control law u_x is obtained by solving the following constrained optimization problem:

$$\begin{aligned} &\text{Given } u_y \text{ and } x, \text{ find } u_x \text{ that minimizes} \\ &\frac{1}{2}\int_0^T r(x, u_x)\, dt \quad \text{subject to} \\ &\Phi(x)B(x)u_x = g(y) + G(y)u_y - \Phi(x)\mathrm{a}(x). \end{aligned} \tag{6.11}$$

In addition to the above equality constraint, we can incorporate inequality constraints on u_x. The latter are particularly important in biological movement control where muscle activations are always nonnegative.

Generally the control energy term has the quadratic form as the following

$$r(x, u_x) = u_x^T R u_x, \tag{6.12}$$

where R is symmetric positive definite. When $r(x, u_x)$ is in the above form, and control inequality constraint is absent, by assuming that $\Phi(x)B(x)$ has full row rank, the low-level controller can be uniquely computed as

$$u_x = \Big(\Phi(x)B(x)\Big)_R^{\dagger} \Big(g(y) + G(y)u_y - \Phi(x)\mathrm{a}(x)\Big). \tag{6.13}$$

One can transform the above constrained optimization problem (6.11) into an unconstrained one, and use a mixed cost that absorbs all constraints. Then the low-level control law u_x is obtained by solving the following unconstrained optimization problem:

$$\begin{aligned} &\text{Given } u_y \text{ and } x, \text{ find } u_x \text{ that minimizes} \\ &\int_0^T \left(r(x, u_x) + \lambda_1 \Big\| g(y) + G(y)u_y - \Phi(x)\Big(\mathrm{a}(x) + B(x)u_x\Big) \Big\|^2 + \lambda_2\, c(t, u_x) \right) dt. \end{aligned} \tag{6.14}$$

The constants λ_1, λ_2 set the relative importance of satisfying constraint versus minimizing energy.

6.3.4 Dynamic compatibility between levels of control

A potential problem of the above method is lack of "dynamic compatibility" between the two levels. This occurs when $\Phi(x)B(x) = 0$. For example, suppose y is end-effector position, u_y is end-effector velocity, and u_x is joint torque. This is problematic because torque cannot affect velocity instantaneously. However torque has a predictable effect when applied over time, in the following, we will present a method for computing temporal prediction $\tilde{\mathrm{a}}, \tilde{B}$ in place of "instantaneous" a and B.

Consider the discrete-time representation of system (6.1) with time step Δ:

$$x(t+\Delta) = x(t) + \Delta\Big(\mathrm{a}\big(x(t)\big) + B\big(x(t)\big)u_x(t)\Big) \tag{6.15}$$

We now analyze this system within a single step, from time t to time $t+\Delta$. Define $x_\tau = x(t+\tau),\ \tau \in [0, \Delta]$, so that $x_0 = x(t)$. Assuming that the control term $\mathrm{b} = B(x)\ u_x(t)$ is constant in the single time step, the changes of $\mathrm{a}(x)$ can be obtained by

$$\mathrm{a}(x_\tau) \approx \mathrm{a}(x_0) + A(x_\tau - x_0), \qquad A = \left.\frac{\partial \mathrm{a}(x)}{\partial x}\right|_{x=x_0} \tag{6.16}$$

Then, we have continuous dynamics

$$\dot{x}_\tau = Ax_\tau + \Big(\mathrm{a}(x_0) + \mathrm{b} - Ax_0\Big) \tag{6.17}$$

The solution of the above ordinary differential equation is

$$x_\Delta = x_0 + \Delta\ T(\Delta A, \infty)\Big(\mathrm{a}(x_0) + \mathrm{b}\Big) \tag{6.18}$$

where

$$T(X) = \lim_{n\to\infty} \frac{1}{n}\left(I + \Big(I + \frac{X}{n}\Big) + \cdots + \Big(I + \frac{X}{n}\Big)^{n-1}\right) = \Big(exp(X) - I\Big)X^{-1}$$

Switching back to continuous notation, we have the following modified integration scheme

$$x(t+\Delta) = x(t) + \Delta T(\Delta A, \infty)\Big(\mathrm{a}\big(x(t)\big) + B\big(x(t)\big)u_x(t)\Big) \tag{6.19}$$

which differs from Euler integration (6.15) only by T. This motivates the construction of the "predictive" dynamics $\tilde{\mathrm{a}}, \tilde{B}$ from the following condition: Euler integration (6.15)

with $\tilde{\mathrm{a}}, \tilde{B}$ should be equivalent to the modified scheme (6.19) with the original a, B. This condition is satisfied by

$$\tilde{\mathrm{a}} = T(\Delta A, \infty)\ \mathrm{a}(x) \tag{6.20}$$

$$\tilde{B} = T(\Delta A, \infty)\ B(x) \tag{6.21}$$

6.4 Application to 2-Link 6-Muscle Arm Movements

6.4.1 The two levels

The approach outlined above will now be applied to the problem of reaching with a realistic arm — 2 links, 6 muscles arm model. The high-level dynamics is defined in end-effector space (hand Cartesian space), and modelled using end-effector equations of motion. Initially, we use a linear model to describe the high-level dynamics by choosing $y = [p\ v\ f]^T$, where the high-level state variable $y \in \mathbb{R}^6$ and control signal $u_y \in \mathbb{R}^2$, and the construction of this end-effector linear model is achieved by expressing the relationships of hand position p, velocities v, as well as the net muscle force f acting on the hand:

$$\dot{p} = v, \tag{6.22}$$

$$\dot{v} = f/m, \tag{6.23}$$

$$\dot{f} = -\alpha(f - mg) + u_y, \tag{6.24}$$

where m is the average hand mass, α is the muscle decay constant. Since the model (6.22)-(6.24) is only an approximation, we need to use some learning processes in the hierarchical control architecture, which can estimate those unknown quantities iteratively online before a successful implementation can be achieved.

The control signal u_y is the output of the high-level loop which specifies a desired change of force in the end-effector space. The objective is to find the control u_y that minimizes the performance criterion

$$V = \frac{1}{2}\int_0^T \left[r_1\|u_y(t)\|^2 + s(t)\ \|p(t) - p^*\|^2\right] dt, \tag{6.25}$$

where p^* is the target, $r_1 = 1e-6$, and $s(t)$ is 0 when the hand is allowed to move, and 1 when the hand is required to be at the target. The optimal feedback controller is designed using the iLQG method developed in chapter 5.

The forward dynamics of the arm can be expressed as

$$\ddot{\theta} = \mathcal{M}(\theta)^{-1}(\tau_{mus} - \mathcal{C}(\theta, \dot{\theta}) - \mathcal{G}(\theta)), \tag{6.26}$$

where $\theta \in \mathbb{R}^2$ is the joint angle vector (shoulder: θ_1, elbow: θ_2), $\mathcal{M}(\theta) \in \mathbb{R}^{2\times 2}$ is a positive definite symmetric inertia matrix, $\mathcal{C}(\theta, \dot{\theta}) \in \mathbb{R}^2$ is a vector centripetal and Coriolis forces, $\mathcal{G}(\theta)$ is the gravity, and $\tau_{mus} \in \mathbb{R}^2$ is the joint torque.

The tension $T(a, l, v)$ produced by a muscle obviously depends on the muscle activation a, but also varies substantially with the length l and velocity v of that muscle

$$T(a, l, v) = aF_{vl}(l(\theta), v(\theta, \dot{\theta})) + F_p(l(\theta)). \tag{6.27}$$

The joint torque generated by the muscles is given by

$$\tau_{mus} = M(\theta)\, T(a, l(\theta), v(\theta, \dot{\theta})), \tag{6.28}$$

Muscle activation a is modelled as the following

$$\dot{a} = \alpha(u_x - a), \tag{6.29}$$

where α is the muscle decay constant. Combining (6.26)-(6.29), the low-level dynamic system can be obtained with state $x = (\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2\ a_1\ \cdots\ a_6)^T \in \mathbb{R}^{10}$, and control $u_x \in \mathbb{R}^6$.

Based on the characteristics of the transformation from the joint space to the end-effector space [54], the mapping $y = \varphi(x)$ —between the state of low-level dynamic model x and the high-level state y (containing hand position p, velocity v, as well as the net muscle force f in hand space)—can be written specifically as

$$p = \mathcal{K}(\theta), \tag{6.30}$$

$$v = J(\theta)\dot{\theta}, \tag{6.31}$$

$$J^T(\theta)f = \tau_{mus}, \tag{6.32}$$

where $\mathcal{K}$ is the transformation of position θ from joint space to the end-effector space (hand space) as described in (3.17); the Jacobian matrix $J(\theta) = \partial\mathcal{K}/\partial\theta$ transforms velocities in joint space, $\dot{\theta}$, to Cartesian velocity v of the end-effector expressed in hand space, as given in (3.19).

Here we use additional assumption and wrote this problem in an analytical formulation. Equation (6.32) represents the fundamental relationship between the net muscle force f and the joint torque τ_{mus} consistent with the end-effector and arm dynamic equations. With (6.27), (6.28), (6.32) and the assumption $\dot{a} \gg \theta, \dot{\theta}$, we can derive $\dot{f} = J^{\dagger}(\theta)M(\theta)F_{vl}(\theta, \dot{\theta})\ \dot{a}$, where $J^{+}(\theta)$ denotes the pseudo inverse. Based on (6.29) and the assumption $\dot{f} = -\alpha(f - mg) + u_y$, we obtain

$$u_y = \alpha Q u_x, \qquad Q = J^{\dagger}(\theta)M(\theta)F_{vl}(\theta, \dot{\theta}). \tag{6.33}$$

And the low-level controller for 2 Link 6 Muscle arm movement can be obtained by solving a constrained quadratic optimization problem

$$\min_{u_x} \int_0^T \frac{1}{2} u_x^T H u_x + b^T u_x dt \tag{6.34}$$

subject to

$$0 \leq u_x \leq 1, \tag{6.35}$$

where $H = \alpha^2 Q^T Q + R,\ b = -\alpha Q^T u_y$. The diagonal matrix R has entries proportional to PCSAi (given in section 3.1.2)—because larger muscles generate more force per unit activation, and therefore consume more metabolic energy. Eq. (6.35) explains that the allowed range of muscle control signals is between 0 and 1. Therefore we can find the low-level control u_x which activates the muscles in a way that achieves the specified change of force acting on the hand.

The low-level has to activate muscles so as to accomplish desired effects in the task-relevant space. Since there are more muscles than task variables, the low-level faces a redundancy problem which will be solved using static optimization. At each point in time, we will obtain a linear relationship between muscle activations and task variables, and then find the minimal feasible muscle activations via Quadratic Programming.

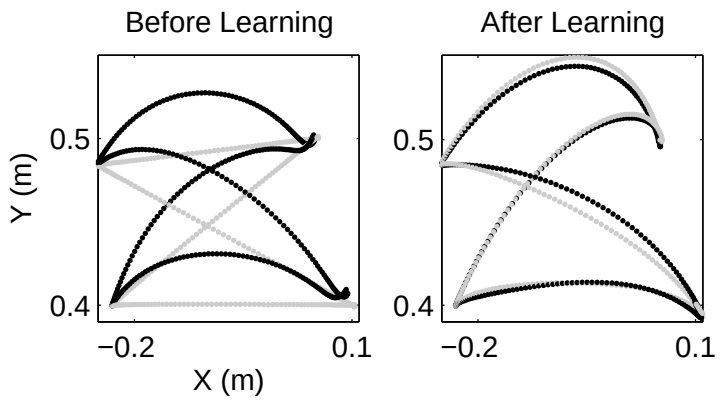


Figure 6.3. Trajectories in Cartesian hand space. Gray lines – trajectories obtained by applying the high-level feedback controller to the virtual dynamics. Black lines – trajectories obtained by applying the hierarchical control scheme to the real plant.

6.4.2 Application to Arm Movements

The task we study here is reaching: the arm starts from rest position and has to reach a specified target with minimal control energy. We chose different pairs of starting postures and targets, and applied the hierarchical control scheme described above. POSTURE 1 : $\theta_1 = 1.5rad, \theta_2 = 1.05rad$; POSTURE 2 : $\theta_1 = 1.3rad, \theta_2 = 1.5rad$; TARGET 1 : $p^* = [0.05m;\ 0.5m]$; TARGET 2 : $p^* = [0.1m;\ 0.4m]$. Hand trajectories are shown in Fig. 6.3. The black curves are the actual trajectories of the hand, that result from the coupling of the two-level hierarchy with the detailed arm model. The gray curves are the trajectories that would have resulted from applying the feedback control law to the virtual dynamical system. Note that before learning the "virtual trajectories" are straight, because we do not have nonlinearities in the initial virtual model. However, after the system identification stage the virtual model is improved, and it now contains nonlinear terms. As a result, both the virtual and real trajectories become curved, and more importantly, they get closer to each other.

Figure 6.4 illustrates the high-level optimal control commands u_y which were sent to the low-level loop, and the low-level optimal control sequences u_x that were used to activate the muscles. It also shows optimal hand trajectory (high level trajectory) in Cartesian space, and shoulder and elbow angle movement trajectory (low level trajectory)

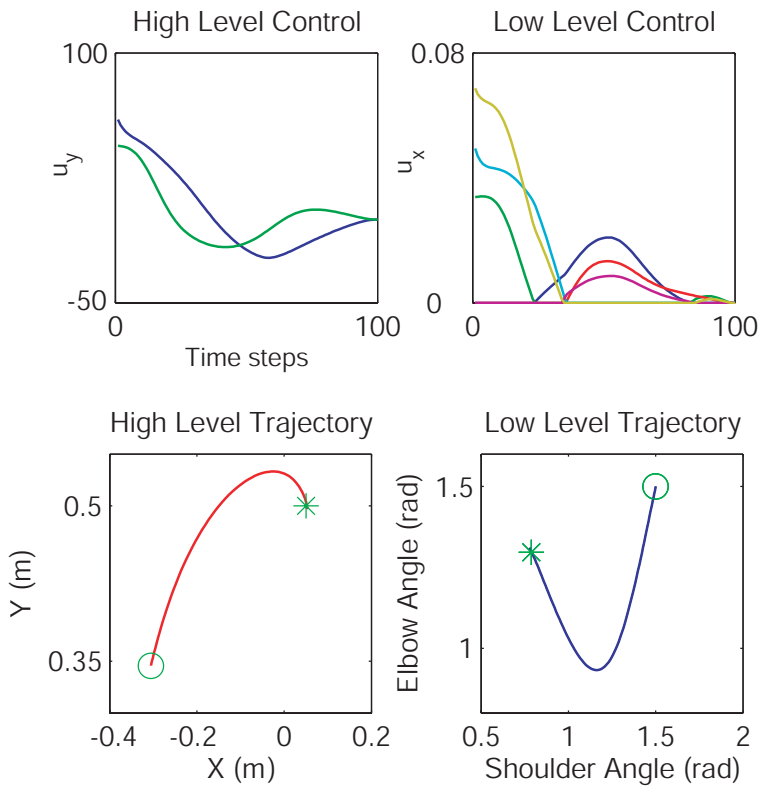


Figure 6.4. Optimal control sequences and trajectories of hierarchical control system (results obtained after learning). Circle: start position. Star: target position.

in joint space. We found that hand arrives to the desired position $X = 0.05m, Y = 0.5m$ respectively. Additionally, the cost achieved by the hierarchical control system after learning was 21% lower compared to its value before learning.

Furthermore, we compared the behavior of the hierarchical control scheme to that of an optimal (non-hierarchical) controller designed for the real plant. Note that the plant we are studying, although quite complex, is still amenable to iterative methods for nonlinear optimal control. We used the control sequences generated by our control scheme as an initial control law, and then improved that control law via gradient descent on the performance criterion. Both before and after learning, we found that the controls generated by our method are close to a local minimum of the unconstrained problem (see Fig. 6.5). After learning, the distance to the nearest local minimum was about 50% less than it was before learning.

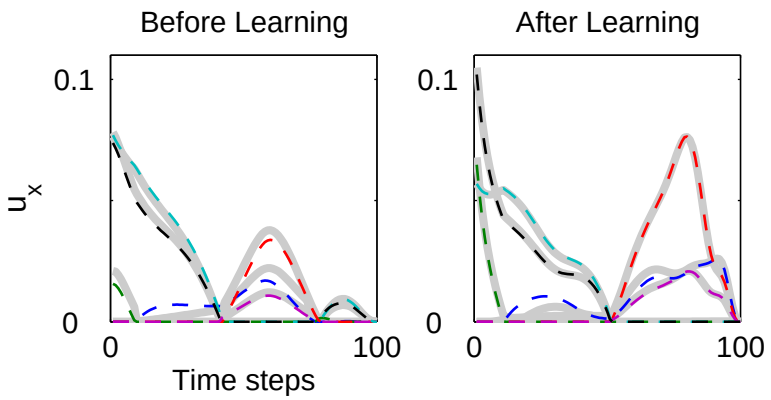


Figure 6.5. Comparison of the muscle control sequences generated by our hierarchical controller (dashed lines) vs. the non-hierarchical optimal controller (thick gray lines).

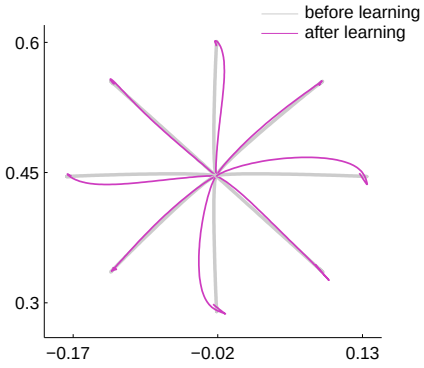


Figure 6.6. Effects of adapting the high-level dynamics to the plant dynamics.

Finally, we applied this hierarchical control scheme to the center-out reaching task which is commonly studied in the Motor Control — the targets are arranged in a circle with 15cm radius around the starting position. Note that the high level dynamics y here is a 6-dimensional vector containing the hand position, velocity and net muscle force. Results are shown in (Fig. 6.6). For the trivial dynamics in the high level (before learning) the end-effector paths were straight as expected, because we did not include nonlinearities in the initial high level virtual model, and we assumed that it was linear. For the adapted dynamics (after learning) the paths became systematically curved. As a result of this strategy, the total cost was reduced by 23%.

Figure 6.7 illustrates the reaching trajectories in hand space both before and after

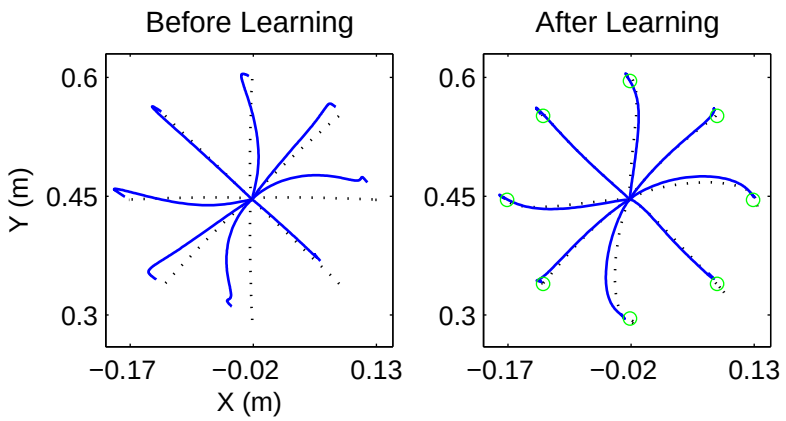


Figure 6.7. Reaching trajectories in hand space obtained before and after learning with y containing hand position, velocity and net muscle force. Dotted lines — trajectories obtained by applying the high-level feedback controller to the virtual dynamics. Solid lines — trajectories obtained by applying the hierarchical control scheme to the real plant. Circles — target positions.

learning. The solid lines are the actual trajectories of the hand, obtained by applying the two-level hierarchy to the real detailed arm model. The dotted lines are the trajectories that would have resulted from applying the feedback control law to the virtual dynamical system. We notice that, in Fig. 6.7, the "expected trajectories" before learning are straight. However, after the system identification stage, the virtual model is improved, and it now contains nonlinear terms. As a result, both the virtual and real trajectories become curved, and more importantly, they get closer to each other. Therefore, the high-level dynamics model that was learned is a good approximation to the dynamics of the real plant. Furthermore, the comparison of hand trajectories (solid blue curves) shows that, after learning the hand can reach the target positions (green circles shown in Fig. 6.7) as accurately as possible.

6.4.3 Different Representation of High-level Feature Space and Comparison to Optimal Control

The key to our hierarchical framework for redundant biomechanical systems is to design the high-level and low-level controllers, as well as the construction of high-level

dynamics that approximately mimic the plant dynamics. How to choose appropriate high-level parameters is a hard question, so we seek that the high-level state contains task-related parameters that enter in the computation of the performance index, and perhaps other closely related parameters. The following simulation will reveal a benefit of including velocity or net muscle force as high-level parameters for the application of reaching movements, even though the state-dependent cost is only a function of position.

Here we designed three hierarchical controllers for different representations of high-level dynamics. In the first case, y is a 6-dimensional vector containing the hand position, velocity and muscle force $y = [p;\ \dot{p};\ f]$. In the second case, y is a 4-dimensional vector containing the hand position and velocity $y = [p;\ \dot{p}]$. In the third case, y is a 2-dimensional vector containing only the hand position $y = p$. The control inequality constraints (the range of muscle control signals is between 0 and 1) made the explicit solutions inapplicable, and therefore, the low-level controller used Quadratic Programming at each time step to compute $u_x(x, u_y)$. In addition to the three hierarchical controllers, we also computed the optimal feedback controller in each case, using our iterative linear-quadratic-Gaussian (iLQG) method derived in chapter 5. Furthermore, the variability of biological movements indicates the presence of large disturbances in the motor system. In particular, it is known that the motor noise is control-dependent [38, 116, 117, 118, 120], with standard deviation proportional to the mean of the control signal. Hence it is also very important for us to take these disturbances into account.

Figure 6.8 shows deterministic and stochastic hand paths for each controller. Stochastic trajectories were simulated by corrupting each control signal with 50% control-multiplicative noise. The differences on the level of kinematics are small among optimal controller and another two hierarchical controllers ($y = [p;\ \dot{p}]$ and $y = [p; \dot{p}; f]$), and the reaching trajectories shown in Fig. 6.8 are similar to the experimental data of human movement. While for the $y = p$ controller, the high-level dynamics only includes the information of hand position, which can't approximate the real plant dynamic very well. Therefore comparing with the other two hierarchical controllers, the performance of the $y = p$ controller is poor, and the hand is hard to reach the targets accurately.

Figure 6.9 shows the muscle activations. Each subplot is one muscle-controller com-

bination. The horizontal axis corresponds to movement direction, while the vertical axis is time during the movement (increasing downward). Dark means higher activation. We now see a much more clear distinction between the three hierarchical controllers. The muscle activations found by the $y = [p;\ \dot{p};\ f]$ controller are quite similar to the optimal muscle activations, and furthermore resemble many features of experimentally observed muscle activations. On the other hand, the $y = p$ controller misses the elaborate temporal pattern of muscle activation, although it still activates the appropriate muscles. The relatively poor performance of the $y = p$ controller suggests that controlling hand position through instantaneous velocity commands is not a good idea—because such commands are too far removed from the muscles that have to carry them out. In contrast, high-level commands related to hand force rather than velocity afford hierarchical control that is much closer to optimal. Overall we conclude that our hierarchical control scheme is reasonably close to the behavior of the optimal controller when the high-level state contains hand position and force.

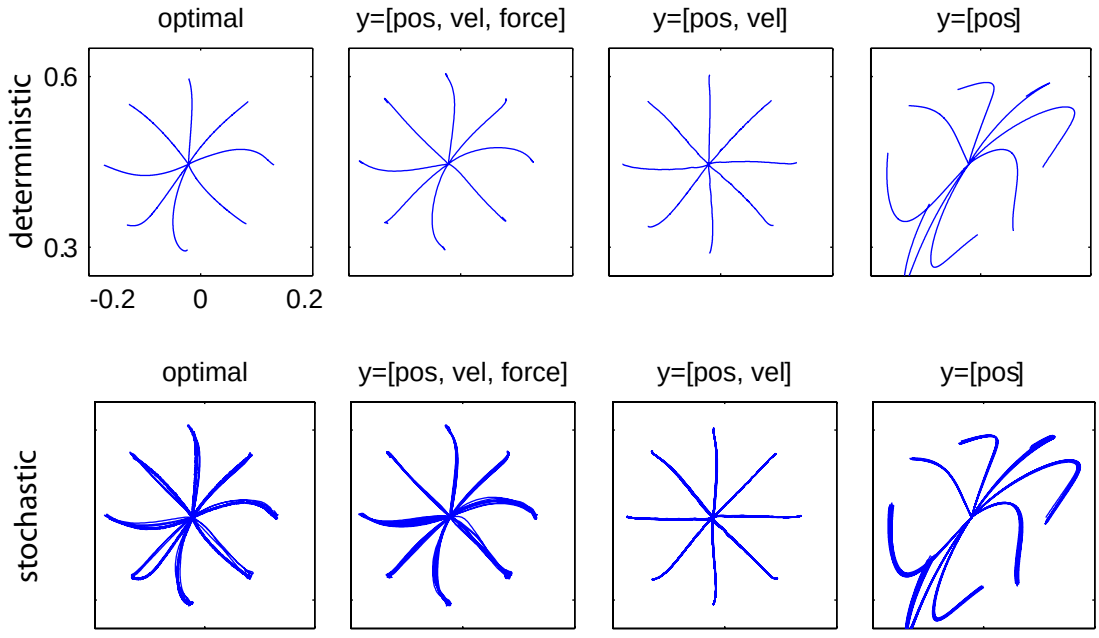


Figure 6.8. Reaching trajectories in hand space for optimal controller and three hierarchical controllers with y containing: hand position, velocity and net muscle force; hand position and velocity; and only hand position.

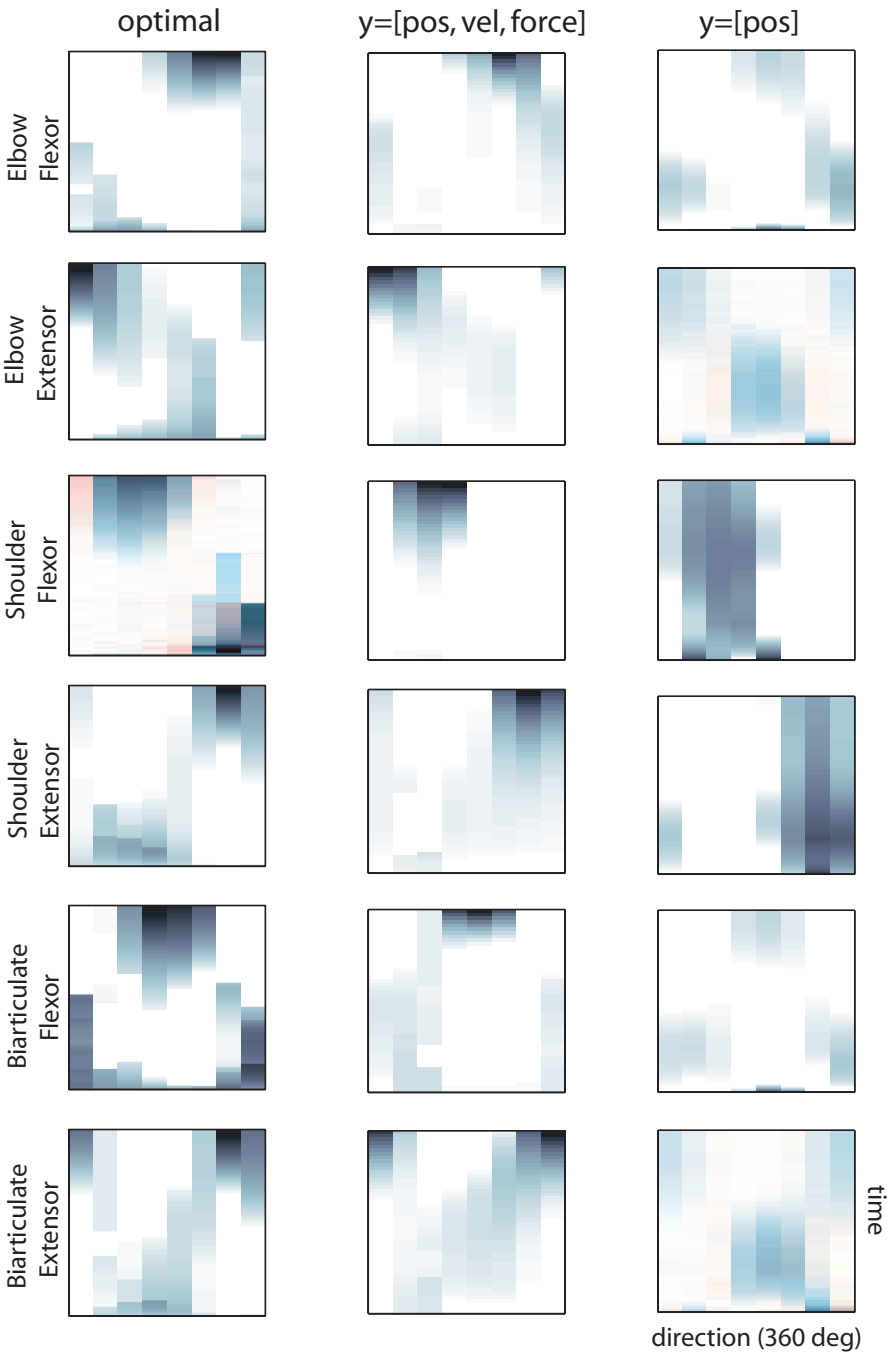


Figure 6.9. Muscle activation patterns for optimal controller and two hierarchical controllers with y containing hand position, velocity and net muscle force; and y containing hand position.

6.5 Application to 5-Link 24-Muscle Arm Movements

Now the hierarchical control approach outlined above will be applied to a much more complicated and realistic arm. This arm model contains 5 links and 24 muscles with varying moment arms, muscle length-velocity-tension curves based on the Virtual Muscle model, and activation dynamics modelled as a nonlinear low-pass filter. The detailed model is given in section 3.2. In this case the state of the high-level virtual dynamics is 6-dimensional, which contains hand position and velocity expressed in Cartesian hand space. We use an initial linear model of the y dynamics, where the velocity is the derivative of position, and the control signal u_y is the derivative of velocity. The high-level controller is designed to implement the goal of the movement: flexing the shoulder to 30 degree and the elbow to 90 degree from the neutral position (where the upper arm is adjacent to the torso, the elbow is fully extended, and the palm of the hand is facing anteriorly). Fig. 6.10 illustrates how the shoulder flexion angle and elbow flexion angle converge by applying the hierarchical control scheme to this complicated 5DOF arm. While the low-level controller is designed to activate the appropriate muscles, and Fig. 6.11 shows the reasonable temporal muscle activation patterns for deltoid, supraspinatus, infraspinatus and subscapularis muscle.

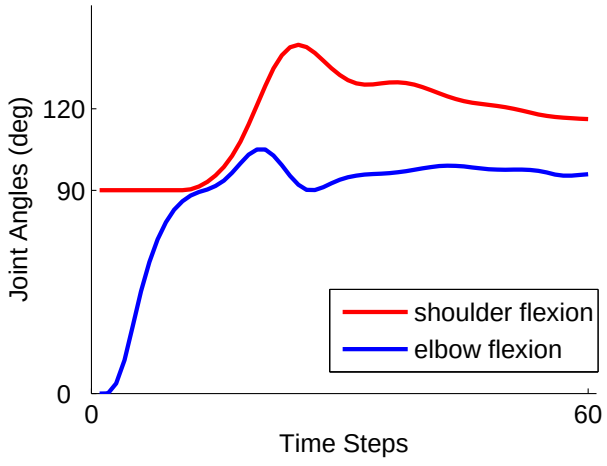


Figure 6.10. Application of hierarchical control to 5-Link 24-Muscle arm movement: shoulder flex to 30 degree, and elbow flexed to 90 degree.

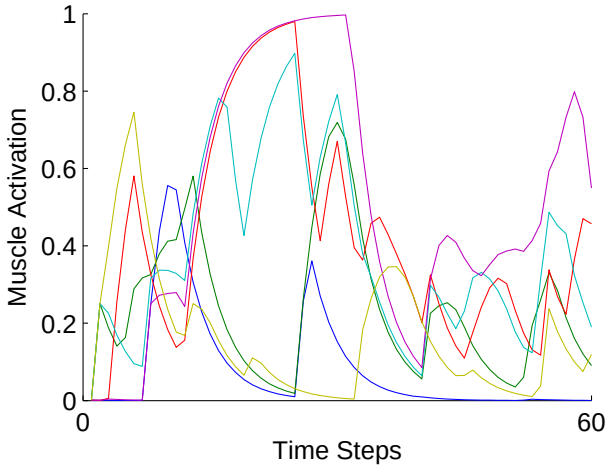


Figure 6.11. Muscle activation patterns of deltoid, supraspinatus, infraspinatus and subscapularis muscle for 5-Link 24-Muscle arm movement.

6.6 Summary

Here we present a general approach to approximately-optimal hierarchical feedback control of redundant systems, and illustrate its application in the context of reaching with a realistic model of the human arm. The design of the proposed control hierarchy involves: (i) specifying a set of task-relevant parameters for constructing the desired high-level dynamics that mimic the plant dynamics but has reduced dimensionality; (ii) designing a low-level feedback controller which yields an augmented plant with the specified input-output behavior; (iii) designing a high-level feedback controller that solves the original control problem but operates on a simplified system. The resulting control hierarchy makes it possible to solve the optimal control problem without running into the curse of dimensionality. Imposing any predefined control structure is likely to result in suboptimal performance. However, our simulations demonstrate that the suboptimality due to the hierarchical structure is negligible (especially after learning). At the same time, the computational demands are much lower than what is required for designing an optimal feedback controller.

While this may be the first comprehensive approach to hierarchical sensorimotor control, the main computational ideas underlying it are not entirely new. Feedback

transformations that create easier-to-control virtual plants have been used in robotics [14]; the Operational Space Framework [54] and Virtual Model Control [92] are particularly relevant. The new computational aspects of our work are: (i) application of optimal feedback control to the virtual plant; (ii) principles for designing the virtual model that make it suitable for optimal control; (ii) continuous improvement of the internal model of virtual dynamics, compensating for unavoidable transformation errors.

Chapter 7

Estimation and Control Design via Linear Matrix Inequalities

7.1 Motivation

The filtering and control problem for the systems with multiplicative noise has recently received a great deal of attentions, and has found applications in many fields of sciences and engineering. Different from the traditional additive noise, multiplicative noise is more practical, since it allows the statistical description of the multiplicative noise be not known *a prior* but depend on the control and state solution. Such models are found in many physical systems, such as signal processing systems [35], biological movement systems [38, 116, 117, 118, 120] and aerospace engineering systems.

One important benefit of the multiplicative noise in a linear control problem is that the controllers for multiplicative systems appear robust. This is in contrast to LQG theory, where the minimum variance occurs at infinite control gain, which renders the solution of problem unstable. Therefore, the multiplicative noise system has a significant effect on the robustness of the overall control system [36, 74, 89, 114].

So far, many researchers have been working on various kinds of analysis of filtering and control in systems with multiplicative noise, and there have been several approaches

for dealing with this problem, including the linear matrix inequality approach [35, 36], the Riccati difference equation method [93, 96, 117, 120, 133], and the game theoretic approach [94]. Since this model reflects more realistic properties in engineering, a complete theory which includes control and estimation should be developed.

Here we focus on the study of estimation problem for multiplicative noise system using LMIs. Reference [36, 74] devised an LMI approach to the robust control of stochastic systems with multiplicative noise. In [66] we study the state estimation with signal-dependent noise model for the continuous time systems. The contribution here is to propose an LMI method to cope with the estimation problem for the discrete time systems with multiplicative noise. We shall show that a mild additional constraint for scaling will make the problem convex. The basic problem solved is to find a state estimator that bounds the estimation error below a specified error covariance.

This chapter is organized as follows. In section 7.2 the filtering and control problem for discrete time system subject to multiplicative noise is formulated. The design of the state feedback controller is developed in section 7.3. In section 7.4, the sufficient conditions for the existence of the state estimator are given, and an algorithm for the filtering design is derived which guarantees the performance requirement. Section 7.5 presents two numerical examples, and some concluding remarks are drawn in section 7.6.

7.2 System Model and Problem Formulation

Consider the following discrete time system with state space representation

$$\begin{aligned} x_{k+1} &= (A_k + A_{s,k}\eta_k)x_k + (B_k + B_{s,k}\varepsilon_k)u_k + D_k\omega_k, \\ z_k &= (H_k + H_{s,k}\zeta_k)x_k + v_k, \\ y_k &= C_k x_k, \end{aligned} \tag{7.1}$$

where $x_k \in R^{n_x}$ is the state variable, $z_k \in R^{n_z}$ is the measurement output; $y_k \in R^{n_y}$ is the output of interest for performance evaluation; $\omega_k \in R^{n_w}$ and $v_k \in R^{n_y}$ are the process and measurement noises; η_k, ε_k and ζ_k are the multiplicative noises; $A_k, A_{s,k}, B_k, B_{s,k}, D_k, H_k, H_{s,k}$ and C_k are constant matrices which have proper dimen-

sions. The independent random variable ω_k, v_k, η_k, ε_k and ζ_k have zero-mean gaussian distributions with covariances W, V, and $\Omega^\eta = \Omega^\varepsilon = \Omega^\zeta = I$ respectively.

Here we consider the following problems. First, we look for the state feedback control law $u_k = Kx_k$ such that the closed loop system (7.1) is mean square stable. And we will determine if there exists a control gain K such that $\varepsilon_\infty\{y_k^T y_k\} < \mu$ is satisfied for the given μ.

Second, we consider (7.1) where B_k and $B_{s,k}$ are zero. The objective is to design a linear filter with the state space representation

$$\hat{x}_{k+1} = A_k\hat{x}_k + F(z_k - H_k\hat{x}_k), \tag{7.2}$$

$$\hat{y}_k = C_k\hat{x}_k, \tag{7.3}$$

where $\hat{x}_k$ is the unbiased estimate of the state x_k, F is the filter gain to be determined such that $(A_k - FH_k)$ is asymptotically stable, and the estimation error has covariance less than a specified matrix. The estimation error is $\tilde{x}_k = x_k - \hat{x}_k$, and the estimation error system is given by

$$\tilde{x}_{k+1} = (A_k - FH_k)\tilde{x}_k + \bar{\omega}_k, \tag{7.4}$$

$$\tilde{y}_k = C_k\tilde{x}_k, \tag{7.5}$$

where

$$\bar{\omega}_k = D_k\omega_k + A_{s,k}\eta_k x_k - FH_{s,k}\zeta_k x_k - Fv_k, \tag{7.6}$$

and $\tilde{y}_k$ denotes the estimation error of particular interests. We will explore the existence condition of the state estimator. And we will be able to provide the sufficient conditions for the existence of the state estimator based on Linear Matrix Inequalities (LMIs). The key idea of this filtering problem is to find the estimate $\hat{x}_k$ of x_k such that the performance criterion $\varepsilon_\infty\{\tilde{y}_k\tilde{y}_k^T\} < \Omega$ is satisfied for the given Ω.

7.3 State-feedback Controller Design

Consider the system (7.1) where z_k is excluded, the state-feedback control law is the form $u_k = Kx_k$, now the closed loop system is the following

$$\begin{aligned} x_{k+1} &= (A_k + B_k K)x_k + (A_{s,k}\eta_k + B_{s,k}K\varepsilon_k)x_k + D_k\omega_k, \\ y_k &= C_k x_k. \end{aligned} \tag{7.7}$$

Definition 1 (**Mean Square Stable**) The system (7.7) with multiplicative and additive noise inputs is mean square stable if the state covariance matrix associated with this system exists and is positive definite.

We first find the state feedback control gain K such that the closed loop system is mean square stable. Assuming that the upper bound of state covariance matrix associated with system (7.7) exists, which is described as $X \geq \varepsilon_\infty\{x_k x_k^T\}$, it should satisfy the following Lyapunov inequality

$$(A_k + B_k K)X(A_k + B_k K)^T - X + A_{s,k}XA_{s,k}^T + B_{s,k}KXK^T B_{s,k}^T < 0. \tag{7.8}$$

Theorem 2 *There exists a state feedback controller gain K such that the closed loop system (7.7) is mean square stable if, and only if, there exist a symmetric positive definite matrix X and a matrix L such that*

$$\begin{pmatrix} X & A_k X + B_k L & A_{s,k}X & B_{s,k}L \\ XA_k^T + L^T B_k^T & X & 0 & 0 \\ XA_{s,k}^T & 0 & X & 0 \\ L^T B_{s,k}^T & 0 & 0 & X \end{pmatrix} > 0. \tag{7.9}$$

And the corresponding feedback control gain is given by $K = LX^{-1}$.

Proof 2 *With the change of variable $L := KX$, (7.8) can be written as*

$$(A_k X + B_k L)X^{-1}(A_k + B_k L)^T - X + A_{s,k}XA_{s,k}^T + B_{s,k}LX^{-1}L^T B_{s,k}^T < 0.$$

Utilizing the Schur complement formula, the above inequality can be immediately written as (7.9).

Theorem 3 *For a given scalar $\mu > 0$, if there exist a symmetric positive definite matrix X and a matrix L such that (7.9) is satisfied and*

$$trace[C_k X C_k^T] < \mu, \tag{7.10}$$

then there exists a state feedback controller gain K such that the closed loop system (7.7) is mean square stable and $\varepsilon_\infty\{y_k^T y_k\} < \mu$. And the corresponding feedback control gain is given by $K = LX^{-1}$.

Proof 3 *The result follows from Theorem 1 and*

$$\varepsilon_\infty\{y_k^T y_k\} = trace[C\varepsilon_\infty\{x_k x_k^T\}C^T] < trace[C_k X C_k^T] < \mu,$$

which concludes the proof.

7.4 Filter Design

7.4.1 Existence Condition

To solve the filter design, we will consider the augmented adjoint system. Combining the filter (7.2) and the estimation error dynamics (7.4), it yields

$$\mathbf{x}_{k+1} = \mathcal{A}\mathbf{x}_k + \mathcal{N}\eta_k\mathbf{x}_k + \mathcal{H}\zeta_k\mathbf{x}_k + \mathcal{D}\mathbf{w}_k, \tag{7.11}$$

where

$$\mathbf{x_k} = \begin{pmatrix} \tilde{x}_k \\ \hat{x}_k \end{pmatrix}, \qquad \mathbf{w}_k = \begin{pmatrix} \omega_k \\ v_k \end{pmatrix}, \tag{7.12}$$

$$\mathcal{A} = \begin{pmatrix} A_k - FH_k & 0 \\ FH_k & A_k \end{pmatrix} = \mathcal{A}_0 + \mathcal{B}_0 F \mathcal{C}_0, \tag{7.13}$$

$$\mathcal{N} = \begin{pmatrix} A_{s,k} & A_{s,k} \\ 0 & 0 \end{pmatrix}, \tag{7.14}$$

$$\mathcal{H} = \begin{pmatrix} -FH_{s,k} & -FH_{s,k} \\ FH_{s,k} & FH_{s,k} \end{pmatrix} = \mathcal{B}_0 F G_0, \tag{7.15}$$

$$\mathcal{D} = \begin{pmatrix} D_k & -F \\ 0 & F \end{pmatrix} = \mathcal{D}_0 + \mathcal{B}_0 F E_0, \tag{7.16}$$

$$\mathcal{A}_0 = \begin{pmatrix} A_k & 0 \\ 0 & A_k \end{pmatrix}, \qquad \mathcal{B}_0 = \begin{pmatrix} -I \\ I \end{pmatrix}, \tag{7.17}$$

$$\mathcal{C}_0 = \begin{pmatrix} H_k & 0 \end{pmatrix}, \qquad \mathcal{D}_0 = \begin{pmatrix} D_k & 0 \\ 0 & 0 \end{pmatrix}, \tag{7.18}$$

$$G_0 = \begin{pmatrix} H_{s,k} & H_{s,k} \end{pmatrix}, \quad E_0 = \begin{pmatrix} 0 & I \end{pmatrix}. \tag{7.19}$$

We start by defining the upper bound of the state covariance matrix of system (7.11) as

$$\mathcal{X} \geq \varepsilon_\infty \{\mathbf{x}_k \mathbf{x}_k^T\}, \tag{7.20}$$

if it exists, it should satisfy the following Lyapunov inequality:

$$0 > \mathcal{A}\mathcal{X}\mathcal{A}^T - \mathcal{X} + \mathcal{N}\mathcal{X}\mathcal{N}^T + \mathcal{H}\mathcal{X}\mathcal{H}^T + \mathcal{D}\mathcal{W}\mathcal{D}^T, \tag{7.21}$$

where $\mathcal{W} = \begin{pmatrix} W & 0 \\ 0 & V \end{pmatrix}$ is symmetric and positive definite. Substitution of (7.13), (7.15)-(7.19) into the above inequality, yields

$$\begin{aligned} 0 \;>\; & (\mathcal{A}_0 + \mathcal{B}_0 F \mathcal{C}_0)\mathcal{X}(\mathcal{A}_0 + \mathcal{B}_0 F \mathcal{C}_0)^T - \mathcal{X} + (\mathcal{B}_0 F G_0)\mathcal{X}(\mathcal{B}_0 F G_0)^T \\ & + \mathcal{N}\mathcal{X}\mathcal{N}^T + (\mathcal{D}_0 + \mathcal{B}_0 F E_0)\mathcal{W}(\mathcal{D}_0 + \mathcal{B}_0 F E_0)^T \end{aligned} \tag{7.22}$$

Lemma 4 *The inequality (7.22) can be rewritten in a form*

$$\Gamma F \Lambda + (\Gamma F \Lambda)^T + \Theta < 0, \tag{7.23}$$

where

$$\Theta = \begin{pmatrix} -\mathcal{X} + \mathcal{N}\mathcal{X}\mathcal{N}^T & 0 & \mathcal{A}_0\mathcal{X} & \mathcal{D}_0\mathcal{W} \\ 0 & -\mathcal{X} & 0 & 0 \\ \mathcal{X}\mathcal{A}_0^T & 0 & -\mathcal{X} & 0 \\ \mathcal{W}\mathcal{D}_0^T & 0 & 0 & -\mathcal{W} \end{pmatrix}, \tag{7.24}$$

$$\Gamma = \begin{pmatrix} \mathcal{B}_0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{7.25}$$

$$\Lambda = \begin{pmatrix} 0 & G_0\mathcal{X} & \mathcal{C}_0\mathcal{X} & E_0\mathcal{W} \end{pmatrix}. \tag{7.26}$$

Proof 4 *We start by using the Schur complement, the inequality (7.22) can be written as*

$$\begin{pmatrix} -\mathcal{X} + \mathcal{N}\mathcal{X}\mathcal{N}^T & (\mathcal{B}_0 F G_0)\mathcal{X} & \mathcal{A}\mathcal{X} & \mathcal{D}\mathcal{W} \\ \mathcal{X}(\mathcal{B}_0 F G_0)^T & -\mathcal{X} & 0 & 0 \\ \mathcal{X}\mathcal{A}^T & 0 & -\mathcal{X} & 0 \\ \mathcal{W}\mathcal{D}^T & 0 & 0 & -\mathcal{W} \end{pmatrix} < 0,$$

where $\mathcal{A}$ is defined in (7.13), $\mathcal{D}$ is defined in (7.16). Utilizing the structure of the above matrix and substituting (7.24) into the above inequality, it obtains

$$\Theta + \begin{pmatrix} 0 & (\mathcal{B}_0 F G_0)\mathcal{X} & (\mathcal{B}_0 F \mathcal{C}_0)\mathcal{X} & (\mathcal{B}_0 F E_0)\mathcal{W} \\ \mathcal{X}(\mathcal{B}_0 F G_0)^T & 0 & 0 & 0 \\ \mathcal{X}(\mathcal{B}_0 F \mathcal{C}_0)^T & 0 & 0 & 0 \\ \mathcal{W}(\mathcal{B}_0 F E_0)^T & 0 & 0 & 0 \end{pmatrix} < 0.$$

With the use of Γ and Λ given in (7.25)-(7.26), the above condition can be equivalently written as (7.23).

It is important to notice that the filtering design problem has been converted into looking for the solution of F in the inequality (7.23). In order to find the existence conditions of the state estimator and the parametrization of all the solutions, the following lemma will be introduced from [108].

Lemma 5 (Finsler's Lemma) *Let $x \in \mathbf{R}^n$, $\Theta = \Theta^T \in \mathbf{R}^{n\times n}$, $\Gamma \in \mathbf{R}^{n\times m}$ and $\Lambda \in \mathbf{R}^{k\times n}$. Let $\Gamma^\perp$ be any matrix such that $\Gamma^\perp\Gamma = 0$. Let $\Lambda^{T^\perp}$ be any matrix such that*

${\Lambda^T}^\perp \Lambda^T = 0$. *The following statements are equivalent:*

$$\text{(i)} \quad x^T \Theta x < 0, \ \forall \Gamma^T x = 0, \Lambda x = 0 \ x \neq 0. \tag{7.27}$$

$$\text{(ii)} \quad \Gamma^\perp \Theta {\Gamma^\perp}^T < 0, \tag{7.28}$$

$${\Lambda^T}^\perp \Theta {\Lambda^T}^{\perp^T} < 0. \tag{7.29}$$

$$\text{(iii)} \quad \exists \mu_1, \mu_2 \in \mathbf{R} : \Theta - \mu_1 \Gamma \Gamma^T < 0, \tag{7.30}$$

$$\Theta - \mu_2 \Lambda^T \Lambda < 0. \tag{7.31}$$

$$\text{(iv)} \quad \exists F \in \mathbf{R}^{m \times k} : \Gamma F \Lambda + (\Gamma F \Lambda)^T + \Theta < 0. \tag{7.32}$$

Note that Finsler's Lemma can be applied to obtain LMI formulations in control and estimation theory. By applying the Finsler's lemma, we obtain the following theorem.

Theorem 4 *The condition (7.28) and (7.29) are equivalent to the following statement: there exist symmetric positive definite matrices* $\mathcal{X}, P \in \mathbf{R}^{2n_x \times 2n_x}$ *that satisfy*

$$\mathcal{X} P = I, \tag{7.33}$$

$$\mathcal{B}_0^\perp (\mathcal{A}_0 \mathcal{X} \mathcal{A}_0^T - \mathcal{X} + \mathcal{N} \mathcal{X} \mathcal{N}^T + \mathcal{D}_0 \mathcal{W} \mathcal{D}_0^T) {\mathcal{B}_0^\perp}^T < 0, \tag{7.34}$$

$$\begin{pmatrix} 0 \\ G_0^T \\ \mathcal{C}_0^T \\ E_0^T \end{pmatrix}^\perp \begin{pmatrix} -P + P\mathcal{N}\mathcal{X}\mathcal{N}^T P & 0 & P\mathcal{A}_0 & P\mathcal{D}_0 \\ 0 & -P & 0 & 0 \\ \mathcal{A}_0^T P & 0 & -P & 0 \\ \mathcal{D}_0^T P & 0 & 0 & -\mathcal{W}^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ G_0^T \\ \mathcal{C}_0^T \\ E_0^T \end{pmatrix}^{\perp^T} < 0. \tag{7.35}$$

Proof 5 *According to Lemma 1 and Finsler's Lemma, we substitute the following matrix*

$$\Gamma^\perp = \begin{pmatrix} \mathcal{B}_0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^\perp = \begin{pmatrix} \mathcal{B}_0^\perp & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and (7.24) into (7.28), it yields

$$\begin{pmatrix} \mathcal{B}_0^\perp (-\mathcal{X} + \mathcal{N}\mathcal{X}\mathcal{N}^T) {\mathcal{B}_0^\perp}^T & 0 & \mathcal{B}_0^\perp (\mathcal{A}_0 \mathcal{X}) & \mathcal{B}_0^\perp \mathcal{D}_0 \mathcal{W} \\ 0 & -\mathcal{X} & 0 & 0 \\ \mathcal{X}\mathcal{A}_0^T {\mathcal{B}_0^\perp}^T & 0 & -\mathcal{X} & 0 \\ \mathcal{W}\mathcal{D}_0^T {\mathcal{B}_0^\perp}^T & 0 & 0 & -\mathcal{W} \end{pmatrix} < 0.$$

A Schur complement of this matrix is

$$\begin{pmatrix} \mathcal{B}_0^{\perp}(\mathcal{A}_0\mathcal{X}\mathcal{A}_0^T - \mathcal{X} + \mathcal{N}\mathcal{X}\mathcal{N}^T + \mathcal{D}_0\mathcal{W}\mathcal{D}_0^T)\mathcal{B}_0^{\perp T} & 0 \\ 0 & -\mathcal{X} \end{pmatrix} < 0,$$

which is equivalent to (7.34) and $\mathcal{X} > 0$. *Furthermore, noting that*

$$\Lambda^{T^{\perp}} = \begin{pmatrix} 0 \\ G_0^T \\ \mathcal{C}_0^T \\ E_0^T \end{pmatrix}^{\perp} \begin{pmatrix} \mathcal{X}^{-1} & 0 & 0 & 0 \\ 0 & \mathcal{X}^{-1} & 0 & 0 \\ 0 & 0 & \mathcal{X}^{-1} & 0 \\ 0 & 0 & 0 & \mathcal{W}^{-1} \end{pmatrix},$$

defining $\mathcal{X}^{-1} = P$, *and substituting (7.24) and the above matrix into (7.29), the inequality (7.35) can be verified which completes the proof.*

The previous theorem provides the existence condition for the state estimator, and the characterization given in Theorem 3 is necessary and sufficient. However, we introduce a nonconvex constraint $\mathcal{X}P = I$, which makes the problem more difficult to solve. The next theorem shows how to rewrite these conditions into convex constraints by using Finsler's Lemma again.

Theorem 5 *There exists a state estimator gain* F *to solve (7.21) if there exist a symmetric matrix* $P \in \mathbf{R}^{2n_x \times 2n_x}$ *and* $\mu_1 < 0, \mu_2 < 0 \in \mathbf{R}$ *that satisfy*

$$P > 0, \tag{7.36}$$

$$\begin{pmatrix} -P & P\mathcal{A}_0 & P\mathcal{N} & P\mathcal{D}_0 & P\mathcal{B}_0 \\ \mathcal{A}_0^T P & -P & 0 & 0 & 0 \\ \mathcal{N}^T P & 0 & -P & 0 & 0 \\ \mathcal{D}_0^T P & 0 & 0 & -\mathcal{W}^{-1} & 0 \\ \mathcal{B}_0^T P & 0 & 0 & 0 & \mu_1^{-1} I \end{pmatrix} < 0, \tag{7.37}$$

$$\begin{pmatrix} -P & 0 & P\mathcal{A}_0 & P\mathcal{D}_0 & 0 & P\mathcal{N} \\ 0 & -P & 0 & 0 & G_0^T & 0 \\ \mathcal{A}_0^T P & 0 & -P & 0 & \mathcal{C}_0^T & 0 \\ \mathcal{D}_0^T P & 0 & 0 & -\mathcal{W}^{-1} & E_0^T & 0 \\ 0 & G_0 & \mathcal{C}_0 & E_0 & \mu_2^{-1} I & 0 \\ \mathcal{N}^T P & 0 & 0 & 0 & 0 & -P \end{pmatrix} < 0. \tag{7.38}$$

Proof 6 *According to Theorem 3 and Finsler's Lemma, we notice that the inequality (7.34) holds if and only if there exists a* $\mu_1 \in \mathbb{R}$ *such that*

$$\mathcal{A}_0 \mathcal{X} \mathcal{A}_0^T - \mathcal{X} + \mathcal{N} \mathcal{X} \mathcal{N}^T + \mathcal{D}_0 \mathcal{W} \mathcal{D}_0^T - \mu_1 \mathcal{B}_0 \mathcal{B}_0^T < 0.$$

Applying the congruence transformation yields

$$\mathcal{X}^{-1}(\mathcal{A}_0 \mathcal{X} \mathcal{A}_0^T - \mathcal{X} + \mathcal{N} \mathcal{X} \mathcal{N}^T + \mathcal{D}_0 \mathcal{W} \mathcal{D}_0^T - \mu_1 \mathcal{B}_0 \mathcal{B}_0^T)\mathcal{X}^{-1} < 0.$$

With the definition $P := \mathcal{X}^{-1} > 0$ *and the assumption* $\mu_1 < 0$*, the above condition can be equivalently written as (7.37) by using the Schur complement.*

Similarly, if the inequality (7.35) holds, it is equivalent to the existence of a $\mu_2 \in \mathbb{R}$ *such that*

$$\begin{pmatrix} -P + P\mathcal{N}\mathcal{X}\mathcal{N}^T P & 0 & P\mathcal{A}_0 & P\mathcal{D}_0 \\ 0 & -P & 0 & 0 \\ \mathcal{A}_0^T P & 0 & -P & 0 \\ \mathcal{D}_0^T P & 0 & 0 & -\mathcal{W}^{-1} \end{pmatrix} - \mu_2 \begin{pmatrix} 0 \\ G_0^T \\ \mathcal{C}_0^T \\ E_0^T \end{pmatrix} \begin{pmatrix} 0 & G_0 & \mathcal{C}_0 & E_0 \end{pmatrix} < 0.$$

By assuming $\mu_2 < 0$ *and applying Schur complements twice on the above inequality, it obtains (7.38) which is the desired conclusion.*

Since the inequality (7.37) and (7.38) are LMIs, the existence of a feasible solution for the state estimator is a convex problem which can be solved with the use of many available algorithms.

7.4.2 Filter Design

In the previous section, a sufficient LMI condition for checking the existence of state estimator has been given. Here it is dedicated to provide the conditions that guarantee

the additional closed loop system performance. We will determine a state estimator F such that the performance criterion, $\varepsilon_\infty\{\tilde{y}_k\tilde{y}_k^T\} < \Omega$, is satisfied. The fundamental algorithm that enables us to solve the filtering problem is derived from the following theorem.

Theorem 6 *There exists a state estimator gain F such that $\varepsilon_\infty\{\tilde{y}_k\tilde{y}_k^T\} < \Omega$ if there exist a positive definite symmetric matrix $P \in \mathbf{R}^{2n_x \times 2n_x}$ and $\mu_1 < 0, \mu_2 < 0 \in \mathbf{R}$ that satisfy (7.37), (7.38) and*

$$\begin{pmatrix} \Omega & \bar{C}_k \\ \bar{C}_k{}^T & P \end{pmatrix} > 0, \tag{7.39}$$

where

$$\bar{C}_k = C_k \begin{bmatrix} I & 0 \end{bmatrix}. \tag{7.40}$$

All the solutions F are given by

$$F = -R^{-1}\Gamma^T\Phi\Lambda^T(\Lambda\Phi\Lambda^T)^{-1} + S^{1/2}L(\Lambda\Phi\Lambda^T)^{-1/2}, \tag{7.41}$$

where

$$S = R^{-1} - R^{-1}\Gamma^T[\Phi - \Phi\Lambda^T(\Lambda\Phi\Lambda^T)^{-1}\Lambda\Phi]\Gamma R^{-1}. \tag{7.42}$$

L is an arbitrary matrix such that $\|L\| < 1$ and R is an arbitrary positive definite matrix such that

$$\Phi = (\Gamma R^{-1}\Gamma^T - \Theta)^{-1} > 0, \tag{7.43}$$

and

$$\Theta = \begin{pmatrix} -P^{-1} + \mathcal{N}P^{-1}\mathcal{N}^T & 0 & \mathcal{A}_0P^{-1} & \mathcal{D}_0\mathcal{W} \\ 0 & -P^{-1} & 0 & 0 \\ P^{-1}\mathcal{A}_0^T & 0 & -P^{-1} & 0 \\ \mathcal{W}\mathcal{D}_0^T & 0 & 0 & -\mathcal{W} \end{pmatrix},$$

$$\Gamma = \begin{pmatrix} \mathcal{B}_0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} 0 & G_0P^{-1} & \mathcal{C}_0P^{-1} & E_0\mathcal{W} \end{pmatrix}.$$

Proof 7 *We know that*

$$\varepsilon_\infty\{\tilde{y}_k\tilde{y}_k^T\} = C_k\,\varepsilon_\infty\{\tilde{x}_k\tilde{x}_k^T\}\,C_k^T = \bar{C}_k\mathcal{X}\bar{C}_k{}^T < \Omega,$$

where the state covariance matrix $\mathcal{X}$ *is defined in (7.20) and* $\bar{C}_k$ *is given in (7.40). With the definition* $P := \mathcal{X}^{-1} > 0$*, the inequality (7.39) can be manipulated by using the Schur complement. And the proof for solving F follows a similar approach in [108].*

We observe that the optimization approach proposed in this theorem is a convex programming problem stated as LMIs, which can be solved by efficient methods.

7.5 Numerical Example

In order to determine the applicability of the method, two examples to solve for the system design are presented next.

7.5.1 Four Mass Mechanical System

Consider the four mass mechanical system with springs and dampers depicted in Fig. 7.1. The discrete-time system dynamics is described in the following state space form

$$x_{k+1} = \begin{bmatrix} I & \Delta I \\ -\Delta M^{-1}K & I - \Delta M^{-1}G \end{bmatrix} x_k + \begin{bmatrix} 0 \\ \Delta M^{-1}D \end{bmatrix} \omega_k, \tag{7.44}$$

with the measurement

$$z_k = (H + H_s\zeta_k)x_k + v_k, \tag{7.45}$$

and the desired output

$$y_k = Cx_k, \tag{7.46}$$

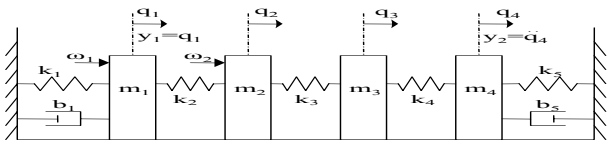


Figure 7.1. Four mass mechanical system with springs and dampers

where

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix}, \qquad G = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_5 \end{bmatrix},$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & 0 \\ -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & -k_4 & k_4 + k_5 \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{k_4}{m_4} & -\frac{k_4+k_5}{m_4} & 0 & 0 & 0 & -\frac{b_5}{m_4} \end{bmatrix}, \qquad C = H,$$

$$H_s = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$m_1 = m_2 = m_4 = 1, \quad m_3 = 2, \quad b_1 = 5, \quad b_5 = 2,$$
$$k_1 = k_3 = k_4 = 1, \quad k_2 = 2, \quad k_5 = 4.$$

Note that ω_k, v_k and ζ_k are uncorrelated zero mean Gaussian white noise sequences with unity covariance. And Δ is the time step (0.01sec). The performance criterion for the filter design is $[\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{i,i} < \Omega$ where $\Omega = 0.01$.

Figure 7.2 and 7.3 demonstrate the performance of the filter introduced in this chapter, where the error of each state variable is plotted. The simulation result shows that the output covariance of the estimation error are $[\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{1,1} = 0.001, [\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{2,2} = 0.0005$, which satisfy the design requirement, since both of 0.001 and $0.0005 < \mathbf{0.01}$.

7.5.2 Biomechanical Hand Movement System

Consider the hand modelled as a point mass ($m = 1kg$) whose one-dimensional position at time t is $p(t)$, and the velocity at time t is $v(t)$. The combined action of all

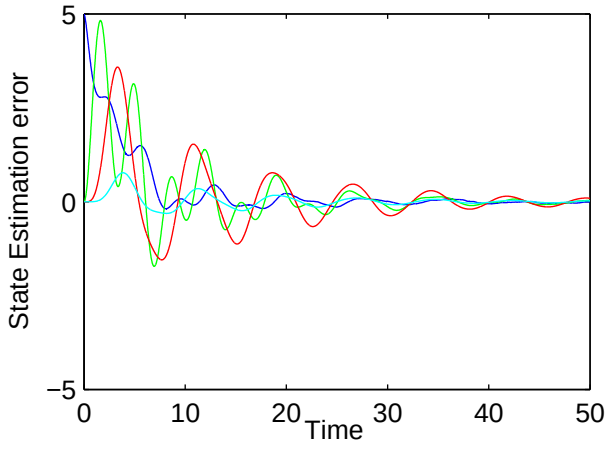


Figure 7.2. Estimation error (from state 1 to 4)

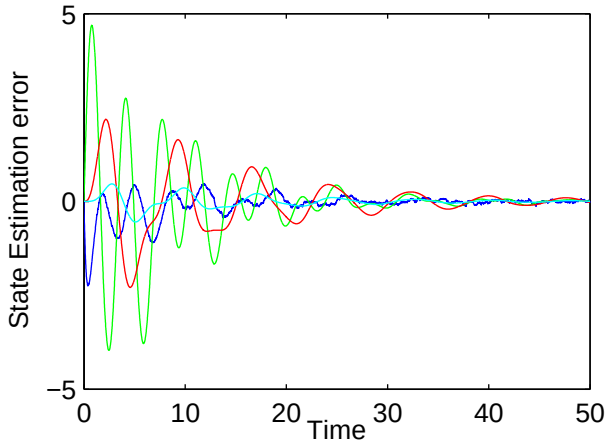


Figure 7.3. Estimation error (from state 5 to 8)

muscles is represented with the force $f(t)$ acting on the hand. The control signal $u(t)$ is transformed into force by adding control-dependent noise and applying a second order muscle-like low-pass filter

$$\tau_1\tau_2\ddot{f(t)} + (\tau_1 + \tau_2)\dot{f(t)} + f(t) = u(t),$$

where $\tau_1 = \tau_2 = 0.04sec$. We know that the above filter can be written as a pair of coupled first-order filters

$$\tau_1\dot{g} + g = u, \quad \tau_2\dot{f} + f = g.$$

The sensory feedback carries the information about position, velocity and force. The discrete-time system dynamics is described as following

$$\begin{aligned} x_{k+1} &= Ax_k + B(1+\sigma_c\varepsilon_k)u_k + \omega_k, \\ z_k &= (H + H_s\zeta_k)x_k + v_k, \\ y_k &= Hx_k. \end{aligned} \tag{7.47}$$

where

$$x_k = \begin{pmatrix} p_k & v_k & f_k & g_k \end{pmatrix}^T, \qquad y_k = \begin{pmatrix} p_k & v_k & f_k \end{pmatrix}^T,$$

$$A = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & \Delta/m & 0 \\ 0 & 0 & 1-\Delta/\tau_2 & \Delta/\tau_2 \\ 0 & 0 & 0 & 1-\Delta/\tau_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\Delta}{\tau_1} \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad H_s = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.125 & 0 \end{bmatrix},$$

and $\omega_k, v_k, \varepsilon_k, \zeta_k$ are independent zero-mean Gaussian white noise sequences with covariance

$$\begin{aligned} \Omega^\omega &= (diag[0.01, 0.001, 0.01, 0.01])^2, & \Omega^\varepsilon &= I, \\ \Omega^v &= (diag[0.01, 0.1, 0.5])^2, & \Omega^\zeta &= I. \end{aligned}$$

Note that $\sigma_c = 0.5$ is a unitless quantity that defines the noise magnitude relative to the control signal magnitude. And the time step $\Delta = 0.01$sec.

Given a controller $u_k = \begin{bmatrix} -1.6032 & -3.0297 & -0.3361 & -2.7793 \end{bmatrix} x_k$ such that the system (7.47) is mean square stable, the objective is to find a state estimator that bounds the estimation error below a specified error covariance: $[\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{i,i} < 0.1$.

Figure 7.4 illustrates the performance of the filter introduced here, where the error of each state variable is plotted. The simulation result shows that the output covariance of the estimation error are $[\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{1,1} = 0.0198$, $[\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{2,2} = 0.0037$, $[\varepsilon_\infty\{\tilde{y_k}\tilde{y_k}^T\}]_{3,3} = 0.0018 < \mathbf{0.1}$, which satisfy the design requirement.

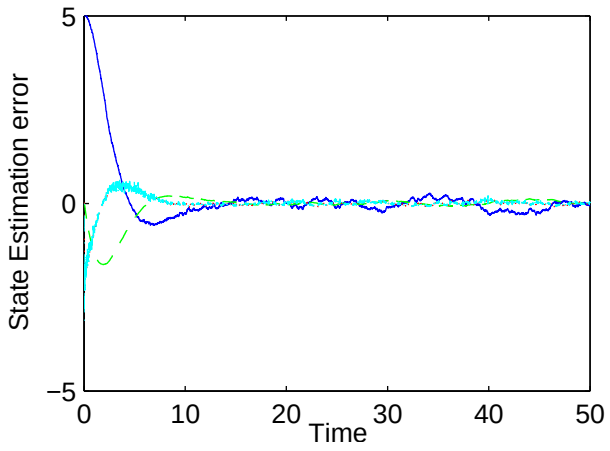


Figure 7.4. Estimation error for hand movement system

7.6 Summary

Multiplicative noise models are more practical than normal additive noise models, since they allow the statistical description of the multiplicative noise be not known *a prior* but depend on the control and state solution. Such models are found in many physical systems, such as signal processing systems, biological movement systems, and aerospace engineering systems.

An LMI based approach is examined in this chapter for the design of the state estimator with multiplicative noise systems. The proposed approach provides the sufficient conditions for the existence of state estimators and a parametrization of all admissible solutions. By adding a mild constrain, the original filtering problem is solved as a convex problem. The simulation results demonstrate the convergence of the system design.

Chapter 8

Conclusions and Future Directions

8.1 Iterative Optimal Estimation and Control Design

Optimal control theory plays a key role in the study of biological movement. One major goal of this thesis is to provide efficient computational methods for optimal control of complex biomechanical system. Given a realistic model of a biomechanical system, and a task goal specified as a quantitative performance criterion, our tools will attempt to design the control strategy that optimizes the performance. Here we present an iterative Linear-Quadratic-Gaussian method for locally-optimal control and estimation of nonlinear stochastic dynamical systems subject to control constraints. The main contribution is that the new method constructs an affine feedback control law obtained by minimizing a novel quadratic approximation to the optimal cost-to-go function. It also constructs a modified extended Kalman filter corresponding to the control law. The control law and filter are iteratively improved until convergence. What the most important is that our approach yields a numerical algorithm with stable convergence achieved through backtracking line search; and convergence in the vicinity of a local minimum is quadratic. Finally, the application of the algorithm is illustrated in the context of reaching movements and obstacle avoidance on a complex biomechanical control prob-

lem involving a stochastic model of the human arm. Although our work is motivated by studying biological movement control, the present results could be of useful to a wider audience.

8.2 Hierarchical Optimal Control Scheme

Sensorimotor control occurs simultaneously on multiple levels, and involves a large number of parallel feedback loops. Here we present a new framework to designing feedback control hierarchies for redundant biomechanical systems, that approximate the (non-hierarchical) optimal control law but have much lower computational demands. This feedback control hierarchy has two levels: the high level is designed as an optimal feedback controller operating on a simplified virtual plant, which is partly isolated from the complexity of the physical plant; the low level is responsible for transforming high-level commands into appropriate muscle activations, and also responsible for converting the dynamics of the true plant into the desired virtual dynamics assumed by the high level. The resulting control hierarchy makes it possible to solve the optimal control problem without running into the curse of dimensionality. Besides its engineering applications, the new hierarchical control framework addresses a key unresolved problem in modelling the neural control of movement, and its performance is demonstrated on the task of reaching movement, using two detailed models of the human arm: 2-Link 6-Muscle arm model including 2 degrees of freedom; and 5-Link 24-Muscle arm model which includes 5 degrees of freedom.

The present approach is also related to the idea of simplifying feedback transformation combined with high-level optimal control [119]. Those feedback transformation was "extracted" from the plant dynamics using unsupervised learning, rather than being designed to capture the task-relevant variables as we did here. As a result they were forced to use inefficient policy-gradient methods for controller optimization, rather than the efficient model-based ILQG method we employed here. Both approaches have advantages, and it is possible to combine them together in the future work. The basic idea is to make the low-level itself hierarchical: starting with a task-independent virtual model that cap-

tures the properties of the plant (and is acquired via unsupervised learning), and then creating higher-level virtual models suitable for specific tasks. Furthermore, it will be particularly interesting to examine the behavioral differences between the present hierarchical control scheme and the (unstructured) optimal controller in light of experimental data.

8.3 Future Directions — Inverse Optimal Control Design

When we are examining the human movement, such as reaching, throwing, orienting, it is usually hypothesized that natural control systems may be implicitly optimizing an objective function. However, it is not known what this performance criterion might be. In order to establish the correspondence between the natural and the artificial systems, we need to ascertain what it is that the natural control system is optimizing. Surprisingly, the novel inverse optimal control approach gives us a clue to determine the cost function by fitting the mathematical model to real behavior in a variety of circumstances.

Over the last few years, great interest has been achieved in optimal control of stochastic nonlinear system. One way of approaching these problems involves solving the Hamilton-Jacobi-Bellman (HJB) partial differential equation, which turn out to be an infeasible task. However, the inverse optimal control approach circumvents the task of solving HJB equation, whenever the optimal cost-to-go function is known (i.e. inferred from the data), it becomes an explicit formula and results in a meaningful cost functional.

The viewpoint of inverse optimum control problem was originated in the early 1960s by Kalman [50]: "Given a control law, find all performance criteria for which this control law is optimal." Kalman considered a precise formulation of this problem for deterministic linear systems and derived a beautiful conclusion for linear quadratic regulators. Given a feedback control $u = -kx$ for the system $\dot{x} = Fx + gu$, which is known to yield an asymptotically stable closed-loop system, then a necessary and sufficient condition for this control law to be optimal with respect to a performance index $\int_{t_0}^{\infty}(u^2 + x^T Qx)dt$ is that the return difference condition is hold for all real frequencies.

The inverse optimal control approach was introduced into nonlinear control in [83, 113, 134], while the main results of [113] are restricted to system processing an optimal performance that is quadratic in the state, and [83] establishes a certain nonlinear return inequality which implies robustness to some input nonlinearities. To our knowledge, this method has not been well-studied until it was recently revived for the design of robust nonlinear controller [102]. The main task of that inverse design method is the construction of control Lyapunov function as optimal value function for some meaningful cost functionals. And this approach has been applied to construct optimal feedback control laws for rigid spacecraft [59] and some other stochastic nonlinear systems [23, 24, 112].

The main goal is to address the problem of inverse optimal control to the biological movement systems, that is, the problem of extracting the objective function that can explain the observed, optimal movement behavior. Such a goal is possible because of significant experimental advances in our understanding of motor control systems in humans, and because of the increased knowledge in computational models for control learning. It will create a new way not only to analyze the natural movement phenomena, but also to derive much more effective methods in modelling all kinds of purposive(or skilled) behavior.

8.3.1 Preliminary Results

Investigations of general optimal control problems assume that the performance criteria are known. Based on this performance criterion and a given dynamic system, an optimal feedback control law is derived to satisfy the HJB equation. The inverse problem of optimal control, on the other hand, solves for the coefficients of cost function for which a given control law is optimal. Consider the problem of minimizing the discounted cost function subject to the nonlinear dynamic system

$$\dot{x} = f(x) + G(x)u, \tag{8.1}$$

$$V(x(0)) = \int_0^\infty e^{-s/\tau} \left[q(x(s)) + \frac{1}{2} u(s)^T R u(s) \right] ds, \qquad R > 0. \tag{8.2}$$

where $q(x)$ and R are unknown, $f(x)$ and $G(x)$ are known nonlinearities, and τ is the time constant for discounting future cost. A necessary and sufficient condition for a control law

$$u^*(t) = -R^{-1}G(x)^T V_x^*(x(t)) \tag{8.3}$$

to be an optimal control law is that it satisfies the following optimality condition

$$\frac{1}{\tau}V^*(x(t)) = \min_{u(t)}\left[q(x(t)) + \frac{1}{2}u(t)^T R u(t) + V_x^{*T}(f(x) + G(x)u(t))\right]. \tag{8.4}$$

where $V^*(x(t))$, known as the optimal value function (optimal cost-to-go), is the accumulated expected cost if the system is initialized in state x at time t, and the optimal control law $u^*(t)$ is applied until the end of optimal control history. By substituting (8.3) into the above equation, we obtain the HJB equation as

$$\frac{1}{\tau}V^*(x(t)) = q(x(t)) + V_x^{*T}f(x) - \frac{1}{2}V_x^{*T}G(x)R^{-1}G^T(x)V_x^*. \tag{8.5}$$

Given the dynamic system (8.1) and a control law $u(t)$, the objective here is to recover the cost function (including $q(x)$ and R) for which the control law is optimal. By using some form of function approximation

$$V(x(t)) \cong \sum_{i=1}^{m} \mathrm{w}_i \Phi_i(x) = \mathrm{w}^T \Phi(x), \tag{8.6}$$

and

$$\Phi_i(x) = exp\left(-\frac{\|x - c_i\|^2}{r_i^2}\right), \tag{8.7}$$

$$\Phi(x) = (\Phi_1(x) \cdots \Phi_m(x))^T, \qquad \mathrm{w} = (\mathrm{w}_1\, \mathrm{w}_2 \cdots \mathrm{w}_m)^T, \tag{8.8}$$

where w_i are weights of Gaussian Basis Function $\Phi_i(x)$, c_i and r_i define the center and width of the i^{th} Gaussian basis function. Actually the basis function $\Phi(\cdot)$ plays an important role in traditional neural networks. The linear parameters w_i can be determined by the linear regression method. By choosing R as a positive definite matrix, the cost function can be recovered by computing

$$q(x) = \frac{1}{\tau}\mathrm{w}^T\Phi(x) + (\mathrm{w}^T\Phi_x(x))f(x) + \frac{1}{2}(\mathrm{w}^T\Phi_x(x))^T G(x)R^{-1}G^T(x)(\mathrm{w}^T\Phi_x(x)), \tag{8.9}$$

where $\Phi_x(x)$ defines the derivative of $\Phi(x)$ over x.

Here we will give a simple example to demonstrate the feasibility of the above idea. Consider a scaler system

$$\dot{x} = -x + u, \tag{8.10}$$

and the performance criterion

$$V(x(0)) = \int_0^\infty e^{-s/\tau} \left[q(x(s)) + \frac{1}{2} r u(s)^2 \right] ds, \qquad r > 0. \tag{8.11}$$

where the discounted factor $\tau = 5$, and $q(x)$ and r are unknown. Given a control law u under a quadratic cost function where $q(x) = \frac{1}{2}(x - x_{target})^2$ and $r = 1$, the main objective here is to find $q(x)$ and r for which this control law u is optimal, which means, we would like to know whether the estimated $q(x)$ can recover the original quadratic performance. Fig. 8.1 illustrates the cost function $q(x)$ for three different targets: $x_{target} = -4$, 0 and 4. Black line describes the original function, circle describes the recovered $q(x)$ based on inverse optimal control approach. The simulation demonstrates that, by changing the targets, the cost function can still be recovered very well.

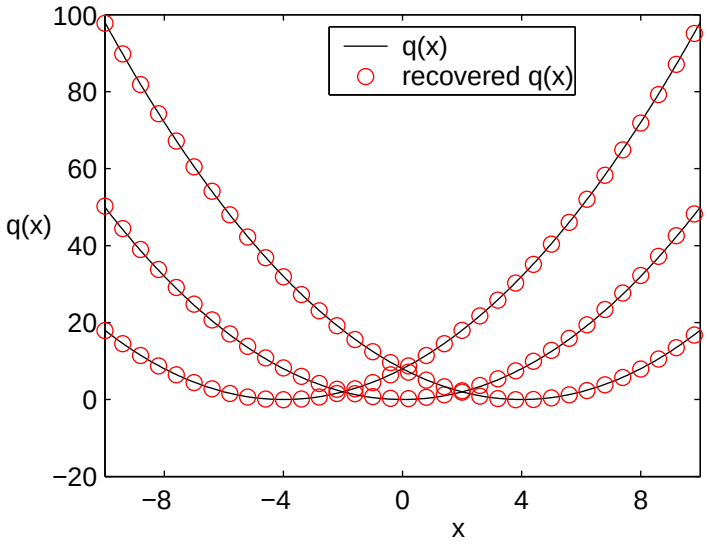


Figure 8.1. Cost function q(x) for three different targets: $x_{target} = -4$, 0 and 4. (Circle describes the recovered q(x) based on inverse optimal control approach, dark line describes the original q(x) function)

8.3.2 Future Work

1. Currently we only show the preliminary results on the inverse optimal control problem of the deterministic linear system under the quadratic cost assumption, how about the nonlinear stochastic system? Given a nonlinear stochastic system and a control law under a non-quadratic cost function, the main objective is to find whether the estimated cost function can recover the original non-quadratic performance for which this control law is optimal. Actually our proposed method used the general HJB equation which can deal with nonlinear dynamic cases, and our approximation to cost function uses Gaussian basis functions which do not assume a quadratic form. However it will be very useful to study how well this inverse optimal control approach works on some specific nonlinear problems.

2. When we apply the inverse optimal control approach to the biological movement system, there are at least two sources of error which can affect the performance of the final result: (a) the data we obtained is noisy (or the biological control system is not entirely optimal); (b) our function approximation has limited capacity which cannot describe the real cost function very well. In both cases, the inferred optimal cost-to-go function will be somewhat inaccurate. How severe are those inaccuracies numerically?

3. To control a dynamic system, you must first know what it is doing. However, it is not possible to measure every variable that you want to control. Back to the biological motor systems, in order to control our movements, the central nervous system needs to know the state variables that it want to control. Suppose there exist an state estimator which provides a means for estimating those information from noisy measurement, therefore, controls will depend on the state estimate $\hat{x}$ rather than the true state x. Another interesting question is: could we formulate the inverse optimal control problem in this kind of setting?

4. In our proposed work, combining (8.3) and (8.6), and fixing R, the weights of Gaussian basis function is determined by solving the following equation using linear regression method: $u^*(t) = -R^{-1}G(x)^T M\mathrm{w}$, where M is the function of $\Phi_x(x)$. What happens if the matrix $R^{-1}G(x)^T M$ is rank deficient? We will obtain a family of solution

for w. In this case it is very hard to infer the optimal cost-to-go function from data. Perhaps we can infer a whole family of optimal cost-to-go functions that are all consistent with the data, and that leads to a family of costs.

5. In the actual psychophysical experiments, one usually records movement kinematics (which could infer the dynamics), but the muscle activations (which are real control commands) are hard to record. Can we formulate the problem so that the control are inferred from motion rather than being measured? Probably we have to infer a probability distribution (or family) of plausible controls, rather than a specific set of controls.

6. In order to evaluate the effectiveness of the inverse optimal control approach, we will design some motor psychophysics experiments where this approach can be validated. For example, we ask the subjects to make planar arm movements through sequences of targets. In condition A, we use five widely spaced targets, whereas in condition B, we include additional targets chosen to fall along the average trajectory produced in condition A. The goal here is to investigate whether the inverse optimal control approach can distinguish these two different task goals with similar average movements.

Bibliography

[1] B. D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, N.J., 1979.

[2] F. C. Anderson and M. G. Pandy. Dynamic optimization of human walking. *Journal of Biomechanical Engineering*, 123, 381-390, 2001.

[3] F. C. Anderson and M. G. Pandy. Static and dynamic optimization solutions for gait are practically equivalent. *Journal of Biomechanics*, 24, 153-161, 2001.

[4] C. G. Atkeson. Using local trajectory optimizers to speed up global optimization in dynamic programming. *Advances in Neural Information Processing Systems 6*, J.D. Cowan, G. Tesauro, and J. Alspector eds., Morgan Kaufmann 1994.

[5] J. J. Beaman. Non-linear quadratic gaussian control. *Int. J. Control* Vol.39, No.2, 343-361, 1984.

[6] A. Beghi and D. D'alessandro. Discrete-time optimal control with control-dependent noise and generalized Riccati difference equations. *Automatica* Vol.34, No.8, 1031-1034, 1998.

[7] A. Bensoussan. *Stochastic Control for Partially Observable Systems*. Cambridge University Press, 1992.

[8] N. Bernstein. *The coordination and regulation of movement*. Oxford, New York: Pergamon Press, 1967.

[9] N. Bernstein. *Dexterity and its development, in Dexterity and its development*. M. Latash and M. Turvey (Editors), Lawrence Erlbaum, Mahwah, NJ, 1996.

[10] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Massachusetts, USA, 2000.

[11] R. R. Bitmead. Iterative control design approaches. *IFAC World Congress*, Sydney, Invited plenary paper, Vol. 9, 381-384, 1993.

[12] R. R. Bitmead. Iterative Optimal Control, in *Lecture Notes on Iterative Identification and Control Design*, P. Albertos and A. Sala (eds), European Science Foundation, Strasbourg France, 153-166, 2000.

[13] S. Boyd , L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, SIAM, 1994.

[14] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1), 14-23, 1986.

[15] I. E. Brown, E. J. Cheng and G. E. Leob. Measured and modeled properties of mammalian skeletal muscle. II. The effects of stimulus frequency on force-length and force-velocity relationships. *J. Muscle* Res Cell Motil 20: 627-643, 1999.

[16] A. E. Bryson and Yu-Chi Ho. *Applied Optimal Control*. Blaisdell Publishing Company, Massachusetts, USA, 1969.

[17] A. E. Bryson. Optimal control — 1950 to 1985. *IEEE Control Systems*, 26-33, June 1996.

[18] C. D. Charalambous and R. J. Elliott. Classes of nonlinear partially observable stochastic optimal control problems with explicit optimal control laws. *SIAM J. Control Optim.* Vol.36, No.2, 542-578, 1998.

[19] C. L. Chen, D. Sun and C. Chang. Numerical solution of time-delayed optimal control problems by iterative dynamic programming. *Optimal control applications and methods*, 21, 91-105, 2000.

[20] P. Corke. A robotics toolbox for Matlab. *IEEE Robotics and Automation Magazine* vol.3, No.1, 2432, 1996.

[21] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.

[22] H. Deng and M. Krstic. Stochastic nonlinear stabilization—PartI: A backstepping design. *Systems and Control Letters*, vol.32, 143-150, 1997.

[23] H. Deng and M. Krstic. Stochastic nonlinear stabilization—PartII: Inverse optimality. *Systems and Control Letters*, vol.32, 151-159, 1997.

[24] H. Deng and M. Krstic. Output-feedback stochastic nonlinear stabilization. *IEEE Transactions on Automatic Control*, vol.44, 328-333, 1999.

[25] K. Doya. Reinforcement learning in continuous time and space. *Neural COmputation*, 12, 219-245, 2000.

[26] K. Doya. Temporal difference learning in continuous time and space. *Advances in Neural Information Processing Systems 8*, Touretzky DS, Mozer MC, Hasselmo ME eds., MIT Press, 1073-1079, 1996.

[27] J. C. Doyle, K. Glover, P. P. Khargonekar and B. A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, vol.34, No.8, 831-847, August 1989.

[28] A. E. Engin and R. D. Peindl. On the biomechanics of human shoulder complex — I. Kinematics for determination of the shoulder complex sinus. *Journal of Biomechanics*, 20, 2, 103-117, 1987.

[29] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, Vol.47, No.6, pp. 381-391, June 1954.

[30] T. Flash and N. Hogan. The coordination of arm movements: An experimentally conformed mathematical model. *The Journal of Neuroscience*, Vol.5, 1688-1703, 1985.

[31] A. Freivalds. *Biomechanics of the Upper Limbs: Mechanics, Modeling, and Musculoskeletal Injuries*. CRC Press, Boca Raton, FL, 2004.

[32] B. A. Garner and M. G. Pandy. A Kinematic Model of the Upper Limb Based on the Visible Human Project (VHP) Image Dataset. *Computer Methods in Biomechanics and Biomedical Engineering*, Vol.2, 107-124, 1999.

[33] B. A. Garner and M. G. Pandy. The Obstacle-Set Method for Representing Muscle Paths in Musculoskeletal Models. *Computer Methods in Biomechanics and Biomedical Engineering*, Vol.3, 1-30, 2000.

[34] B. A. Garner and M. G. Pandy. Musculoskeletal model of the upper limb based on the visible human male dataset. *Computer Methods in Biomechanics and Biomedical Engineering*, Vol.4, 93-126, 2001.

[35] E. Gershon, U. Shaked and I. Yaesh. H_∞ control and filtering of discrete-time stochastic systems with multiplicative noise. *Automatica* Vol.37, 409-417, 2001.

[36] L. El Ghaoui. State-feedback control of systems with multiplicative noise via linear matrix inequalities. *Systems and Control Letters*, February 1995.

[37] M. Hardt, J. W. Helton and K. Kreutz-Delgado. Numerical Solution of Nonlinear H_2 and H_∞ Control Problems with Application to Jet Engine Compressors. *IEEE Control Systems Technology*, Vol.8, No.1, 98-111, 2000.

[38] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, Vol.394, August 1998.

[39] H. Hatze and J. D. Buys. Energy-optimal controls in the mammalian neuromuscular system. *Biological Cybernetics* 27, 9-20, 1977.

[40] F.C.T. Van der Helm. A finite element musculoskeletal model of the shoulder mechanism. *J. Biomechanics*, 27, 551-569, 1994.

[41] B. Hoff. A Computational Description of the Organization of Human Reaching and Prehension. Ph.D. Thesis, University of Southern California, 1992.

[42] N. Hogan. An organizing principle for a class of voluntary movement. *Journal of Neuroscience*, Vol.4, No.11, 2745-2754, 1984.

[43] C. Högfors, G. Sigholm and P. Herberts. Biomechanical model of the human shoulder — I. Elements. *J. Biomechanics*, 20, 157-166, 1987.

[44] A. Isidori. *Nonlinear Control Systems*. Springer, New York, USA, 1995.

[45] D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier Publishing Company, New York, USA, 1970.

[46] M. Jamshidi. *Large-scale Systems, Modeling and Control*. North-Holland, New York, 1983.

[47] J. C. Jimenez and T. Ozaki. Linear estimation of continuous-discrete linear state space models with multiplicative noise. *Systems and Control Letters*, 47, 91-101, 2002.

[48] M. I. Jordan and D. M. Wolpert. Computational motor control. In M. Gazzaniga (Ed.), *The Cognitive Neurosciences*, 2nd edition, Cambridge: MIT Press, 1999.

[49] F. Lacquaniti, C. Terzuolo, and P. Viviani. The law relating kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54:115130, 1983.

[50] R. E. Kalman. When is a linear control system optimal? *Journal of Basic Engineering*, 51-60, March 1964.

[51] M. Kawato. Internal model for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9: 718-717, 1999.

[52] D. L. Kern and F. B. Hanson. Filtering approximation using systematic perturbations of a discrete-time stochastic dynamical system. *Proceedings of 1999 American Control Conference*, June 1999.

[53] M. A. Khan, R. CHua, D. Elliott, J. Coull and J. Lyons. Optimal control strategies under different feedback schedules: Kinematic evidence. *Journal of Motor Behavior*, Vol.34, No.1, 45-57, 2002.

[54] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, Vol. RA-3, No.1, 1987.

[55] D. E. Kirk. *Optimal Control Theory An Introduction*. Prentice Hall Inc., New Jersey, 1970.

[56] D. L. Kleinman. Optimal stationary control of linear systems with control-dependent noise. *IEEE Transactions on Automatic Control*, Vol.AC-14, No.6, 673-677, December 1969.

[57] K. P. Körding and D. M. Wolpert. Bayesian integration in sensorimotor learning. *Nature* 427:244-247, 2004.

[58] M. Krstic and Z. H. Li. Inverse optimal design of input-to-state stabilizing nonlinear controllers. *IEEE Transactions on Automatic Control*, vol. 43, 336-351, 1998.

[59] M. Krstic and P. Tsiotras. Inverse optimal stabilization of a rigid spacecraft. *IEEE Transactions on Automatic Control*, vol.44, 1042-1049, 1999.

[60] A. D. Kuo. An optimal control model for analyzing human postural balance. *IEEE Transactions on Biomedical Engineering*, Vol.42, No.1, January 1995.

[61] H. Kushner and P. Dupuis. *Numerical methods for stochastic control problems in continuous time (2nd edition)*. Spinger, New York, USA, 2001.

[62] T. Lahdhiri and H. A. Elmaraghy. Design of optimal feedback linearizing-based controller for an experimental flexible-joint robot manipulator. *Optimal Control Applications and Methods*, Volume 20, 165-182, 1999.

[63] F. L. Lewis and V. L. Syrmos. *Optimal Control*. John Wiley and Sons, New York, USA, 1995.

[64] L. Z. Liao and C. Shoemaker. Convergence in unconstrained discrete-time differential dynamic programming. *IEEE Transactions on Automatic Control*, Vol. 36, No. 6, 692-706, 1991.

[65] D. Li. Hierarchical control for large-scale systems with general multiple linear-quadratic structure. *Automatica*, Vol. 29, No. 6, 1451-1461, 1993.

[66] W. Li and R. E. Skelton. State estimation with finite signal-to-noise models. *Proceedings of the 42nd IEEE Conference on Decision and Control*, 5378-5383, Hawaii, December 2003.

[67] W. Li and E. Todorov. Iterative linear quadratic regulator applied to nonlinear biological movement systems. *In Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics*, Vol. 1, 222-229, 2004.

[68] W. Li, E. Todorov, and X. Pan. Hierarchical optimal control of redundant biomechanical systems. *Proc. of the 26th IEEE Conference on Engineering in Medicine and Biology Society*, 4618-4621, September 2004.

[69] W. Li, E. Todorov and X. Pan. Hierarchical feedback and learning for multi-joint arm movement control, *Proceedings of the 27th IEEE Conference on Engineering in Medicine and Biology Society*, September 2005.

[70] W. Li, E. Todorov and R. E. Skelton. Estimation and control of systems with multiplicative noise via linear matrix inequalities, *Proceedings of 2005 American Control Conference*, 1811-1816, June 2005.

[71] A. E. B. Lim, Y. Q. Liu, K. L. Teo and J. B. Moore. Linear-quadratic optimal control with integral quadratic constraints. *Optimal control applications and methods*, 20, 79-92, 1999.

[72] G. E. Loeb, W. S. Levine and J. He. Understanding sensorimotor feedback through optimal control. *Cold Spring Harbor Symposia on Quantitative Biology*, Vol.55, 1990.

[73] G. E. Loeb, I. E Brown and E. Cheng. A hierarchical foundation for models of sensorimotor control. *Experimental Brain Research*, 126, 1-18, 1999.

[74] J. Lu, R. E. Skelton. Robust variance control for systems with finite signal-to-noise uncertainty. *Automatica* 36, 2000, pp. 511-525.

[75] R. Luus. *Iterative Dynamic Programming*. Chapman & Hall/CRC, USA, 2000.

[76] Y. Mao and Z. Liu. The optimal feedback control of the linear-quadratic control problem with a control inequality constraint. *Optimal control applications and methods*, 22, 95-109, 2001.

[77] W. Maurel and D. Thalmann. A case study on human upper limb modelling for dynamic simulation. *Computer Methods in Biomechanics and Biomedical Engineering*, Vol. 2, No. 1, 65-82, 1999.

[78] P. J. Mclane. Optimal stochastic control of linear systems with state- and control-dependent disturbances. *IEEE Transactions on Automatic Control*, Vol. AC-16, No.6, 793-798, December 1971.

[79] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright and J. E. K. Smith. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95, 340-370, 1988.

[80] R. C. Miall and D. M. Wolpert. Forward models for physiological motor control. *Neural Network*, Vol.9, No.8, 1265-1279, 1996.

[81] J. B. Moore, X. Zhou and A. Lim. Discrete time LQG controls with control dependent noise. *Sytem and Control Letters*, Vol.36, 199-206, 1999.

[82] P. Morasso. Spatial control of arm movements. *Exp Brain Res* 42: 223-227, 1981.

[83] P. J. Moylan and B. D. Anderson. Nonlinear regulator theory and an inverse optimal control problem. *IEEE Transactions on Automatic Control*, Vol.AC-18, No.5, 460-465, October 1973.

[84] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka and M. Kawato. Quantitative examinations of internal representations for arm trajectory planning: minimum commanded torque change model. *Journal of Neurophysiology* 81, 2140-2155, 1999.

[85] Y. Niho and J. H. Makin. A solution to the inverse problem of optimal control: Note. *Journal of Money, Credit and Banking*, Vol.10, No.3, 371-377, August 1978.

[86] A. Y. Ng, D. Harada and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. *In Proceedings of the Sixteenth International Conference on Machine Learning*, 1999.

[87] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. *In Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[88] C. K. Ng, L. Z. Liao and D. Li. A globally convergent and efficient method for unconstrained discrete-time optimal control. *Journal of Global Optimization*, 23, 401-421, 2002.

[89] M. C. de Oliveria and R. E. Skelton. State feedback control of linear systems in the presence of devices with finite signal-to-noise ratio. *International Journal of Control*, 2001, vol.74, No.15, pp. 1501-1509.

[90] K. Ohta, M. M. Svinin, Z. W. Luo and S. Hosoe. On the trajectory formation of the human arm constrained by the external environment. *Proceeding of the 2003 IEEE International COnference on Robotics and Automation*, pp. 2884-2891, 2003.

[91] J. Pantoja. Differential dynamic programming and newton's method. *International Journal of Control*, 47, 1539-1553, 1988.

[92] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth and G. Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, Vol. 20, No. 2, pp. 129-143, February 2001.

[93] Y. A. Phillis. Controller design of systems with multiplicative noise. *IEEE Transactions on Automatic Control*, vol.AC-30, No.10, October 1985.

[94] Y. A. Phillis. Estimation and control of systems with unknown covariance and multiplicative noise. *IEEE Transactions on Automatic Control*, vol.34, No.10, pp 1075-1078, October 1989.

[95] M. A. Rami and L. El Ghaoui. LMI optimization for nonstandard Riccati equations arising in stochasitc control. *IEEE Transactions on Automatic Control*, vol.41, No.11, November 1996.

[96] M. A. Rami, X. Chen, J. B. Moore and X. Zhou. Solvability and asympototic bahavior of generalized Riccati equations arising in indefinite stochasitc LQ controls. *IEEE Transactions on Automatic Control*, vol.46, No.3, pp 428-440, March 2001.

[97] P. D. Roberts and V. M. Becerra. Optimal control of a class of discrete-continuous non-linear systems—decomposition and hierarchical structure. *Automatica*, vol.37, 1757-1769, 2001.

[98] A. Santos and R. Bitmead. Nonlinear control design for an autonomous underwater vehicle (AUV) preserving linear design capabilities, *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, 1995, 3817-3823.

[99] C. W. Scherer, P. Gahinet and M Chilali. Multi-objective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control* 42, 896-911, 1997.

[100] R. A. Schmidt, H. Zelaznik, B. Hawkin, J. S. Frank and J. T. Quinn. Motor-output variability: a theory for the accuracy of rapid motor tasks. *Psychological Review* 86, 415-451, 1979.

[101] J. P. Scholz A1 and G. Schöner. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, vol.126, No.3, 289-306, 1999.

[102] R. Sepulchre, M. Janković and P. Kokotović. *Constructive Nonlinear Control*, Springer, 1997.

[103] R. Shadmehr and F. A. Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14: 3208-3224, May 1994.

[104] G. Shi and R. E. Skelton. State feedback covariance control for linear finite signal-to-noise ratio models. *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, December 1995.

[105] M. G. Singh and A. Titli. *Systems: Decomposition, Optimisation and Control*. Pergamon Press, Oxford, 1978.

[106] R. E. Skelton. *Dynamic Systems Control*. John Wiley and Sons, New York, USA, 1988.

[107] R. E. Skelton. Robust control of aerospace systems. *Proceeding IFAC Symposium on Robust Control Design*, Rio de Janeiro, Brazil, 24-32, September 1994.

[108] R. E. Skelton, T. Iwasaki and K. Grigoriadis. *A Unified Algebraic Aproach to Linear Control Design*. Taylor and Francis, London, UK, 1997.

[109] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[110] W. Son, K. Kim, N. Amato and J. C. Trinkle. A generalized framework for interactive dynamic simulation for multirigid bodies. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.34, No.2, 2004.

[111] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. Wiley, New York, 1989.

[112] C. Tang and T. Basar. Inverse optimal controller design for strict-feedback stochastic systems with esponential-of-integral cost. *Proceeding of the 15th IFAC World Congress*, Barcelona, Spain, July 2002.

[113] F. E. Thau. On the inverse optimal control problem for a class of nonlinear autonomous systems. *IEEE Transactions on Automatic Control*, Vol.AC-12, No.6, 674-681, December 1967.

[114] U. H. Thygesen and R. E. Skelton. Linear system with finite signal-to-noise ratios: a robust approach. *Proceedings of the 34th IEEE Conference on Decision and Control*, pp 4157-4162, New Orleans, December 1995.

[115] E. Todorov and M. Jordan. Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *Journal of Neurophysiology* 80, 696-714, 1998.

[116] E. Todorov and M. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, Vol.5, No.11, 1226-1235, 2002.

[117] E. Todorov and M. Jordan. A minimal intervention principle for coordinated movement. *Advances in Neural Information Processing Systems*, 15: 27-34, Becker, Thrun, Obermayer (eds), MIT Press, 2003.

[118] E. Todorov and W. Li. Optimal control methods suitable for biomechanical systems. *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Biology and Medicine Society*, September 2003.

[119] E. Todorov and Z. Ghahramani. Unsupervised learning of sensory-motor primitives. *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Biology and Medicine Society*, September 2003.

[120] E. Todorov. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural Computation*, 17(5), 2005.

[121] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9), 907-915.

[122] E. Todorov. On the role of primary motor cortex in arm movement control. *In Progress in Motor Control III*, ch 6, pp 125-166, Latash and Levin (eds), Human Kinetics, 2003.

[123] E. Todorov and W. Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. *Proceedings of 2005 American Control Conference*, pp. 300-306, 2005.

[124] E. Todorov, W. Li and X. Pan. From Task Parameters to Motor Synergies: A Hierarchical Framework for Approximately-optimal Control of Redundant Manipulator. *Journal of Robotic Systems* 22(11), 691-710, 2005.

[125] 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.

[126] Y. Wada, Y. Kaneko, E. Nakano, R. Osu, and M. Kawato. Quantitative examinations for multi joint arm trajectory planning by a strict and robust calculation algorithm of minimum commanded torque change trajectory. *Neural Networks*, 14(4):381-393, 2001.

[127] F. Wang and V. Balakrishnan. Robust estimators for systems with determinstic and stochastic uncertainties. *Proceedings of the 38th IEEE Conference on Decision and Control*, Arizona, December 1999.

[128] P. Whittle. *Risk-Sensitive Optimal Control*. John Wiley and Sons, 1990.

[129] J. L. Willems and J. C. Willems. Feedback stabilizability for stochastic systems with state and control dependent noise. *Automatica*, 12, 277-283, 1976.

[130] D. M. Wolpert, Z. Ghahramani and M. I. Jordan. An internal model for sensorimotor integration. *Science*, Vol.269, 1880-1882, September 1995.

[131] D. M. Wolpert. Computational approaches to motor control. *Trends in Cognitive Science*, Vol.1, No.6, 209-216, September 1997.

[132] D. M. Wolpert and Z. Ghahramani. Computational principles of movement neuroscience. *Nature Neuroscience*, Vol.3, 1212-1217, November 2000.

[133] F. Yang, Z. Wang and Y. S. Hung. Robust kalman filtering for discrete time-varying uncertain systems with multiplicative noises. *IEEE Transactions on Automatic Control*, vol.47, No.7, 1179-1183, July 2002.

[134] R. Yokoyama and E. Kinnen. The inverse problem of the optimal regulator. *IEEE Transactions on Automatic Control*, Vol.AC-17, No.4, 497-504, August 1972.

[135] H. Zhang and M. Ishikawa. Structure determination of a criterion function by dynamic inverse optimization. *Proceeding of the International Conference on Neural Information Processing*, 662-666, Taejon, Korea 2000.