Automating Stochastic Control

Krishnamurthy Dvijotham

Automating Stochastic Control

Doctor of Philosophy

University of Washington

2014

Reading Committee:

Emanuel Todorov, Chair

Maryam Fazel, Chair

Mehran Mesbahi

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Automating Stochastic Control

Krishnamurthy Dvijotham

Co-Chairs of the Supervisory Committee:
Associate Professor Emanuel Todorov
Computer Science and Engineering & Applied Mathematics

Associate Professor Maryam Fazel
Electrical Engineering

Stochastic Optimal Control is an elegant and general framework for specifying and solving control problems. However, a number of issues have impeded its adoption in practical situations. In this thesis, we describe algorithmic and theoretical developments that address some of these issues. In the first part of the thesis, we address the problem of designing cost functions for control tasks. For many tasks, the appropriate cost functions are difficult to specify and high-level cost functions may not be amenable to numerical optimization. We adopt a data-driven approach to solving this problem and develop a convex optimization based algorithm for learning costs given demonstrations of desirable behavior. The next problem we tackle is modelling risk-aversion. We develop a general theory of linearly solvable optimal control capable of modelling all these preferences in a computationally tractable manner. We then study the problem of optimizing parameterized control policies. The study presents the first convex formulation of control policy optimization for arbitrary dynamical systems. Using algorithms for stochastic convex optimization, this approach leads to algorithms that are guaranteed to find the optimal policy efficiently. We describe applications of these ideas to multiple problems arising in energy systems. Finally, we outline some future possibilities for combining policy optimization and cost-learning into an integrated data-driven cost shaping framework.

TABLE OF CONTENTS

Page

List of Figures iii

Glossary v

Chapter 1: Introduction 1

Chapter 2: Background & Fundamentals 7
2.1 Linearly Solvable Optimal Control Problems 10
2.2 Properties and algorithms 18

Chapter 3: Designing Costs: Inverse Optimal Control 22
3.1 Discrete problems 24
3.2 Continuous problems 30
3.3 Summary 35
3.4 Appendix : Convexity of OptQ 35

Chapter 4: Modeling Risk: A Unified Theory of Linearly Solvable Optimal Control 36
4.1 Game Theoretic Control : Competitive Games 36
4.2 Conclusions 43
4.3 Proofs 43

Chapter 5: Convex Policy Optimization 48
5.1 Convex Policy Optimization Based on Bode's Sensitivity Integral 49
5.2 Convex Stochastic Policy Optimization for General Dynamical Systems . . . 75
5.3 Applications 79

Chapter 6: Applications to Electric Energy Systems 85
6.1 Storage Sizing and Placement to Operational and Uncertainty-Aware Simulations 85
6.2 Introduction 85
6.3 Related Work 86

6.4 Mathematical Formulation 89
6.5 Simulations 96
6.6 Discussion 100
6.7 Conclusions and Future Work 101
6.8 Distributed Control of Frequency in a Grid with a High Penetration of Renewables 102

Chapter 7: A vision for Automated Stochastic Control 118
7.1 Convex Data-Driven Cost-Shaping for LMDPs 119
7.2 Convex Data-Driven Policy Optimization with Cost Shaping 126

Chapter 8: Conclusions & Future Work 128

LIST OF FIGURES

Figure Number — Page

1.1 General Stochastic Control Problem 2

2.1 Probability Shift 11

2.2 Continuous problems. Comparison of our MDP approximation and a traditional MDP approximation on a continuous car-on-a-hill problem. (A) Terrain, (B) Z iteration (ZI) (blue), policy iteration (PI) (red), and value iteration (VI) (black) converge to control laws with identical performance; ZI is 10 times faster than PI and 100 times faster than VI. Horizontal axis is on log-scale. (C) Optimal cost-to-go for our approximation. Blue is small, red is large. The two black curves are stochastic trajectories resulting from the optimal control law. The thick magenta curve is the most likely trajectory of the optimally controlled stochastic system. (D) The optimal cost-to-go is inferred from observed state transitions by using our algorithm for inverse optimal control. Figure taken from Todorov [2009b]. 17

3.1 Comparison of OptV and prior IRL algorithms on a grid-world problem. Black rectangles are obstacles. 29

3.2 **(A)** Control policies in the first-exit (inverted pendulum) problem. Each subplot shows the CPU time and the policy found given the optimal transition probabilities. The policy found by OptV was indistinguishable from the optimal policy and achieved average cost of 13.06, as compared to 57.21 for Syed and 41.15 for Abbeel. **(B)** Value functions in the infinite-horizon (metronome) problem. Here the algorithms have access to finite data (12,000 transitions) thus the optimal value function can no longer be recovered exactly. OptV with a lookup table representation does quite poorly, indicating the need for smoothing/generalization. The result of OptVA with the initial bases vaguely resembles the correct solution, and is substantially improved after basis adaptation. The ellipses show the location and shape of the Gaussian bases before normalization. **(C)** Performance of OptVA over iterations of basis adaptation for 12,000 samples (left), and as a function of the sample size at the last iteration of basis adaptation (right). We plot the difference between the optimal and inferred z functions (expressed as KL divergence), and the log average cost of the resulting control policy. The curves are scaled and shifted to fit on the same plot. 34

4.1 Terrain and Cost Function for LMG example 40

4.2 Logarithm of Stationary Distribution under Optimal Control vs α 40

5.1 Convex Surrogate vs Original Objective (rescaled to lie in [0,1]): $q_\infty(k)$ vs $UB_\infty(k)$ (top),$q_2(k)$ vs $UB_2(k)$ (bottom) 54
5.2 Comparison of Algorithms for q_∞-norm Controller Synthesis. The blue bars represent histograms and the red curves kernel density estimates of the distribution of values. 64
5.3 Comparison of Algorithms for q_2-norm Controller Synthesis. The blue bars represent histograms and the red curves kernel density estimates of the distribution of values. 65
6.3 Iterations of our Algorithm on the BPA System. Red Corresponds to Low Storage Capacity and Purple to High 99
6.4 Storage Placement (colored circles) relative to Wind Farms/Interties (shown as blue diamonds). 99
6.5 Total Energy and Power Capacity Relative to Placement at Renewables and Interties 100
6.6 Our model of the BPA Transmission Network 104
6.7 Comparison of control schemes. a) Aggregate wind generation from a period with significant ramping events. b) *Worst-case* frequency deviations over the control period for 18 validation scenarios not used in the control design. . . 112
7.1 Data-Driven Cost-Shaping Controller Design 118

GLOSSARY

MDP: Markov Decision Process: A model for sequential decision making problems.

LMDP: Linearly Solvable MDP: A Markov Deceision Process with special structure for which the Bellman equation can be made linear.

IRL: Inverse Reinforcement Learning: A branch of reinforcement learning that tries to recover a near-optimal policy given demonstrations from an expert for a particular control or reinforcement learning task.

IOC: Inverse Optimal Control: A branch of optimal control theory and machine learning that studies the problem of recover cost functions for optimal control given demonstrations of optimal or near-optimal behavior.

FH: Finite Horizon: Refers to MDPs for which the objective is stagewise costs summed over a finite horizon.

IH: Infinite Horizon Average Costs: Refers to MDPs for which the objective is the limiting average value of stagewise costs over an infinite horizon.

BE: Bellman Equation: This refers to the equation describing the optimal solution to an MDP.

PIC: Path Integral Control: This refers to an approach to stochastic control that exploits the linearity of the HJB PDE to develop sampling approximations based on the Feynman-Kac lemma.

ACKNOWLEDGMENTS

This thesis would not have been possible without the continued support and encouragement of my advisers, Emo and Maryam. I was fortunate to hear about Emo's work and meet him in the middle of my first year before he moved to UW. Serendipitously, we were thinking about the same problem (inverse optimal control) at the time and I naturally got involved in research with him subsequently. Emo has always given me freedom to explore my ideas, and I thank him for his patience and support even at times when my research progress was slow. I met Maryam when I took her course on convex optimization, a course that taught me many of the tools that have been used in deriving the results of this thesis. I wrote the first paper of my PhD with her and have enjoyed having her to talk to about the more theoretical aspects of my research. Both my advisers have been approachable and have generously provided valuable guidance on both technical and non-technical issues.

I thank Misha Chertkov and Scott Backhaus for hosting me at Los Alamos National Labs (LANL) multiple times during my PhD. My first internship at Los Alamos came at a time when I was struggling to make progress on my research, and provided the change and impetus needed for me to get excited about research again. I also got to make valuable contacts through the smart grid conferences and seminar series at Los Alamos (including Prof. Steven Low, my to-be postdoc mentor). I learned a great deal about the challenges of control in power systems, a topic I have continued to work on and intend to focus on during my postdoc. Chapter 6 of this thesis resulted from work I did while at Los Alamos.

I thank Prof. Daniel Kirschen for serving on my committee and providing feedback on my power systems work. He was also responsible for getting me invited to the "Next Generation of Researchers on Power Systems" conference, a conference where I learned a lot and made valuable contacts. I thank Prof. Mehran Mesbahi for serving on my committee and providing valuable feedback on my work on decentralized control of linear systems.

My undergraduate mentors, Professors Soumen Chakrbarti and Subhasis Chaudhuri, were my first research advisers. It is their encouragement and support that made me decide to go to graduate school. The UW CSE department has provided an excellent student-friendly environment for research. I wish to thank Lindsay Michimoto and the CSE staff for prompt and timely help with all administrative issues. My labmates Vikash, Igor, Tom, Yuval, Svet, Akshay, Mingyuan, Mikala, Karthik, Brian, Dennis, Amin and Reza have provided valuable feedback on different parts of my thesis work and on practice talks. Special thanks go to Evangelos Theodorou who was as a mentor and collaborator during his postdoc at UW.

My friends have ensured that I have a life outside of work and provided support and encouragement during difficult times. I have discovered several hobbies during my PhD years, and it can be largely ascribed to being with adventurous friends who pushed me to try new things. I am thankful to them (too many to list here) for all the good times and the amazing trips we've done all over the US. I wish to thank Hindu Yuva, an organization that has taught me much and given me opportunities to interact with amazing people.

I cannot describe in words my gratitude towards my music teacher, Arijit Mahalanabis. I am forever indebted to him for sharing the beautiful art of Dhrupad with me, something I will hold on to and cherish for the rest of my life. It has been a source of joy and comfort that has seen me through the highs and lows of graduate school. Meghann Gerber introduced me to mindfulness meditation. I have immensely benefited from having a regular meditation practice and being part of the meditation group organized by her.

Not a single page of this thesis would have been written without the unconditional love and support of my parents. They taught me the value of knowledge and encouraged me to follow my passions in all aspects of life. The lives of my grandmother (Patti) and uncle (Kannan mama) are a testimony to the adage "simple living and high thinking", a value I have cherished and tried to imbibe in my own life.

DEDICATION

To Amma, Appa, Kannan mama, Patti and Sanath.

Chapter 1

INTRODUCTION

From single-cell organisms to the largest mammals, from nanotechnology to the power grid, there are systems, both natural and man-made, all around us that respond dynamically to changes in their environment. The best responses are often not myopic. In order to achieve a certain goal, a system needs to respond well in advance in a careful manner. Most systems involve repeated interactions and require a constant cycle of gathering information (sensing) and making decisions (controls). Control theory studies repeated interactions in dynamical environments. It aims to provide tools and techniques to analyze the performance of an interaction scheme (a "control policy") and further, to automatically synthesize an interaction scheme to achieve a given objective. This can be abstracted into the well known sense-actuate-control loop, as shown in figure 1.1.

Real-world environments often involve uncertainty, due to modeling imperfections and external disturbances. Stochastic optimal control deals with control in uncertain environments. It is a conceptually elegant framework for specifying control problems. One simply specifies an abstract cost function encoding the control task, and leaves the details of synthesizing control to an optimization algorithm. At a conceptual level, stochastic optimal control can in general be formulated as:

$$\begin{array}{ll} \underset{\text{Controls}}{\text{Minimize}} & \underset{\text{noise}}{\text{E}}\left[\text{Cost}(\text{Trajectory})\right] \\ \text{Subject to} & \text{Trajectory} = \text{Dynamics}\left(\text{Controls}, \text{Noise}\right) \end{array}$$

We give a brief explanation of the terms appearing in the above optimization problem:

1 Trajectory: Most dynamical systems are modeled as having a state, which is a mathematical representation of the variables in the dynamical system sufficient to predict the future evolution of the system in the absence of noise. For example, for a point

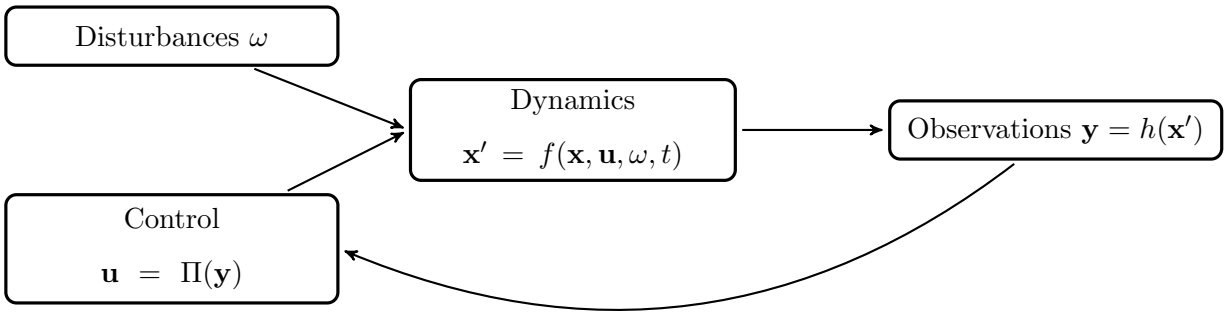


Figure 1.1: General Stochastic Control Problem

mass acting under Newton's laws, the state is given by its (3-dimensional) position and velocity. A trajectory is the sequence of states that a dynamical system visits over a fixed period of interest (the horizon).

2 Cost: This refers to a real valued function of the trajectory that encodes the control task. For example, if we want to control a point mass to remain at the origin, an appropriate cost is a quadratic penalty applied to the position and velocity of the point mass.

3 Controls: This refers to the tunable parameters of the dynamical system that we can modify in order achieve the control objective (minimize the cost function).

4 Noise: This refers to external disturbances, either from un-modeled dynamics, random perturbations or sensor noise.

Because of its elegance and generality, stochastic optimal control has been used in several domains including finance Steele [2001], motor control Todorov [2004b] and robotics Morimoto et al. [2003]. However, a number of issues have impeded the widespread adoption of stochastic optimal control as a practical control design methodology. These include:

1 *Computational Complexity*: This refers to the complexity of solving the optimization problems arising in stochastic optimal control. The only generally applicable technique is dynamic programming, which scales exponentially with the dimensionality of the state space. Tractable special cases are known (Linear-Quadratic-Gaussian (LQG) control) but even small changes to these problems (enforcing decentralization, for example) makes them intractable.

2 *Modeling Dynamical Systems*: Real-systems often have complicated and hard-to-model dynamics. The level of modeling detail required depends on the particular control task and dynamical system: Some control problems may be very sensitive to modeling details while others may not. However, even systematically exploring this trade-off is difficult since solving control problems, even with a fixed model, is difficult.

3 *Designing Costs*: Appropriate cost for control problems is difficult to specify in a mathematically precise manner. Further, simple-to-specify abstract costs (for example, incur a cost unless you reach the goal) are often not amenable to numerical optimization. Further, in an uncertain environment, a trade-off must be made between performance (accomplishing the control task efficiently) and robustness (dealing with perturbations etc.). Achieving the right trade-off again relies on specifying appropriate objective functions, and this can be difficult to get right in many situations.

In this thesis, we describe theoretical and algorithmic advances that go alleviate some of these problems using techniques and tools from recent developments in optimization and stochastic control. The two major tools we will draw upon are:

a The theory of Linearly Solvable Markov Decision Processes (LMDPs) Todorov [2007] and the related framework of Path Integral Control Kappen [2005]. We describe this framework in detail in chapter 2. The essential idea in this framework is the interchangeability of control and noise in a dynamical system. By injecting noise into the control input, one can regard controls as distributions rather than single input. This idea plays a key role in most of this thesis.

b The other set of tools comes from convex optimization. Several applications of convex optimization have sprung up over the years, including control problems Boyd [1994], Dullerud and Paganini [2000]. Further advances have shown that even problems that are not convex can be solved through convex optimization techniques Chandrasekaran et al. [2010]. An overarching theme in this thesis is to adapt techniques from convex optimization to push the boundary on control problems that can be solved efficiently and with provable performance guarantees. This is important because of the degree of automation it brings to the control design process. Alternate heuristic approaches often require careful manual tuning, relying heavily on experience and insight into the control problem being solved. However, once a problem has been convexified (i.e, formulated as a convex optimization problem), tuning is not a major issue and there standard algorithms that are guaranteed to converge to the optimal solution quickly, both in theory and in practice. This makes the adoption of advanced control design methodologies simpler, since the end-user need can simply use them as a black box without worrying about the internal details.

In chapter 3, we study the problem of inverse optimal control, that is, the problem of recovering cost functions for optimal control given demonstrations of optimal or near-optimal behavior. There are several control problems for which the appropriate cost function is difficult to specify. In computer graphics for example, one often desires the motion of animated characters to be human-like. However, simple objective functions generally fail to capture the nuances of human motion. Further, physics laws are not typically enforced explicitly so one needs to indirectly capture "realness" of the motion using an appropriate objective function. Demonstrations of appropriate behavior on the other hand, are generally available more easily (using motion capture data for example). One problem with most prior works on inverse optimal control is that they require repeated solution of the forward problem as a subroutine. This makes them computationally intractable for most continuous-state control problems, since solving optimal control problems in general is hard. This has limited applications of inverse optimal control to discrete domains or required drastic approximations. By exploiting the properties of LMDPs, we develop the first algorithm

that can uniquely recover the objective function from demonstrated behaviors, without having to solve the forward problem as a subroutine. Effectively, the data collected provides information about what the optimal control policy should be which allows us to avoid solving the forward problem as a subroutine. This requires parameterizing the value function (or the cost-to-go function) rather than the value function, but in LMDPs, we show that the cost function can be uniquely recovered from the value function (and vice-versa).

Stochastic control problems naturally have a trade-off between risk and return. In the face of uncertainty, aggressive control strategies may work well under normal conditions, but can fail catastrophically under certain kinds of disturbances. On the other hand, safer strategies have less catastrophic failure but can be painstakingly slow at accomplishing the control task or even fail. Consider the problem of getting a robot lying on the floor to get up and walk to a certain spot. One could consider a very careful motion where the robot drags itself to the destination by passing itself. This strategy would require considerably larger effort than simply getting up and walking to the spot. However, the second strategy carries the risk of tripping and falling, causing catastrophic damage to the robot. This simple example shows that the trade-off between risk and return is critical to achieving desired behavior out of a stochastic control problem. In chapter 4, we extend the framework of linearly solvable MDPs Todorov [2007] to the risk-sensitive setting. This enables us to model risk-sensitivity in a computationally tractable manner while retaining all the elegant properties of the LMDP setting.

In chapter 5, we discuss the problem of tractable control policy synthesis. There are several approaches to dealing with the curse of dimensionality that is the root of the computational intractability of dynamic programming approaches. There is been a lot of research in trying to find approximate solutions to the Bellman equations. LMDPs make this problem easier since the Bellman equation one tries to approximate becomes linear and amenable to classical methods from function approximation. A recent thesis that explored this approach in great detail is Zhong [2013]. While some encouraging results were obtained, in general, approaches that directly try to approximate the solution to the Bellman equation suffer from the fact that a reduction in error in the solution to the Bellman equation (measured by some metric) is not monotonically related to the performance of the control policy

derived from that approximation. In this thesis, we study an alternate approach of directly parameterizing the control policy and searching for the best solution within this parameterized class of controllers using gradient-descent like techniques. This has been studied under the name of "policy gradient" methods Sutton et al. [2000], Baxter and Bartlett [2001], Todorov [2010a] in the context of model-free reinforcement learning. However, in general, the resulting optimization problems are nonconvex and computationally difficult to solve. In this thesis, we propose the first formulation of control policy optimization that leads to convex optimization problems. Combining this with stochastic gradient methods for convex optimization gives us the *first known polytime algorithms* for synthesizing near-optimal policies for arbitrary continuous-state dynamical systems with differentiable dynamics and cost functions.

In chapter 6, we discuss applications of some of these ideas to control of electric power systems. Both of the problems considered are closely related to the framework described in chapter 5, and they formed the inspiration for the problems studied in chapters 5 and 7. Although the algorithms developed in chapters 5 and 7 are not used in the work presented here, the heuristic algorithms developed there work well for the applications: Decentralized Frequency Control in Power Systems and Placement of Energy Storage.

In chapter 7, we outline some ideas for combining the works from chapters 3 and 5 into an integrated cost-shaping and policy synthesis framework. This brings us pretty close to the aim of automating stochastic control: Given a family of relevant costs, data and a parameterized family of control policies, this integrated framework provides algorithms to design a cost function and a control policy using stochastic convex optimization algorithms with guaranteed polytime convergence to a near-optimal solution. Although these ideas are preliminary at the time of writing this thesis and haven't been tested extensively, we believe that they are an important step towards realising the vision of this thesis: Automating Control Design via the framework of stochastic optimal control.

Chapter 2

BACKGROUND & FUNDAMENTALS

This chapter introduces basic concepts from stochastic optimal control. After going over the general theory, we describe the framework of linearly solvable optimal control, which forms the basis of chapters 3 and 4. We end with a description of some ideas from reinforcement learning, which are relevant to the rest of this thesis as well.

2.0.1 Markov Decision Processes (MDPs)

Markov Decision Processes (MDPs) are a widely used framework for specifying and solving optimal control problems. MDPs are formally defined by specifying:

- A state space $\mathcal{X}$. We use $\mathbf{x}$ to denote states, $\mathbf{x} \in \mathcal{X}$. This could be continuous (subset of $\Re^n$), discrete (set of nodes in a graph) or a mixture of both.

- A control space $\mathcal{U}(\mathbf{x})$ for each state. Controls are denoted by $\mathbf{u}$. Policies are mappings from states to controls and denoted by $\Pi(\mathbf{x}) \in \mathcal{U}(\mathbf{x})$. Note that in the reinforcement learning Sutton and Barto [1998b] literature, controls are often called actions.

- A stochastic dynamics $\mathbb{P}(\mathbf{x}, \mathbf{u})$, which is the probability distribution over the next state given the current state $\mathbf{x}$ and action $\mathbf{u} \in \mathcal{U}(\mathbf{x})$.

- An immediate cost function $\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})$.

At any time t, an action $\mathbf{u}$ is chosen depending on the current state and the system transitions into a new state sampled from the stochastic dynamics. The objective of the control is to minimize the expected cost accumulated over time. The precise notion of accumulation can vary, giving rise to different problem formulations as follows:

- Finite Horizon (FH): These problems are specified by a horizon N, a running cost $\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})$ and a terminal cost $\ell_{\mathrm{f}}(\mathbf{x}, \mathbf{u})$. The overall optimization problem can be written as:

$$\min_{\{\Pi_t(\mathbf{x})\}} \mathop{\mathrm{E}}_{\mathbf{x}_{t+1} \sim \mathbb{P}(\mathbf{x}_t, \Pi_t(\mathbf{x}))} \left[\left(\sum_{t=0}^{\mathrm{N}-1} \ell_{\mathrm{t}} \left(\mathbf{x}_t, \Pi_t \left(\mathbf{x}_t \right) \right) \right) + \ell_{\mathrm{f}} \left(\mathbf{x}_{\mathrm{N}} \right) \right]. \tag{2.1}$$

  Note that here policies are indexed by time, since the optimal policy for a finite horizon problem is always time-varying. For all other problems listed below, it suffices to consider time-invariant policies (these can be shown to be optimal).

- First exit (FE) problems are specified by a set of terminal states $\mathcal{T} \subset \mathcal{X}$, a running cost $\ell(\mathbf{x}, \mathbf{u})$ and a terminal cost $\ell_{\mathrm{f}} : \mathcal{T} \to \mathbf{R}$. The objective is given by:

$$\mathop{\mathrm{E}}_{\mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \Pi(\mathbf{x})),\, \mathrm{N_e} = \min\{t:\, \mathbf{x}_t \in \mathcal{T}\}} \left[\left(\sum_{t=0}^{\mathrm{N_e}} \ell \left(\mathbf{x}_t, \Pi \left(\mathbf{x}_t \right) \right) \right) + \ell_{\mathrm{f}} \left(\mathbf{x}_{\mathrm{N_e}} \right) \right]$$

  Here the end-time $\mathrm{N_e}$ is also a random variable - it refers to the first time step at which the state $\mathbf{x}_t$ is a terminal state. Thus, the expectation is with respect to the stochastic dynamics and the end-time.

- Infinite Horizon Average Cost (IH) problems are specified just by a running cost $\ell(\mathbf{x}, \mathbf{u})$ and the objective is the limiting average cost:

$$\mathop{\mathrm{E}}_{\mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \Pi(\mathbf{x}))} \left[\lim_{\mathrm{N} \to \infty} \left(\frac{\sum_{t=0}^{\mathrm{N}} \ell \left(\mathbf{x}_t, \Pi \left(\mathbf{x}_t \right) \right)}{\mathrm{N}} \right) \right]$$

- Infinite Horizon Discounted Cost problems are specified by a running cost $\ell(\mathbf{x}, \mathbf{u})$ and a discount factor γ. The objective is:

$$\mathop{\mathrm{E}}_{\mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \Pi(\mathbf{x}))} \left[\lim_{\mathrm{N} \to \infty} \left(\sum_{t=0}^{\mathrm{N}} \gamma^t \, \ell \left(\mathbf{x}_t, \Pi \left(\mathbf{x}_t \right) \right) \right) \right]$$

We do not go into the details of the Bellman equations for the different formulations here. Instead, we focus on the simplest case of finite horizon (FH) problems. The optimal cost-to-go function (or optimal value function) $v_t\left(\mathbf{x}\right)$ is defined as the expected cumulative cost for

starting at state $\mathbf{x}$ at time t and acting optimally thereafter. This function is characterized by the Bellman equation (BE) Bellman [1957]:

$$v_t(\mathbf{x}) = \min_{\mathbf{u}} \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) + \mathop{\mathrm{E}}_{\mathbb{P}(\mathbf{x},\mathbf{u})}[v_{t+1}], \tag{2.2}$$

$$\Pi_t^*(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{u}} \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) + \mathop{\mathrm{E}}_{\mathbb{P}(\mathbf{x},\mathbf{u})}[v_{t+1}]$$

where $\mathrm{E}_{\mathbb{P}(\mathbf{x},\mathbf{u})}[v_{t+1}] = \mathrm{E}_{\mathbf{x}' \sim \mathbb{P}(\mathbf{x},\mathbf{u})}[v_{t+1}(\mathbf{x}')]$. $\Pi_t^*(\cdot)$ is called the *optimal policy* and is the solution to the optimization problem (2.1). The Bellman equation has an intuitive meaning. $v_{t+1}(\mathbf{x})$ represents the minimum accumulated cost starting at state $\mathbf{x}$ at time $t+1$. Thus, $v_t(\mathbf{x})$ must be the minimum over immediate actions $\mathbf{u}$ of the sum of the immediate cost $\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})$ and the minimum accumulated cost starting at the next state $\mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \mathbf{u})$, which is precisely $v_{t+1}(\mathbf{x}')$. Since transitions are probabilistic and objectives are measured in expectation, we take the expected value $\mathrm{E}_{\mathbf{x}' \sim \mathbb{P}(\mathbf{x},\mathbf{u})}[v_{t+1}(\mathbf{x}')]$. The Bellman equation approach to solving optimal control is also called *Dynamic Programming*, since its an optimization method that recursively constructs a solution backward in time.

Although the dynamic programming approach is an elegant and general solution to stochastic optimal control problems, for most problems of practical interest, solving the Bellman equation is computationally intractable. This is because one needs to store the value function at each state $\mathbf{x}$ and the number of states could be very large (infinite if $\mathcal{X}$ is a continuous domain). This has led to a variety of approximation schemes. Many of these rely on solving the BE approximately. However, getting such schemes to work often requires a lot of problem-specific tuning, and even then may not scale to genuinely hard problems. Part of the difficulty is the highly nonlinear nature of the BE which is a result of the $\min_{\mathbf{u}}$ term. A key advantage of linearly-solvable MDPs (see below) is that the minimization over actions can be done analytically given the value function. The minimized Bellman equation can then be made linear by exponentiating the value function.

2.1 Linearly Solvable Optimal Control Problems

2.1.1 Probability shift: A an alternative view of control

Conventionally, we think of control signals as quantities that modify the system behavior in some pre-specified manner. In our framework it is more convenient to work with a somewhat different notion of control, which is nevertheless largely equivalent to the conventional notion, allowing us to model problems of practical interest. To motivate this alternative view, consider a control-affine diffusion:

$$d\mathbf{x} = (\mathrm{a}(\mathbf{x}) + \mathrm{B}(\mathbf{x})\mathbf{u})\, dt + \mathrm{C}(\mathbf{x})d\omega$$

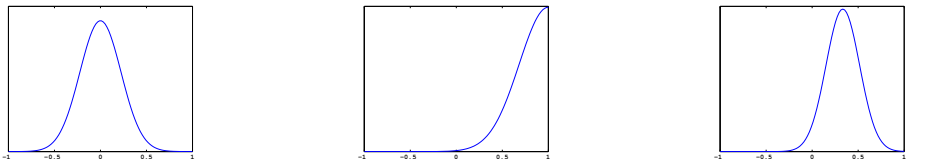
This is a stochastic differential equation specifying the infinitesimal change in the state $\mathbf{x}$, caused by a passive/uncontrolled drift term $\mathrm{a}(\mathbf{x})$, a control input $\mathbf{u}$ scaled by a control gain $\mathrm{B}(\mathbf{x})$, and Brownian motion noise with amplitude $\mathrm{C}(\mathbf{x})$. Subject to this system dynamics, the controller seeks to minimize a cost function of the form

$$\ell(\mathbf{x}) + \frac{1}{2}\mathbf{u}^T\mathbf{u}$$

In terms of MDPs, the transition probability may be written as

$$\mathbb{P}(\mathbf{x}, \mathbf{u}) = \mathcal{N}(\mathbf{x} + \delta(\mathrm{a}(\mathbf{x}) + \mathrm{B}(\mathbf{x})\mathbf{u}), \Sigma)$$

where we have discretized time using a time step δ. Thus, one way of thinking of the effect of control is that it changes the distribution of the next state from $\mathcal{N}(\mathbf{x} + \delta a(\mathbf{x}), \Sigma)$ to $\mathcal{N}(\mathbf{x} + \delta(\mathrm{a}(\mathbf{x}) + \mathrm{B}(\mathbf{x})\mathbf{u}), \Sigma)$. In other words, the controller shifts probability mass from one region of the state space to another. More generally, we can think of the system as having an uncontrolled dynamics which gives a distribution p over future states. The controller acts by modifying this distribution by probability shift to get a new distribution: $u \otimes p = \frac{pu}{\mathrm{E}_p[u]}$. This causes the probability mass in p to shift towards areas where u is large (figure 2.1). The controllers in our framework will act on the system dynamics by performing such probability shifts. The control signals will be positive scalar functions over the state space, rather than vectors or discrete symbols.



(a) p (b) u (c) $u \times p$

Figure 2.1: Probability Shift

2.1.2 Linearly-solvable Markov Decision Processes (LMDPs)

Here we introduce the framework of linearly-solvable optimal control in discrete time. Such problems, called LMDPs, can be viewed in two mathematically equivalent ways. We shall describe both, since they both offer useful perspectives and illustrate the relationship to traditional MDPs in complementary ways.

In traditional MDPs the controller chooses a control signal or action $\mathbf{u}$ which determines the distribution of the next state $\mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \mathbf{u})$. In LMDPs, we assume that there is an uncontrolled or passive dynamics $\Pi^0(\mathbf{x})$ for each state $\mathbf{x}$ that gives the distribution of the next state. The controller can change this distribution by picking a probability shift $u \in \mathcal{X}^{\mathbf{R}^+}$. This causes the distribution of the next state to change: $\mathbf{x}' \sim u \otimes \Pi^0(\mathbf{x})$. However, the controller must pay a price for doing so, given by the KL divergence between the controlled distribution $u \otimes \Pi^0(\mathbf{x})$ and the uncontrolled distribution $\Pi^0(\mathbf{x})$, which is a measure of the amount of change in the dynamics due to the controller. The Bellman equation for LMDPs is nonlinear in terms of the value function, but using an exponential transformation $z_t = \exp(-v_t)$ yields a linear equation in z. We call this the desirability function, since it is inversely related to the cost-to-go. The desirability function also gives the optimal shift

policy $\Pi_t^*(\mathbf{x}) = z_{t+1}$, so the optimal controller is always trying to shift the uncontrolled dynamics towards more desirable states. The key results and their analogs for traditional MDPs are summarized in the following table:

	MDPs	LMDPs
Policy	$\Pi : \mathcal{X} \to \mathcal{U}$	$\Pi : \mathcal{X} \to \mathcal{X}^{\mathbf{R}^+}$
Dynamics	$\mathbf{x} \xrightarrow{\Pi} \mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \Pi(\mathbf{x}))$	$\mathbf{x} \xrightarrow{\Pi} \mathbf{x}' \sim \Pi(\mathbf{x}) \otimes \Pi^0(\mathbf{x})$
Cost	$\ell_{\mathrm{t}}(\mathbf{x}, \Pi(\mathbf{x}))$	$\ell_{\mathrm{t}}(\mathbf{x}) +$ $\mathrm{KL}\left(\Pi(\mathbf{x}) \otimes \Pi^0(\mathbf{x}) \parallel \Pi^0(\mathbf{x})\right)$
Bellman Equation	$v_t(\mathbf{x}) = \min_{\mathbf{u}} \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) + \mathop{\mathrm{E}}_{\mathbb{P}(\mathbf{x}, \Pi(\mathbf{x}))}[v_{t+1}]$	$z_t(\mathbf{x}) = \exp(-\ell_{\mathrm{t}}(\mathbf{x})) \mathop{\mathrm{E}}_{\Pi^0(\mathbf{x})}[z_{t+1}]$
Optimal Policy	$\Pi_t^*(\mathbf{x}) =$ $\operatorname{argmin}_{\mathbf{u}} \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) + \mathop{\mathrm{E}}_{\mathbb{P}(\mathbf{x}, \Pi(\mathbf{x}))}[v_{t+1}]$	$\Pi_t^*(\mathbf{x}) = z_{t+1}$

2.1.3 An alternate view of LMDPs

In the alternate view, LMDPs are almost the same as traditional MDPs with deterministic dynamics and stochastic policies, except for two differences: we impose an additional cost that encourages policies with high entropy, and we compute the cost based not on the action that happened to be sampled from the stochastic policy, but by taking an expectation over all actions that could have been sampled. In this view, the relation between traditional deterministic MDPs and LMDPs is summarized as:

	Deterministic MDPs with Stochastic Policies	LMDPs
Policy	$\Pi : \mathcal{X} \to \mathcal{P}\left[\mathcal{U}\right]$	$\Pi : \mathcal{X} \to \mathcal{P}\left[\mathcal{U}\right]$
Dynamics	$\mathbf{u} \sim \Pi(\mathbf{x})$ $\mathbf{x}' = \mathrm{f}(\mathbf{x}, \mathbf{u})$	$\mathbf{u} \sim \Pi(\mathbf{x})$ $\mathbf{x}' = \mathrm{f}(\mathbf{x}, \mathbf{u})$
Cost	$\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})$	$\underset{\mathbf{u} \sim \Pi(\mathbf{x})}{\mathrm{E}} \left[\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})\right] - H(\Pi(\mathbf{x}))$
Bellman Equation	$v_t\left(\mathbf{x}\right) =$ $\min\limits_{\mathbf{u} \sim \Pi(\mathbf{x})} \mathrm{E}\left[\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) + v_{t+1}\left(\mathrm{f}(\mathbf{x}, \mathbf{u})\right)\right]$	$z_t\left(\mathbf{x}\right) =$ $\sum\limits_{\mathbf{u}} \exp\left(-\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})\right) z_{t+1}\left(\mathrm{f}(\mathbf{x}, \mathbf{u})\right)$
Optimal Policy	$\Pi_t^*\left(\mathbf{x}\right) = \delta(\mathbf{u}^*)$ $\mathbf{u}^* =$ $\underset{\mathbf{u}}{\mathsf{argmin}}\, \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) + v_{t+1}\left(\mathrm{f}(\mathbf{x}, \mathbf{u})\right)$	$\Pi_t^*\left(\mathbf{x}\right) = z_{t+1}$

We can rewrite the BE for LMDPs in this interpretation as:

$$v_t\left(\mathbf{x}\right) = -\log\left(\sum_{\mathbf{u}} \exp\left(-\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}) - v_{t+1}\left(\mathrm{f}(\mathbf{x}, \mathbf{u})\right)\right)\right)$$

The relationships between MDPs and LMDPs is now clear: the hard minimum in the Bellman equation for MDPs is replaced by a soft minimum for LMDPs, namely $-\log(\sum(\exp(-\ldots)))$. If we replace the cost $\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})$ by a scaled version $\gamma\, \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})$, as γ increases we move closer and closer to the hard minimum, and in the limit $\gamma \to \infty$ we recover the Bellman equation for MDPs. Thus any deterministic MDP can be obtained as a limit of LMDPs.

The relationship between the two interpretations can be understood as follows. Define a passive dynamics with support only on the states immediately reachable from $\mathbf{x}$ under some action $\mathbf{u}$:

$$\Pi^0(\mathrm{f}(\mathbf{x}, \mathbf{u})|\mathbf{x}) \propto \exp\left(-\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u})\right)$$

For states not immediately reachable from $\mathbf{x}$, the probability under the passive dynamics is 0. Given any control (probability shift) $\Pi \in \mathcal{X}^{\mathbf{R}^+}$, we have:

$$\begin{aligned}\mathrm{KL}\left(\Pi \otimes \Pi^0(\mathbf{x}) \parallel \Pi^0(\mathbf{x})\right) &= -H\left[\Pi \otimes \Pi^0(\mathbf{x})\right] + \mathop{\mathrm{E}}_{\Pi \otimes \Pi^0(\mathbf{x})}\left[-\log\left(\Pi^0(\mathbf{x})\right)\right] \\ &= -H\left[\Pi \otimes \Pi^0(\mathbf{x})\right] + \mathop{\mathrm{E}}_{\Pi \otimes \Pi^0(\mathbf{x})}\left[\ell_{\mathrm{t}}(\mathbf{x},\mathbf{u})\right] - \ell_{\mathrm{t}}(\mathbf{x})\end{aligned}$$

where $\ell_{\mathrm{t}}(\mathbf{x}) = -\log\left(\sum_{\mathbf{u}} \exp\left(-\ell_{\mathrm{t}}(\mathbf{x},\mathbf{u})\right)\right)$. Thus, the alternate interpretation is equivalent to the original interpretation with passive dynamics proportional to $\exp\left(-\ell_{\mathrm{t}}(\mathbf{x},\mathbf{u})\right)$ and cost function $-\log\left(\sum_{\mathbf{u}} \exp\left(-\ell_{\mathrm{t}}(\mathbf{x},\mathbf{u})\right)\right)$.

2.1.4 Other Problem Formulations

Thus far we focused on the FH problem formulation. We can obtain linearly-solvable problems with other problem formulations as well. The corresponding BEs are

$$\begin{aligned}\mathrm{FE}z\left(\mathbf{x}\right) &= \exp\left(-\ell(\mathbf{x})\right) \mathop{\mathrm{E}}_{\Pi^0(\mathbf{x})}\left[z\right] \text{ if } \mathbf{x} \notin \mathcal{T} \\ z\left(\mathbf{x}\right) &= \exp\left(-\ell_{\mathrm{f}}(\mathbf{x})\right) \text{ if } \mathbf{x} \in \mathcal{T}\end{aligned} \tag{2.3}$$

$$\mathrm{IH}z\left(\mathbf{x}\right) = \exp\left(c - \ell(\mathbf{x})\right) \mathop{\mathrm{E}}_{\Pi^0(\mathbf{x})}\left[z\right] \tag{2.4}$$

c is the Optimal Average Cost

In the IH case the linear BE becomes an eigenvalue problem, with eigenvalue $\exp(-c)$ where c is the average cost. It can be shown that the solution to the optimal control problem corresponds to the principal eigenpair. The optimal policy in both cases is given by

$$\Pi^*\left(\mathbf{x}\right) = z \tag{2.5}$$

2.1.5 Applications

We now give some examples of how commonly occurring control problems can be modeled as LMDPs.

Shortest paths: Consider the shortest path problem defined on a graph. We can view this as an MDP with nodes corresponding to states and edges corresponding to actions. A stochastic version of this problem is one where the action does not take you directly where you intend, but possibly to the end of one of the other outgoing edges from that node. We can define an LMDP with passive dynamics at a node to be the uniform distribution over all nodes reachable in one step. The cost is a constant cost per unit time and the problem is a FE problem with the goal state as the state to which the shortest path is being computed. By scaling up the constant cost by ρ, in the limit as $\rho \to \infty$ we recover the traditional deterministic shortest paths problem. This yields an efficient approximation algorithm for the shortest paths problem, by solving an LMDPs with sufficiently large ρ, see Todorov [2009b].

Discretizing continuous problems: We can construct efficient solutions to problems with continuous state spaces and continuous time, provided the state space can be discretized to a reasonable size (LMDPs can easily handle problems with millions of discrete states). We consider a simple problem that has been a standard benchmark in the Reinforcement Learning literature, the mountain-car problem. In this problem, the task is to get a car to drive down from a hill into a valley and park on another hill on the other side of the valley. The control variable is the acceleration of the car, and the state consists of the position and velocity of the car. We impose limits on all these quantities and discretize the state space to within those limits. The dynamics is completely determined by gravity and the shape of the hill. We plot results in figure 2.2 comparing the LMDP discretization and a iterative solution of the LMDP to a standard MDP discretization and using policy/value iteration to solve that. It can be seen that the LMDP solution converges faster to the optimal policy. See Todorov [2009b].

2.1.6 Linearly-solvable controlled diffusions (LDs)

Although the focus of this chapter is on discrete-time problems (i.e. LMDPs), here we summarize related results in continuous time. The linearly-solvable optimal control problems

in continuous time are control-affine diffusions with dynamics

$$\mathrm{d}\,\mathbf{x} = \mathrm{a}(\mathbf{x})\,\mathrm{d}\,t + \mathrm{B}(\mathbf{x})\mathbf{u}\,\mathrm{d}\,t + \sigma\,\mathrm{B}(\mathbf{x})\,\mathrm{d}\,\omega$$

and cost rate

$$\ell_{\mathrm{t}}(\mathbf{x}) + \frac{1}{2\sigma^2}\,\|\mathbf{u}\|^2$$

The unusual aspects of this problem are that: (i) the noise and the control act in the same subspace spanned by the columns of $\mathrm{B}(\mathbf{x})$; (ii) the control cost is scaled by σ^{-2}, thus increasing the noise in the dynamics makes the controls cheaper.

For problems in this class one can show that the optimal control law is

$$\Pi_t^*\left(\mathbf{x}\right) = \frac{\sigma^2}{z_t\left(\mathbf{x}\right)}\mathrm{B}(\mathbf{x})^T\frac{\partial z_t\left(\mathbf{x}\right)}{\partial \mathbf{x}}$$

and the Hamilton-Jacobi-Bellman (HJB) equation expressed in terms of z becomes linear and is given by

$$\frac{\partial z_t\left(\mathbf{x}\right)}{\partial t} = \ell_{\mathrm{t}}(\mathbf{x})\,z_t\left(\mathbf{x}\right) - \mathcal{L}\left[z_t\right]\left(\mathbf{x}\right) \tag{2.6}$$

Here $\mathcal{L}$ is a 2nd-order linear differential operator known as the generator of the passive dynamics:

$$\mathcal{L}\left[f\right]\left(\mathbf{x}\right) = a(\mathbf{x})^T\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + \frac{\sigma^2}{2}\mathrm{tr}\left(\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^T}\,\mathrm{B}(\mathbf{x})\mathrm{B}(\mathbf{x})^T\right) \tag{2.7}$$

This operator computes expected directional derivatives of functions along trajectories of the passive dynamics. We call problems of this kind linearly solvable controlled diffusions (LDs).

2.1.7 Relationship between discrete and continuous-time problems

If we take the first view of LMDPs that uses the notion of a stochastic passive dynamics, we can interpret the above linearly solvable diffusion as a continuous-time limit of LMDPs. This can be done by discretizing the time axis of the diffusion process with time step h using the Euler approximation:

$$\mathbf{x}(t+h) = \mathbf{x}(t) + h\,\mathrm{a}(\mathbf{x}) + h\,\mathrm{B}(\mathbf{x})\mathbf{u} + \epsilon$$

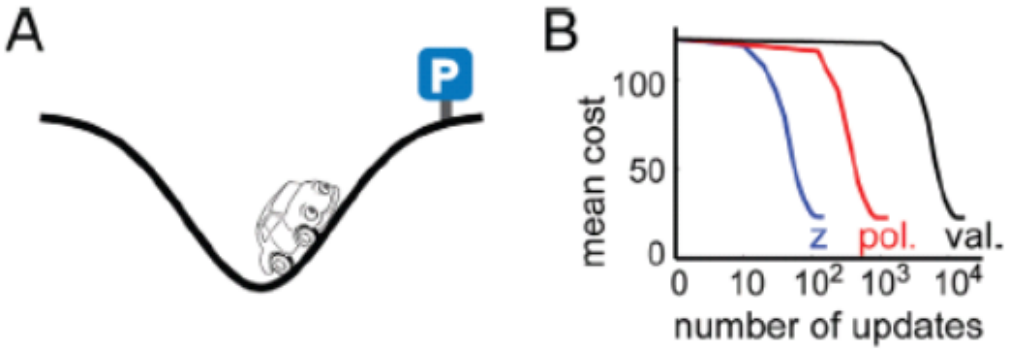


Figure 2.2: Continuous problems. Comparison of our MDP approximation and a traditional MDP approximation on a continuous car-on-a-hill problem. (A) Terrain, (B) Z iteration (ZI) (blue), policy iteration (PI) (red), and value iteration (VI) (black) converge to control laws with identical performance; ZI is 10 times faster than PI and 100 times faster than VI. Horizontal axis is on log-scale. (C) Optimal cost-to-go for our approximation. Blue is small, red is large. The two black curves are stochastic trajectories resulting from the optimal control law. The thick magenta curve is the most likely trajectory of the optimally controlled stochastic system. (D) The optimal cost-to-go is inferred from observed state transitions by using our algorithm for inverse optimal control. Figure taken from Todorov [2009b].

where $\epsilon \sim \mathcal{N}\left(0, h\sigma^2\,\mathrm{B}(\mathbf{x})\mathrm{B}(\mathbf{x})^T\right)$. The covariance is scaled by h since for Brownian noise the standard deviation grows as the square root of time. The discrete-time cost becomes $h\,\ell_{\mathrm{t}}(\mathbf{x}) + h\frac{1}{2\sigma^2}\mathbf{u}^T\mathbf{u}$. We will now construct an LMDP that resembles this time-discretized LD. To do this, we define the passive dynamics at state $\mathbf{x}$ to be the Euler approximation of the distribution of $\mathbf{x}(t+h)$ given $\mathbf{x}(t) = \mathbf{x}$:

$$\Pi^0(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} + h\,\mathrm{a}(\mathbf{x}), h\sigma^2\,\mathrm{B}(\mathbf{x})\mathrm{B}(\mathbf{x})^T\right).$$

This converges to the continuous time LD dynamics with $\mathbf{u} = 0$ as $h \to 0$. Now, consider a family of probability shifts $u^{\mathbf{u}}$ parameterized by $\mathbf{u}$ such that

$$u^{\mathbf{u}} \otimes \Pi^0(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} + h\,\mathrm{a}(\mathbf{x}) + h\,\mathrm{B}(\mathbf{x})\mathbf{u}, h\sigma^2\,\mathrm{B}(\mathbf{x})\mathrm{B}(\mathbf{x})^T\right).$$

This distribution is the Euler discretization of the LD dynamics under control $\mathbf{u}$. It can be shown that $\mathrm{KL}\left(u^{\mathbf{u}} \otimes \Pi^0(\mathbf{x}) \parallel \Pi^0(\mathbf{x})\right) = h\frac{1}{2\sigma^2}\mathbf{u}^T\mathbf{u}$. Thus, for every $\mathbf{u}$, there is a probability shift $u^{\mathbf{u}}$ that matches the Euler approximation of the LD dynamics under control $\mathbf{u}$ and also matches the time-discretized control cost. We define the state cost to be $h\,\ell_{\mathrm{t}}(\mathbf{x})$. This LMDP is very close to the MDP corresponding to the time discretized LD, the only difference being that we allow probability shifts that are not equal to $u^{\mathbf{u}}$ for any $\mathbf{u}$. However, it turns out that this extra freedom does not change the optimal control law, at least in the limit $h \to 0$. The BE corresponding to this LMDP is:

$$z_t(\mathbf{x}) = \exp\left(-h\,\ell_{\mathrm{t}}(\mathbf{x})\right) \mathop{\mathrm{E}}_{\mathcal{N}\left(\mathbf{x}+h\,\mathrm{a}(\mathbf{x}), h\sigma^2\,\mathrm{B}(\mathbf{x})\mathrm{B}(\mathbf{x})^T\right)}\left[z_{t+h}\right]$$

It can be shown that after some algebra and taking the limit $h \to 0$, we recover the linear HJB equation (2.6).

2.2 Properties and algorithms

2.2.1 Sampling approximations and path-integral control

For LMDPs , it can be shown that the FH desirability function equals the expectation

$$z_0(\mathbf{x}_0) = \mathop{\mathrm{E}}_{\mathbf{x}_{t+1}\sim\Pi^0(\mathbf{x}_t)}\left[\exp\left(-\ell_{\mathrm{f}}(\mathbf{x}_T) - \sum\nolimits_{t=1}^{T-1}\ell_{\mathrm{t}}(\mathbf{x}_t)\right)\right] \tag{2.8}$$

over trajectories $\mathbf{x}_1 \cdots \mathbf{x}_T$ sampled from the passive dynamics starting at $\mathbf{x}_0$. This is also known as a path-integral. It was first used in the context of linearly-solvable controlled diffusions Kappen [2005] to motivate sampling approximations. This is a model-free method for Reinforcement Learning Sutton and Barto [1998a], however unlike Q-learning (the classic model-free method) which learns a Q-function over the state-action space, here we only learn a function over the state space. This makes model-free learning in the LMDP setting much more efficient Todorov [2009b].

One could sample directly from the passive dynamics, however the passive dynamics are very different from the optimally-controlled dynamics that we are trying to learn. Faster convergence can be obtained using importance sampling:

$$z_0(\mathbf{x}_0) = \underset{\mathbf{x}_{t+1} \sim \Pi^1(\cdot|\mathbf{x})}{\mathrm{E}} \left[\exp\left(-\ell_{\mathrm{f}}(\mathbf{x}_T) - \sum\nolimits_{t=1}^{T-1} \ell_{\mathrm{t}}(\mathbf{x}_t) \right) \frac{\Pi^0(\mathbf{X}|\mathbf{x}_0)}{\Pi^1(\mathbf{X}|\mathbf{x}_0)} \right]$$

Here $\Pi^1(\mathbf{x}_{t+1}|\mathbf{x}_t)$ is a proposal distribution and $\Pi^0(\mathbf{X}|\mathbf{x}_0), \Pi^1(\mathbf{X}|\mathbf{x}_0)$ denote the trajectory probabilities under Π^0, Π^1. The proposal distribution would ideally be Π^*, the optimally controlled distribution, but since we do not have access to it, we use the approximation based on our latest estimate of the function z. We have observed that importance sampling speeds up convergence substantially Todorov [2009b]. Note however that in order to evaluate the importance weights $\Pi^1(\cdot)/\Pi^0(\cdot)$, one needs a model of the passive dynamics. In Theodorou et al. [2010a], the authors develop an iterative version of this algorithm called policy improvement with path integrals (PI^2).

2.2.2 Natural policy gradient

The residual in the Bellman equation is not monotonically related to the performance of the corresponding control law. Thus many researchers have focused on policy gradient methods that optimize control performance directly Williams [1992], Sutton et al. [2000], J. and S. [2008]. The remarkable finding in this literature is that, if the policy is parameterized linearly and the Q-function for the current policy can be approximated, then the gradient of the average cost is easy to compute.

Within the LMDP framework, we have shown Todorov [2010a] that the same gradient can be computed by estimating only the value function. This yields a significant improve-

ment in terms of computational efficiency. The result can be summarized as follows. Let $g(\mathbf{x})$ denote a vector of bases, and define the control law

$$\Pi^{(s)}(\mathbf{x}) = \exp\left(-s^{\mathsf{T}} g(\mathbf{x})\right)$$

This coincides with the optimal control law when $s^{\mathsf{T}} g(\mathbf{x})$ equals the optimal value function $v(\mathbf{x})$. Now let $v^{(s)}(\mathbf{x})$ denote the value function corresponding to control law $\Pi^{(s)}$, and let $v(\mathbf{x}) = r^{\mathsf{T}} g(\mathbf{x})$ be an approximation to $v(\mathbf{x})$, obtained by sampling from the optimally controlled dynamics $u^{(s)} \otimes \Pi^0$ and following a procedure described in Todorov [2010a]. Then it can be shown that the natural gradient Amari [1998] of the average cost with respect to the Fisher information metric is simply $s - r$. Note that these results do not extend to the LMG case since the policy-specific Bellman equation is nonlinear in this case.

2.2.3 Compositionality of optimal control laws

One way to solve hard control problems is to use suitable primitives Precup et al. [1998], Mahadevan and Maggioni [2007]. The only previously known primitives that preserve optimality were Options Precup et al. [1998], which provide temporal abstraction. However what makes optimal control hard is space rather than time, i.e. the curse of dimensionality. The LMDP framework for the first time provided a way to construct spatial primitives, and combine them into provably-optimal control laws Todorov [2009a], Da Silva et al. [2009]. This result is specific to FE and FH formulations. Consider a set of LMDPs (indexed by k) which have the same dynamics and running cost, and differ only by their final costs ${\ell_{\mathrm{f}}}^{(k)}(\mathbf{x})$. Let the corresponding desirability functions be $z^{(k)}(\mathbf{x})$. These will serve as our primitives. Now define a new (composite) problem whose final cost can be represented as

$${\ell_{\mathrm{f}}}(\mathbf{x}) = -\log\left(\sum_k w_k \exp\left(-{\ell_{\mathrm{f}}}^{(k)}(\mathbf{x})\right)\right)$$

for some constants w_k. Then the composite desirability function is

$$z(\mathbf{x}) = \sum\nolimits_k w_k \, z^{(k)}(\mathbf{x})$$

and composite optimal control law is

$$\Pi^*(\mathbf{x}) = \sum\nolimits_k w_k \, \Pi^{*(k)}(\mathbf{x})$$

One application of these results is to use LQG primitives – which can be constructed very efficiently by solving Riccati equations. The composite problem has linear dynamics, Gaussian noise and quadratic cost rate, however the final cost no longer has to be quadratic. Instead it can be the log of any Gaussian mixture. This represents a substantial extension to the LQG framework. These results can also be applied in infinite-horizon problems where they are no longer guaranteed to yield optimal solutions, but nevertheless may yield good approximations in challenging tasks such as those studied in Computer Graphics Da Silva et al. [2009].

Chapter 3

DESIGNING COSTS: INVERSE OPTIMAL CONTROL

Inverse optimality has attracted considerable attention in both control engineering and machine learning. Unlike the forward problem of optimal control which is well-defined, the inverse problem can be posed in multiple ways serving different purposes.

Inverse optimality was first studied for control-theoretic purposes in relation to stability Kalman [1964]. This idea later inspired a constructive approach (e.g. Deng and Krstic [1997]) where one designs a control-Lyapunov function, treats it as an optimal value function (and derives the corresponding control law) and finds the cost for which this value function is optimal. Apart from the guesswork involved in designing control-Lyapunov functions, this is easier than solving the forward problem because for many nonlinear systems (see Section 3) the Hamilton-Jacobi-Bellman (HJB) equation gives an explicit formula for the cost once the value function is known. Linearly Solvable MDPs (LMDPs) we will be working with Todorov [2007, 2009b] also have this property, and it will play a key role here.

It is notable that the above control-theoretic approach does not actually use data. In contrast, Inverse Reinforcement Learning (IRL) methods in machine learning rely on data in the form of state transitions (and possibly actions) obtained from an expert performing some task. In general there are two things that one could do with such data: infer the costs/values of the expert, or build a controller which mimics the expert. The former is relevant to cognitive and neural science, where researchers are interested in "theories of mind" Baker et al. [2007] as well as in identifying the cost functions being optimized by the sensorimotor system Todorov [2004a], Kording and Wolpert [2004]. While many existing IRL algorithms Ng and Russell [2000], Abbeel and Ng [2004], Syed et al. [2008] use cost features and infer weights for those features, they do not actually aim to recover the cost or value function but only the control law. Indeed in generic MDPs there is a continuum of cost and value functions for which a given control law is optimal Ng and Russell [2000].

This ill-posedness is removed in the LMDP framework, making our algorithms much more applicable to cognitive and neural science.

Now consider the task of building a control law from data – which is what the above IRL methods do. One reason to use data (instead of solving the forward problem directly) is that an appropriate cost function which captures the control objectives may be hard to design. But we believe this difficulty is negligible compared to the second reason – which is that we lack algorithms capable of solving forward optimal control problems for complex systems. Take for example the control domain where data has been used most extensively, namely locomotion and other full-body movements seen in movies and games. A sensible cost function for locomotion is not hard to design: it should require the center of mass to remain a certain distance above ground (to prevent falling), move at a certain speed towards the goal, and at the same time conserve energy. Indeed open-loop optimization of similar costs for simplified models can predict various features of human walking and running Srinivasan and Ruina [2006]. If we could find feedback control laws which optimize such costs for realistic systems, this would constitute a major breakthrough both in animation and in robotics. Unfortunately this is not yet possible, and thus many researchers are exploring ways to build controllers using motion capture data (e.g. Treuille et al. [2007]). We emphasize this point here because all prior IRL methods we are aware of, including the MaxEntIRL method discussed later Ziebart et al. [2008a], end up solving the forward problem repeatedly in an inner loop. While one can construct problems with moderate numbers of discrete states where such an approach is feasible, scaling it to control problems that involve interesting physical systems is unlikely. A case in point is the elegant work of Abbeel and colleagues on aerobatic helicopter flight Abbeel et al. [2007]. After trying to apply their apprenticeship learning framework Abbeel and Ng [2004] to this problem, they eventually gave up and simply recorded reference trajectories from human radio-pilots. If future IRL algorithms are to avoid this fate, they should avoid solving the forward problem. Here we develop the first inverse method which avoids solving the forward problem. This is done by parameterizing and inferring the value function rather than the cost function, and then computing the cost function using an explicit formula.

Finally, all IRL methods including ours use linear combinations of features to represent

the costs (or values in our case). However, previous work has left the choice of features to manual design. This is arguably one of the biggest unsolved problems not only in IRL but in AI and machine learning in general. Here we consider automatic methods for initializing the parameterized features and methods for adapting their parameters. When the number of features is small relative to the size of the state space (which is always the case in high dimensional problems), feature adaptation turns out to be essential. While the problem of feature adaptation in IRL is far from being solved in its generality, the present work is an important first step in this direction.

3.1 Discrete problems

We consider problems with discrete state space in this section, and problems with continuous state space in the next section. In both cases we derive IRL algorithms from the recently-developed framework of linearly-solvable stochastic optimal control Todorov [2007, 2009b], Kappen [2005].

3.1.1 Parameterizing the value function (OptV)

Unlike prior IRL algorithms which require trajectory data, our algorithms work with any dataset of transitions $\{\mathbf{x}_n, \mathbf{x}'_n\}_{n=1 \cdots N}$ sampled from the optimal control law:

$$\mathbf{x}'_n \sim \Pi^* (\mathbf{x}_n) \otimes \Pi^0(\mathbf{x}_n) \tag{3.1}$$

We are also given the passive dynamics Π^0. Our objective is to estimate the cost ℓ, the desirability function z, the optimal value function v and the optimal control law Π^*. Conveniently we have explicit formulas relating these quantities, thus it is sufficient to infer one of them. For reasons explained below it is most efficient to infer v. Once we have an estimate $\widehat{v}$, we can obtain $\widehat{z} = \exp(-\widehat{v})$, $\widehat{\Pi^*}$ from (2.5), and $\widehat{\ell}$ from (2.3).

The inference method is maximum likelihood. Think of the optimal control law $\Pi^*_t(\cdot)$ as being parameterized by the desirability function $z(\cdot)$ as given by (2.5). Then the negative log-likelihood is

$$L[z()] = -\log(z(\mathbf{x})) + \log\left(\mathop{\mathrm{E}}_{\mathbf{x}' \sim \Pi^0(\mathbf{x})}\left[z(\mathbf{x}')\right]\right) \tag{3.2}$$

We have omitted the term $\sum_n \log\left(\Pi^0(\mathbf{x}'_n|\mathbf{x}_n)\right)$ because it does not depend on z, although this term could be used in future work attempting to learn p under some regularizing assumptions. Now L could be minimized w.r.t. z, however it is not a convex function of z. We have experimented with such minimization and found it to be slower as well as prone to local minima.

If however we write L in terms of v it becomes convex – because it is a positive sum of log-sum-exp functions plus a linear function. One additional improvement, which enables us to compute L faster when the number of data points exceeds the number of states, is to write L in terms of the visitation counts $a(\mathbf{x}')$ and $b(\mathbf{x})$ defined as the number of times $\mathbf{x}'_n = \mathbf{x}'$ and $\mathbf{x}_n = \mathbf{x}$ respectively. It is interesting that the likelihood depends only on these counts and not on the specific pairings of states in the dataset. We now have

$$L[v\cdot] = \sum_{\mathbf{x}'\in\mathcal{X}} a(\mathbf{x}')v\left(\mathbf{x}'\right) + \sum_{\mathbf{x}\in\mathcal{X}} b(\mathbf{x}) \log\left(\mathop{\mathrm{E}}_{\mathbf{x}'\sim\Pi^0(\mathbf{x})}\left[\exp\left(-v\left(\mathbf{x}'\right)\right)\right]\right) \tag{3.3}$$

Thus inverse optimal control in the linearly-solvable MDP framework reduces to unconstrained convex optimization of an easily-computed function. We will call the resulting algorithm **OptV**. In our current implementation we compute the gradient and Hessian of (3.3) analytically and apply Newton's method with backtracking linesearch.

We did not distinguish between first-exit and average-cost problems because the algorithm is the same in all three cases; the only differences are in how the data are sampled and how $\widehat{\ell}$ is subsequently computed from $\widehat{v}$. This is an advantage over other IRL methods which are usually derived for a single problem formulation.

Finally, the above discussion implied lookup-table representations, however it is easy to use features as well. Consider a linear function approximator in v-space:

$$v(\mathbf{x}) = \textstyle\sum_i w_i \phi_i(\mathbf{x}) \tag{3.4}$$

where $\phi_i(\mathbf{x})$ are given features and w_i are unknown weights. Then $L(\mathbf{w})$ is again convex and can be optimized efficiently. In section 3.2, we consider methods for initializing and adapting the features automatically when the state space is continuous.

3.1.2 Learning the cost directly (OptQ)

We can also express L as a function of ℓ and infer ℓ directly (algorithm **OptQ**). When using lookup-table representations the two algorithms yield identical results, however the results are generally different when using features. This is because the transformation between v and ℓ given by (2.3) is nonlinear, thus a linear function approximator in v-space does not correspond to a linear function approximator in ℓ-space. A second reason to explore direct inference of ℓ is because this turns out to reveal an interesting relationship to the MaxEntIRL algorithm Ziebart et al. [2008a].

For simplicity we focus on first-exit problems where we have the explicit formula (2.8) relating z and ℓ. This formula enables us to express L as a function of ℓ and compute the gradient analytically – which is cumbersome due to the matrix inverse, but doable. Computing the Hessian however is too cumbersome, so we use a BFGS method which approximates the Hessian. L turns out to be convex in ℓ (see Appendix). Nevertheless the OptQ algorithm is much slower than the OptV algorithm. This is because computing $L\left[\ell\left(\cdot\right)\right]$ requires solving the forward problem at every step of the minimization. Therefore learning ℓ directly is not a good idea. If one wants to use features in ℓ-space, it may be better to do the learning in v-space (perhaps with a different set of features) and then fit the function approximator for ℓ using linear regression.

The function $L\left[\ell\left(\cdot\right)\right]$ can be written in an alternative form using the trajectory probabilities (**??**). Suppose the transitions are sampled along trajectories $\varsigma^{(k)}$ with lengths $T\left(k\right)$, and let $\mathbf{x}_t^{(k)}$ denote the state at time t along trajectory k. Using (**??**) and omitting the p-dependent term which does not involve ℓ, we have

$$L\left[\ell\left(\cdot\right)\right] = \sum_k \left(\log z\left(\mathbf{x}_0^{(k)}\right) + \sum_{t=0}^{T(k)} \ell\left(\mathbf{x}_t^{(k)}\right)\right) \tag{3.5}$$

Again we see that computing $L\left[\ell\left(\cdot\right)\right]$ requires $z\left(\cdot\right)$.

3.1.3 Relationship with MaxEntIRL

The MaxEntIRL algorithm Ziebart et al. [2008a] is derived using features (which can also be done in OptV and OptQ) but for simplicity we discuss the lookup-table case with one

delta function "feature" per state. MaxEntIRL is a density estimation algorithm: it looks for the maximum-entropy distribution consistent with the observed state visitation counts (or feature counts more generally). It is known that the maximum-entropy distribution under moment-matching constraints is in the exponential family. Thus MaxEntIRL comes down to finding $\ell(\cdot)$ which maximizes the probability of the observed trajectories within the family

$$p_{\text{MaxEnt}}(\mathbf{X}|\mathbf{x}_0) \propto \exp\left(-\sum_{t=0}^{T} \ell(\mathbf{x}_t)\right) \tag{3.6}$$

The bottleneck is in computing the partition function at each step of the optimization, which is done using a recursive procedure.

Intuitively MaxEntIRL resembles an IRL method. However until now it was unclear what forward optimal control problem is being inverted by MaxEntIRL, and whether such a problem exists in the first place. We can now answer these questions. Comparing (3.6) to (**??**), we see that the trajectory probabilities are identical when the passive dynamics are uniform. Therefore MaxEntIRL is an inverse method for LMDPs with uniform passive dynamics. Indeed the recursion used in Ziebart et al. [2008a] to compute the partition function is very similar to the iterative method for computing the desirability function in Todorov [2007, 2009b]. Both recursions are computationally equivalent to solving the forward problem. As a result both MaxEntIRL and OptQ are slower than OptV, and furthermore MaxEntIRL is a special case of OptQ. MaxEntIRL's restriction to uniform passive dynamics is particularly problematic in modeling physical systems, which often have interesting passive dynamics that can be exploited for control purposes Collins et al. [2005].

3.1.4 Embedding Arbitrary IRL Problems

In this section we show how an IRL problem for a traditional MDP can be embedded in the LMDP framework. This is almost the same as the embedding described in Todorov [2009b], except that here we do not know the cost function during the embedding, thus we need some additional assumptions. We assume that the MDP cost is in the form $l(\mathbf{x}) + r(\mathbf{u})$ where $r(\mathbf{u})$ is a known control cost while $l(\mathbf{x})$ is an unknown state cost. Let $\mathbf{x}' \sim \mathbb{P}(\mathbf{x}, \mathbf{u})$ be the

(known) transition probabilities in the MDP, and assume that the number of actions per state equals the number of possible next states. Let $\ell(\mathbf{x})$ and $\Pi^0(\mathbf{x}'|\mathbf{x})$ be the unknown state cost and passive dynamics in the corresponding LMDP. The embedding Todorov [2009b] comes down to matching the costs for all $\mathbf{x}, \mathbf{u}$:

$$l(\mathbf{x}) + r(\mathbf{u}) = \ell(\mathbf{x}) + \sum_{\mathbf{x}'} \mathbb{P}\left(\mathbf{x}'|\mathbf{x}, \mathbf{u}\right) \log\left(\frac{\mathbb{P}\left(\mathbf{x}'|\mathbf{x}, \mathbf{u}\right)}{\Pi^0(\mathbf{x}'|\mathbf{x})}\right) \tag{3.7}$$

These equations are linear in $\log(\Pi^0\left(\mathbf{x}'|\mathbf{x}\right))$. Let us fix $\mathbf{x}$ and suppose k states are reachable from $\mathbf{x}$ in one step. Then $\Pi^0\left(\mathbf{x}'|\mathbf{x}\right)$ has at most k non-zeros (to ensure finite KL divergence). Let the non-zeros be stacked into the vector $p_{\mathbf{x}}$. Thus we have k linear equations in $k+1$ variables $\log(p_{\mathbf{x}}), l(\mathbf{x}) - \ell(\mathbf{x})$. The additional degree of freedom is removed using $1^T p_{\mathbf{x}} = 1$. We can then solve the LMDP IRL problem, and use the solution as an approximation to the MDP IRL problem. There are no guarantees on the quality of the recovered solution, but we observe that it gives good results experimentally in section 3.1.5.

3.1.5 Numerical results

We compared OptV to three prior IRL algorithms labeled in Figure 3.1 according to the name of their first author: Syed Syed et al. [2008], Abbeel Abbeel and Ng [2004], and Ng Ng and Russell [2000]. The forward problem is a traditional MDP: a grid world with obstacles (black rectangles), a state-action cost which only depends on the state, and discrete actions causing transitions to the immediate neighbors (including diagonals). There is one action per neighbor and it causes a transition to that neighbor with probability 0.9. The rest of the probability mass is divided equally among the remaining neighbors. The problem is in a discounted-cost setting with discount factor 0.5.

All four IRL methods were implemented in Matlab in the most efficient way we could think of. Rather than sampling data from the optimal control policy, we gave them access to the true visitation frequencies under the optimal policy of the traditional MDP (equivalent to infinite sample size). Using the embedding from section 3.1.4, we get an embedded LMDP IRL problem with passive dynamics that is uniform over possible next states and run OptV on this.

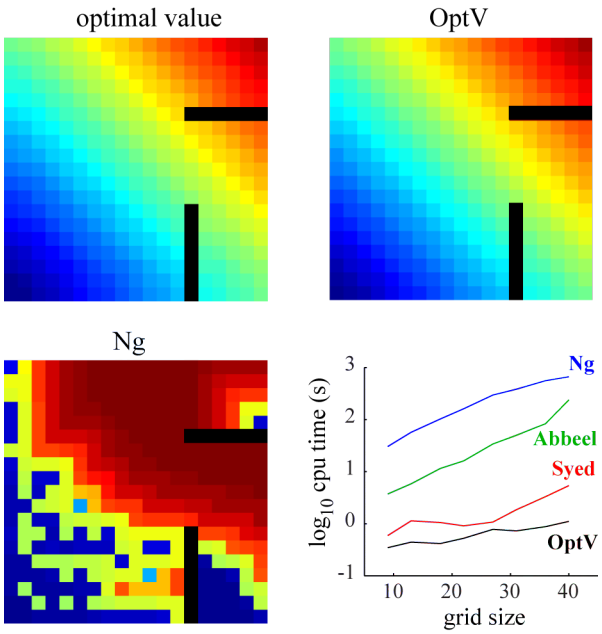


Figure 3.1: Comparison of OptV and prior IRL algorithms on a grid-world problem. Black rectangles are obstacles.

As expected, OptV was substantially faster than all other algorithms for all grid sizes we tested. Even though the forward problem which generated the data is a traditional MDP while OptV is trying to invert an LMDP, it infers a value function very similar to the solution to the forward problem. Since the passive dynamics here are uniform, MaxEntIRL/OptQ produce the same result but about 20 times slower. Although such close similarity is not guaranteed, it is common in our experience. Ng and Russell [2000] proposes a heuristic to select a cost function – which we then translated into a value function by solving the forward problem, while the other algorithms only recover a policy, not a cost function. As shown in the figure, the result is quite different from the correct value function.

Two of the prior IRL algorithms (Syed and Abbeel) are guaranteed to recover the control policy given the true visitation counts, and indeed they do. Since OptV is solving a different problem it does not recover the control policy exactly (which it would if the forward problem was an LMDP). Nevertheless the result is very close, and actually improves when the grid size increases. The expected cost achieved by the inferred policy was 6% above optimal for the 9-size grid, and only 0.3% above optimal for the 40-size grid. Thus we pay a small penalty in terms of performance of the inferred policy, but we recover costs/values and do so faster than any other algorithm.

3.2 Continuous problems

We now focus on optimal control problems in continuous space and time. Such problems lead to PDEs which in our experience are difficult to handle numerically. Therefore the new IRL method we derive below (OptVA) uses time-discretization, along with adaptive bases to handle the continuous state space. We also consider state discretization as a way of obtaining large MDPs on which we can further test the algorithms from the previous section (see Figure 3.2A below).

3.2.1 Linearly-solvable controlled diffusions

Consider the control-affine Ito diffusion

$$d\mathbf{x} = \mathbf{a}(\mathbf{x})\,dt + B(\mathbf{x})\,(\mathbf{u}dt + \sigma d\omega) \tag{3.8}$$

where $\mathbf{a}(\mathbf{x})$ is the drift in the passive dynamics (including gravity, Coriolis and centripetal forces, springs and dampers etc), $B(\mathbf{x})\mathbf{u}$ is the effect of the control signal (which is now a more traditional vector instead of a probability distribution), and $\omega(t)$ is a Brownian motion process. The cost function is in the form

$$\ell(\mathbf{x},\mathbf{u}) = \ell(\mathbf{x}) + \frac{1}{2\sigma^2}\|\mathbf{u}\|^2 \tag{3.9}$$

The relationship between the noise magnitude and the control cost is unusual but can be absorbed by scaling ℓ. The only restriction compared to the usual control-affine diffusions studied in the literature is that the noise and controls must act in the same space.

It can be shown Kappen [2005], Todorov [2009b] that the HJB equation for such problems reduces to a 2nd-order linear PDE when expressed in terms of the desirability z, just like the Bellman equation (2.3) is linear in z. This similarity suggests that the above problem and the linearly-solvable MDPs are somehow related. Indeed it was shown in Todorov [2009b] that problem (3.8, 3.9) can be obtained from a discrete-time continuous-state LMDP by taking a certain limit. The passive dynamics for this MDP are constructed using explicit Euler discretization of the time axis: $\Pi^0(\mathbf{x}'|\mathbf{x})$ is Gaussian with mean $\mathbf{x}+h\,\mathrm{a}(\mathbf{x})+h\,\mathrm{B}(\mathbf{x})\mathbf{u}$ and covariance $h\sigma^2\,\mathrm{B}(\mathbf{x})\,\mathrm{B}(\mathbf{x})^{\mathsf{T}}$, where h is the time step. The state cost in the MDP is $h\,\ell(\mathbf{x})$. It can be shown that the quadratic control cost in (3.9) is the limit of the KL divergence control cost in (3.7) when $h \to 0$.

Thus the continuous optimal control problem (3.8, 3.9) is approximated by the LMDP described above, and IRL methods for this LMDP approximate the continuous inverse problem. The approximation error vanishes when $h \to 0$. However, time-discretization allows us to use larger h, which usually leads to better performance for a given number of samples and bases.

3.2.2 Inverse optimal control with adaptive bases (OptVA)

The inverse method developed here is similar to OptV, however it uses a function approximator with adaptive bases. We represent the value function as

$$v(\mathbf{x};\mathbf{w},\theta) = \textstyle\sum_i w_i \phi_i(\mathbf{x};\theta) = \mathbf{w}^T\phi(\mathbf{x};\theta) \tag{3.10}$$

where $\mathbf{w}$ is a vector of linear weights while θ is a vector of parameters that affect the shape and location of the bases ϕ_i. The bases are normalized Gaussian RBFs:

$$\phi_i(\mathbf{x};\theta) = \frac{\exp\left(\theta_i^{\mathsf{T}}\mathbf{s}(\mathbf{x})\right)}{\sum_j \exp\left(\theta_j^{\mathsf{T}}\mathbf{s}(\mathbf{x})\right)} \tag{3.11}$$

Here θ_i denotes the part of θ specific to ϕ_i, and $\mathbf{s}(\mathbf{x}) = [1; \mathbf{x}_k; \mathbf{x}_k\mathbf{x}_l]$ for all $k \leq l$. Thus $\exp\left(\theta_i^{\mathsf{T}}\mathbf{s}(\mathbf{x})\right)$ is Gaussian. In the language of exponential families, θ_i are the natural parameters and $\mathbf{s}(\mathbf{x})$ the sufficient statistics. We chose normalized RBFs because they often produce better results than unnormalized RBFs – which we also found to be the case here.

Similar to the discrete case, the negative log-likelihood of a dataset $\{\mathbf{x}_n, \mathbf{x}'_n\}$ is

$$L(\mathbf{w},\theta) = \sum_n \mathbf{w}^T\phi\left(\mathbf{x}'_\mathbf{n};\theta\right) + \log\left(\mathop{\mathrm{E}}_{\mathbf{x}'\sim\Pi^0(\mathbf{x})}\left[\exp\left(-\mathbf{w}^T\phi(\mathbf{x};\theta)\right)\right]\right). \tag{3.12}$$

Thus L is convex in $\mathbf{w}$ and can be minimized efficiently for fixed θ. The optimization of θ, or in other words the basis function adaptation, relies on gradient descent – LBFGS or Conjugate Gradients as implemented in the off-the-shelf optimizer Schmidt. We take advantage of the convexity in $\mathbf{w}$ by optimizing $\widetilde{L}(\theta) = \min_\mathbf{w} L(\mathbf{w},\theta)$. Each evaluation of $\widetilde{L}(\theta)$ involves computing the optimal $\mathbf{w}^*(\theta)$ by Conjugate Gradients (which converges very quickly). Then we compute the gradient of $\widetilde{L}$ using

$$\frac{\partial\widetilde{L}(\theta)}{\partial\theta} = \frac{\partial L(\mathbf{w}^*(\theta),\theta)}{\partial\mathbf{w}}\frac{\partial\mathbf{w}^*(\theta)}{\partial\theta} + \frac{\partial L(\mathbf{w}^*(\theta),\theta)}{\partial\theta}$$

The first term on the right vanishes because L has been optimized w.r.t. $\mathbf{w}$. The only complication here is the computation of $\mathrm{E}_{\mathbf{x}'\sim\Pi^0(\mathbf{x})}\left[\exp\left(-w^T\phi(\mathbf{x};\theta)\right)\right]$. In our current implementation we do this by discretizing the state space around $\mathrm{E}_{\mathbf{x}'\sim\Pi^0(\mathbf{x})\otimes\exp(-w^T\phi(\cdot;\theta))}[\mathbf{x}']$ and replacing the integral with a sum. In high-dimensional problems such discretization will not be feasible. However the passive dynamics $\Pi^0(\mathbf{x}'|\mathbf{x})$ are Gaussian, and numerical approximation methods for Gaussian integrals have been studied extensively, resulting in so-called cubature formulas which can be applied here.

The optimization problem is convex in w but non-convex in the basis parameters θ, thus we need good initialization for θ. We developed an automated procedure for this. The intuition is that the optimal controller frequently visits "good" parts of the state space

where the function approximator should have the highest resolution. Thus the centers of the Gaussians are initialized using K-means on the data. The function approximator can also benefit from initializing the covariances properly. We do this by finding the nearest Gaussians, computing the covariance of their means, and scaling it by a constant.

We argued earlier that data makes the inverse problem generally easier than the forward problem. Is this still true in the LMDP case given that the forward problem is linear? For fixed features/bases the two computations are comparable, however basis adaptation is much easier in the inverse problem. This is because the data provides good initialization, and good initialization is key when optimizing a non-convex function.

3.2.3 Numerical results

Here we study inverted pendulum dynamics in the form (3.8, 3.9), with $\sigma = 1$. The state space is 2D: $\mathbf{x} = [\mathbf{x}_p; \mathbf{x}_v]$. We consider a first-exit formulation where the goal is to reach a small region around the vertical position with small velocity. We also consider an infinite-horizon average-cost formulation corresponding to a metronome. The cost $\ell(\mathbf{x})$ only depends on $\mathbf{x}_v$. It is small when $\mathbf{x}_v = \pm 2.5$ and increases sigmoidally away from these values. Thus the pendulum is required to move in either direction at constant speed. The system has positional limits; when these limits are hit the velocity drops to zero. The discretization time step is $h = 0.1$. In the first-exit problem the state space is also discretized, on a 70-by-70 grid.

Figure 3.2A shows further comparison to prior IRL algorithms in a discretized state space using lookup table representaiton (pendulum first exit problem). OptV is faster by orders of magnitude, and recovers the optimal policy almost exactly (relative error $< 10^{-9}$), while prior ILR algorithms recover different policies with significantly worse performance. We used the LP solver Gurobi [Gurobi Optimization, 2014] to implement the Syed-Schapire algorithm. We discretized the actions space with a grid of 70 points for Syed-Schapire and 10 points for Abbeel-Ng (discretization was coarse to limit running time). Figure 3.2B illustrates the performance of the OptVA algorithm on the infinite horizon metronome problem with finite data. A small number of bases (10) is sufficient to recover the optimal value

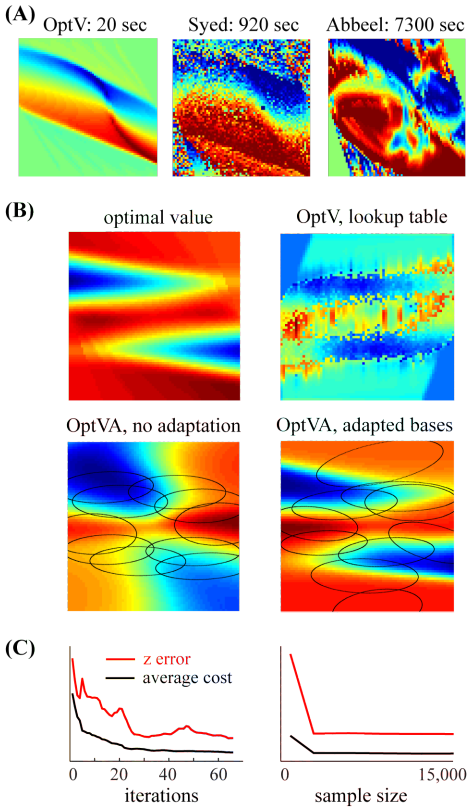


Figure 3.2: **(A)** Control policies in the first-exit (inverted pendulum) problem. Each subplot shows the CPU time and the policy found given the optimal transition probabilities. The policy found by OptV was indistinguishable from the optimal policy and achieved average cost of 13.06, as compared to 57.21 for Syed and 41.15 for Abbeel.

(B) Value functions in the infinite-horizon (metronome) problem. Here the algorithms have access to finite data (12,000 transitions) thus the optimal value function can no longer be recovered exactly. OptV with a lookup table representation does quite poorly, indicating the need for smoothing/generalization. The result of OptVA with the initial bases vaguely resembles the correct solution, and is substantially improved after basis adaptation. The ellipses show the location and shape of the Gaussian bases before normalization.

(C) Performance of OptVA over iterations of basis adaptation for 12,000 samples (left), and as a function of the sample size at the last iteration of basis adaptation (right). We plot the difference between the optimal and inferred z functions (expressed as KL divergence), and the log average cost of the resulting control policy. The curves are scaled and shifted to fit on the same plot.

function quite accurately after basis adaptation. The effects of sample size and iterations of the basis adaptation algorithm are illustrated in Figure 3.2C.

3.3 Summary

Here we presented new algorithms for inverse optimal control applicable to LMDPs with discrete and continuous state. They outperform prior IRL algorithms. The new algorithms are solving a restricted class of problems, but this class is broad enough to include or approximate many control problems of interest. It is particularly well suited for modeling the physical systems commonly studied in nonlinear control.

Apart from the benefits arising from the LMDP framework, key to the efficiency of our algorithms is the insight that recovering values is easier than recovering costs because solving the forward problem is avoided. This of course means that we need features over values rather than costs. Cost features are generally easier to design, which may seem like an advantage of prior IRL algorithms. However prior IRL algorithms need to solve the forward problem – therefore they need features over values (or policies, or state-values, depending on what approximation method is used for solving the forward problem) in addition to features over costs. Thus the feature selection problem in prior IRL work is actually harder.

3.4 Appendix : Convexity of OptQ

The convexity of $L[\ell]$ follows from the following **Lemma:** Let $\mathbf{x} \in \mathrm{R}^m$ and $M(\mathbf{x}) \in \mathrm{R}^{n \times n}$ be such that $M(\mathbf{x})_{ij} = \exp(a_{ij}^T x + b_{ij})$. Suppose that $\sum_j \exp(b_{ij}) < 1 \, \forall i$. Then for any $c, d \in \mathrm{R}_+^n$, the function $\phi(\mathbf{x}) = c^T \log((I - M(\mathbf{x}))^{-1} d)$ is convex on the domain $\mathcal{X} = \{\mathbf{x} : a_{ij}^T x \leq 0 \quad \forall i, j\}$.

Proof: $M(\mathbf{x})$ is a matrix with positive entries and row sums smaller than 1. Thus, the spectral radius of $M(\mathbf{x})$ is smaller than 1. Hence, we use a series expansion of $(I - M(\mathbf{x}))^{-1}$ to get $\phi(\mathbf{x}) = c^T \log\left(\sum_{k=0}^{\infty} M(\mathbf{x})^k d\right)$. For $k \geq 1$, letting $l_0 = i, l_{k+1} = j$, we have $[M(\mathbf{x})^k]_{ij} = \sum_{l_1, l_2, \ldots, l_{k-1}} \prod_{p=0}^{k} [M(\mathbf{x})]_{l_p l_{p+1}}$ Since each entry of $M(\mathbf{x})^k$ is a positive linear combination of terms of the kind $\exp(a^T x + b)$, so is $\sum_k M(\mathbf{x})^k d$ (since $d > 0$). Thus, $\log(\sum_k M(\mathbf{x})^k d)$ is a log-sum-exp function of $\mathbf{x}$ and is hence convex. Since $c > 0$, $c^T \log(\sum_k M(\mathbf{x})^k d)$ is a positive linear combination of convex functions and is hence convex.

Chapter 4

MODELING RISK: A UNIFIED THEORY OF LINEARLY SOLVABLE OPTIMAL CONTROL

Traditional MDPs are formulated as minimizing expected accumulated costs (over a finite or infinite time horizon). However, in many applications, one cares about higher order moments of the accumulated cost (like its variance) that depend on the amount of noise in the system. This is particularly relevant for noisy underactuated systems near unstable equilibria, since it can be hard to recover from even small amounts of perturbations. In this chapter, we develop a class of linearly solvable risk-sensitive optimal control problems. The risk-sensitivity can also be interpreted in a game-theoretic fashion, which is the view we present in this chapter and hence we call these problems Linearly Solvable Markov Games (LMGs).

The results in this chapter can be seen as a generalization of two lines of work: one on Linearly Solvable MDPs(LMDPs) Todorov [2009b] and the other on risk-sensitive path-integral control Broek et al. [2010]. We obtain the LMDP results in the risk-neutral limit $\alpha \to 0$ and the results from Broek et al. [2010] by taking a continuous-time limit. To the best of our knowledge, LMGs are the broadest class of linearly solvable control problems known and include all previous work as a special case.

4.1 Game Theoretic Control : Competitive Games

Here we briefly introduce the notion of game theoretic control or robust control Başar and Bernhard [1995]. In this setting, the system can be influenced by another agent (adversary) in addition to the controller. The controller needs to design a strategy that achieves the control objective in spite of the adversarial disturbances. We shall focus on the simplest case of two-player zero-sum dynamic games, where the adversary is trying to maximize the same cost that the controller is trying to minimize. The game proceeds as follows: 1) The adversary and controller pick actions $\mathbf{u}_{\mathrm{a}}, \mathbf{u}_{\mathrm{c}}$ respectively. 2) The controller pays

cost $\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}_{\mathrm{c}}, \mathbf{u}_{\mathrm{a}})$ and adversary pays $-\ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}_{\mathrm{c}}, \mathbf{u}_{\mathrm{a}})$. 3) The system transitions to state $\mathbf{x}' \sim \mathbb{P}(\mathbf{x}'|\mathbf{x}, \mathbf{u}_{\mathrm{c}}, \mathbf{u}_{\mathrm{a}})$. The solution to such a game can be formulated using the Shapley equations Shapley [1953]:

$$v_t(\mathbf{x}) = \max_{\mathbf{u}_{\mathrm{a}} \in \mathcal{U}_{\mathrm{a}}(\mathbf{x}, u_c)} \min_{\mathbf{u}_{\mathrm{c}} \in \mathcal{U}(\mathbf{x})} \ell_{\mathrm{t}}(\mathbf{x}, \mathbf{u}_{\mathrm{c}}, \mathbf{u}_{\mathrm{a}}) + \mathop{\mathrm{E}}_{\mathbb{P}(\mathbf{x}, \mathbf{u}_{\mathrm{c}}, \mathbf{u}_{\mathrm{a}})} [v_{t+1}]$$

We call such problems Markov Games or MGs. If the min, max can be interchanged without changing the optimal policies for either the controller or the adversary, we say that the game has a *saddle-point equilibrium*. If not, then it matters which player plays first and we have corresponding *upper* and *lower* value functions.

In this chapter, we describe a class of linearly-solvable Markov games (LMGs), where the Shapley equation can be made linear as explained below. But first, we need to introduce a class of divergence measures between probability distributions that will play a key role in LMGs.

4.1.1 Renyi divergence

Renyi divergences are a generalization of the KL divergence. For distributions $p, q \in \mathcal{P}[\mathcal{X}]$, the Renyi divergence of order α is defined as

$$\mathbb{D}_\alpha(p \parallel q) = \frac{\mathrm{sign}(\alpha)}{\alpha - 1} \log \left(\mathop{\mathrm{E}}_{p} \left[\left(\frac{q}{p} \right)^{1-\alpha} \right] \right)$$

For any fixed p, q, it is known that $\mathbb{D}_\alpha$ is always non-negative, decreasing for $\alpha < 0$, and increasing for $\alpha > 0$. It is also known that $\lim_{\alpha \to 1} \mathbb{D}_\alpha(p \parallel q) = \mathrm{KL}(p \parallel q)$.

4.1.2 Linearly Solvable Markov Games (LMGs)

An LMG proceeds as follows: 1) The system in state $\mathbf{x}$ at time t.

2) The adversary picks controls $u_{\mathrm{a}} \in \mathcal{X}^{\mathbf{R}^+}$.

3) The controller picks controls $u_c \in \mathcal{X}^{\mathbf{R}^+}$.

4) The system transitions into a state $\mathbf{x}' \sim u_c \otimes u_\mathrm{a} \otimes \Pi^0(\mathbf{x})$ 5) The cost function is given by:

$$\ell_\mathrm{t}(\mathbf{x}, u_c, u_\mathrm{a}) = \underbrace{\ell_\mathrm{t}(\mathbf{x})}_{\text{State Costs}} + \underbrace{\mathrm{KL}\left(u_c \otimes u_\mathrm{a} \otimes \Pi^0(\mathbf{x}) \parallel u_\mathrm{a} \otimes \Pi^0(\mathbf{x})\right)}_{\text{Control Costs}}$$
$$\underbrace{- \mathbb{D}_{\frac{1}{\alpha}}\left(\Pi^0(\mathbf{x}) \parallel u_\mathrm{a} \otimes \Pi^0(\mathbf{x})\right)}_{\text{Control Cost for Adversary}}$$

We focus on competitive games and require that $\alpha > 0, \alpha \neq 1$. Also, the dynamics of the game is such that the adversary plays first, so the controller has a chance to respond to the adversarial disturbance. Thus, it is a maximin problem where we work with the lower value function. Later, we describe the case $\alpha < 0$ which leads to cooperative games. It turns out that for this class of games, we can prove that the Shapeley equation becomes linear **??**.

The differences between standard MGs and LMGs can be summarized as follows:

	MGs	LMGs
Pol	$\boldsymbol{\Pi_\mathbf{c}} : \mathcal{X} \times \mathcal{U}_\mathrm{a} \to \mathcal{U}$	$\Pi_\mathrm{c} : \mathcal{X} \times \mathcal{X}^{\mathbf{R}^+} \to \mathcal{X}^{\mathbf{R}^+}$
	$\boldsymbol{\Pi}_\mathrm{a} : \mathcal{X} \to \mathcal{U}_\mathrm{a}$	$\Pi_\mathrm{a} : \mathcal{X} \to \mathcal{X}^{\mathbf{R}^+}$
Dyn	$\mathbf{u}_\mathrm{a} = \boldsymbol{\Pi}_\mathrm{a}(\mathbf{x}), \mathbf{u}_\mathrm{c} = \boldsymbol{\Pi_\mathbf{c}}(\mathbf{x}, \mathbf{u}_\mathrm{a})$	$u_\mathrm{a} = \Pi_\mathrm{a}(\mathbf{x}), u_c = \Pi_\mathrm{c}(\mathbf{x}, u_\mathrm{a})$
	$\mathbf{x} \xrightarrow{\Pi_\mathrm{c}, \Pi_\mathrm{a}} \mathbf{x}' \sim \mathbb{P}\left(\mathbf{x}' \middle| \mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a}\right)$	$\mathbf{x} \xrightarrow{\Pi_\mathrm{c}, \Pi_\mathrm{a}} \mathbf{x}' \sim u_c \otimes u_\mathrm{a} \otimes \Pi^0(\mathbf{x})$
Co	$\ell_\mathrm{t}(\mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a})$	$\ell_\mathrm{t}(\mathbf{x}) - \mathbb{D}_{\frac{1}{\alpha}}\left(\Pi^0(\mathbf{x}) \parallel u_\mathrm{a} \otimes \Pi^0(\mathbf{x})\right)$ $+ \mathrm{KL}\left(u_c \otimes u_\mathrm{a} \otimes \Pi^0(\mathbf{x}) \parallel u_\mathrm{a} \otimes \Pi^0(\mathbf{x})\right)$
BE	$v_t(\mathbf{x}) = \max_{\mathbf{u}_\mathrm{a}} \min_{\mathbf{u}_\mathrm{c}} \ell_\mathrm{t}(\mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a})$ $+ \mathrm{E}_{\mathbb{P}(\mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a})}\left[v_{t+1}\right]$	$z_t(\mathbf{x}) = \mathcal{Q}_\mathrm{t}(\mathbf{x}) \mathop{\mathrm{E}}_{\Pi^0(\mathbf{x})}\left[z_{t+1}\right]$ $z_t(\mathbf{x}) = \exp\left((\alpha - 1) v_t(\mathbf{x})\right)$ $\mathcal{Q}_\mathrm{t}(\mathbf{x}) = \exp\left((\alpha - 1)\, \ell_\mathrm{t}(\mathbf{x})\right)$
OP	$\boldsymbol{\Pi_\mathbf{c}}^*(\mathbf{x}, \mathbf{u}_\mathrm{a}; t) = \mathop{\mathsf{argmin}}_{\mathbf{u}_\mathrm{c}} \ell_\mathrm{t}(\mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a})$ $+ \mathrm{E}_{\mathbb{P}(\mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a})}\left[v_{t+1}\right]$	$\Pi_\mathrm{c}{}^*(\mathbf{x}, u_\mathrm{a}; t) = z_{t+1}^{\frac{1}{1-\alpha}}$

LMDPs as a special case of LMGs:

As $\alpha \to 0$, we recover the LMDP Bellman equation. We can explain this by looking at the cost function. It is known that $\lim_{\alpha \to 0} \mathbb{D}_{1/\alpha}(p \parallel q) \to \log\left(\sup_{\mathbf{x}} p(\mathbf{x})/q(\mathbf{x})\right)$. For this cost, the optimal strategy for the adversary is to always leave the passive dynamics unchanged, that is ${\Pi_{\mathrm{a}}}^*{}_t(\mathbf{x}) = 1$. Intuitively, this says that the control cost for the adversary is high enough and the optimal strategy for him is to do nothing. Thus the problem reduces to the LMDP setting.

Effect of α :

As α increases, the relative control cost of the controller with respect to the adversary increases, so, effectively, the adversary becomes more powerful. This makes the controller more conservative (or risk-averse), since it is fighting a stronger adversary.

Cooperative LMGs:

We have also derived a cooperative LMG where two agents collaborate to accomplish the same control task. The game proceeds similar to a competitive game, however now both agents pay the same cost and are trying to minimize it in collaboration. The cost function for cooperative LMGs (for both agents) is:

$$\ell(\mathbf{x}) + \mathbb{D}_{1/\alpha}\left(u_{\mathrm{a}} \otimes \Pi^0(\mathbf{x}) \parallel \Pi^0(\mathbf{x})\right) + \mathrm{KL}\left(u_c \otimes u_{\mathrm{a}} \otimes \Pi^0(\mathbf{x}) \parallel u_{\mathrm{a}} \otimes \Pi^0(\mathbf{x})\right)$$

where $\alpha < 0$. As $|\alpha|$ gets bigger, the control cost for the helper gets smaller and the helper contributes more towards accomplishing the control task while the controller contributes less. The resulting BE is similar to the competitive case. The BE for IH is:

$$z(\mathbf{x}) = \exp\left((\alpha - 1)\left(\ell(\mathbf{x}) - c\right)\right) \mathop{\mathrm{E}}_{\Pi^0(\mathbf{x})}[z] \tag{4.1}$$

In this case, again we can recover LMDPs by taking $\alpha \to 0$ and making the control cost for the helper effectively large enough that he always chooses not to change the passive dynamics.

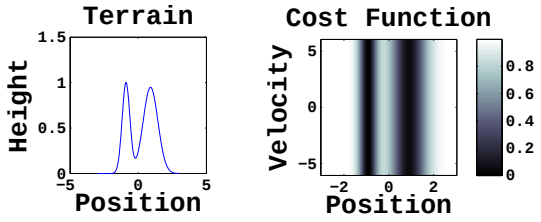


Figure 4.1: Terrain and Cost Function for LMG example

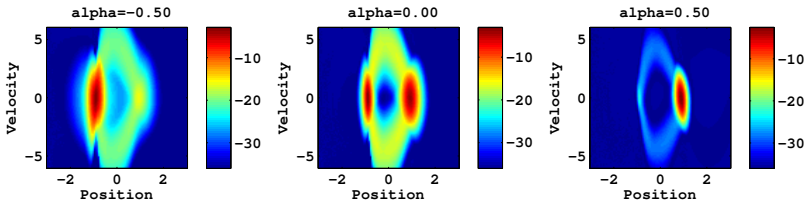


Figure 4.2: Logarithm of Stationary Distribution under Optimal Control vs α

Examples:

We illustrate the effect of α with a simple control problem that requires one to drive up as high as possible on a hilly terrain. The cost function encourages one to drive up to the highest point, but the highest point is the peak of a steep hill, so that even a small perturbation from the adversary can push one downhill quickly. On the other hand, there is a shorter but less steep hill, where the adversary cannot have as much of an effect. The problem is formulated in the IH setting, so we are looking for a control strategy that achieves low average cost over a very long horizon. The terrain and cost function are plotted in figure 4.1. The stationary distributions over $\mathcal{X}$ under optimal control for different values of α are plotted in 4.2. It can be seen that when $\alpha < 0$ (cooperative case), the controller places more probability on the riskier but more rewarding option (steeper/higher hill) but when $\alpha > 0$, the controller is more conservative and chooses the safer but less rewarding option (shorter/less steep hill). In the LMDP case, the solution splits its probability more or less evenly between the two options.

4.1.3 Linearly Solvable Differential Games (LDGs)

In this section we consider differential games (DGs) which are continuous-time versions of MGs. A differential game is described by a stochastic differential equation

$$\mathrm{d}\,\mathbf{x} = \big(\mathrm{a}(\mathbf{x}) + \mathrm{B}(\mathbf{x})\mathbf{u}_\mathrm{c} + \sqrt{\alpha}\,\mathrm{B}(\mathbf{x})\mathbf{u}_\mathrm{a}\big)\,\mathrm{d}\,t + \sigma\,\mathrm{B}(\mathbf{x})\,\mathrm{d}\,\omega$$

The infinitesimal generator $\mathcal{L}[\cdot]$ for the uncontrolled process ($\mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a} = 0$) can be defined similarly to (2.7). We also define a cost rate

$$\ell_\mathrm{t}(\mathbf{x}, \mathbf{u}_\mathrm{c}, \mathbf{u}_\mathrm{a}) = \underbrace{\ell_\mathrm{t}(\mathbf{x})}_{\text{State Cost}} + \underbrace{\frac{1}{2\sigma^2}{\mathbf{u}_\mathrm{c}}^T\mathbf{u}_\mathrm{c}}_{\text{Control Cost for Controller}} - \underbrace{\frac{1}{2\sigma^2}{\mathbf{u}_\mathrm{a}}^T\mathbf{u}_\mathrm{a}}_{\text{Control Cost for Adversary}}$$

Like LMGs, these are two-player zero-sum games, where the controller is trying to minimize the cost function while the adversary tries to maximize the same cost. It can be shown that the optimal solution to differential games based on diffusion processes is characterized by a nonlinear PDE known as the Isaacs equation Başar and Bernhard [1995].

However, for the kinds of differential games we described here, the Isaacs equation expressed in terms of $z_t = \exp\left((\alpha - 1)v_t\right)$ becomes linear and is given by:

$$\frac{\partial z_t\left(\mathbf{x}\right)}{\partial t} = \left(1 - \alpha\right)\ell_{\mathrm{t}}(\mathbf{x})\, z_t\left(\mathbf{x}\right) - \mathcal{L}\left[z_t\right]\left(\mathbf{x}\right)$$

$$\mathbf{\Pi_c}^*\left(\mathbf{x}; t\right) = \frac{\sigma^2}{\left(\alpha - 1\right) z_t\left(\mathbf{x}\right)} \mathrm{B}(\mathbf{x})^T \frac{\partial\, z_t\left(\mathbf{x}\right)}{\partial \mathbf{x}}$$

$$\mathbf{\Pi_a}^*{}_t\left(\mathbf{x}\right) = \frac{-\sqrt{\alpha}\sigma^2}{\left(\alpha - 1\right) z_t\left(\mathbf{x}\right)} \mathrm{B}(\mathbf{x})^T \frac{\partial\, z_t\left(\mathbf{x}\right)}{\partial \mathbf{x}}$$

When $\alpha = 0$, the adversarial control $\mathbf{u}_{\mathrm{a}}$ has no effect and we recover LDs. As α increases, the adversary's power increases and the control policy becomes more conservative.

There is a relationship between LDGs and LMGs. LDGs can be derived as the continuous time limit of LMGs that solve time-discretized versions of differential games. This relationship is analogous to the one between LMDPs and LDs.

Connection to Risk-Sensitive Control

Both LMGs and LDGs can be interpreted in an alternate manner, as solving a sequential decision making problem with an alternate objective: Instead of minimizing expected total cost, we minimize the expectation of the exponential of the total cost:

$$\mathop{\mathrm{E}}_{\mathbf{x}_{t+1}\sim\Pi_{\mathrm{c}}(\mathbf{x}_t)\otimes\Pi^0(\mathbf{x}_t)}\left[\exp\left(\sum_{t=0}^{\mathrm{N}} \alpha\, \ell_{\mathrm{t}}(\mathbf{x}_t) + \sum_{t=0}^{\mathrm{N}-1} \mathbb{D}_\alpha\left(\Pi_{\mathrm{c}}(\mathbf{x}_t)\otimes\Pi^0(\mathbf{x}_t) \parallel \Pi^0(\mathbf{x}_t)\right)\right)\right]$$

This kind of objective is used in risk-sensitive control Marcus et al. [1997] and it has been shown that this problem can also be solved using dynamic programming giving rise to a risk-sensitive Bellman equation. It turns out that for this objective, the Bellman equation is exactly the same as that of an LMG. The relationship between risk-sensitive control and game theoretic or robust control has been studied extensively in the literature Başar and Bernhard [1995], and it also shows up in the context of linearly solvable control problems.

4.1.4 Relationships among the different formulations

Linearly Solvable Markov Games (LMGs) are the most general class of linearly solvable control problems, to the best of our knowledge. As the adversarial cost increases ($\alpha \to 0$),

we recover Linearly Solvable MDPs (LMDPs) as a special case of LMGs. When we view LMGs as arising from the time-discretization of Linearly Solvable Differential Games (LDGs), we recover LDGs as a continuous time limit ($\mathrm{d}\,t \to 0$). Linearly Solvable Controlled Diffusions(LDs) can be recovered either as the continuous time limit of an LMDP , or as the non-adversarial limit ($\alpha \to 0$) of LDGs. The overall relationships between the various classes of linearly solvable control problems is summarized in the figure below:

$$\begin{array}{ccc} LMGs & \xrightarrow[\alpha \to 0]{} & LMDPs \\ \Big\downarrow{\scriptstyle dt \to 0} & & \Big\downarrow{\scriptstyle dt \to 0} \\ LDGs & \xrightarrow{\alpha \to 0} & LDs \end{array}$$

4.2 Conclusions

We have developed a very general family of linearly solvable control problems. To the best of our knowledge, all previous work on linearly solvable control are special cases. Also, the use of Renyi divergences in control is novel. An interesting theoretical question is whether LMGs are the most general family of linearly solvable control problems possible.

In terms of practical applicability, LMGs could be very useful for tuning controllers to be more conservative (risk-averse) or more aggressive (risk-taking). We have seen that the resulting behavior can be substantially different for different α. The linearity makes LMGs easier to solve, but we still need to develop function approximation techniques that scale to high dimensional state spaces and nonlinear dynamics. If Π^0 is Gaussian and z is represented as a mixtures of Gaussians $\times$ polynomials $\times$ trigonometric functions, one can use a power-iteration like algorithm to solve the linear Bellman equation with each step being analytical. Combined with the flexibility of LMGs we believe that these techniques are very promising and can potentially solve hard control problems.

4.3 Proofs

Let $\mu \in \mathcal{P}[\mathcal{X}]$ and $f : \mathcal{X} \to \mathbf{R}$. Define $\Psi_\mu^\kappa[f] = \frac{1}{\kappa} \log\left(\mathrm{E}_\mu\left[\exp\left(\kappa f\right)\right]\right)$.

definition 1. A *saddle point equilibrium* of an LMG is a pair of feedback policies ${\Pi_\mathrm{c}}^*, {\Pi_\mathrm{a}}^* :$

$\mathcal{X} \to \mathcal{P}[\mathcal{X}]$ that achieve the following extremum

$$\min_{\Pi_c} \max_{\Pi_a} \mathcal{J}(\mathbf{x}, \Pi_c, \Pi_a) = \max_{\Pi_a} \min_{\Pi_c} \mathcal{J}(\mathbf{x}, \Pi_c, \Pi_a)$$

at every state $\mathbf{x} \in \mathcal{X}$, where $\mathcal{J}(\mathbf{x}, \Pi_c, \Pi_a)$ depends on the problem formulation:

$$\text{Finite Horizon (FH): } \mathcal{J}(\mathbf{x}, \Pi_c, \Pi_a) = \mathop{\mathrm{E}}_{\Pi_c, \Pi_a, \mathbf{x}_0 = \mathbf{x}} \left[\sum_{t=0}^{\mathrm{N}-1} \ell(\mathbf{x}_t, \Pi_c(\mathbf{x}_t), \Pi_a(\mathbf{x}_t)) + \ell_f(\mathbf{x}_\mathrm{N}) \right]$$

$$\text{First Exit (FE): } \mathcal{J}(\mathbf{x}, \Pi_c, \Pi_a) = \mathop{\mathrm{E}}_{\Pi_c, \Pi_a, \mathbf{x}_0 = \mathbf{x}, \mathrm{N_e} = \min[t : \mathbf{x}_t \in \mathcal{T}]} \left[\sum_{t=0}^{\mathrm{N_e}-1} \ell(\mathbf{x}_t, \Pi_c(\mathbf{x}_t), \Pi_a(\mathbf{x}_t)) + \ell_f(\mathbf{x}_{\mathrm{N_e}}) \right]$$

$$\text{Infinite Horizon (IH): } \mathcal{J}(\mathbf{x}, \Pi_c, \Pi_a) = \lim_{\mathrm{N} \to \infty} \frac{1}{\mathrm{N}} \mathop{\mathrm{E}}_{\Pi_c, \Pi_a, \mathbf{x}_0 = \mathbf{x}} \left[\sum_{t=0}^{\mathrm{N}} \ell(\mathbf{x}_t, \Pi_c(\mathbf{x}_t), \Pi_a(\mathbf{x}_t)) \right]$$

where the expectations are under the stochastic dynamics of the system defined by Π^0, Π_c, Π_a.

Theorem 1. *If $\alpha < 1$, the LMG always has a saddle point equilibrium. The saddle point equilibrium is given by*

$$\Pi^*{}_a(\mathbf{x}) = (z)^{\frac{\alpha}{\alpha-1}}, \Pi_c{}^*(\mathbf{x}) = (z)^{\frac{1}{1-\alpha}}$$

for IH,FE.

Proof. We do the proof for the IH case, the proof for the other cases is similar.

$$\begin{aligned} \lambda + v(\mathbf{x}) &= \min_{u_c} \max_{u_a} \left[\ell(\mathbf{x}) - \mathbb{D}_{1/\alpha}\left(\Pi^0(\mathbf{x}) \parallel u_a\right) + \mathrm{KL}\left(u_c \parallel u_a\right) + \mathop{\mathrm{E}}_{u_c}[v] \right] \\ &= \max_{u_a} \min_{u_c} \left[\ell(\mathbf{x}) - \mathbb{D}_{1/\alpha}\left(\Pi^0(\mathbf{x}) \parallel u_a\right) + \mathrm{KL}\left(u_c \parallel u_a\right) + \mathop{\mathrm{E}}_{u_c}[v] \right] \end{aligned}$$

then the saddle point equilibrium is attained for the feedback policies setting u_c, u_a to the values attaining the extremum in the Shapely equation above. We are given that the linear BE (4.1) has a solution, say z. Consider the function $v = \frac{1}{\alpha-1}\log(z)$. This is bounded and continuous function since z is continuous,bounded above and below away from 0. Using theorem 2, we know that

$$\begin{aligned} &\max_{u_a} \min_{u_c} \left[\ell(\mathbf{x}) - \mathbb{D}_{1/\alpha}\left(\Pi^0(\mathbf{x}) \parallel u_a\right) + \mathrm{KL}\left(u_c \parallel u_a\right) + \mathop{\mathrm{E}}_{u_c}[v] \right] \\ &= \ell(\mathbf{x}) - \Psi_{\Pi^0(\mathbf{x})}^{\frac{\frac{1}{\alpha}-1}{\frac{1}{\alpha}}}[-v] = \ell(\mathbf{x}) + \Psi_{\Pi^0(\mathbf{x})}^{1-\alpha}[v] \end{aligned}$$

and the same result holds if the min, max are interchanged. Since z satisfies the linear BE, we know that v satisfies

$$v(\mathbf{x}) = \ell(\mathbf{x}) + \frac{1}{\alpha - 1} \log \left(\underset{\Pi^0(\mathbf{x})}{\mathrm{E}} \left[\exp\left((\alpha - 1) v \right) \right] \right) = \ell(\mathbf{x}) + \Psi_{\Pi^0(\mathbf{x})}^{1-\alpha} \left[v \right].$$

Combining the results above, we have that v is a bounded continuous solution to the Shapely equation. Hence the result. □

Theorem 2. *If $\alpha > 1$ and $f : \Omega \mapsto \Re$ is a bounded measurable function, then the problem*

$$\min_{\mu} \max_{\nu} - \mathbb{D}_\alpha \left(\mu_0 \parallel \nu \right) + \mathrm{KL} \left(\mu \parallel \nu \right) + \underset{\mu}{\mathrm{E}} \left[f \right]$$
$$\textit{Subject to } \mu, \nu \in \mathcal{P}\left[\mathcal{X} \right]$$

has optimum $- \Psi_{\mu_0}^{\frac{\alpha-1}{\alpha}} \left[-f \right]$ with the optimum attained at $\nu^* = \mu_0 \otimes \exp\left(\frac{f}{\alpha} \right)$,$\mu^* = \nu^* \otimes \exp\left(-f \right)$. The results hold even if the order of the min, max *are reversed.*

Proof. Let μ, ν be any distributions satisfying the constraints of the problem. Then by lemma 3, we have $\mathbb{D}_\alpha \left(\mu_0 \parallel \nu \right) - \mathrm{KL} \left(\mu \parallel \nu \right) \geq \frac{\alpha \, \mathrm{KL}(\mu \| \mu_0)}{1-\alpha}$. This bound is attained when $\nu = \mu \otimes \left(\frac{\partial \mu}{\partial \mu_0} \right)^{\frac{\alpha}{1-\alpha}}$. Also, it is easy to see that this choice for ν satisfies the constraints. So we have

$$\max_{\nu} - \mathbb{D}_\alpha \left(\mu_0 \parallel \nu \right) + \mathrm{KL} \left(\mu \parallel \nu \right) = - \min_{\nu} \mathbb{D}_\alpha \left(\mu_0 \parallel \nu \right) - \mathrm{KL} \left(\mu \parallel \nu \right) = \frac{\alpha}{1 - \alpha} \mathrm{KL} \left(\mu \parallel \mu_0 \right)$$

We are left with $\frac{\alpha}{1-\alpha} \mathrm{KL} \left(\mu \parallel \mu_0 \right) + \mathrm{E}_\mu \left[f \right]$, which can be lower bounded using lemma 1:

$$\frac{\alpha}{1 - \alpha} \left(\mathrm{KL} \left(\mu \parallel \mu_0 \right) + \underset{\mu}{\mathrm{E}} \left[\frac{(1-\alpha) f}{\alpha} \right] \right) \geq \frac{\alpha}{\alpha - 1} \Psi_{\mu_0} \left[\frac{(1-\alpha) f}{\alpha} \right] = - \Psi_{\mu_0}^{\frac{\alpha-1}{\alpha}} \left[-f \right]$$

Since f is bounded, the probability distribution $\mu = \mu_0 \otimes \exp\left(\frac{(1-\alpha) f}{\alpha} \right)$ exists and attains the lower bound above. Thus, the minimum is attained at the distribution $\mu^* = \mu_0 \otimes \exp\left(\frac{(1-\alpha) f}{\alpha} \right)$ which satisfies the constraints and the minimum value is $- \Psi_{\mu_0}^{\frac{\alpha-1}{\alpha}} \left[f \right]$. Thus, the optimal value of the overall minimax problem is $- \Psi_{\mu_0}^{\frac{\alpha-1}{\alpha}} \left[-f \right]$ and the saddle point is given by $\mu^* = \mu_0 \otimes \exp\left(\frac{(1-\alpha) f}{\alpha} \right), \nu^* = \mu^* \otimes \left(\frac{\partial \mu^*}{\partial \mu_0} \right)^{\frac{\alpha}{1-\alpha}} = \mu_0 \otimes \exp\left(\frac{f}{\alpha} \right)$. This can be rewritten as in the statement of the theorem, $\nu^* = \mu_0 \otimes \exp\left(\frac{f}{\alpha} \right)$ and $\mu^* = \nu^* \otimes \exp\left(-f \right)$.

If we switch the max and min, we can write $\min_\mu \mathrm{KL}(\mu \parallel \nu) + \mathrm{E}_\mu[f] = -\Psi_\nu[-f]$ with the minimum attained at $\mu^* = \nu \otimes \exp(-f)$ by lemma 1. We are then left with

$$\max_\nu -\mathbb{D}_\alpha(\mu_0 \parallel \nu) - \Psi_\nu[\exp(-f)] = -\min_\nu \mathbb{D}_\alpha(\mu_0 \parallel \nu) + \Psi_\nu[-f]$$

By lemma 2, we know that $\mathbb{D}_\alpha(\mu_0 \parallel \nu) + \Psi_\nu[-f] \geq -\Psi_{\mu_0^\alpha}^{\frac{1-\alpha}{\alpha}}[f]$ with the minimum attained at $\nu^* = \mu_0 \otimes \exp\left(\frac{f}{\alpha}\right)$. □

Lemma 1. *Let $\mu, \mu_0 \in \mathcal{P}[\mathcal{X}]$ and $f : \mathcal{X} \to \mathbf{R}$. Then,* $\mathrm{KL}(\mu \parallel \mu_0) + \mathrm{E}_\mu[f] \geq -\Psi_{\mu_0}[-f]$.

Proof. The objective can be rewritten as

$$\mathop{\mathrm{E}}_{\mu}\left[\log\left(\frac{\mu}{\mu_0}\right) + f\right] = \mathop{\mathrm{E}}_{\mu}\left[-\log\left(\frac{\mu_0}{\mu}\right) + f\right] = \mathop{\mathrm{E}}_{\mu}\left[-\log\left(\frac{\mu_0}{\mu}\exp(-f)\right)\right].$$

By Jensen's inequality, since $-\log$ is convex, the RHS is larger than $-\log\left(\mathrm{E}_\mu\left[\frac{\mu_0}{\mu}\exp(-f)\right]\right) = -\log\left(\mathrm{E}_{\mu_0}[\exp(-f)]\right)$, establishing the result. □

Lemma 2. *Let $\mu, \mu_0 \in \mathcal{P}[\mathcal{X}]$ and $f : \mathcal{X} \to \mathbf{R}$. Then,*

$$\operatorname{sgn}(\alpha)\,\mathbb{D}_\alpha(\mu_0 \parallel \mu) + \Psi_\mu[f] \geq \Psi_{\mu_0^\alpha}^{\frac{\alpha-1}{\alpha}}[f] \quad \text{if } \alpha > 0$$

$$\operatorname{sgn}(\alpha)\,\mathbb{D}_\alpha(\mu_0 \parallel \mu) + \Psi_\mu[f] \leq \Psi_{\mu_0^\alpha}^{\frac{\alpha-1}{\alpha}}[f] \quad \text{if } \alpha > 0$$

Proof. Letting $g = \exp(f)$, the LHS can be rewritten as

$$\frac{\log\left(\mathrm{E}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)}{\alpha - 1} + \log\left(\mathop{\mathrm{E}}_{\mu}[g]\right) = \log\left(\left(\mathop{\mathrm{E}}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)^{\frac{1}{\alpha-1}} \mathop{\mathrm{E}}_{\mu_0}\left[\frac{\mu}{\mu_0}g\right]\right).$$

First suppose $\alpha < 0$. Then, using Holder's inequality, we have

$$\mathop{\mathrm{E}}_{\mu_0}\left[\frac{\mu}{\mu_0}g\right] \leq \left(\mathop{\mathrm{E}}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)^{\frac{1}{1-\alpha}} \left(\mathop{\mathrm{E}}_{\mu_0}\left[(g)^{\frac{1-\alpha}{-\alpha}}\right]\right)^{\frac{-\alpha}{1-\alpha}} \implies$$

$$\mathop{\mathrm{E}}_{\mu_0}\left[\frac{\mu}{\mu_0}g\right]\left(\mathop{\mathrm{E}}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)^{\frac{1}{\alpha-1}} \leq \left(\mathop{\mathrm{E}}_{\mu_0}\left[(g)^{\frac{1-\alpha}{-\alpha}}\right]\right)^{\frac{-\alpha}{1-\alpha}} = \left(\mathop{\mathrm{E}}_{\mu_0}\left[\exp\left(\frac{(\alpha-1)f}{\alpha}\right)\right]\right)^{\frac{\alpha}{\alpha-1}}.$$

Taking log on both sides, we have the result. The distribution $\mu_0 \otimes (g)^{\frac{\alpha-1}{\alpha}}$ exists since f is bounded. Also note that $h(x) = x^{\frac{1}{1-\alpha}}$ is convex if $\alpha > 0, \alpha \neq 1$. The first term inside the

log can be bounded as follows:

$$\begin{aligned}\left(\mathop{\mathrm{E}}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)^{\frac{1}{1-\alpha}} &= \left(\mathop{\mathrm{E}}_{\mu_0 \otimes g^{\frac{\alpha-1}{\alpha}}}\left[g^{\frac{1-\alpha}{\alpha}}\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)^{\frac{1}{1-\alpha}}\left(\mathop{\mathrm{E}}_{\mu_0}\left[g^{\frac{\alpha-1}{\alpha}}\right]\right)^{\frac{1}{1-\alpha}} \\ &\leq \mathop{\mathrm{E}}_{\mu_0 \otimes g^{\frac{\alpha-1}{\alpha}}}\left[g^{\frac{1}{\alpha}}\frac{\mu}{\mu_0}\right]\left(\mathop{\mathrm{E}}_{\mu_0}\left[g^{\frac{\alpha-1}{\alpha}}\right]\right)^{\frac{1}{1-\alpha}} \text{ (Jensen's Inequality)} \\ &= \mathop{\mathrm{E}}_{\mu_0}\left[g^{\frac{\alpha-1}{\alpha}+\frac{1}{\alpha}}\frac{\mu}{\mu_0}\right]\left(\mathop{\mathrm{E}}_{\mu_0}\left[g^{\frac{\alpha-1}{\alpha}}\right]\right)^{\frac{1}{1-\alpha}-1} = \mathop{\mathrm{E}}_{\mu_0}\left[g\frac{\mu}{\mu_0}\right]\left(\mathop{\mathrm{E}}_{\mu_0}\left[g^{\frac{\alpha-1}{\alpha}}\right]\right)^{\frac{\alpha}{1-\alpha}}\end{aligned}$$

Rewriting the last inequality, we get $\left(\mathrm{E}_{\mu_0}\left[g^{\frac{\alpha-1}{\alpha}}\right]\right)^{\frac{\alpha}{\alpha-1}} \leq \mathrm{E}_{\mu}\left[g\right]\left(\mathrm{E}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right]\right)^{\frac{1}{1-\alpha}}$. Taking log on both sides gives the result.

□

Lemma 3. *Let $\mu, \nu, \mu_0 \in \mathcal{P}\left[\mathcal{X}\right]$ and $f : \mathcal{X} \to \mathbf{R}$, $\alpha > 1$. Then, $\mathbb{D}_\alpha\left(\mu_0 \parallel \nu\right) - \mathrm{KL}\left(\mu \parallel \nu\right) \geq \frac{\alpha}{1-\alpha}\mathrm{KL}\left(\mu \parallel \mu_0\right)$.*

Proof. By definition,

$$\mathrm{KL}\left(\mu \parallel \nu\right) = \mathop{\mathrm{E}}_{\mu}\left[-\log\left(\frac{\nu}{\mu}\right)\right], \mathbb{D}_\alpha\left(\mu_0 \parallel \nu\right) = \frac{\log\left(\mathrm{E}_{\mu_0}\left[\left(\frac{\nu}{\mu_0}\right)^{1-\alpha}\right]\right)}{\alpha-1}.$$

Now, $\mathrm{E}_{\mu_0}\left[\left(\frac{\nu}{\mu_0}\right)^{1-\alpha}\right] = \mathrm{E}_{\mu}\left[\frac{\mu_0}{\mu}\left(\frac{\nu}{\mu_0}\right)^{1-\alpha}\right] = \mathrm{E}_{\mu}\left[\frac{\nu}{\mu}\left(\frac{\nu}{\mu_0}\right)^{1-\alpha}\right]\frac{\nu}{\mu_0} = \mathrm{E}_{\mu}\left[\frac{\nu}{\mu}\left(\frac{\nu}{\mu_0}\right)^{-\alpha}\right]$. This gives us

$$\mathbb{D}_\alpha\left(\mu_0 \parallel \nu\right) - \mathrm{KL}\left(\mu \parallel \nu\right) = \frac{\log\left(\mathrm{E}_{\mu}\left[\frac{\nu}{\mu}\left(\frac{\partial\nu}{\partial\mu_0}\right)^{-\alpha}\right]\right)}{\alpha-1} + \mathop{\mathrm{E}}_{\mu}\left[\log\left(\frac{\nu}{\mu}\right)\right]$$

When $\alpha > 1$, $\frac{\log}{\alpha-1}$ is concave and by Jensen's inequality the first term is $\geq \frac{\mathrm{E}_{\mu}\left[\log\left(\frac{\nu}{\mu}\left(\frac{\nu}{\mu_0}\right)^{-\alpha}\right)\right]}{\alpha-1} = \frac{\mathrm{E}_{\mu}\left[-\alpha\log\left(\frac{\nu}{\mu_0}\right)+\log\left(\frac{\nu}{\mu}\right)\right]}{\alpha-1}$. This implies:

$$\mathbb{D}_\alpha\left(\mu_0 \parallel \nu\right) - \mathrm{KL}\left(\mu \parallel \nu\right) \geq \frac{\alpha}{\alpha-1}\mathop{\mathrm{E}}_{\mu}\left[\log\left(\frac{\frac{\nu}{\mu}}{\frac{\nu}{\mu_0}}\right)\right] = \frac{\alpha}{\alpha-1}\mathop{\mathrm{E}}_{\mu}\left[\log\left(\frac{\mu_0}{\mu}\right)\right] = \frac{\alpha\,\mathrm{KL}\left(\mu \parallel \mu_0\right)}{1-\alpha}.$$

□

Chapter 5

CONVEX POLICY OPTIMIZATION

So far, we have looked at a special class of control problems (LMDPs and extensions) that simplify the Bellman Equations that characterize the optimal solution to a stochastic optimal control problem. However, even with these simplified Bellman Equations, it is not obvious how to solve high-dimensional control problems for arbitrary dynamical systems in an automated way. Function approximation methods Todorov [2009c]Zhong [2013] have been applied to solve the simplified Bellman equations approximately, but successful applications have required careful tuning of parameters and combining the results with approaches like Model Predictive Control.

In this chapter, we look at an alternative approach: We ignore the Bellman equation and look at stochastic control purely as a (stochastic) optimization problem. Of course, searching over the space of all possible policies leads to an infinite dimensional optimization problem that cannot be solved on a computer. Instead, we parameterize the control policy and search for the optimal policy within a parametric class. This is now a finite dimensional (stochastic) optimization problem and can be attacked using techniques from nonlinear and stochastic optimization. Indeed, we do apply these approaches with success to problems in electric power systems in chapter 6. However, for many problems, direct policy optimization is fraught with numerical difficulties and the optimization algorithms applied to this problems are susceptible to slow convergence and local minima. On the other hand, tremendous progress has been made in convex optimization Boyd and Vandenberghe [2004], leading to generic algorithms that can solve convex optimization problems with thousands of variables and special purpose algorithms for problems with upto a million variables. Furthermore, applications such as large scale machine learning have lead to the development of efficient algorithms for stochastic convex optimization Nemirovski et al. [2009], where the optimization objective or gradient cannot be evaluated exactly but only estimated through noisy

samples. Unfortunately, under standard formulations, stochastic optimal control problems are generically non-convex optimization problems, even if the dynamics and control policies are linear, which is one of the reasons direct optimization algorithms tend to work poorly when applied to stochastic control problems.

In this chapter, we look at non-standard formulations of stochastic control problems that lead to convex optimization problems. The first formulation (section 5.1) looks closely at linear systems and develops convex approximations to Linear Quadratic Regulator (LQR) and Linear Quadratic Games (LQ Games) problems in finite horizon, based on the Bode Sensitivity Integral, a classical result in control theory that characterizes fundamental limitations in control systems. The second formulation (section 5.2) looks at a risk-averse control formulation and develops algorithms based on stochastic convex optimization to solve control problems with arbitrary dynamics and control policies, subject to a condition on the control costs and noise that is closely related to the Linearly Solvable MDPs and Path Integral Control formalisms discussed in the previous chapters.

5.1 Convex Policy Optimization Based on Bode's Sensitivity Integral

Linear feedback control synthesis is a classical topic in control theory and has been extensively studied in the literature. From the perspective of stochastic optimal control theory, the classical result is the existence of an optimal linear feedback controller for systems with linear dynamics, quadratic costs and gaussian noise (LQG systems) that can be computed via dynamic programming Kalman et al. [1960]. However, if one imposes additional constraints on the feedback matrix (such as a sparse structure arising from the need to implement control in a decentralized fashion), the dynamic programming approach is no longer applicable. In fact, it has been shown that the optimal control policy may not even be linear Witsenhausen [1968] and that the general problem of designing linear feedback gains subject to constraints is NP-hard Blondel and Tsitsiklis [1997].

Previous approaches to synthesizing structured controllers can be broadly categorized into three types: Frequency Domain ApproachesRotkowitz and Lall [2002]Qi et al. [2004]Rotkowitz and Lall [2006]Shah [2013], Dynamic Programming Approaches Fan et al. [1994]Swigart and Lall [2010]Lamperski and Lessard [2013] and Nonconvex optimization methods Lin et al.

[2013]Apkarian et al. [2008]Burke et al. [2006]. The first two classes of approaches find *exact* solutions to structured control problems for special cases. The third class of approaches tries to directly solve the optimal control problem (minimizing the $\mathcal{H}_2$,$\mathcal{H}_\infty$ norm) subject to constraints on the controller, using nonconvex optimization techniques. These are generally applicable, but are susceptible to local minima and slow convergence (especially for nonsmooth norms such as $\mathcal{H}_\infty$).

In this work, we take a different approach: We reformulate the structured control problem using a family of new control objectives (section 5.1.1). We develop these bounds as follows: The $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms can be expressed as functions of singular values of the linear mapping from disturbance trajectories to state trajectories. This mapping is a highly nonlinear function of the feedback gains. However, the inverse of this mapping has a simple linear dependence on the feedback gains. Further, the determinant of the mapping has a fixed value independent of the closed loop dynamics - this is in fact a finite horizon version of Bode's sensitivity integral and has been studied in Iglesias [2001]. By exploiting both these facts, we develop upper bounds on the $\mathcal{H}_2, \mathcal{H}_\infty$ norms in terms of the singular values of the inverse mapping. We show that these upper bounds have several properties that make them desirable control objectives. For the new family of objectives, we show that the resulting problem of designing an optimal linear state feedback matrix, under arbitrary convex constraints, is convex (section 5.2.1). Further, we prove suboptimality bounds on how the solutions of the convex problems compare to the optima of the original problem. Our approach is directly formulated in state space terminology and does not make any reference to frequency domain concepts. Thus, it applies directly to time-varying systems. We validate our approach numerically and show that the controllers synthesized by our approach achieve good performance (section 5.1.4).

5.1.1 Problem Formulation

Consider a finite-horizon discrete-time linear system in state-space form:

$$\mathbf{x}_1 = D_0\omega_0$$

$$\mathbf{x}_{t+1} = A_t\mathbf{x}_t + B_t u_t + D_t\omega_t, \quad t = 1, 2, \ldots, N-1.$$

Here $t = 0, 1, 2, \ldots, N$ is the discrete time index, $\mathbf{x}_t \in \mathbf{R}^n$ is the plant state, $\omega_t \in \mathbf{R}^n$ is an exogenous disturbance and $u_t \in \mathbf{R}^{n_u}$ is the control input. We employ static state feedback:

$$u_t = K_t \mathbf{x}_t.$$

Let $\mathbf{K} = \{K_t : t = 1, 2, \ldots, N-1\}$ and denote the closed-loop system dynamics by

$$\tilde{A}_t(K_t) = A_t + B_t K_t.$$

Let $\lambda_{\max}(M)$ denote the maximum eigenvalue of an $l \times l$ symmetric matrix M, $\lambda_{\min}(M)$ the minimum eigenvalue and $\lambda_i(M)$ the i-th eigenvalue in descending order:

$$\lambda_l(M) = \lambda_{\min}(M) \leq \lambda_{l-1}(M) \leq \ldots \leq \lambda_{\max}(M) = \lambda_1(M).$$

Similarly, singular values of a general rank l matrix M are:

$$\sigma_l(M) = \sigma_{\min}(M) \leq \sigma_2(M) \leq \ldots \leq \sigma_{\max}(M) = \sigma_1(M).$$

I denotes the identity matrix. Boldface smallcase letters denote trajectories:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \omega_0 \\ \vdots \\ \omega_{N-1} \end{pmatrix}$$

For $z \in \mathbf{R}^n$,

$$\mathrm{Var}(z) = \frac{1}{n} \sum_{i=1}^{n} \left(z_i - \frac{\sum_{i=1}^{n} z_i}{n} \right)^2.$$

$z_{[i]}$ is the i-th largest component of z and $|z|$ the vector with entries $|z_1|, \ldots, |z_n|$. Finally, $\mathcal{N}(\mu, \Sigma)$ denotes a Gaussian distribution with mean μ and covariance matrix Σ.

There is a linear mapping between disturbance and state trajectories for a linear system. This will play a key role in our work, and we denote it by

$$\mathbf{X} = F(\mathbf{K})\boldsymbol{\epsilon},$$

$$\text{where } F(\mathbf{K}) = \begin{bmatrix} D_0 & 0 & \ldots & 0 \\ \tilde{A}_1 D_0 & D_1 & \ldots & 0 \\ \tilde{A}_2 \tilde{A}_1 D_0 & \tilde{A}_2 D_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \prod_{\tau=1}^{N-1} \tilde{A}_{N-\tau} D_0 & \prod_{\tau=2}^{N-1} \tilde{A}_{N-\tau} D_1 & \ldots & D_{N-1} \end{bmatrix}.$$

Our formulation differs from standard control formulations in the following ways:

1 We assume that the controller performance is measured in terms the norm of the system trajectory $\mathbf{X}^T\mathbf{X}$ (see section 5.1.8 for an extension that includes control costs).

2 As mentioned earlier, we restrict ourselves to have static state feedback $u_t = K_t\mathbf{x}_t$ (section 5.1.8 discusses dynamic output feedback).

3 We assume that D_t is square and invertible.

Finite-horizon versions of the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of the system are given by:

$$\begin{aligned} q_2(\mathbf{K}) &= \underset{\omega_t \sim \mathcal{N}(0,I)}{\mathrm{E}} \left[\sum_{t=1}^{N} {\mathbf{x}_t}^T \mathbf{x}_t \right] = \mathrm{E}\left[\mathrm{tr}\left(\mathbf{X}\mathbf{X}^T \right) \right] \\ &= \mathrm{tr}\left(F\left(\mathbf{K}\right)^T F\left(\mathbf{K}\right) \right) = \sum_{i=1}^{nN} \sigma_i \left(F\left(\mathbf{K}\right) \right)^2 \end{aligned} \tag{5.1}$$

$$\begin{aligned} q_\infty(\mathbf{K}) &= \sqrt{\max_{\boldsymbol{\epsilon} \neq 0} \frac{\sum_{t=1}^{N} {\mathbf{x}_t}^T \mathbf{x}_t}{\sum_{t=0}^{N-1} {\omega_t}^T \omega_t}} \\ &= \max_{\boldsymbol{\epsilon} \neq 0} \frac{\left\| F\left(\mathbf{K}\right) \boldsymbol{\epsilon} \right\|}{\left\| \boldsymbol{\epsilon} \right\|} = \sigma_{\max}\left(F\left(\mathbf{K}\right) \right). \end{aligned} \tag{5.2}$$

If there are no constraints on $\mathbf{K}$, these problems can be solved using standard dynamic programming techniques. However, we are interested in synthesizing structured controllers. We formulate this very generally: We allow *arbitrary* convex constraints on the set of feedback matrices: $\mathbf{K} \in \mathcal{C}$ for some convex set $\mathcal{C}$. Then, the control synthesis problem becomes

$$\underset{\mathbf{K} \in \mathcal{C}}{\mathrm{Minimize}}\, q_2(\mathbf{K}) \tag{5.3}$$

$$\underset{\mathbf{K} \in \mathcal{C}}{\mathrm{Minimize}}\, q_\infty(\mathbf{K}) \tag{5.4}$$

The general problem of synthesizing stabilizing linear feedback control, subject even to simple bound constraints on the entries of K, is known to be hard Blondel and Tsitsiklis [1997]. Several hardness results on linear controller design can be found in Blondel and Tsitsiklis [2000]. Although these results do not cover the problems (5.3)(5.4), they suggest that (5.3)(5.4) are hard optimization problems. In this work, we propose an alternate objective function based on the singular values of the inverse mapping $F\left(\mathbf{K}\right)^{-1}$ and prove

that this objective can be optimized using convex programming techniques under *arbitrary* convex constraints on the feedback matrices $\mathbf{K} = \{K_t\}$. Given the above hardness results, it is clear that the optimal solution to the convex problem will not match the optimal solution to the original problem. However, we present theoretical and numerical evidence to suggest that the solutions of the convex problem we propose ((5.5),(5.6)) approximate the solution to the original problems ((5.3),(5.4)) well for several problems.

Control Objective

The problems (5.3),(5.4) are non-convex optimization problems, because of the nonlinear dependence of $F(\mathbf{K})$ on $\mathbf{K}$. In this section, we will derive convex upper bounds on the singular values of $F(\mathbf{K})$ that can be optimized under arbitrary convex constraints $\mathcal{C}$. We have the following results (section 5.1.8, theorems 5.1.4, 5.1.5):

$$q_\infty(\mathbf{K}) \leq \left(\prod_{t=0}^{N-1} \det(D_t)\right) \left(\frac{\sum_{i=1}^{n_s N-1} \sigma_i\left(F(\mathbf{K})^{-1}\right)}{n_s N-1}\right)^{n_s N-1}$$

$$q_2(\mathbf{K}) \leq nN \left(\prod_{t=0}^{N-1} \det(D_t)\right)^2 \left(\sigma_{\max}\left(F(\mathbf{K})^{-1}\right)\right)^{2(n_s N-1)}$$

To illustrate the behavior of these upper bounds (denoted UB_∞, UB_2), we plot them for a scalar linear system

$$\mathbf{x}_{t+1} = u_t + \omega_t, \mathbf{x}_t, u_t = k\mathbf{x}_t, k \in \mathbf{R}$$

over a horizon $N = 100$ in figure 5.1. When $|k| < 1$, this system is unstable and otherwise it is stable. Thus, $|k|$ is a measure of the "degree of instability" of the system. As expected, the original objectives grow slowly to the point of instability and then blow up. The convex upper bounds are fairly loose upper bounds and increase steadily. However, the rate of growth increases with degree of instability. Similar results are observed for $n > 1$.

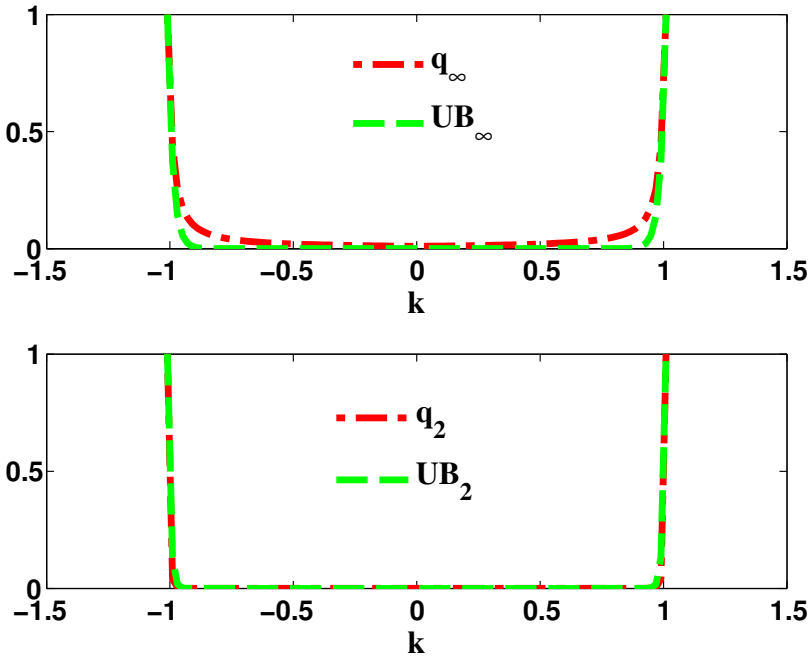


Figure 5.1: Convex Surrogate vs Original Objective (rescaled to lie in [0,1]): $q_\infty(k)$ vs $UB_\infty(k)$ (top),$q_2(k)$ vs $UB_2(k)$ (bottom)

A General Class of Control Objectives

Inspired by the upper bounds of the previous section, we formulate the controller design problem as follows:

$$\underset{\mathbf{K}\in\mathcal{C}}{\text{Minimize}}\, q_2^c(\mathbf{K}) = \sigma_{\max}\left(F(\mathbf{K})^{-1}\right) \text{ (surrogate to } q_2) \tag{5.5}$$

$$\underset{\mathbf{K}\in\mathcal{C}}{\text{Minimize}}\, q_\infty^c(\mathbf{K}) = \sum_{i=1}^{n_s N-1} \sigma_i\left(F(\mathbf{K})^{-1}\right) \text{ (surrogate to } q_\infty) \tag{5.6}$$

The objectives (5.5),(5.6) are just two of the control objectives that are allowed in our framework. We can actually allow a general class of objectives that can be minimized for control design. From Lewis [1995], we know that for any *absolutely invariant* convex function $f(x)$ on $\mathbf{R}^n$, the function $g(X) = f(\mathbf{s}X)$ on $\mathbf{R}^{n\times n}$ is convex. This motivates us to consider a generalized control objective:

$$\begin{aligned}\underset{\mathbf{K}}{\text{Minimize}} \quad & \underbrace{f\left(\mathbf{s}F(\mathbf{K})^{-1}\right)}_{\text{Controller Performance}} + \underbrace{R(\mathbf{K})}_{\text{Minimize Control Effort}} \\ \text{Subject to } & \mathbf{K}\in\mathcal{C}\end{aligned} \tag{5.7}$$

where $\mathcal{C}$ is a convex set encoding the structural constraints on $\mathbf{K}$ and $R(\mathbf{K})$ is a convex penalty on the feedback gains $\mathbf{K}$. We show (in theorem 5.2.1) that this problem is a convex optimization problem. Common special cases for f are:

1 $f(x) = \|x\|_\infty$ which gives rise to the spectral norm $\left\|(F(\mathbf{K}))^{-1}\right\| = \sigma_{\max}\left((F(\mathbf{K}))^{-1}\right)$, the same as (5.5).

2 $f(x) = \|x\|_1$ which gives rise to the nuclear norm $\left\|(F(\mathbf{K}))^{-1}\right\|_* = \sum_i \sigma_i\left((F(\mathbf{K}))^{-1}\right)$.

3 $f(x) = \sum_{i=1}^k |x|_{[i]}$ which gives rise to the Ky Fan k-norm $\sum_{i=1}^k \sigma_i\left((F(\mathbf{K}))^{-1}\right)$. In particular $f(x) = \sum_{i=1}^{nN-1} |x|_{[i]}$ corresponds to (5.6).

A common choice for $R(\mathbf{K})$ is $\|\mathbf{K}\|^2$. For decentralized control, $\mathcal{C}$ would be of the form $\mathcal{C} = \{\mathbf{K} : \mathbf{K}_t \in S\}$ where S is the set of matrices with a certain sparsity pattern corresponding to the decentralization structure required. We now present our main theorem proving the convexity of the generalized problem (5.7).

5.1.2 Main Technical Results

Proof of Convexity

Theorem 5.1.1. *If f is an absolutely symmetric lower-semicontinuous convex function, $R(\mathbf{K})$ is a convex function and $\mathcal{C}$ is a convex set, then the problem* (5.7) *is a convex optimization problem.*

Proof. The proof relies on the structure of $F(\mathbf{K})^{-1}$. Rewriting the discrete-time dynamics equations, we have:

$$\omega_0 = {D_0}^{-1}\mathbf{x}_1, \omega_t = {D_t}^{-1}\mathbf{x}_{t+1} - {D_t}^{-1}\tilde{A}_t\mathbf{x}_t \text{ for } t \geq 1.$$

It can be shown that $F(\mathbf{K})^{-1}$ is given by

$$\begin{bmatrix} {D_0}^{-1} & 0 & \dots & \dots & 0 \\ -{D_1}^{-1}\tilde{A}_1 & {D_1}^{-1} & \dots & \dots & 0 \\ 0 & -{D_2}^{-1}\tilde{A}_2 & {D_2}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \dots & {D_{N-1}}^{-1} \end{bmatrix}$$

This can be verified by simple matrix multiplication. Now, the convexity is obvious since $\tilde{A}_t = A_t + B_t K_t$ is a linear function of $\mathbf{K}$, and so is $F(\mathbf{K})^{-1}$. Since f is an absolutely symmetric lower-semicontinuous convex function, $f(\mathbf{s}X)$ is a convex function of X Lewis [1995]. Thus, $f(\mathbf{s}(F(\mathbf{K}))^{-1})$ is the composition of an affine function in $\mathbf{K}$ with a convex function, and is hence convex. The function $R(\mathbf{K})$ is known to be convex and so are the constraints $\mathbf{K} \in \mathcal{C}$. Hence, the overall problem is a convex optimization problem. □

Suboptimality Bounds

We are using convex surrogates for the q_2, q_∞ norms. Thus, it makes sense to ask the question: How far are the optimal solutions to the convex surrogates from those of the original problem? We answer this question by proving multiplicative suboptimality bounds: We prove that the ratio of the q_2 norm of the convex surrogate solution and the q_2-optimal solution is bounded above by a quantity that decreases as the variance of the singular

vector of $(F(\mathbf{K}))^{-1}$ at the optimum. Although these bounds may be quite loose, they provide qualitative guidance about when the algorithm would perform well.

Theorem 5.1.2. *Let the solution to the convex optimization and original problem be:*

$$\mathbf{K}_c{}^* = \underset{\mathbf{K}\in\mathcal{C}}{\operatorname{argmin}}\, \sigma_{\max}\left((F(\mathbf{K}))^{-1}\right) \text{ (Convex Opt)}$$

$$\mathbf{K}^* = \underset{\mathbf{K}\in\mathcal{C}}{\operatorname{argmin}} \sum_i \left(\sigma_i\left(F(\mathbf{K})\right)\right)^2 \text{ (Original Opt)}$$

respectively. Let $F^* = (F(\mathbf{K}^*))^{-1}, F_c{}^* = (F(\mathbf{K}_c{}^*))^{-1}$. *Let*

$$\sigma_c{}^* = \left[\left(\frac{\sigma_2\left(F_c{}^*\right)}{\sigma_{nN}\left(F_c{}^*\right)}\right)^2, \ldots, \left(\frac{\sigma_2\left(F_c{}^*\right)}{\sigma_2\left(F_c{}^*\right)}\right)^2\right]$$

$$\sigma^* = \left[\left(\frac{\sigma_{nN}\left(F^*\right)}{\sigma_{nN}\left(F^*\right)}\right)^2, \ldots, \left(\frac{\sigma_{nN}\left(F^*\right)}{\sigma_2\left(F^*\right)}\right)^2\right]$$

Then,

$$\frac{q_2(\mathbf{K}_c{}^*)}{q_2(\mathbf{K}^*)} \leq \left(\frac{nN}{nN-1}\right)\exp\left(\frac{\operatorname{Var}\left(\sigma_c{}^*\right) - \operatorname{Var}\left(\sigma^*\right)}{2}\right)$$

Proof. The proof relies on Holder's defect formula which quantifies the gap in the AM-GM inequality Becker [2012]. For any numbers $0 < a_m \leq \ldots \leq a_1$, we have:

$$\left(\frac{\sum_{i=1}^m a_i}{m}\right)\exp\left(-\frac{\mu}{2}\operatorname{Var}(a)\right) = \left(\prod_{i=1}^m a_i\right)^{1/m}$$

where $\mu \in \left[\left(\frac{1}{a_1}\right)^2, \left(\frac{1}{a_m}\right)^2\right]$. Plugging in the lower and upper bounds for μ, we get

$$\left(\frac{\sum_{i=1}^m a_i}{m}\right)\exp\left(-\frac{\operatorname{Var}\left(a/a_1\right)}{2}\right) \geq \left(\prod_{i=1}^m a_i\right)^{1/m}$$
$$\left(\frac{\sum_{i=1}^m a_i}{m}\right)\exp\left(-\frac{\operatorname{Var}\left(a/a_m\right)}{2}\right) \leq \left(\prod_{i=1}^m a_i\right)^{1/m}.$$

Using this inequality with $a_i = \left(\sigma_{nN-i+1}\left(F^*\right)\right)^{-2}, i = 1, 2, 3, \ldots, nN-1$, we get

$$\begin{aligned}
\frac{q_2(\mathbf{K}^*)}{nN-1} &\geq \frac{1}{nN-1}\sum_{i=2}^{nN}\frac{1}{\left(\sigma_i\left(F^*\right)\right)^2} \\
&\geq \exp\left(\frac{\mathrm{Var}\left(\sigma^*\right)}{2}\right)\left(\prod_{i=2}^{nN}\frac{1}{\left(\sigma_i\left(F^*\right)\right)^2}\right)^{\frac{1}{nN-1}} \\
&= c\exp\left(\frac{\mathrm{Var}\left(\sigma^*\right)}{2}\right)\left(\sigma_{\max}\left(F^*\right)\right)^{\frac{2}{nN-1}}
\end{aligned}$$

where $c = \left(\prod_{t=0}^{N-1}\det\left(D_t\right)\right)^{\frac{2}{nN-1}}$ and the last equality follows since $\det\left(F^*\right) = \prod_{t=0}^{N-1}\det\left(D_t\right)$. Since $\mathbf{K}_c{}^*$ minimizes $\sigma_{\max}\left(F\left(\mathbf{K}\right)^{-1}\right)$, we have

$$\begin{aligned}
\frac{q_2(\mathbf{K}^*)}{nN-1} &\geq c\exp\left(\frac{\mathrm{Var}\left(\sigma^*\right)}{2}\right)\left(\sigma_{\max}\left(F_c{}^*\right)\right)^{\frac{2}{nN-1}} \\
&\geq \exp\left(\frac{\mathrm{Var}\left(\sigma^*\right)}{2}\right)\left(\prod_{i=2}^{nN}\left(\frac{1}{\sigma_i\left(F_c{}^*\right)}\right)^2\right)^{\frac{1}{nN-1}} \\
&\geq \exp\left(\frac{\mathrm{Var}\left(\sigma^*\right)}{2} - \frac{\mathrm{Var}\left(\sigma_c{}^*\right)}{2}\right)\left(\frac{\sum_{i=2}^{nN}\frac{1}{\left(\sigma_i\left(F_c{}^*\right)\right)^2}}{nN-1}\right) \\
&\geq \left(\frac{nN-1}{nN}\right)\exp\left(\frac{\mathrm{Var}\left(\sigma^*\right)}{2} - \frac{\mathrm{Var}\left(\sigma_c{}^*\right)}{2}\right)\frac{q_2(\mathbf{K}_c{}^*)}{nN-1}.
\end{aligned}$$

The result follows from simple algebra now. □

Theorem 5.1.3. *Let the solution to the convex optimization and original problem be:*

$$\mathbf{K}_c{}^* = \underset{\mathbf{K}\in\mathcal{C}}{\mathsf{argmin}}\sum_{i=1}^{nN-1}\sigma_i\left(\left(F\left(\mathbf{K}\right)\right)^{-1}\right) \quad \textit{(Convex Opt)}$$

$$\mathbf{K}^* = \underset{\mathbf{K}\in\mathcal{C}}{\mathsf{argmin}}\,\sigma_{\max}\left(F\left(\mathbf{K}\right)\right) \quad \textit{(Original Opt)}$$

respectively. Let $F^* = \left(F\left(\mathbf{K}^*\right)\right)^{-1}, F_c{}^* = \left(F\left(\mathbf{K}_c{}^*\right)\right)^{-1}$. *Let*

$$\sigma_c{}^* = \left[\frac{\sigma_{nN-1}\left(F_c{}^*\right)}{\sigma_1\left(F_c{}^*\right)}, \ldots, \frac{\sigma_1\left(F_c{}^*\right)}{\sigma_1\left(F_c{}^*\right)}\right]$$

$$\sigma^* = \left[\frac{\sigma_{nN-1}\left(F^*\right)}{\sigma_{nN-1}\left(F^*\right)}, \ldots, \frac{\sigma_1\left(F^*\right)}{\sigma_{nN-1}\left(F^*\right)}\right]$$

Then,

$$\frac{q_\infty(\mathbf{K}_c{}^*)}{q_\infty(\mathbf{K}^*)} \leq \exp\left((nN-1)\left(\frac{\mathrm{Var}\left(\sigma^*\right) - \mathrm{Var}\left(\sigma_c{}^*\right)}{2}\right)\right)$$

Proof. The proof follows a similar structure as the previous theorem and relies on Holder's defect formula. Let $c = \prod_{t=0}^{N-1} \det(D_t)$. Using the same inequalities with $a_i = \sigma_i(F^*), i = 1, 2, \ldots, nN-1$, we get

$$\begin{aligned}(q_\infty(\mathbf{K}^*))^{\frac{1}{nN-1}} &= c\left(\prod_{i=1}^{nN-1} \sigma_i(F^*)\right)^{\frac{1}{nN-1}} \\ &\geq \frac{c\exp\left(-\frac{\mathrm{Var}(\sigma^*)}{2}\right)}{nN-1}\left(\sum_{i=1}^{nN-1} \sigma_i(F^*)\right)\end{aligned}$$

where $c = \left(\prod_{t=0}^{N-1} \det(D_t)\right)^{\frac{2}{nN-1}}$. Since $\mathbf{K}_c{}^*$ minimizes $\sum_{i=1}^{nN-1} \sigma_i\left(F(\mathbf{K})^{-1}\right)$, we have

$$\begin{aligned}(q_\infty(\mathbf{K}^*))^{\frac{1}{nN-1}} &\geq \frac{c\exp\left(\frac{-\mathrm{Var}(\sigma^*)}{2}\right)}{nN-1}\left(\sum_{i=1}^{nN-1} \sigma_i(F_c{}^*)\right) \\ &\geq c\exp\left(\frac{\mathrm{Var}(\sigma_c{}^*)}{2} - \frac{\mathrm{Var}(\sigma^*)}{2}\right)\left(\prod_{i=1}^{nN-1} \sigma_i(F_c{}^*)\right)^{\frac{1}{nN-1}} \\ &= \exp\left(\frac{\mathrm{Var}(\sigma_c{}^*)}{2} - \frac{\mathrm{Var}(\sigma^*)}{2}\right)(q_\infty(\mathbf{K}_c{}^*))^{\frac{1}{nN-1}}.\end{aligned}$$

The result follows from simple algebra now. □

Interpretation of Bounds

The bounds have the following interpretation: Since the product of singular values is constrained to be fixed, stable systems (with small $\mathcal{H}_2, \mathcal{H}_\infty$ norm) would have all of their singular values close to each other. Thus, if the singular values at the solution discovered by our algorithm are close to each other, we can expect that our solution is close to the true optimum. Further, the bounds say that the only thing that matters is the spread of the singular values relative to the spread of singular values at the optimal solution. A side-effect of the analysis is that it suggests that the spectral norm of $(F(\mathbf{K}))^{-1}$ be used as a surrogate for the q_2 norm and the nuclear norm be a surrogate for the q_∞ norm, since optimizing these surrogates produces solutions with suboptimality bounds on the original objectives.

Finally note that although the bounds depend on the (unknown) optimal solution $\mathbf{K}^*$, we can still get a useful bound for the q_2 case by simply dropping the effect of the negative

term so that

$$\frac{q_2(\mathbf{K}_c{}^*)}{q_2(\mathbf{K}^*)} \leq \left(\frac{nN}{nN-1}\right) \exp\left(\frac{\mathrm{Var}\left(\sigma_c{}^*\right)}{2}\right).$$

which can be computed after solving the convex problem to get $\mathbf{K}_c{}^*$. A finer analysis may be possible by looking at the minimum possible value of $\mathrm{Var}\left(\sigma^*\right)$, just based on the block-bidiagonal structure of the matrix $F\left(\mathbf{K}\right)^{-1}$, but we leave this for future work.

5.1.3 Algorithms and Computation

In this work, our primary focus is to discuss the properties of the new convex formulation of structured controller synthesis we developed here. Algorithms for solving the resulting convex optimization problem (5.7) is a topic we will investigate in depth in future work. In most cases, problem (5.7) can be reformulated as a semidefinite programming problem and solved using off-the-shelf interior point methods. However, although theoretically polynomial time, off-the-shelf solvers tend to be inefficient in practice and do not scale. In this section, we lay out some algorithmic options including the one we used in our numerical experiments (section 5.1.4).

When the objective used is the nuclear norm, $\sum_{i=1}^{nN} \sigma_i \left(\left(F\left(\mathbf{K}\right)\right)^{-1}\right)$, we show that it is possible to optimize the objective using standard Quasi-Newton approaches. The nuclear norm is a nonsmooth function in general, but given the special structure of the matrices appearing in our problem, we show that it is differentiable. For a matrix X, the subdifferential of the nuclear norm $\|X\|_*$ at X is given by

$$\{UV^T + W : U^TW = 0 \text{ or } WV = 0, \|W\|_2 \leq 1\}$$

where $X = U\Sigma V^T$ is the singular value decomposition of X. For our problem $X = F\left(\mathbf{K}\right)^{-1}$, which has a non-zero determinant and hence is a nonsingular square matrix irrespective of the value of $\mathbf{K}$. Thus, the subdifferential is a singleton ($U^TW = 0 \implies W = 0$ as U is full rank and square). This means that the nuclear norm is a differentiable function in our problem and one can use standard gradient descent and Quasi Newton methods to minimize it. These methods are orders of magnitude more efficient than other approaches (reformulating as an SDP and using off-the-shelf interior point methods). They still require computing

the SVD of an $nN \times nN$ matrix at every iteration, which will get prohibitively expensive when nN is of the order of several thousands. However, the structure of $F(\mathbf{K})^T F(\mathbf{K})$ is block-tridiagonal and efficient algorithms have been proposed for computing the eigenvalues of such matrices (see Sandryhaila and Moura [2013] and the references therein). Since the singular values of $F(\mathbf{K})$ are simply square roots of eigenvalues of $F(\mathbf{K})^T F(\mathbf{K})$, this approach could give us efficient algorithms for computing the SVD of $F(\mathbf{K})$.

When the objective is the spectral norm $\sigma_{\max}\left((F(\mathbf{K}))^{-1}\right)$, we can reformulate the problem as a semidefinite programming problem (SDP):

$$\begin{aligned} &\underset{t,\mathbf{K}\in\mathcal{C}}{\text{Minimize}}\, t + R(\mathbf{K}) \\ &\text{Subject to } tI \geq \begin{pmatrix} 0 & (F(\mathbf{K}))^{-1^T} \\ (F(\mathbf{K}))^{-1} & 0 \end{pmatrix} \end{aligned}$$

The log-barrier for the semidefinite constraint can be rewritten as $\log\left(\det\left(t^2 - F(\mathbf{K})^{-1^T} F(\mathbf{K})^{-1}\right)\right)$ using Schur complements. The matrix $(F(\mathbf{K}))^{-1^T}(F(\mathbf{K}))^{-1}$ is a symmetric positive definite block-tridiagonal matrix, which is a special case of a chordal sparsity pattern Andersen et al. [2010]. This means that computing the gradient and Newton step for the log-barrier is efficient, with complexity growing as $O(N)$. Thus, at least for the case where the objective is the spectral norm, we can develop efficient interior point methods.

5.1.4 Numerical Results

Comparing Algorithms: Decentralized Control

In this section, we compare different approaches to controller synthesis. We work with discrete-time LTI systems over a fixed horizon N with $A_t = A, B_t = B = I, D_t = D = I$. Further, we will use $\mathcal{C} = \{K : K_{ij} = 0 \notin S\}$, where S is the set of non-zero indices of K. The control design methodologies we compare are:

NCON: This refers to nonconvex approaches for both the q_2 and q_∞ norms. The q_2 norm is a differentiable function and we use a standard LBFGS method Schmidt [2012] to minimize it. The q_∞ norm is nondifferentiable, but only at points where the maximum singular value of $F(\mathbf{K})$ is not unique. We use a nonsmooth Quasi Newton method Lewis and Overton

[2012] to minimize it (using the freely available software implementation HANSO Overton).
CON: The convex control synthesis described here. In the experiments described here, we use the following objective:

$$\underset{K}{\text{Minimize}} \frac{1}{nN}\left(\sum_{i=1}^{m}\sigma_i\left(F\left(K\right)^{-1}\right)\right) \tag{5.8}$$

where

$$F\left(K\right)^{-1} = \begin{bmatrix} I & 0 & \dots & 0 \\ -(A+BK) & I & \dots & 0 \\ 0 & -(A+BK) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & I \end{bmatrix}$$

Subject to

$K_{ij} = 0 \quad \forall (i,j) \notin S$

with $m = nN-1$ as a surrogate for the q_∞ norm and $m = 1$ for the q_2 norm. Although these objectives are non differentiable, we find that an off-the-shelf LBFGS optimizer Schmidt [2012] works well and use it in our experiments here.
OPT: The optimal solution to the problem in the absence of the constraint $\mathcal{C}$. This is simply the solution to a standard LQR problem for the q_2 case. For the q_∞ norm, this is computed by solving a series of LQ games with objective:

$$\sum_{t=1}^{N}{\mathbf{x}_t}^T\mathbf{x}_t - \sum_{t=0}^{N-1}\gamma^2{\omega_t}^T\omega_t$$

where the controller chooses u to minimize the cost while an adversary chooses ω_t so as to maximize the cost. There is critical value of γ below which the upper value of this game is unbounded. This critical value of γ is precisely the q_∞ norm and the resulting policies for the controller at this value of γ is the q_∞-optimal control policy. For any value of γ, the solution of the game can be computed by solving a set of Ricatti equations backward in time Başar and Bernhard [2008].
We work with a dynamical system formed by coupling a set of systems with unstable

dynamics $A^i \in \mathbf{R}^{2\times 2}$.

$$\mathbf{x}_{t+1}^i = A^i \mathbf{x}_t^i + \sum_j \eta_{ij} \mathbf{x}_t^j + u_t^i + \omega_t^i$$

where $\mathbf{x}^i$ denotes the state of the i-th system and η_{ij} is a coupling coefficient between systems i and j. The objective is to design controls $u = \{u^i\}$, in order to stabilize the overall system. In our examples, we use $N = 5$ systems giving us a 10 dimensional state space. The A^i, η_{ij} are generated randomly, with each entry having a Gaussian distribution with mean 0 and variance 10. The sparsity pattern S is also generated randomly by picking 20% of the off-diagonal entries of K and setting them to 0. For both the CON, NCON problems, we initialize the optimizer at the same point $\mathbf{K} = 0$. For the q_∞ norm, we present results comparing the approaches over 100 trials. The q_∞ norm of the solution obtained by the CON approach to that found by NCON, OPT in figure 5.2. We plot histograms of how the q_∞ compares between the CON, NCON and OPT approaches. The red curves show kernel-density estimates of the distribution of values being plotted. The results show that CON consistently outperforms NCON and often achieves performance close to the centralized OPT solution. The x-axis denotes the ratio between objectives on a log scale. The y-axis shows the frequency with which a particular ratio is attained (out of a 100 trials). We also plot a histogram of computation times with the log of ratio of CPU times for the CON and NCON algorithms on the x-axis. Again, in terms of CPU times, the CON approach is consistently superior except for a small number of outliers. For the q_2 norm, we plot the results in figure 5.3. Here, the NCON approach does better and beats the CON approach for most trials. However, in more than 70% of the trials the q_2 norm of the solution found by CON is within 2% of that found by NCON. In terms of computation time, the CON approach retains superiority.

The numerical results indicate that the convex surrogates work well in many cases. However, they do fail in particular cases. In general, the surrogates seem to perform better on the q_∞ norm than the q_2 norm. The initial results are promising but we believe that further analytical and numerical work is required to exactly understand when the convex objectives proposed in this work are good surrogates for the original nonconvex q_2 and q_∞ objectives.

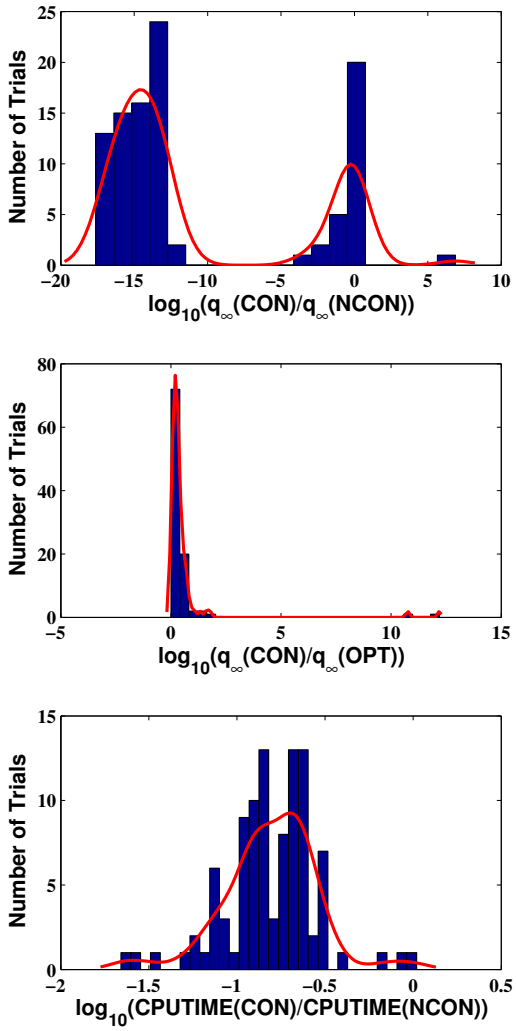


Figure 5.2: Comparison of Algorithms for q_∞-norm Controller Synthesis. The blue bars represent histograms and the red curves kernel density estimates of the distribution of values.

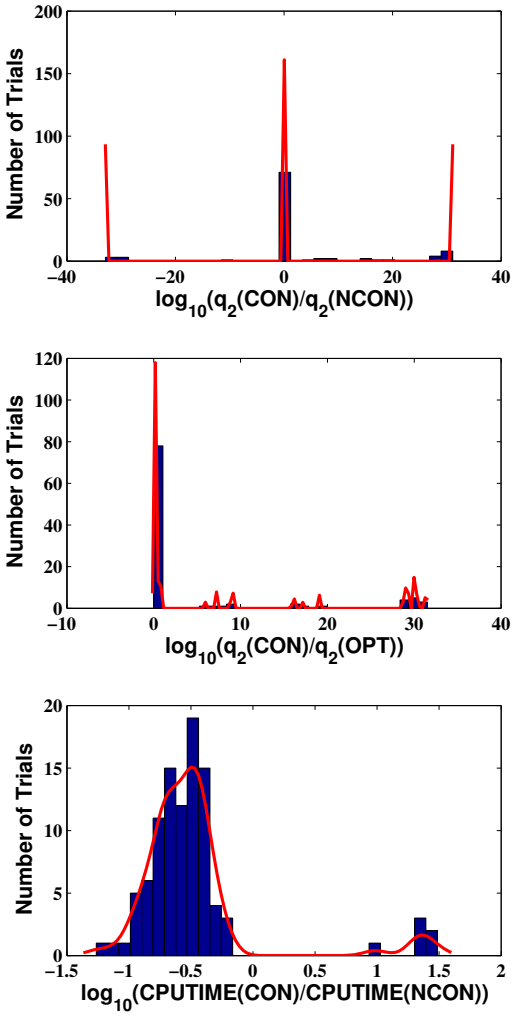


Figure 5.3: Comparison of Algorithms for q_2-norm Controller Synthesis. The blue bars represent histograms and the red curves kernel density estimates of the distribution of values.

5.1.5 Generalization to Nonlinear Systems

We now present a generalization of our approach to nonlinear systems. The essential idea is to study a nonlinear system in terms of sensitivities of system trajectories with respect to disturbances. Consider a control-affine nonlinear discrete-time system:

$$\begin{aligned}\mathbf{x}_1 &= D_0\omega_0\\ \mathbf{x}_{t+1} &= \mathrm{a}_t(\mathbf{x}_t) + \mathrm{B}_t(\mathbf{x}_t)u_t + D_t\omega_t \quad (1 \le t \le N-1)\end{aligned}$$

where $\mathrm{a}_t : \mathbf{R}^n \mapsto \mathbf{R}^n$ and $\mathrm{B} : \mathbf{R}^n \mapsto \mathbf{R}^{n\times n_u}$, $D_t \in \mathbf{R}^{n\times n}, \mathbf{x}_t \in \mathbf{R}^n, w_t \in \mathbf{R}^n, u_t \in \mathbf{R}^{n_u}$. Suppose that 0 is an equilibrium point (if not, we simply translate the coordinates to make this the case). Now we seek to design a controller $u_t = K_t\phi(\mathbf{x}_t)$ where ϕ is any set of fixed "features" of the state on which we want the control to depend that minimizes deviations from the constant trajectory $[0, 0, \ldots, 0]$. We can look at the closed loop system:

$$\mathbf{x}_{t+1} = \mathrm{a}_t(\mathbf{x}_t) + \mathrm{B}_t(\mathbf{x}_t)K_t\phi_t(\mathbf{x}_t) + D_t\omega_t$$

where $\phi_t(\mathbf{x}_t) \in \mathbf{R}^{\mathrm{m}}, K_t \in \mathbf{R}^{n_u\times \mathrm{m}}$. As before, let $\mathbf{K} = \{K_t : 1 \le t \le N-1\}$. Let $F(\mathbf{K})(\boldsymbol{\epsilon})$ denote the (nonlinear) mapping from a sequence of disturbances $\boldsymbol{\epsilon} = [\omega_0, \ldots, \omega_{N-1}]$ to the state space trajectory $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$. The finite-horizon q_∞ norm for a nonlinear system can be defined analogously as for a linear system.

$$\max_{\boldsymbol{\epsilon}\neq 0} \frac{\|F(\mathbf{K})(\boldsymbol{\epsilon})\|}{\|\boldsymbol{\epsilon}\|}. \tag{5.9}$$

Given a state trajectory $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, we can recover the noise sequence as

$$\begin{aligned}\omega_0 &= {D_0}^{-1}\mathbf{x}_1\\ \omega_t &= {D_t}^{-1}\left(\mathbf{x}_{t+1} - \mathrm{a}_t(\mathbf{x}_t) - \mathrm{B}_t(\mathbf{x}_t)K_t\phi_t(\mathbf{x}_t)\right), t > 0\end{aligned} \tag{5.10}$$

Thus the map $F(\mathbf{K})$ is invertible. Let $F(\mathbf{K})^{-1}$ denote the inverse. It can be shown (theorem 5.1.7) that the objective (5.9) (assuming it is finite) can be bounded above by

$$\sup_{\mathbf{X}} \left(\frac{\sum_{i=1}^{nN-1} \sigma_i\left(\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}}\right)\right)}{nN-1} \right)^{nN-1}$$

In the linear case, the maximization over $\mathbf{X}$ is unnecessary since the term being maximized is independent of $\mathbf{X}$. However, for a nonlinear system, the Jacobian of $(F(\mathbf{K}))^{-1}(\mathbf{X})$ is a function of $\mathbf{X}$ and an explicit maximization needs to be performed to compute the objective. Thus, we can formulate the control design problem as

$$\min_{\mathbf{K}\in\mathcal{C}} \sup_{\mathbf{X}} \left(\frac{\sum_{i=1}^{nN-1} \sigma_i \left(\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right) \right)}{nN-1} \right)^{nN-1} \tag{5.11}$$

The convexity of the above objective follows using a very similar proof as the linear case (see theorem 5.1.6). Computing the objective (maximizing over $\mathbf{X}$) in general would be a hard problem, so this result is only of theoretical interest in its current form. However, in future work, we hope to explore the computational aspects of this formulation more carefully.

5.1.6 Discussion and Related Work

There have been three major classes of prior work in synthesizing structured controllers: Frequency domain approaches, dynamic programming and nonconvex optimization approaches. We compare the relative merits of the different approaches in this section.

In frequency domain approaches, problems are typically formulated as follows:

$$\begin{aligned} &\underset{K}{\text{Minimize}} \parallel \text{Closed loop system with feedback K} \parallel \\ &\text{Subject to } K \text{ Stabilizing }, K \in \mathcal{C} \end{aligned}$$

where $\|\cdot\|$ is typically the $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm. In general, these are solved by reparameterizing the problem in terms of a Youla parameter (via a nonlinear transformation), and imposing special conditions on $\mathcal{C}$ (like quadratic invariance) that guarantee that the constraints $\mathcal{C}$ can be translated into convex constraints on the Youla parameter Rotkowitz and Lall [2006]Qi et al. [2004]. There are multiple limitations of these approaches:

(1) Only specific kinds of constraints can be imposed on the controller. Many of the examples have the restriction that the structure of the controller mirrors that of the plant.

(2) They result in infinite dimensional convex programs in general. One can solve them using a sequence of convex programming problems, but these approaches are susceptible to numerical issues and the degree of the resulting controllers may be ill-behaved, leading to

practical problems in terms of implementing them.
(3) The approaches rely on frequency domain notions and cannot handle time-varying systems.
In the special case of poset-causal systems (where the structure of the plant and controller can be described in terms of a partial order Shah [2013]), the problem can be decomposed when the performance metric is the $\mathcal{H}_2$ norm and explicit state-space solutions are available by solving Ricatti equations for subsystems and combining the results. For the $\mathcal{H}_\infty$ norm, a state-space solution using an LMI approach was developed in Scherer [2013].

Another thread of work on decentralized control looks at special cases where dynamic programming techniques can be used in spite of the decentralization constraints. The advantage of these approaches is that they directly handle finite horizon and time-varying approaches. For the LEQG cost-criterion, a dynamic programming approach was developed in Fan et al. [1994] for the case of 1-step delay in a 2-agent decentralized control problem. In Swigart and Lall [2010], the authors show that for the case of 2 agents (a block-lower triangular structure in A, B with 2 blocks) can be solved via dynamic programming. In Lamperski and Lessard [2013], the authors develop a dynamic programming solution that generalizes this and applies to general "partially-nested" systems allowing for both sparsity and delays.

All the above methods work for special structures on the plant and controller (quadratic invariance/partial nestedness) under which decentralized controllers can be synthesized using either convex optimization or dynamic programming methods.

In very recent work Lavaei [2013], the authors pose decentralized control (in the discrete-time, finite horizon, linear quadratic setting) as a rank-constrained semidefinite programming problem. By dropping the rank constraint, one can obtain a convex relaxation of the problem. The relaxed problem provides a solution to the original problem only when the relaxed problem has a rank-1 solution. However, it is unknown when this can be guaranteed, and how a useful controller can be recovered from a higher-rank solution. Further, the SDP posed in this work grows very quickly with the problem dimension.

Our work differs from these previous works in one fundamental way: Rather than looking for special decentralization structures that can be solved tractably under standard control

objectives, we formulate a new control objective that helps us solve problems with *arbitrary* decentralization constraints. In fact, we can handle *arbitrary convex constraints* - decentralization constraints that impose a sparsity pattern on $\mathbf{K}$ are a special case of this. We can also handle time-varying linear systems. Although the objective is nonstandard, we have provided theoretical and numerical evidence that it is a sensible control objective. The only other approaches that handle all these problems are nonconvex approaches Zhai et al. [2001], Apkarian et al. [2008], Lin et al. [2013]. We have shown that our approach outperforms a standard nonconvex approach, both in terms of performance of resulting controller and in computation times.

We also believe that this was the first approach to exploit a fundamental limitation (Bode's sensitivity integral) to develop efficient control design algorithms. The fact that the spectrum of the input output map satisfies a conservation law (the sum of the logs of singular values is fixed) is a limitation which says that reducing some of the singular values is bound to increase the others. However, this limitation allows us to approximate the difficult problem of minimizing the $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm with the easier problem of minimizing a convex surrogate, leading to efficient solution.

5.1.7 Conclusion

We have argued that the framework developed seems promising and overcomes limitations of previous works on computationally tractable approaches to structured controller synthesis. Although the control objective used is non-standard, we have argued why it is a sensible objective, and we also presented numerical examples showing that it produces controllers outperforming other nonconvex approaches. Further, we proved suboptimality bounds that give guidance on when our solution is good even with respect to the original ($\mathcal{H}_2/\mathcal{H}_\infty$) metrics. There are three major directions for future work: 1) Investigating the effect of various objectives in our family of control objectives, 2) Developing efficient solvers for the resulting convex optimization problems and 3) Deriving computationally efficient algorithms for nonlinear systems.

5.1.8 Appendix

Penalizing Control Effort

A more direct approach is to augment the state to include the controls. We define an augmented problem with $\bar{\mathbf{x}}_t \in \mathbf{R}^{n_s+n_u}, \overline{\omega}_t \in \mathbf{R}^{n_s+n_u}$.

$$\overline{A}_t = \begin{pmatrix} A_t & 0 \\ 0 & 0 \end{pmatrix}, \overline{B}_t = \begin{pmatrix} B_t \\ R_t \end{pmatrix}, \overline{D}_t = \begin{pmatrix} D_t & 0 \\ 0 & \gamma I \end{pmatrix}$$

$$\overline{\mathbf{x}}_{t+1} = \overline{A}_t \overline{\mathbf{x}}_t + \overline{B}_t u_t + \overline{D}_t \overline{\omega}_t$$

Partitioning the new state $\overline{\mathbf{x}}_t = \begin{pmatrix} \mathbf{x}_t \\ \tilde{\mathbf{x}}_t \end{pmatrix}$, $\overline{\omega}_t = \begin{pmatrix} \omega_t \\ \tilde{\omega}_t \end{pmatrix}$, we have:

$$\mathbf{x}_{t+1} = A_t \mathbf{x}_t + B_t u_t + D_t \omega_t, \tilde{\mathbf{x}}_{t+1} = R_t u_t + \gamma \tilde{\omega}_t$$

Given this,

$$\sum_{t=1}^{N} \overline{\mathbf{x}}_t{}^T \overline{\mathbf{x}}_t = \sum_{t=1}^{N} \mathbf{x}_t{}^T \mathbf{x}_t + \sum_{t=1}^{N-1} \left(R_t u_t + \gamma \tilde{\omega}_t\right)^T \left(R_t u_t + \gamma \tilde{\omega}_t\right) + \gamma^2 \omega_0{}^T \omega_0$$

In the limit $\gamma \to 0$, we recover the standard LQR cost. However, setting $\gamma = 0$ violates the condition of invertibility. Thus, solving the problem with an augmented state $\overline{\mathbf{x}} \in \mathbf{R}^{n_u+\nu}$, $\overline{\omega} \in \mathbf{R}^{n_u+\nu}$,

$$\overline{A}_t = \begin{pmatrix} A_t & 0 \\ 0 & 0 \end{pmatrix}, \overline{B}_t = \begin{pmatrix} B_t \\ R_t \end{pmatrix}, \overline{D}_t = \begin{pmatrix} D_t & 0 \\ 0 & \gamma I \end{pmatrix}$$

solves the problem with a quadratic control cost in the limit $\gamma \to 0$. The caveat is that the problems (5.5)(5.6) become increasingly ill-conditioned as $\gamma \to 0$. However, we should be able to solve the problem for a small value of γ, which models the quadratic controls cost closely but still leads to a sufficiently well-conditioned problem that we can solve numerically.

Dynamic Output Feedback

So far, we have described the problem in terms of direct state feedback $u_t = K_t \mathbf{x}_t$. However, we can also model output feedback $u_t = K_t C_t \mathbf{x}_t$ by simply defining $\tilde{K}_t = K_t C_t$ where

$C_t \in \mathbf{R}^{\mathrm{m} \times n_s}$ is a measurement matrix that produces m measurements given the state. Convex constraints on K_t will translate into convex constraints on $\tilde{K_t}$, since $\tilde{K_t}$ is a linear function of K_t. If we wanted to allow our controls to depend on the previous k measurements (dynamic output feedback), we simply create an augmented state $\overline{\mathbf{x}_t} = \begin{pmatrix} \mathbf{x}_t \\ \vdots \\ \mathbf{x}_{t-\mathrm{k}} \end{pmatrix}$. Then, we can define $K_t \in \mathbf{R}^{n_u \times \mathrm{km}}$ and

$$\tilde{K_t} = K_t \begin{pmatrix} C_t & 0 & \dots & 0 \\ 0 & C_{t-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & C_{t-\mathrm{k}} \end{pmatrix}$$

and an augmented dynamics

$$\overline{A}_t = \begin{pmatrix} A_t & 0 & \dots & 0 & 0 \\ I & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{pmatrix}, \overline{B}_t = \begin{pmatrix} B_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\overline{D}_t = \begin{pmatrix} D_t & 0 & \dots & 0 & 0 \\ 0 & \gamma I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \gamma I \end{pmatrix}$$

Again, we need to set $\gamma = 0$ to exactly match the standard output feedback problem but that violates the assumption of invertibility. We can consider taking $\gamma \to 0$ and recovering the solution as a limiting case, as in the previous section.

Proofs

Theorem 5.1.4.

$$\begin{aligned} & q_\infty(\mathbf{K}) = \sigma_{\max}\left(F\left(\mathbf{K}\right)\right) \\ & \leq \prod_{t=0}^{N-1} \det\left(D_t\right) \left(\frac{\sum_{i=1}^{n_s N-1} \sigma_i \left(F\left(\mathbf{K}\right)^{-1} \right)}{n_s N - 1} \right)^{n_s N-1} \end{aligned}$$

Proof. Since $F\left(\mathbf{K}\right)$ is a block lower triangular matrix (a reflection of the fact that we have a causal linear system) , its determinant is simply the product of determinants of diagonal blocks: $\det\left(F\left(\mathbf{K}\right)\right) = \prod_t \det\left(D_t\right) = c$ *independent* of the values of $\tilde{A}_t$. In fact, this result is a generalization of Bode's classical sensitivity integral result and has been studied in Iglesias [2001]. Since the product of singular values is equal to the determinant, we have

$$\sigma_{\max}\left(F\left(\mathbf{K}\right)\right) = \frac{c}{\prod_{i=2}^{nN} \sigma_i\left(F\left(\mathbf{K}\right)\right)} = c \prod_{i=1}^{nN-1} \sigma_i\left(F\left(\mathbf{K}\right)^{-1}\right)$$

where the last equality follows because the singular values of $F\left(\mathbf{K}\right)^{-1}$ are simply reciprocals of the singular values of $F\left(\mathbf{K}\right)$. The result now follows using the AM-GM inequality. □

Theorem 5.1.5.

$$q_2(\mathbf{K}) \leq nN \left(\prod_{t=0}^{N-1} \det\left(D_t\right) \right)^2 \left(\sigma_{\max}\left(F\left(\mathbf{K}\right)^{-1} \right) \right)^{2(n_s N-1)}$$

Proof. Let $\prod_{t=0}^{N-1} \det\left(D_t\right) = c$. From the above argument, we can express $\sigma_i\left(F\left(\mathbf{K}\right)\right)$ as

$$c \prod_{j \neq nN-i+1} \sigma_j\left(\left(F\left(\mathbf{K}\right)\right)^{-1} \right) \leq c \left(\sigma_{\max}\left(F\left(\mathbf{K}\right)^{-1} \right) \right)^{(n_s N-1)}.$$

The expression for $q_2(\mathbf{K})$ is

$$\sum_{i=1}^{nN} \left(\sigma_i\left(F\left(\mathbf{K}\right)\right)\right)^2 \leq nNc^2 \left(\sigma_{\max}\left(F\left(\mathbf{K}\right)^{-1} \right) \right)^{2(n_s N-1)}.$$

□

Theorem 5.1.6. *For the nonlinear system described in* (5.10)*, the function*

$$\sup_{\mathbf{X}} \left(\frac{\sum_{i=1}^{nN-1} \sigma_i \left(\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right) \right)}{nN-1} \right)^{nN-1}$$

is convex in $\mathbf{K}$*.*

Proof. First fix $\boldsymbol{\epsilon}$ to an arbitrary value. From (5.10), we know that $\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right)$ is of the form

$$\begin{bmatrix} {D_0}^{-1} & 0 & \dots & \dots & 0 \\ -{D_1}^{-1} \frac{\partial \omega_1}{\partial \mathbf{x}_1} & {D_1}^{-1} & \dots & \dots & 0 \\ 0 & -{D_2}^{-1} \frac{\partial \omega_2}{\partial \mathbf{x}_2} & {D_2}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & {D_{N-1}}^{-1} \end{bmatrix}$$

Since $\omega_t = {D_t}^{-1} \left(\mathbf{x}_{t+1} - \mathrm{a}_t(\mathbf{x}_t) - \mathrm{B}_t(\mathbf{x}_t) K_t \phi_t(\mathbf{x}_t) \right)$, ω_t is an affine function of $\mathbf{K}$. Hence, so is $\frac{\partial \omega_t}{\partial \mathbf{x}_t}$, for any t. Thus, the overall matrix $\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right) = M(\mathbf{K})$ is an affine function of $\mathbf{K}$. Thus, by composition properties, $\left(\frac{\sum_{i=1}^{nN-1} \sigma_i(M(\mathbf{K}))}{nN-1} \right)^{nN-1}$ is a convex function of $\mathbf{K}$ for any fixed $\mathbf{X}$. Taking a supremum over all $\mathbf{X}$ preserves convexity, since the pointwise supremum of a set of convex functions is convex. □

Theorem 5.1.7. *Consider the nonlinear system described in* (5.10)*. Suppose that* $\sup_{\boldsymbol{\epsilon} \neq 0} \frac{\|F(\mathbf{K})(\boldsymbol{\epsilon})\|}{\|\boldsymbol{\epsilon}\|}$ *is finite and the supremum is achieved at* $\boldsymbol{\epsilon}^* \neq 0$ *for all values of* $\mathbf{K}$*. Then,* $\sup_{\boldsymbol{\epsilon} \neq 0} \frac{\|F(\mathbf{K})(\boldsymbol{\epsilon})\|}{\|\boldsymbol{\epsilon}\|}$ *is bounded above by*

$$\sup_{\mathbf{X}} \left(\frac{\sum_{i=1}^{nN-1} \sigma_i \left(\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right) \right)}{nN-1} \right)^{nN-1}$$

Proof. By theorem 5.1.8, $\sup_{\boldsymbol{\epsilon} \neq 0} \frac{\|F(\mathbf{K})(\boldsymbol{\epsilon})\|}{\|\boldsymbol{\epsilon}\|}$ is bounded above by

$$\sup_{\boldsymbol{\epsilon} \neq 0} \sigma_{\max} \left(\frac{\partial F(\mathbf{K})(\boldsymbol{\epsilon})}{\partial \boldsymbol{\epsilon}} \right).$$

Now, $M(\mathbf{K}) = \frac{\partial F(\mathbf{K})(\boldsymbol{\epsilon})}{\partial \boldsymbol{\epsilon}}$ is a lower-triangular matrix (since we have a causal system) and the diagonal blocks are given by D_t. Thus, $\det(M(\mathbf{K})) = \prod_{t=0}^{N-1} \det(D_t) = c$, and we can rewrite $\sigma_{\max}(M(\mathbf{K}))$ as $c \prod_{i=1}^{nN-1} \sigma_i \left((M(\mathbf{K}))^{-1} \right)$. By the rules of calculus, we know that

$$(M(\mathbf{K}))^{-1} = \left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right)_{\mathbf{X} = F(\mathbf{K})(\boldsymbol{\epsilon})}$$

Thus, the above objective reduces to

$$\sup_{\boldsymbol{\epsilon}\neq 0} \prod_{i=1}^{nN-1} \sigma_i \left(\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right)_{\mathbf{X}=F(\mathbf{K})(\boldsymbol{\epsilon})} \right).$$

Given any $\mathbf{X}$, we can find $\boldsymbol{\epsilon}$ such that $\mathbf{X} = F(\mathbf{K})(\boldsymbol{\epsilon})$ (simply choose $\boldsymbol{\epsilon} = (F(\mathbf{K}))^{-1}(\mathbf{X})$). Thus, the above quantity is equal to

$$\sup_{\mathbf{X}} \prod_{i=1}^{nN-1} \sigma_i \left(\left(\frac{\partial (F(\mathbf{K}))^{-1}(\mathbf{X})}{\partial \mathbf{X}} \right) \right).$$

The result now follows using the AM-GM inequality. □

Theorem 5.1.8. *Let $g(\mathbf{y}) : \mathbf{R}^l \mapsto \mathbf{R}^p$ be any differentiable function. If the function $\frac{\|g(\mathbf{y})\|_2}{\|\mathbf{y}\|_2}$ attains its maximum at $\mathbf{y}^*$,*

$$\sup_{y} \frac{\|g(\mathbf{y})\|_2}{\|\mathbf{y}\|_2} \leq \sup_{\mathbf{y}\neq 0} \sigma_{\max} \left(\frac{\partial g(\mathbf{y})}{\partial \mathbf{y}} \right) \quad \forall\, \mathbf{y} \neq 0.$$

Proof. $\log\left(\frac{\|g(\mathbf{y})\|^2}{\|\mathbf{y}\|^2}\right)$ is differentiable at any $\mathbf{y} \neq 0$ and hence at $\mathbf{y} = \mathbf{y}^*$. Since this is an unconstrained optimization problem, we can write the optimality condition (0 gradient):

$$\frac{2\left(\frac{\partial g(\mathbf{y})}{\partial \mathbf{y}}\right)_{\mathbf{y}=\mathbf{y}^*} g(\mathbf{y}^*)}{\|g(\mathbf{y}^*)\|^2} = \frac{2\,\mathbf{y}^*}{\|\mathbf{y}^*\|^2}$$

Taking the ℓ_2 norm on both sides, we get

$$\frac{\left\| \left(\frac{\partial g(\mathbf{y})}{\partial \mathbf{y}}\right)_{\mathbf{y}=\mathbf{y}^*} g(\mathbf{y}^*) \right\|}{\|g(\mathbf{y}^*)\|} = \frac{\|g(\mathbf{y}^*)\|}{\|\mathbf{y}^*\|}$$

Since $\mathbf{y}^*$ maximizes $\frac{\|g(\mathbf{y})\|}{\|\mathbf{y}\|}$, for any $\mathbf{y} \neq 0$, we have:

$$\begin{aligned} \frac{\|g(\mathbf{y})\|}{\|\mathbf{y}\|} &\leq \frac{\|g(\mathbf{y}^*)\|}{\|\mathbf{y}^*\|} = \frac{\left\| \left(\frac{\partial g(\mathbf{y})}{\partial \mathbf{y}}\right)_{\mathbf{y}=\mathbf{y}^*} g(\mathbf{y}^*) \right\|}{\|g(\mathbf{y}^*)\|} \\ &\leq \sigma_{\max} \left(\frac{\partial g(\mathbf{y})}{\partial \mathbf{y}} \right)_{\mathbf{y}=\mathbf{y}^*} \leq \max_{\mathbf{y}\neq 0} \sigma_{\max} \left(\frac{\partial g(\mathbf{y})}{\partial \mathbf{y}} \right). \end{aligned}$$

Taking supremum over y on both sides, we get the result. □

5.2 Convex Stochastic Policy Optimization for General Dynamical Systems

In this section, we present a new convex-optimization based approach to the control of discrete-time dynamical systems. Stochastic Optimal Control of nonlinear systems in general is a hard problem and the only known general approach is based on dynamic programming, which scales exponentially with the size of the state space. Algorithms that approximate the solution of the dynamic program directly (approximate dynamic programming) have been successful in various domains, but scaling these approaches to high dimensional continuous state control problems has been challenging [Zhong, 2013]. In this section, we pursue the alternate approach of policy search or policy gradient methods [Baxter and Bartlett, 2001]. These algorithms have the advantage that they are directly optimizing the performance of a control policy (using gradient descent) as opposed to a surrogate measure like the error in the solution to the Bellman equation. They have been used successfully for applications in robotics [Peters and Schaal, 2008] and are closely related to the recent framework of path integral control [**?**Theodorou et al., 2010b]. However, in all of these approaches, there were no guarantees made regarding the optimality of the policy that the algorithm converges to (even in the limit of infinite sampling) or the rate of convergence.

In this work, we develop the *first* policy gradient algorithms that achieve the *globally* optimal solutions to policy optimization problems. We do this by proving that under certain assumptions, the policy optimization problem is a convex optimization problem. This can then be solved used stochastic convex optimization methods, which have guaranteed convergence to the optimal solution (in expectation and with high probability) in polynomial time. There are two ways of taking gradients in this approach: One of them leads to model-free updates and is very similar to the updates in path integral control [Theodorou et al., 2010b]. The other approach leads to a model-based algorithm, which typically converges faster than the model-free variant but requires a model of the system dynamics. All of the approaches work in both finite and infinite horizon settings.

Problem Setup

We deal with arbitrary discrete-time dynamical systems of the form

$$\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_{N-1} \end{pmatrix} \sim \mathbb{P}_\epsilon$$

$$\mathbf{x}_1 = \epsilon_0, \mathbf{x}_{t+1} = \mathcal{F}\left(\mathbf{x}_t, \mathbf{y}_t, \epsilon_t, t\right) \quad t = 1, 2, \ldots, N-1 \tag{5.12}$$

$$\mathbf{y}_t = u_t + \omega_t, \omega_t \sim \mathcal{N}\left(0, \Sigma_t\right) \quad t = 1, 2, \ldots, N-1 \tag{5.13}$$

where $\mathbf{x}_t \in \mathbf{R}^{n_s}$ denotes the state, $\mathbf{y}_t \in \mathbf{R}^{n_u}$ the effective control input, $\epsilon_t \in \mathbf{R}^p$ external disturbances, $u_t \in \mathbf{R}^{n_u}$ the actual control input, $\omega_t \in \mathbf{R}^{n_u}$ the control noise, $\mathcal{F} : \mathbf{R}^{n_s} \times \mathbf{R}^{n_u} \times \mathbf{R}^p \times \{1, \ldots, N-1\} \mapsto \mathbf{R}^{n_s}$ and $\mathcal{N}\left(\mu, \Sigma\right)$ a Gaussian distribution with mean μ and covariance matrix Σ. Equation (6.10) can model any noisy discrete-time dynamical system, since $\mathcal{F}$ can be any function of the current state, control input and external disturbance (noise). However, we require that all the control dimensions are affected by Gaussian noise as in (5.13). This can be thought of either as real actuator noise or artificial exploration noise.

We will work with costs that are a combination of arbitrary state costs and quadratic control costs:

$$\sum_{t=1}^{N} \ell_{\mathrm{t}}(\mathbf{x}_t) + \sum_{t=0}^{N-1} \frac{{u_t}^T R_t u_t}{2} \tag{5.14}$$

Further, we will assume that the control-noise is non-degenerate, that is Σ_t is full rank for all $0 \leq t \leq N-1$. We denote $S_t = {\Sigma_t}^{-1}$.

We seek to design feedback policies

$$u_t = K_t \phi\left(\mathbf{x}_t, t\right), \phi : \mathbf{R}_s^n \times \{1, 2, \ldots, N-1\} \mapsto \mathbf{R}^r, K_t \in \mathbf{R}^{\nu \times r} \tag{5.15}$$

to minimize the accumulated cost (5.14). We will assume that the features ϕ are fixed and we seek to optimize the policy parameters

$$\mathbf{K} = \{K_t : t = 1, 2, \ldots, N-1\}.$$

The stochastic optimal control problem we consider is defined as follows:

$$\begin{aligned}
&\underset{\mathbf{K}}{\text{Minimize}} \underset{\boldsymbol{\epsilon}\sim\mathbb{P}_\epsilon,\omega_t\sim\mathcal{N}(0,\Sigma_t)}{\mathrm{E}} \left[\exp\left(\alpha\left(\sum_{t=1}^{N}\ell_{\mathrm{t}}(\mathbf{x}_t)+\sum_{t=1}^{N-1}\frac{{u_t}^T R_t u_t}{2}\right)\right)\right] \\
&\text{Subject to } \mathbf{x}_1=\epsilon_0, \mathbf{x}_{t+1}=\mathcal{F}\left(\mathbf{x}_t,\mathbf{y}_t,\epsilon_t,t\right) \quad t=1,2,\ldots,N-1 \\
&\qquad \mathbf{y}_t=u_t+\omega_t, \omega_t\sim\mathcal{N}\left(0,\Sigma_t\right) \quad t=1,2,\ldots,N-1
\end{aligned} \tag{5.16}$$

This is exactly the same as the formulation in Risk Sensitive Markov Decision Processes Marcus et al. [1997], the only change being that we have explicitly separated the noise appearing in the controls from the noise in the dynamical system overall. In this formulation, the objective depends not only on the average behavior of the control policy but also on variance and higher moments (the tails of the distribution of costs). This has been studied for linear systems under the name of LEQG control Speyer et al. [1974]. α is called a risk factor: Large positive values of α result in strongly risk-averse policies while large negative values result in risk-seeking policies. In our formulation, we will need a certain minimum degree of risk-aversion for the resulting policy optimization problem to be convex.

5.2.1 Main Technical Results

Theorem 5.2.1. *If $\alpha R_t \succeq (\Sigma_t)^{-1} = S_t$ for $t=1,\ldots,N-1$, then the optimization problem* (6.9) *is convex.*

Proof. We first show that for a fixed $\boldsymbol{\epsilon}$, the quantity

$$\underset{\omega_t\sim\mathcal{N}(0,\Sigma_t)}{\mathrm{E}} \left[\exp\left(\alpha\left(\sum_{t=1}^{N}\ell_{\mathrm{t}}(\mathbf{x}_t)+\sum_{t=1}^{N-1}\frac{{u_t}^T R_t u_t}{2}\right)\right)\right]$$

is a convex function of $\mathbf{K}$. Then, by the linearity of expectation, so is the original objective. We can write down the above expectation as:

$$\begin{aligned}
&\underset{\mathbf{y}_t\sim\mathcal{N}(K_t\phi(\mathbf{x}_t,t),\Sigma_t)}{\mathrm{E}} \left[\exp\left(\alpha\left(\sum_{t=1}^{N}\ell_{\mathrm{t}}(\mathbf{x}_t)+\sum_{t=1}^{N-1}\frac{{u_t}^T R_t u_t}{2}\right)\right)\right] \\
&= \int \frac{\exp\left(-\sum_{t=1}^{N-1}\frac{\|\mathbf{y}_t-K_t\phi\left(\mathbf{x}_t,t\right)\|^2_{S_t}}{2}\right)}{\prod_{t=1}^{N-1}\sqrt{(2\pi)^{n_u}\det\left(\Sigma_t\right)}} \exp\left(\alpha\left(\sum_{t=1}^{N}\ell_{\mathrm{t}}(\mathbf{x}_t)+\sum_{t=1}^{N-1}\frac{\|K_t\phi\left(\mathbf{x}_t,t\right)\|^2_{R_t}}{2}\right)\right) d\mathbf{Y}
\end{aligned}$$

In the above integral, $\mathbf{x}_t$ can be written as a deterministic function of $\boldsymbol{\epsilon}, \mathbf{Y}$ for any $t \in \{1, \ldots, N\}$ using (6.10). The term inside the exponential can be written as

$$-\left(\sum_{t=1}^{N-1} \frac{\|\mathbf{y}_t\|_{S_t}^2}{2}\right) + \alpha\left(\sum_{t=1}^{N} \ell_{\mathrm{t}}(\mathbf{x}_t)\right) + \sum_{t=1}^{N-1} \frac{\operatorname{tr}\left(\left(K_t{}^T\left(\alpha R_t - S_t\right) K_t\right) \phi\left(\mathbf{x}_t, t\right) \phi\left(\mathbf{x}_t, t\right)^T\right)}{2} - \mathbf{y}_t{}^T S_t \phi\left(\mathbf{x}_t, t\right) K_t$$

The terms on the first line don't depend on $\mathbf{K}$. The function $\left(K_t{}^T\left(\alpha R_t - S_t\right) K_t\right)$ is $\succeq$-convex when $\alpha R_t - S_t \succeq 0$ and hence the first term on the second line is convex in $\mathbf{K}$. The second term is linear in $\mathbf{K}$ and hence convex. Since exp is a convex and increasing function, the composed function (which is the integrand) is convex as well in $\mathbf{K}$. Thus, the integral is convex in $\mathbf{K}$. □

We can add arbitrary further convex constraints and penalties on $\mathbf{K}$ without affecting convexity.

Corollary 1. *The problem*

$$\min_{\mathbf{K}} \quad \mathop{\mathrm{E}}_{\boldsymbol{\epsilon}\sim\mathbb{P}_\epsilon, \omega_t \sim \mathcal{N}(0,\Sigma_t)} \left[\exp\left(\alpha\left(\sum_{t=1}^{N} \ell_{\mathrm{t}}(\mathbf{x}_t) + \sum_{t=1}^{N-1} \frac{u_t{}^T R_t u_t}{2}\right)\right)\right]$$

$$\textit{Subject to} \qquad (6.10), (5.13), \mathbf{K} \in \mathcal{C} \tag{5.17}$$

is a convex optimization problem for any arbitrary convex set $\mathcal{C} \subset \mathbf{R}^{n_u \times r \times (N-1)}$ *if* $\alpha R_t \succeq \Sigma_t \quad \forall t$.

Extension to Infinite Horizon Systems

The results of the previous section can also be proven using a dynamic programming approach. For a given policy parameterized by $\mathbf{K}$, the expected cost can be written using a recursive relationship.

Before establishing the convexity results, we prove the following lemma from which all the subsequent results follow easily.

lemma 1. *If* $h\left(\mathbf{x}, \mathbf{K}\right)$ *is a convex function of* $\mathbf{K}\, \forall \mathbf{x}$ *and* $\alpha R_t \succeq {\Sigma_t}^{-1}$, *then so is*

$$\frac{\left(K_t \mathbf{x}_t\right)^T R_t \left(K_t \mathbf{x}_t\right)}{2} + \frac{1}{\alpha} \log\left(\mathop{\mathrm{E}}_{\omega_t, \epsilon_t}\left[\exp\left(\alpha h\left(\mathcal{F}\left(\mathbf{x}_t, K_t \mathbf{x}_t + \omega_t, \epsilon_t, t\right), \mathbf{K}\right)\right)\right]\right) \forall \mathbf{x}_t$$

Proof. The RHS is equal to

$$\frac{1}{\alpha}\log\left(\underset{\mathbf{y}_t\sim\mathcal{N}(K_t\mathbf{x}_t,\Sigma),\epsilon_t}{\mathrm{E}}\left[\exp\left(\alpha h\left(\mathcal{F}\left(\mathbf{x}_t,\mathbf{y}_t,\epsilon_t,t\right),\mathbf{K}\right)+\frac{\left(K_t\mathbf{x}_t\right)^T\left(\alpha R_t\right)\left(K_t\mathbf{x}_t\right)}{2}\right)\right]\right)=$$

$$\frac{1}{\alpha}\log\left(\int\frac{\exp\left(-\frac{\left(\mathbf{y}_t-K_t\mathbf{x}_t\right)^T{\Sigma_t}^{-1}\left(\mathbf{y}_t-K_t\mathbf{x}_t\right)}{2}+\alpha h\left(\mathcal{F}\left(\mathbf{x}_t,\mathbf{y}_t,\epsilon_t,t\right),\mathbf{K}\right)+\frac{\left(K_t\mathbf{x}_t\right)^T\left(\alpha R_t\right)\left(K_t\mathbf{x}_t\right)}{2}\right)}{\sqrt{\left(2\pi\right)^{n_u}\det\left(\Sigma_t\right)}}\,\mathrm{d}\,\mathbf{y}_t\right).$$

The term inside the exponent depends on is a convex function of $\mathbf{K}$ since $\alpha R_t \succeq {\Sigma_t}^{-1}$, the term inside the exponent is a convex function of $\mathbf{K}$. Hence, by composition, so is the overall function for every value of $\mathbf{x}_t$. □

Theorem 5.2.1 (Infinite Horizon Convexity)**.** *The following problems are convex optimization problems:*

$$\mathrm{IH}:\min_{\mathbf{K}}\lim_{N\to\infty}\frac{1}{\alpha N}\log\left(\underset{\boldsymbol{\epsilon}\sim\mathbb{P}_\epsilon,\omega_t\sim\mathcal{N}(0,\Sigma_t)}{\mathrm{E}}\left[\exp\left(\alpha\left(\sum_{t=1}^{N}\ell_{\mathrm{t}}(\mathbf{x}_t)+\sum_{t=1}^{N-1}\frac{{u_t}^T R_t u_t}{2}\right)\right)\right]\right) \tag{5.18}$$

$$\mathrm{FE}:\min_{\mathbf{K}}\frac{1}{\alpha}\log\left(\underset{\substack{\boldsymbol{\epsilon}\sim\mathbb{P}_\epsilon,\omega_t\sim\mathcal{N}(0,\Sigma_t)\\ \mathrm{N_e}=\min\{t:\mathbf{x}_t\in\mathcal{T}\}}}{\mathrm{E}}\left[\exp\left(\alpha\left(\sum_{t=1}^{\mathrm{N_e}-1}\ell_{\mathrm{t}}(\mathbf{x}_t)+\ell_{\mathrm{f}}\left(\mathbf{x}_{\mathrm{N_e}}\right)+\sum_{t=1}^{\mathrm{N_e}-1}\frac{{u_t}^T R_t u_t}{2}\right)\right)\right]\right),\mathcal{T}\subset\mathcal{X} \tag{5.19}$$

subject to the constraints

$$\mathbf{x}_1=\epsilon_0,\mathbf{x}_{t+1}=\mathcal{F}\left(\mathbf{x}_t,\mathbf{y}_t,\epsilon_t,t\right),\mathbf{y}_t=u_t+\omega_t,\omega_t\sim\mathcal{N}\left(0,\Sigma_t\right)\quad t=1,2,\ldots,N-1$$

$$\mathbf{K}\in\mathcal{C}$$

Proof. Follows by writing the policy specific Bellman equation for each problem and invoking lemma (1). □

5.3 Applications

5.3.1 Structured Controller Design for Linear Systems

Consider a linear dynamical systems $\mathbf{x}_{t+1}=A_t\mathbf{x}_t+B_t\mathbf{y}_t,\mathbf{x}_1=0$ (we assume a noiseless system here for brevity). We seek to synthesize linear state feedback policies $u_t=K_t\mathbf{x}_t$ with structural constraints $\mathbf{K}\in\mathcal{C}$. If the state costs are quadratic: $\ell_{\mathrm{t}}(\mathbf{x}_t)=\frac{{\mathbf{x}_t}^T\mathrm{Q}_t\,\mathbf{x}_t}{2}$, the expectations in (5.17) can be computed analytically. We can write down

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ B_1 & 0 & 0 & \dots & 0 \\ A_1B_1 & B_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \prod_{t=N-1}^{1} A_tB_1 & \prod_{t=N-1}^{2} A_tB_2 & \prod_{t=N-1}^{3} A_tB_3 & \dots & B_{N-1} \end{pmatrix}}_{\mathcal{M}} \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{N-1} \end{pmatrix} = \mathcal{M}\mathbf{Y}$$

where $\mathcal{M}$ is an $\mathbf{R}^{Nn_s \times (N-1)n_u}$ matrix that is independent of $\mathbf{K}$. Also, define:

$$\mathcal{S} = \begin{pmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & S_{N-1} \end{pmatrix} \in \mathbf{R}^{m(N-1)\times m(N-1)}, \mathcal{K} = \begin{pmatrix} K_1 & 0 & \dots & 0 & 0 \\ 0 & K_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & K_{N-1} & 0 \end{pmatrix} \in \mathbf{R}^{m(N-1)\times nN}$$

$$\mathcal{R} = \begin{pmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & R_{N-1} \end{pmatrix} \in \mathbf{R}^{m(N-1)\times m(N-1)}, \mathcal{Q} = \begin{pmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & Q_N \end{pmatrix} \in \mathbf{R}^{nN\times nN}$$

Let $c = \sqrt{(2\pi)^{n_u(N-1)} \prod_{t=1}^{N-1} \det(\Sigma_t)}$. The expectation becomes

$$\begin{aligned}
&\frac{1}{c}\int \exp\left(-\sum_{t=1}^{N-1}\frac{\|\mathbf{y}_t - K_t\mathbf{x}_t\|_{S_t}^2}{2}\right)\exp\left(\alpha\left(\sum_{t=1}^{N-1}\frac{\mathbf{x}_t{}^T\left(Q_t + K_t{}^T R_t K_t\right)\mathbf{x}_t}{2} + \frac{\mathbf{x}_N{}^T Q_N \mathbf{x}_N}{2}\right)\right)\mathrm{d}\,\mathbf{Y} \\
&= \frac{1}{c}\int \exp\left(-\sum_{t=1}^{N-1}\frac{\|\mathbf{y}_t - K_t\mathbf{x}_t\|_{S_t}^2}{2}\right)\exp\left(\alpha\left(\sum_{t=1}^{N-1}\frac{\mathbf{x}_t{}^T\left(Q_t + K_t{}^T R_t K_t\right)\mathbf{x}_t}{2} + \frac{\mathbf{x}_N{}^T Q_N \mathbf{x}_N}{2}\right)\right)\mathrm{d}\,\mathbf{Y} \\
&= \frac{1}{c}\int \exp\left(-\frac{1}{2}\mathbf{Y}^T\mathcal{S}\mathbf{Y} + \mathbf{Y}^T\mathcal{S}\mathcal{K}\mathbf{X} - \frac{1}{2}\mathbf{X}^T\mathcal{K}^T\mathcal{S}\mathcal{K}\mathbf{X} + \frac{1}{2}\alpha\mathbf{X}^T\left(\mathcal{K}^T\mathcal{R}\mathcal{K} + \mathcal{Q}\right)\mathbf{X}\right)\mathrm{d}\,\mathbf{Y} \\
&= \frac{1}{c}\int \exp\left(-\frac{1}{2}\mathbf{Y}^T\left(\mathcal{S} + \mathcal{S}\mathcal{K}\mathcal{M} + \mathcal{M}^T\mathcal{K}^T\mathcal{S}^T + \mathcal{M}^T\left(\mathcal{K}^T\left(\alpha\mathcal{R} - \mathcal{S}\right)\mathcal{K} + \alpha\mathcal{Q}\right)\mathcal{M}\right)\mathbf{Y}\right)\mathrm{d}\,\mathbf{Y} \\
&= \begin{cases} \infty \text{ if } \mathcal{S} - \mathcal{S}\mathcal{K}\mathcal{M} - \mathcal{M}^T\mathcal{K}^T\mathcal{S}^T \succeq \mathcal{M}^T\left(\mathcal{K}^T\left(\alpha\mathcal{R} - \mathcal{S}\right)\mathcal{K} + \alpha\mathcal{Q}\right)\mathcal{M} \\ \dfrac{\det\left(\left(\mathcal{S} - \mathcal{S}\mathcal{K}\mathcal{M} - \mathcal{M}^T\mathcal{K}^T\mathcal{S}^T - \mathcal{M}^T\left(\mathcal{K}^T(\alpha\mathcal{R} - \mathcal{S})\mathcal{K} + \alpha\mathcal{Q}\right)\mathcal{M}\right)^{-1}\right)}{\sqrt{(2\pi)^{n_u(N-1)}\prod_{t=1}^{N-1}\det(\Sigma_t)}} \text{ otherwise} \end{cases}
\end{aligned}$$

When $\alpha\mathcal{R} \succeq \mathcal{S}$, the above problem is a convex optimization problem and is equivalent to the following determinant maximization problem:

$$\begin{aligned} \underset{\mathbf{K}}{\text{Maximize}} \log\left(\det\left(\mathcal{S} - \mathcal{S}\mathcal{K}\mathcal{M} - \mathcal{M}^T\mathcal{K}^T\mathcal{S}^T - \mathcal{M}^T\left(\mathcal{K}^T\left(\alpha\mathcal{R} - \mathcal{S}\right)\mathcal{K} + \alpha\mathcal{Q}\right)\mathcal{M}\right)\right) \\ \text{Subject to } \mathcal{S} - \mathcal{S}\mathcal{K}\mathcal{M} - \mathcal{M}^T\mathcal{K}^T\mathcal{S}^T - \mathcal{M}^T\left(\mathcal{K}^T\left(\alpha\mathcal{R} - \mathcal{S}\right)\mathcal{K} + \alpha\mathcal{Q}\right)\mathcal{M} \succeq 0 \end{aligned}$$

5.3.2 Learning Neural Networks

This approach can also be applied to learning neural networks. In this approach, the time-steps corresponds to layers of the neural network. In each step, the state (neural activations) are passed through a linear mapping and an component-wise nonlinear transfer function:

$$\mathbf{x}_{t+1} = h(K_t\mathbf{x}_t), \mathbf{x}_1 = \alpha$$

where h is applied component-wise (common examples include the sigmoid and tanh functions). To put this in our framework, we assume that the neural inputs are subject to noise, so that

$$\mathbf{x}_{t+1} = h(\mathbf{y}_t), \mathbf{y}_t = u_t + \omega_t, u_t = K_t\mathbf{x}_t.$$

Note that in our framework this corresponds to an $\mathcal{F}\left(\mathbf{x}_t, \mathbf{y}_t, \omega_t, t\right) = h(\mathbf{y}_t)$, so the dynamics only depends on the (noisy) control input. The training data for the neural network are considered as external disturbances sampled from the true data distribution. The input is the initial state α and we would like to adjust the network weights $\mathbf{K}$ so that the final state $\mathbf{x}_N$ matches the desired output, β.

The neural network training problem can be then phrased as a problem in our framework:

$$\underset{\mathbf{K}}{\text{Minimize}} \underset{(\alpha,\beta)\sim\mathbb{P}_{\text{data}},\omega_t\sim\mathcal{N}(0,\Sigma_t)}{\text{E}}\left[\exp\left(\alpha\,\ell\left(\mathbf{x}_N, \beta\right) + \sum_{i=1}^{N-1}\alpha\frac{{u_t}^T R_t u_t}{2}\right)\right]$$

where ℓ is a loss function measuring the discrepancy between the desired output and that of the neural network. In practice the true distribution $\mathbb{P}_{\text{data}}$ is unknown and will be replaced by a sample average over training data. Our result says that as long as $\alpha R_t \succeq {\Sigma_t}^{-1}$, the resulting problem is a convex optimization problem. Thus, we have a convex optimization problem for training neural networks with arbitrary architecture, transfer functions and data

distributions! The way this differs from standard formulations of neural network training are:

a The inputs to each layer must be made noisy by adding Gaussian noise $\mathcal{N}(0, \Sigma_t)$.

b The performance of the network is measured as the expectation over the added noise of an exponentiated augmented loss function: This has both a "data" term $\ell(\mathbf{x}_N, \beta)$ and a "regularization" term: $\sum_{t=1}^{N-1} \frac{\mathbf{x}_t{}^T K_t{}^T R_t K_t \mathbf{x}_t}{2}$ which penalizes networks that have large activations or large weights.

c The noise added at each layer has to be "sufficient", that is, $\Sigma_t \succeq \frac{{R_t}^{-1}}{\alpha}$

5.3.3 Convex Reinforcement Learning

Stochastic gradient methods can be used to solve the convex optimization problems described in this work, leading to stochastic gradient methods for reinforcement learning and control. Stochastic approximation is an old idea in reinforcement learning and forms the basis of popular reinforcement learning algorithms like TD-learning, Q-learning and policy gradient algorithms [Szepesvári, 2010]. However, thus far, there were no guarantees made about the speed of convergence of these algorithms, particularly for continuous state problems.

Based on the work presented here, we can derive policy optimization algorithms based on stochastic convex optimization that are guaranteed to converge to the optimal policy in polynomial time. The results can give rise to both model-free and model-based reinforcement learning algorithms, based on the type of gradient update used. We then develop 2 policy gradient learning algorithms for finite horizon problems (algorithms 1,2). Note that algorithm 1 requires gradients of the dynamics (and hence a model of the dynamics) while algorithm 2 is completely model-free and just uses roll-outs in order to compute policy gradients.

Algorithm 1 Convex Stochastic Policy Gradient Method for (5.17)

$\mathbf{K} \leftarrow \mathbf{K}_0$

for $i = 1, \ldots, l$ **do**

$\hat{\ell} \leftarrow 0$ (Cost Estimate for Rollout)

for $t = 1, \ldots, N-1$ **do** (Policy Rollout)

$u_t \leftarrow K_t \phi(\mathbf{x}_t)$

$\hat{\ell} \leftarrow \hat{\ell} + \ell_{\mathrm{t}}(\mathbf{x}_t) + \frac{{u_t}^T R_t u_t}{2}$

$\omega_t \sim \mathcal{N}(0, \Sigma_t)$ (Sample Control Noise)

$\epsilon_t \sim \mathbb{P}_\epsilon$ (Sample System Noise)

$\mathbf{x}_{t+1} \leftarrow \mathcal{F}(\mathbf{x}_t, u_t + \omega_t, \epsilon_t, t)$ (Simulate Dynamical System Step)

end for

$\hat{\ell} \leftarrow \hat{\ell} + \ell_N(\mathbf{x}_N)$

$\mathbf{G} \leftarrow 0 * \mathbf{K}$

$\lambda \leftarrow \nabla \ell_{\mathrm{t}N}(\mathbf{x}_N)$

for $t = N-1, \ldots, 1$ **do** (Policy Gradient)

$G_t \leftarrow \frac{(R_t K_t \mathbf{x}_t)(\phi(\mathbf{x}_t))^T + \mathbf{x}_t (R_t K_t \phi(\mathbf{x}_t))^T}{2} + \frac{\partial \mathbf{x}_{t+1}}{\partial u_t} \lambda \phi(\mathbf{x}_t)^T$

$u_x \leftarrow \left(K_t \frac{\partial \phi(\mathbf{x}_t)}{\partial \mathbf{x}_t} \right)^T$

$\lambda \leftarrow \frac{\partial \ell(\mathbf{x}_t)}{\partial \mathbf{x}_t} + u_x R_t K_t \phi(\mathbf{x}_t) + \left(\frac{\partial \mathbf{x}_{t+1}}{\partial \mathbf{x}_t} + {u_x}^T \frac{\partial \mathbf{x}_{t+1}}{\partial u_t} \right) \lambda$

end for

$\mathbf{K} \leftarrow \mathbf{K} - \eta_i \left(\exp\left(\alpha \hat{\ell} \right) \right) \mathbf{G}$

end for

Algorithm 2 Derivative-Free Stochastic Gradient Method for (5.17)

$\mathbf{K} \leftarrow \mathbf{K}_0$

for $i = 1, \ldots, l$ **do**

$\hat{\ell} \leftarrow 0$ (Cost Estimate for Rollout)

for $t = 1, \ldots, N-1$ **do** (Policy Rollout)

$u_t \leftarrow K_t \phi(\mathbf{x}_t)$

$\hat{\ell} \leftarrow \hat{\ell} + \ell_{\mathrm{t}}(\mathbf{x}_t) + \frac{{u_t}^T R_t u_t}{2}$

$\mathbf{y}_t \sim \mathcal{N}(u_t, \Sigma_t)$ (Sample Noisy Control)

$\epsilon_t \sim \mathbb{P}_\epsilon$ (Sample System Noise)

$\mathbf{x}_{t+1} \leftarrow \mathcal{F}(\mathbf{x}_t, \mathbf{y}_t, \epsilon_t, t)$ (Simulate Dynamical System Step)

end for

$\hat{\ell} \leftarrow \hat{\ell} + \ell_N(\mathbf{x}_N)$

$\mathbf{G} \leftarrow 0 * \mathbf{K}$

$\lambda \leftarrow \nabla \ell_{\mathrm{t}N}(\mathbf{x}_N)$

for $t = N-1, \ldots, 1$ **do** (Policy Gradient)

$G_t \leftarrow \left({\Sigma_t}^{-1}(\mathbf{y}_t - u_t) + \alpha R_t u_t\right)(\phi(\mathbf{x}_t))^T$

end for

$\mathbf{K} \leftarrow \mathbf{K} - \eta_i \left(\exp\left(\alpha\hat{\ell}\right)\right)\mathbf{G}$

end for

Chapter 6

APPLICATIONS TO ELECTRIC ENERGY SYSTEMS

In this chapter, we describe applications of stochastic control to problems in energy systems. We will look at two problems:

1 Optimal placement and sizing of Energy Storage (Large-Scale Batteries) in order to mitigate fluctuations in intermittent generation like wind energy.

2 Distributed control of frequency in a transmission power grid.

6.1 Storage Sizing and Placement to Operational and Uncertainty-Aware Simulations

6.2 Introduction

Electrical grid planning has traditionally taken two different forms; operational planning and expansion or upgrade planning. The first is concerned with the relatively short time horizon of day-ahead unit commitment or hour-ahead or five-minute economic dispatch. The focus is on controlling assets that are already present within the system to serve loads at minimum cost while operating the system securely. The second typically looks out many years or decades and is focused on optimal addition of new assets, with a focus on minimizing the cost of electricity over the long time horizon. When a system consists entirely of controllable generation and well-forecasted loads, the network power flows do not deviate significantly or rapidly from well-predicted patterns. In this case, expansion planning can be reasonably well separated from operational planning. In the latter case, expansions may be optimized against only a handful of extreme configurations.

As the penetration of time-intermittent renewables increases, expansion and operational planning will necessarily become more coupled. For an electrical grid with large spatial extent, renewable generation fluctuations at well-separated sites will be uncorrelated on

short time scalesGibescu et al. [2009], Mills and Wiser [2010], and the intermittency of this new non-controllable generation will cause the patterns of power flow to change on much faster time scales than before, and in unpredictable ways. This new paradigm shift calls for accounting of multiple diverse configurations of uncertain resources in many operational as well as planning tasks. New equipment (e.g. combustion turbines or energy storage) and control systems may have to be installed to mitigate the network effects of renewable generation fluctuations to maintain generation-load balance. The optimal placement and sizing of the new equipment depends on how the rest of the network and its controllable components respond to the fluctuations of the renewable generation. Overall, we desire to install a minimum of new equipment by placing it at network nodes where controlled power injection and/or consumption have a significant impact on the network congestion introduced by the renewable fluctuations. From the outset, it is not clear which nodes provide the best controllability. Placing a minimum of new equipment is desirable since the investment and installation costs and costs associated with overcoming regulatory barriers. Thus, it makes sense to minimize the number of sites at which storage is placed for economic reasons.

6.3 Related Work

Before discussing our initial approach at integrating operational planning and expansion planning, we summarize a few methods for mitigating the intermittency of renewable generation. When renewable penetration is relatively low and the additional net-load fluctuations are comparable to existing load fluctuations, a power system may continue to operate "as usual" with primary and secondary regulation reservesHirst and Kirby [1999] being controlled via a combination of distributed local control, i.e. frequency droop, and centralized control, i.e. automatic generation control (AGC). In this case, *planning* for renewables may simply entail increasing the level of reserves to guard against the largest expected fluctuation in *aggregate renewable output*.

As the penetration level grows, simply increasing the reserve levels will generally result in increased renewable integration costsMeibom et al. [2010] which are usually spread over the rate base. Alternatively, operational planning can be improved by using more

accurate techniques for renewable generation forecasting to better schedule the controllable generation (energy and reserves) to meet net load and operate reliably Meibom et al. [2010], Bouffard and Galiana [2008], Hirst [2002]. Simulations using rolling unit commitment Tuohy et al. [2007], Meibom et al. [2010], where updated wind forecasts are used modify the unit commitment more frequently, have resulted in lower overall renewable integration costs.

Both unit commitment and economic dispatch seek to minimize the cost of electricity, however, they must also respect system constraints including generation/ramping limits, transmission line thermal limits, voltage limits, system stability constraints, and N-1 contingencies. Previous worksMeibom et al. [2010], Bouffard and Galiana [2008], Tuohy et al. [2007], Hirst [2002] have generally looked at the effects of stochastic generation on the economics and adequacy of *aggregate* reserves while not considering such network constraints. These constraints may be respected for a dispatch based on a *mean* renewable forecast. However, if the number of renewable generation sites and their contribution to the overall generation is significant, verifying the system security of all probable renewable fluctuations (and the response of the rest of the system) via enumeration is a computationally intractable problem.

The approaches summarized above do not consider network constraints or the behavior of the system on time scales shorter than the time between economic dispatches (one hour in the case of Meibom et al. [2010]). In particular, they do not model how fast changes in renewable generation and the compensating response of regulation reserves interact with network constraints. In this manuscript, extending our initial study Dvijotham et al. [2011], we augment the approaches summarized above by focusing on the behavior of the electrical network at a finer time resolution and investigate how the control of energy storage affects its placement and sizing.

We presume that the unit commitment problem has been solved, and at the start of a time period, we perform time-varying (every 5 minutes) lookahead dispatch of controllable generation and storage based on an operational scenario (spatial and temporal profiles of wind generation, load and net interchange) while trying to minimize the storage capacity used (in terms of both energy and power) —this gives us the minimum level of storage at each bus required for a particular operational scenario. We perform this optimization for

several different scenarios (based on historical data, if available, or data generated using an appropriate statistical model). The statistics from simulated system operations are then coupled to the expansion planning process by developing a heuristic to guide the optimal placement and sizing of storage throughout the network—a result that cannot be achieved with the previous approaches described above.

A new approach, applying convex relaxations to traditional operations (like Optimal Power Flow (OPF)) including uncertain (wind) resources and storage was recently proposed in Bose et al. [2012]. The idea was to solve a version of the OPF problem with certain constraints relaxed (permitting potentially inadmissible solutions) so that the resulting problem is a convex optimization problem and can be solved to global optimality efficiently. Further, the authors provide conditions under which the solution to the relaxed problem satisfies all the constraints of the original problem, so that the relaxed problem can be used as a computationally efficient proxy. The approach was also extended to the storage placement problem in Bose et al. [2012], Gayme and Topcu [2013], which concluded, that placement of storage on, or close to, renewable sites is far from optimal. Although innovative and theoretically interesting, the convex relaxation approach of Bose et al. [2012], Gayme and Topcu [2013] lacks scalability and was only illustrated on a very small 14 bus system. This is due to the high computational complexity of the semidefinite programming approach used in Bose et al. [2012]. Further, the authors in Bose et al. [2012] need to assume periodicity of renewable generation in order to solve the storage placement problem. In contrast, our work is the first resolving the storage placement problem over realistically sized networks. We run our algorithm on a 2209-node model of the Boneville Power Administration (BPA), accounting for actual operational data and multiple (more than hundred) wind patterns.

As discussed earlier, there are several reasons to place energy storage at a small number of sites. However, choosing the optimal set of sites is a *combinatorial* problem and cannot be solved by convex programming techniques. In this paper, we develop a greedy heuristic that attempts to solve the storage placement problem directly. While we can no longer guarantee optimality of this algorithm, we demonstrate that our approach is robust and works across different network topologies leading to more economical placements that

obvious alternatives.

We first set up notation in subsection 6.4.1, formulate the look-ahead dispatch problem in section 6.4.2, and finally describe our heuristic algorithm for storage placement in 6.4.4. We present numerical results in section 6.5 and section 6.7 wraps up with some conclusions and directions for future work.

6.4 Mathematical Formulation

6.4.1 Background and Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the graph representing the topology of the power system, with $\mathcal{V}$ being set of buses and $\mathcal{E}$ being the set of transmission lines. Let n denote the number of buses. At each bus, we can have three types of elements: Loads which consumer active power, Traditional Generators which generate active power and whose output can be controlled (within limits) and unconventional generators (renewables like wind) which generate active power, but whose output cannot be controlled.

For any $S \subset \mathcal{V}$ and any vector $v \in \mathbf{R}^n$, we denote $v_S = \{v_i : i \in S\}$. We will sometimes abuse this notation slightly to also denote the n dimensional vector with zeros everywhere except in S. We denote by $\mathbf{p}$ the vector of net-injections at each bus, and by $\mathbf{p}^s, \mathbf{p}^r, \mathbf{p}^l, \mathbf{p}^g$ the vector of injections at every node due to storage, renewables, loads and traditional generators, respectively. For any quantity y that is a function of time, we denote by $y(t)$ its value at time t. In this paper, we will use integer-valued time $t = 0, 1, \ldots, T_f$ where T_f is the time horizon of interest.

Let $\mathcal{N}(j)$ denotes the set of neighbors of node j in the network. We define the graph Laplacian to be a $|V| \times |V|$ matrix with entries:

$$\mathbf{L}_{ij} = -\frac{1}{\mathbf{x}_{ij}}, \mathbf{L}_{ii} = \sum_{j \in \mathcal{N}(i)} \frac{1}{\mathbf{x}_{ij}}$$

where $\mathbf{x}_{ij}$ is the reactance on the transmission line between node i and j. Then, the DC power flow equations are given by:

$$\mathbf{f}_{ij} = \frac{\theta_i - \theta_j}{\mathbf{x}_{ij}}, \theta = \mathbf{L}\mathbf{p}$$

where θ denotes the voltage phase and $\mathbf{f}_{ij}$ the active power flow between node i and j.

We will consider placing energy storage at nodes in the network. We denote the by $\mathbf{s}$ the vector of energy stored at each node in the network. The energy capacity of storage (maximum energy that can be stored) is denoted $\bar{\mathbf{s}}$ and the maximum power that can be withdrawn from or supplied to the energy storage units $\bar{\mathbf{p}^s}$. We denote by $\bar{\mathbf{p}}$ the maximum power output of traditional generators and $\bar{\mathbf{p}_{gr}}$ the corresponding limit on the ramping limits. $\bar{\mathbf{f}}_{ij}$ the limit on the flow on the line between i, j.

6.4.2 Lookahead Dispatch of Generation and Storage

In the presence of energy storage, the Optimal Power Flow(OPF)-based dispatch problem gets coupled over time (since energy stored at some time can be used later). Our approach to sizing and placing energy storage relies on operational simulations of the system under realistic load and renewable generation profiles. The operational simulation is formulated as a lookahead-dispatch problem: This is very similar to what the system operator would do to dispatch energy storage given a forecast of renewable generation and load. However, since we are interested in sizing and placement of energy storage, we additionally optimize over the energy capacity $\bar{\mathbf{s}}$ and power capacity $\bar{\mathbf{p}^s}$ of the energy storage needed to ameliorate the fluctuations in renewables and loads.

$$\min_{\mathbf{p}^s(t),\mathbf{p}^g(t)} \underbrace{\sum_{t=0}^{T_f} {\ell_{\mathrm{t}_g}}^T \mathbf{p}^g(t)}_{\text{Generation Costs}} + \underbrace{{\ell_{\mathrm{t}_s}}^T \bar{\mathbf{s}} + {\ell_{\mathrm{t}_s}^p}^T \bar{\mathbf{p}^s}}_{\text{Storage Investment Costs}}$$

subject to

$$0 \leq \mathbf{p}^g(t) \leq \bar{\mathbf{p}}_g \quad \text{(Generation Capacities)} \tag{6.1}$$

$$\mathbf{p}(t) = \mathbf{p}^g(t) + \mathbf{p}^r(t) + \mathbf{p}^l(t) + \mathbf{p}^r(t) \quad \text{(Net Injection)} \tag{6.2}$$

$$\mathbf{L}\theta(t) = \mathbf{p}(t) \quad \text{(DC Power Flow)} \tag{6.3}$$

$$\left| \mathbf{f}_{ij}(t) = \frac{\theta_i(t) - \theta_j(t)}{\mathbf{x}_{ij}} \right| \leq \bar{\mathbf{f}}_{ij} \quad \text{(Flow Limits)} \tag{6.4}$$

$$|\mathbf{p}^g(t+1) - \mathbf{p}^g(t)| \leq \bar{\mathbf{p}_{gr}} \quad \text{(Generation Ramping Limits)} \tag{6.5}$$

$$0 \leq \mathbf{s}(t) \leq \bar{\mathbf{s}} \quad \text{(Energy Capacity of Storage)} \tag{6.6}$$

$$0 \leq \mathbf{p}^s(t) \leq \bar{\mathbf{p}^s} \quad \text{(Power Capacity of Storage)} \tag{6.7}$$

$$\mathbf{s}(t) = \mathbf{s}(0) - \sum_{\tau=0}^{t-1} \mathbf{p}^s(t)\Delta \quad \text{(Energy Conservation)} \tag{6.8}$$

$$\mathbf{1}^T \mathbf{s}(T_f) = \mathbf{1}^T \mathbf{s}(0) \quad \text{(0 Net Energy Supply)} \tag{6.9}$$

The objective models operational costs of generation (fuel etc.) and *amortized* investment costs of placing energy storage in the grid. The constraints (6.1),(6.2),(6.3) and (6.4) are standard constraints appearing in a DCOPF formulation. The fifth constraint (6.6) is relevant in scenarios where wind generation undergoes a ramp event (sudden drop or increase) and traditional generators need to increase or decrease their output at rates close to their ramping limits. The constraints (6.6), (6.7), (6.8) are standard constraints for storage. The final constraint (6.9) models the fact that we want to use energy storage as a hedge over time - to store energy when too much power is being produced in the grid and supply it at a later time. Thus, over the horizon of interest, we do not want a net energy supply to/from the energy storage. This optimization problem is a Linear Program (LP) (like a standard DCOPF) and can be solved using off-the-shelf linear programming packages. We use the gurobi package in our work here Gurobi Optimization [2014].

6.4.3 Modeling Assumptions

Since this is a preliminary study meant to illustrate the value of coupling planning and operations, we made a number of simplifying assumptions that may not hold for a real power system. The first one is to use the DC Power Flow equations rather than the full nonlinear AC equations. The second one is to assume that dispatch is based on perfect forecasts of wind and loads over a 2-hour period. We outline the justifications for these assumptions in this section.

DC vs AC OPF

The DC power flow equations are an approximation to the nonlinear AC power flow equations. They are frequently used in the context of power markets although system operators would use the nonlinear ACOPF to perform actual dispatch of generators in a grid. In general, there can be significant discrepancies between DC and AC power flow results that make the DC solution unacceptable in an operational setting. In this paper, however, we stick with the DCOPF formulation. There are multiple reasons for this:

1 Since our interest in this work was to concentrate on the novel aspect of integrating planning and operational studies, we were not interested in building a nonlinear ACOPF solver. Freely available solvers like MATPOWER Zimmerman et al. [2005] do not generalize to the lookahead dispatch setting, that is, they are unable to deal with the time-coupling introduced by storage. However the storage placement algorithm (Algorithm 3) we develop in this paper can be used with any OPF solver. In particular, a more complete commercial-grade ACOPF solver should work better. Additionally, the extra computational burden of the ACOPF is not an issue here since we are performing offline planning studies which does not impose strict real-time requirements on the computation time (we could allow the algorithm to run for days if required).

2 We are mostly concerned with *long-term planning* and use operational information to inform the planning process. Hence, we are only interested in the accuracy of

the OPF to the extent that it captures all possible patterns of flows observed in typical operational scenarios. In numerical studies we performed, we observed that the DCOPF suffices for this purpose, at least for this preliminary study meant to illustrate the value of coupling planning and operations.

3 In general, the DCOPF becomes less accurate as the system gets under more stress. While we consider systems with high penetrations of renewable energy, we do not aim to deal with critical scenarios where the grid is under stress (close to voltage/ frequency instability). The challenge of high renewable penetration (which we aim to handle here) is that of non-predictable patterns of power flows. Thus, we are looking at the system under stable operating conditions, but with fluctuating patterns of power flows. When the grid is under stress, we assume that appropriate emergency control actions will be taken to protect the system. We do not aim to use energy storage to perform emergency control actions.

Perfect Forecasts

Note that in our DCOPF formulation $\mathbf{p}^r(t), \mathbf{p}^l(t)$ are assumed to be known functions of time. This is like performing lookahead dispatch with perfect forecasts. Although this differs from a real operational scenario (imperfect forecasts), we believe that the discrepancy will not break our analysis here for the following reasons:

- We consider time horizons of about $T_f = 2$ hours. Over such a time-scale, loads are well-predictable for sure, although wind may not be. However, we use operational simulations to develop a heuristic for placement of energy storage: Hence changes due to forecast errors, while important in an operational context, are less important from the context of deciding placement of energy storage.

- Several system operators today perform periodic redispatch of the grid resources (generation/storage) at fairly short intervals of time (5-15 mins) and hence can easily adapt to and cope-with forecast errors.

Further, we note that our heuristic for storage placement is *independent* of the specific dispatch scheme (OPF) used. Thus, we can perform a robust or chance-constrained version of DCOPF Bienstock et al. [2012] which would allow us to incorporate the effect of forecast uncertainty into the dispatch, and hence into the storage placement decision.

6.4.4 Optimal Sizing and Placement of Storage

We seek to develop heuristics to decide how to place storage and size its energy and power capacity. However, we must first define some metrics to evaluate a given storage placement. Let $\mathbf{S}$ denote the set of nodes with non-zero storage. For a given scenario δ_i (renewable/load profiles) and $\mathbf{S}$, the energy and power capacities resulting from the optimization (6.9) are $\bar{\mathbf{s}}_i$ and $\bar{\mathbf{p}^s}_i$. We define the energy in the renewable fluctuations to be $\mathbf{s}^r(t) = \sum_{\tau=0}^{t-1} \mathbf{p}^r(\tau)\Delta$, i.e. $\mathbf{s}^r(t)$ is the energy stored in a (hypothetical) battery that is connected directly to a renewable node and eliminates all fluctuations about the mean renewable generation. Then, plausible metrics can be defined according to the following criteria:

Normalized Power Capacity: This quantifies the total power capacity of the storage relative to the sum of maximal power fluctuations over the renewables:

$$\frac{\sum_{j\in\mathbf{S}} \max_t \left|\mathbf{p}^{s*}_j(t)\right|}{\sum_i (\max_t \mathbf{p}^r_i(t) - \min_t \mathbf{p}^r_i(t))}$$

Normalized Energy Capacity: This quantifies the total energy capacity of the storage relative to the sum of maximal energy fluctuations over the renewables:

$$\frac{\sum_{j\in\mathbf{S}} (\max_t \mathbf{s}^*_j(t) - \min_t \mathbf{s}^*_j(t))}{\sum_i (\max_t \mathbf{s}^r_i(t) - \min_t \mathbf{s}^r_i(t))}$$

Overall Performance: We denote a weighted combination of the above metrics by perf($\mathbf{S}$). In this study, we choose this to be the total normalized energy capacity plus a fixed cost for each site at which storage needs to be placed.

Renewable Penetration: The fraction of load served by renewables over the time horizon T.

The high-level pseudocode given in Algorithm 3. The algorithm is a greedy pruning heuristic that starts with $\mathbf{S} = \mathcal{V}$, i.e. storage at all nodes, and seeks to shrink $\mathbf{S}$ while improving performance at least by some minimum amount ϵ at each iteration. Then, the

same procedure is repeated, each time shrinking the target number of nodes as long as the performance metric is improving. Note that this repetition is required (and critical) because the dispatch based on restricted storage would be different, since there are a smaller set of controllable resources.

Algorithm 3 Greedy Heuristic for Optimal Placement

Input: Collection of Scenarios $\{\delta_k\}$, Threshold ϵ

$\mathbf{S} \leftarrow \{1, 2, \ldots, n\}$.

repeat

for $k = 1 \rightarrow N$ **do**

Solve (6.9) for scenario δ_k to get $\bar{\mathbf{s}}_k, \bar{\mathbf{p}^s}_k$

end for

$\bar{\mathbf{s}} \leftarrow \max_k \bar{\mathbf{s}}_k$

$\bar{\mathbf{p}^s} \leftarrow \max_k \bar{\mathbf{p}^s}_k$

$\gamma \leftarrow \max\{\gamma : \{\text{perf}(\{i \in \mathbf{S} : \bar{\mathbf{s}}_i \geq \gamma \max(\bar{\mathbf{s}})\}) < \text{perf}(\mathbf{S}) - \epsilon\}\}$.

$\mathbf{S} \leftarrow \{i \in \mathbf{S} : \bar{\mathbf{s}}_i \geq \gamma \max(\bar{\mathbf{s}})\}$.

until $1 - \gamma \leq \epsilon'$

6.4.5 Justification for Greedy Algorithm

The choice of the greedy algorithm is motivated by the theory of submodular function maximization Krause and Golovin [2012]. Submodular functions are functions with diminishing marginal returns. Mathematically, if one had a function F defined on subsets A of $S = \{1, 2, \ldots, m\}$ that satisfied:

$$F(A \cup \{i\}) - F(A) \leq F(B \cup \{i\}) - F(B), B \subset A \subset S, i \notin A.$$

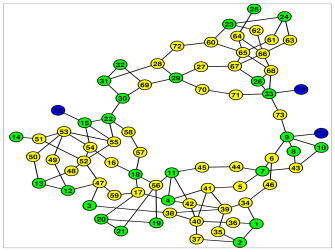
In our context, this simply means: The additional performance gain obtained by adding storage at a new node when there is already storage at a large number of nodes is smaller than the performance gain obtained by adding to storage when there is storage at only a few nodes. Although we have not been able to prove that this property holds for the

storage placement problem, it definitely makes intuitive sense - at some point one would expect to observe diminishing returns for additional placement. It can be shown that for a submodular objective function, the greedy algorithm achieves an objective that is within $1 - \frac{1}{e}$ of the optimal solution Krause and Golovin [2012]. This motivated us to consider a greedy algorithm to solve the problem of storage placement.

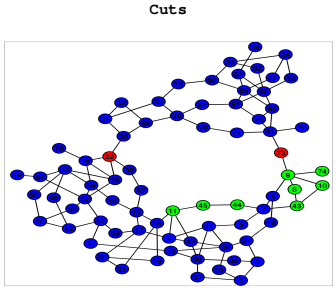
We have some preliminary results (not included in this paper) regarding the submodular property for certain simplified versions of the objective presented here and hope to pursue this line of investigation further in future work.

6.5 Simulations

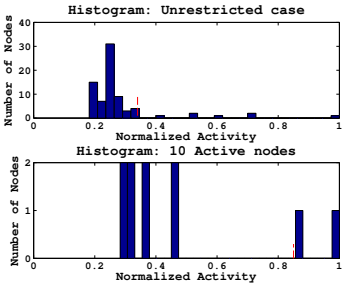
6.5.1 RTS-96+Synthetic Wind Data



(a) Our modified version of RTS-96. The added renewables are blue, loads are yellow and controllable generators are green

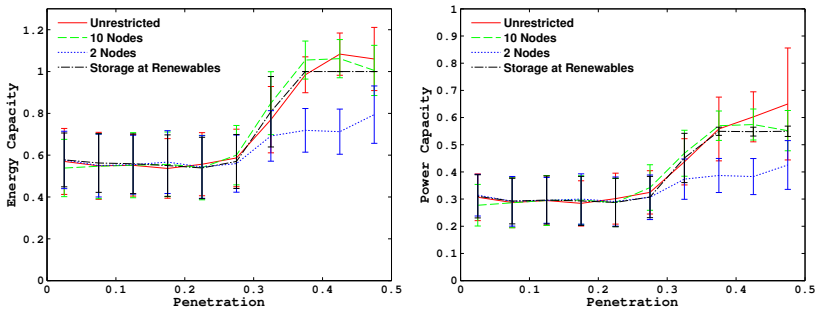


(b) Sets of nodes identified by our heuristic: The minimal set (two nodes) is shown in red, additional 8 nodes, of the 10 node set, are show in green, all other nodes are shown in blue.



(c) Storage Capacity Histograms: Red lines mark thresholds used for the reduction in the storage node set

We tested our optimal control and heuristic for storage placement and sizing on a modified versions of RTS-96Grigg et al. [1999]. The grid is shown in Fig. 6.1a. Our modification includes the addition of three renewable generation nodes shown Fig. 6.1a in blue. The capacities of the new lines connecting the renewables to their immediate neighbors are set



(a) The normalized energy capacity of storage in the entire network vs the penetration of renewable generation

(b) The normalized power capacity of storage in the entire network vs the penetration of renewable generation

higher than the capacity of the added renewable generation, otherwise, these lines would be overloaded in nearly every trial.

In each iteration of algorithm 3), we generate $N = 2000$ time series profiles for the renewables. These are chosen so that we can control the penetration of wind in the system and study the effect of penetration of intermittent sources on storage sizing and placement.

In the first iteration, storage is available at all nodes in the network. The histogram of storage capacities is plotted in Fig. 6.1c. We then shrink the set of nodes having storage until the performance metric perf($\mathbf{S}$) (defined in Section 6.4.4) fails to improve significantly (by more than ϵ). For this example, we were able to shrink down to 10 nodes in the first iteration. Using these 10 nodes, we rerun the optimal control algorithm and again accumulate statistics of the storage activity (plotted in figure 6.1c). Based on the updated statistics, we can again shrink the set of storage nodes down to 2 nodes and this is the final output of the algorithm (we cannot shrink any further without performance degradation). The optimally chosen sets of 10 and then 2 nodes are shown in Fig. 6.1b.

The method for generating the renewable profiles is described in details in Dvijotham et al. [2011]. The evaluation metrics defined in Section 6.4.4 are shown as functions of penetration in Figs. 6.2a,6.2b, with storage at all the nodes and sets of shrunken nodes discovered by the Algorithm 3.

6.5.2 BPA System with Historical Wind Data

We also apply our algorithm to real data from the BPA network covering Washington and Oregon. By overlaying the grid on the US map, we were able to locate the major wind farms and inter-ties (to California) in the system. Loads were divided roughly in proportion to population densities. Mapping this onto data published on the BPA web-site in The BPA Balancing Authority, we were able to create realistic wind, load and interchange profiles. We considered data from 100 different wind configurations during 2012 (each of length about 2 hours, spread uniformly throughout the year). We also ensured that we pick particularly challenging operating conditions, for example, periods with high ramping conditions in wind generation, i.e. these pushing the storage dispatch to its limits, and thus to enable sizing storage so as to be prepared for the worst contingencies.

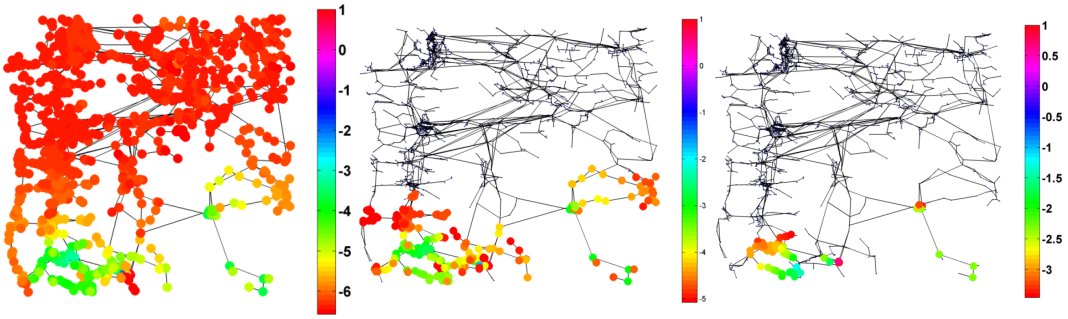
We plot the iterations of our algorithm on the BPA system in Fig. 6.3. The nodes at which storage is present are colored—red marking the nodes with least storage capacity and purple marking the nodes with the highest storage capacity - The capacities are color coded in a log-scale:

$$\log\left(\frac{\text{Storage Capacity at a node}}{\text{Maximum Storage Capacity over all nodes}}\right)$$

so as to improve visual discriminability. Our sequential algorithm is able to discover a relatively small subset of 37 nodes at which to place storage. Reducing this number any further leads to a significant increase in the overall storage capacity required.

In Fig. 6.4, we plot the locations of the storage nodes relative to the locations of the wind farms and inter-ties. We note that our algorithm does not place the storage near either the wind sites or the interties We also compared our strategy to placing storage directly at the wind farms or inter-ties (which are the “sources” and ”sinks” which contribute most to fluctuations in the generation/load). The overall storage capacity required by this naive approach is twice the storage capacity required for the placement discovered by our algorithm. In Fig. 6.5,we plot the total energy and power capacity of the storage placements discovered by our algorithm relative to the naive strategy of placing storage directly at the renewables and interties.

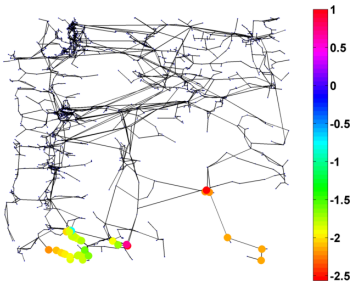
This result shows that the storage placement discovered by the algorithm, although



(a) BPA - Algorithm Iteration 1

(b) BPA - Algorithm Iteration 2

(c) BPA - Algorithm Iteration 3



(d) BPA - Algorithm Iteration 4

Figure 6.3: Iterations of our Algorithm on the BPA System. Red Corresponds to Low Storage Capacity and Purple to High

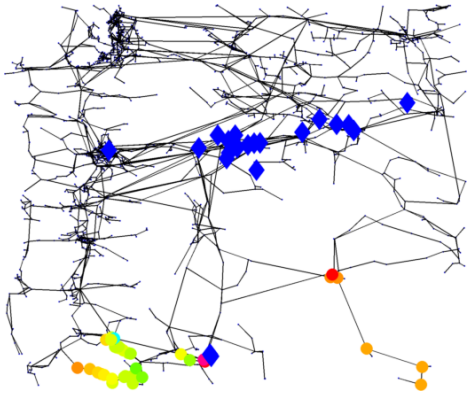


Figure 6.4: Storage Placement (colored circles) relative to Wind Farms/Interties (shown as blue diamonds).

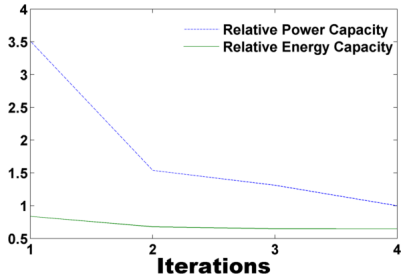


Figure 6.5: Total Energy and Power Capacity Relative to Placement at Renewables and Interties

intuitive, is non-trivial since the many of the nodes picked are not precisely at the renewables or interties, but rather at critical nearby nodes which are critical for controlling the power flows in that region of the network.

6.5.3 Computation Times

For the BPA system, the entire algorithm took about 10 minutes on a i7 2.9 GHz CPU to produce the optimal placement of storage. The optimal dispatch for a particular scenario takes about 5 seconds.

6.6 Discussion

We have presented an efficient and effective heuristic for sizing and placing energy storage in a transmission network. Our essential insight in this paper was to couple operational simulations with planning, and use statistics from operational simulations to inform the planning procedure. For any realistic engineered network, operational simulations will contain valuable information about the various flow patterns, congestion, ramping restrictions etc. in the network and provide an effective heuristic for making planning decisions - we have observed this in the above simulations as well. With an unoptimized matlab implementation, our approach takes about 10 minutes to discover an effective storage placement for the BPA system. For an offline planning problem, this is perfectly acceptable.

An alternate approach would be to formulate this directly as a mixed integer linear program: Choose a small number of sites to place energy storage so as to minimize investment and operational costs over a large set of possible scenarios. However, this approach fails to take advantage of the above observation, and quickly becomes computationally infeasible for realistically sized networks.

We use the DCOPF approximation in our work, but as mentioned in section **??**, the approach can be easily used with an ACOPF solver.

For both the BPA and RTS-96 network, by using the greedy pruning algorithm, we are able to reduce the number of energy storage sites to a very small number compared to the total number of nodes in the network. In both cases, the storage is placed far from the renewable generation. Instead, the storage appears to be placed at a few critical nodes suggesting that the storage is being used not only to buffer fluctuations, but also to assist with controlling flows in the rest of the network. For the BPA system, these may seem geographically close to the renewables or interties. However, the precise placement of storage is non-trivial and the discovered placement uses particular nodes that offer a large degree of controllability on the power flow patterns in that region of the network. Thus, in effect, our algorithm is designing the grid control system by finding the nodes with the highest controllability over the network congestion.

This conclusion is supported by the plot of iteration-by-iteration storage energy and power capacity in Fig. 3. The energy capacity of the storage is not dramatically reduced by during the pruning. Instead, storage capacity that was dispersed throughout the network is concentrated at fewer nodes resulting in larger but sparser storage installations. However, the storage power capacity drops significantly. This seems to indicate that the wind fluctuations require a certain amount of energy capacity for buffering on a network wide basis. However, better placement of that energy capacity enables it to be used just as effectively with a much smaller power capacity.

6.7 Conclusions and Future Work

Somewhat unexpectedly, our algorithm chooses to place storage at nodes at critical junctions between major subcomponents of the network rather than at the sites of renewable

generation. We conjecture that these nodes provide for enhanced controllability because, in addition to simply buffering the fluctuations of the renewables, controlled power injections at these nodes can modify overall network flows and direct fluctuating power flows to regions that are better positioned to mitigate them.

There is much follow on work needed to expand the concept presented in this manuscript and to verify some of its conjectures. We also plan to extend this approach to allow for stochastic, robust and/or chance-constrained optimization, as in Bienstock et al. [2012], to provide for a better representation and more accurate modeling of the wind uncertainty. Finally, we performed lookahead dispatch assuming perfect information about loads and renewable generation. Although this assumption is reasonable (for reasons described in section 6.4.2), a more thorough study is required to determine the exact effect of forecast errors, particularly on storage sizing (we would expect the placement to be robust to reasonable forecast errors). This would require modeling standard generation response mechanisms (primary control and AGC) which modify generator outputs in response to changes in renewable generation and loads. These mechanisms are well studied for generation, but need to be extended for storage systems as well. We plan to build on recent work in this direction Dvijotham et al. [2012] for this.

6.8 Distributed Control of Frequency in a Grid with a High Penetration of Renewables

6.8.1 Problem Setup and Brief Statement of Results

In today's power systems, the system operator performs an Optimal Power Flow (OPF) dispatch periodically with typical time interval being 5, 15, or 60 minutes depending on the Balancing Area Kundur [1994]. The OPF sets the power outputs of the committed generation to match power demand and minimize generation cost while respecting the capacity limits on lines, ramping constraints and limits on generators and sometimes taking into account the N-1 security constraints. In between two successive OPFs, the system is automatically controlled by a combination of two mechanisms. The faster of the two, acting on the scale of seconds, is primary frequency control–a fully distributed proportional feedback on locally-measured frequency deviations that may also include a deadband. The

slower mechanism, acting on the scale of minutes, is automatic generation control (AGC), also called secondary control–a centralized feedback on the integral of a weighted sum of a centrally measured frequency and tie line flows to neighboring balancing areas.Tomsovic et al. [2005]

These combined controls correct deviations in the generation-load balance driven by fluctuations in loads, renewables and other disturbances in the system. However, these mechanisms do not explicitly incorporate line-flow limits, generators ramping limits, or time-integral constraints like those on run-of-river hydro generation or energy storage. For systems with relatively low levels of fluctuations, these limits are not frequently violated and it is not necessary incorporate them directly. However, higher levels of time-intermittent generation will create larger fluctuations and ramping events and the associated constraint violations will become more common. Standard primary and secondary controls are limited in their ability to balance these fluctuations, and better control design is needed to manage these larger fluctuations. Because these fluctuations are intimately connected to frequency deviations, they are of special concern because they my result in system-wide instabilities and loss of synchrony Eto et al. [2010].

Other considerations for real-time power grid control systems are communication constraints and communication securityTomsovic et al. [2005]. Mechanisms that rely on central aggregation of the entire grid state followed by a centrally computed response will be vulnerable to communication failures and attacks on the communication network, making the overall system less robust. On the other hand, with significant renewable penetration, it is difficult to control a system purely based on local feedback, since under some conditions, it may be necessary to control distant generators in a correlated manner.

In this preliminary work, we explore a hybrid approach that combines the speed and security of fully distributed control with the extensive system visibility provided by centralized control. Our method performs a centralized lookahead dispatch that also computes optimal local feedback parameters for all controllable generation, thus enabling the system

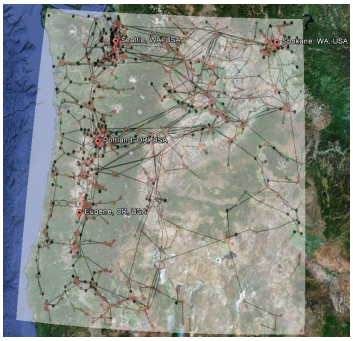


Figure 6.6: Our model of the BPA Transmission Network

to respond to fluctuations based only on local observables. We expand our definition of local observables to include not just frequency but also real power flows to neighboring nodes. We use an ensemble of forecasts that capture various possible scenarios for the wind generation and loads over the next intra-dispatch period (5 min/15 min/ 1 hour) to design an optimal time-varying dispatch for all the generators, as well as local feedback functions that enable the generators to respond to fluctuations based on the local observables.

Our control design is split into 2 phases:

a An off-line optimization phase where the distributed control gains are optimized jointly for the whole network in a central computer using extensive simulation of possible future wind generation and forecast scenarios. These gains are then communicated to each flexible resource (controllable device) in the transmission network. This off-line optimization would need to be re-run every time the statistics of possible future scenarios change significantly. In general, we expect this optimization to be run every time the generation re-dispatch changes.

b An online response phase where each device implements its purely local control in response to local observables (local frequency, line flows etc.) on the pace of the standard primary controls.

We test our algorithm on historical data from the Bonneville Power Administration (BPA) system BPA [a], an ideal test system for our algorithm as it has significant amounts of both hydro and wind generation. We show that our algorithm performs well, even in cases of significant wind ramps.

Our results (detailed in section 6.8.3) lead to the following important observations:

a Local control based on response to frequency deviations and local line flows at each generation can keep frequency deviations down to the about 10 mHz while maintaining all the security and capacity constraints.

b Proportional control on frequency deviations and feedback on line flows is sufficient. Adding a frequency-deviation integral response is unnecessary, which is advantageous because a distributed implementation of an integral term may cause instabilities due to errors in local frequency measurement, and also because it limits communication requirements.

c Joint optimization of feedback parameters for frequency deviation and line flows is necessary. Independent optimization or removal of either term leads to poor control performance.

d Optimization over a finite but representative set of future scenarios enables the generalization of the control to new unseen scenarios.

The rest of the paper is organized as follows: Section 6.8.2 describes the mathematical setting of the underlying control/optimization problem; we describe and discuss results of our numerical BPA experiments in Section 6.8.3; and Section 6.8.5 presents conclusions and explains our path forward.

6.8.2 Mathematical Formulation

Preliminaries

The power system is described by an undirected graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ with edges $\mathcal{E}$ and n vertices $\mathcal{V}$. The grid is composed of loads ($\mathbf{l}$), conventional generators ($\mathbf{g}$) and renewable

generators ($\mathbf{R}$). The flexible resources in our grid are the online conventional generators $\mathbf{go}$. We denote by $\mathbf{p}$ the $|\mathcal{V}| \times 1$ vector of net active power injections at each node in the network and by $\mathbf{p}^g, \mathbf{p}^r, \mathbf{p}^l$ the net active power injections due to online conventional generators, renewable generators and loads: $\mathbf{p} = \mathbf{p}^g + \mathbf{p}^r + \mathbf{p}^l$. Each of these vectors is of size $|\mathcal{V}| \times 1$ with the convention that $\mathbf{p}^g_i = 0$ or $\mathbf{p}^r_i = 0$ if there is no conventional or renewable generator at node i. Note here that we make the assumption that there is only one generator or load at a given node. If there are multiple, we replace them by an equivalent single generator or load. For a vector v with indices in $\mathcal{V}$, v_i denotes a particular component for $i \in \mathcal{V}$ and v_S denotes the sub-vector $\{v_i : i \in S\}$ for a subset $S \subset \mathcal{V}$.

Scenarios

The control algorithm proceeds by analyzing an ensemble possible future scenarios and designs control strategies that optimize the system cost (defined below) across all scenarios. We define a scenario χ to be a collection of the following quantities:

a Renewable generation over the time horizon of interest: $\mathbf{p}^r(t)$.

b Load profile over the time horizon of interest: $\mathbf{p}^{\mathbf{l}}_0(t)$.

c A unit commitment (configuration of generators which are online, i.e. available for re-dispatch) $\mathbf{go}$.

To define the control problem, we require a collection of scenarios Ξ and estimates for the probability of each scenario, i.e. $\Xi = \{\chi_i, \text{Prob}(\chi_i)\}$. We note that for a given collection Ξ, $\mathbf{p}^r(t)$ and $\mathbf{p}^{\mathbf{l}}_0(t)$ (items a and b from above) will vary across the ensemble of scenarios, however, we take $\mathbf{go}$ (the unit commitment from c) fixed because we are designing the time-dependent dispatch and local feedback parameter for that particular $\mathbf{go}$. In this work, we assume that the collection Ξ is finite. Typically, Ξ will be built up from load and wind forecasts from different forecasting methodologies weighted by confidences in each of these forecasts. Ξ could also include samples from a stochastic forecasting model based on climate models, historical data, meteorological sensors etc.

Control Formulation

We ignore electro-mechanical dynamical transients and work with a discrete-time quasi-static approximation of the system dynamics with fixed time step δ and integer time indices $t = 0, 1, \ldots, T$: at each time step the power flows over lines are re-computed for configuration of consumption/generation at nodes evolving in discrete time. In general, the feedback can depend on any of the system variables, but we limit ourselves to local observables so that the control can be implemented in a completely distributed fashion at each generator after the dispatch and feedback parameters have been communicated.

For each generator $g \in \mathbf{go}$, we compute a time-varying dispatch $\mathbf{p}_g^0(t) : 0 \leq t \leq T$, proportional frequency response coefficient α_g^P, integral frequency response coefficient α_g^I, and a response coefficient to local flows $\{\alpha_{g\to i}^F : i \in \mathrm{Neb}(g)\}$. Further, we denote by $\omega(t)$ the frequency deviation from the nominal frequency (50/60 Hz) at time t and by $\Omega(t)$ the integral of the frequency deviation, which in discrete-time is approximated by $\Omega(t) = \sum_{\tau=0}^{t} \gamma^{\tau-t}\omega(\tau)$ where $0 < \gamma < 1$ is a discount factor. In other words, the integral frequency term is simply a weighted sum of frequency deviations in the past, where frequency deviations that are further in the past receive a geometrically smaller weight. With the time varying dispatch and feedback parameters determined, the output of the generators is given by:

$$\mathbf{p}_g^g(t) = \mathbf{p}_0^{\mathbf{go}}{}_g(t) + \alpha_g^P\omega(t) + \alpha_g^I\Omega(t) + \sum_{i\in\mathrm{Neb}(g)} \alpha_{g\to i}^F \mathbf{p}_{g\to i}(t).$$

Although our algorithm can incorporate nonlinear feedback, we choose feedback which is linear in the local observables for this initial work. In addition to generators, the real power consumption of loads responds to frequency changes, and we assume a simple linear load-frequency response given by

$$\mathbf{p}^l(t) = \mathbf{p}_0^{\mathbf{l}}(t) + \beta^l\omega(t),$$

where the β^l are known from measurement where $\mathbf{p}_0^{\mathbf{l}}$ is the load at the nominal frequency (60 Hz). Combining the load and generator frequency response and the generators' time varying

dispatch, the system's equilibrium frequency is computed by enforcing power balance in the system:

$$\sum_{i\in\mathcal{V}} \mathbf{p}_i^g(t) + \mathbf{p}_i^l(t) + \mathbf{p}_i^r(t) = 0 \implies$$
$$\omega(t) = -\frac{\sum_i \mathbf{p}_{0i}^{\mathbf{l}}(t) + \mathbf{p}_i^r(t) + \mathbf{p}_i^g(t)}{\sum_i \beta_i^l}.$$

To compute power flow from the injections $\mathbf{p}(t) = \mathbf{p}^l(t) + \mathbf{p}^r(t) + \mathbf{p}^g(t)$, we use a modified version of the DC Power flow equations based on a linearization of the AC Power flow equations around the nominal dispatch at the beginning of the control period $\mathbf{p}(0)$. The linearization gives us dynamic impedances $\mathbf{s}_{i\to j}^d$ that substitute for the line reactances in the DC power flow equations:

$$\mathbf{p}_i(t) = \mathbf{p}^0 + \sum_{j\in \mathrm{Neb}(i)} \frac{\theta_i(t) - \theta_j(t)}{\mathbf{s}_{i\to j}^d},$$
$$\mathbf{p}_{i\to j}(t) = \mathbf{p}^0{}_{i\to j} + \frac{\theta_i(t) - \theta_j(t)}{\mathbf{s}_{i\to j}^d}.$$

Such a linearization is reasonable assuming that the flow patterns do not change too much during the course of the control period.

In addition to several other constraints discussed below, we will also imposed a constraint on the total energy extracted from generators in the control period. Such constraints can represent the water discharge constraints on run-of-river hydro systems or state-of-charge constraints on energy storage devices. Therefore, we must also include the total energy extracted from each generator into the system state:

$$\mathbf{p}^I(t) = \sum_{\tau=0}^{t} \mathbf{p}^g(\tau).$$

The overall system state consists of $\mathbf{x}(t) = [\Omega(t); \mathbf{p}^g(t); \mathbf{p}^I(t)]$ (in Matlab notation), and

the system evolution can be summarized by:

$$\begin{aligned}
\omega(t) &= -\frac{\sum_i \mathbf{p}^{\mathbf{l}}_{0i}(t) + \mathbf{p}^r_i(t) + \mathbf{p}^g_i(t)}{\sum_i \beta^l_i} \qquad (6.10)\\
\Omega(t+1) &= \omega(t) + \gamma\Omega(t)\\
\mathbf{p}^g{}_g(t) &= \mathbf{p}^{\mathbf{go}}_0{}_g(t) + \alpha^P_g\omega(t) + \alpha^I_g\Omega(t)\\
&\quad + \sum_{i\in \mathrm{Neb}(g)} \alpha^F_{g\to i}\mathbf{p}_{g\to i}(t)\\
\mathbf{p}^l(t) &= \mathbf{p}^{\mathbf{l}}_0(t) + \beta^l\omega(t)\\
\mathbf{p}_i(t) &= \sum_{j\in \mathrm{Neb}(i)} \frac{\theta_i(t)-\theta_j(t)}{\mathbf{s}^d_{i\to j}}, \mathbf{p}_{i\to j}(t) = \frac{\theta_i(t)-\theta_j(t)}{\mathbf{s}^d_{i\to j}}
\end{aligned}$$

Cost Functions

We consider a stochastic setting with many possible features, and it is unclear whether it is feasible to satisfy all constraints across all scenarios in Ξ. Therefore, we use a penalty function to enforce our constraints in a smooth manner. The penalty function has a magnitude of zero in a dead-band around the most feasible region and grows cubically with the magnitude of constraint violation:

$$\mathrm{Pen}(a,l,u) = \begin{cases} 10^7((a-u)/(0.1*(u+1)))^3 & \text{if } a \geq u \\ 10^7((l-a)/(0.1*(l+1)))^3 & \text{if } a \leq l \\ 0 & \text{otherwise} \end{cases}.$$

Here, a is the value of the constrained quantity and l and u are the lower and upper bounds on a, respectively. We also adopt the convention that when a, l, u can be vectors (of the same size) and the penalty in this case is applied element-wise and added up. The penalty function is designed so that the resulting cost function is smooth (twice differentiable). However, if a is violates the upper bound by 10%, a penalty of approximately 10^7 is incurred–a high enough penalty so that if a feasible solution exists across all scenarios, it will be found.

The cost function $\mathrm{Cost}(\mathbf{x}(t), \mathbf{x}(t+1), t)$ is computed at each time step in the control period, but it requires state information from both t and $t+1$ so it can incorporate generator

ramping limits. The cost includes seven terms that penalize both economic cost of supplying generation and deviations of the system state outside of normal operational bounds. The individual terms are:

1 Generation costs

$$\text{GenCost}(\mathbf{p_{go}}(t)) = \sum_{g \in \mathbf{go}} c_{g1}(\mathbf{p}_g^g)^2 + c_{g2}\mathbf{p}_g^g + c_{g3}.$$

2 Generation limit penalties

$$\text{Pen}(\mathbf{p}^g(t), \underline{\mathbf{p}^g}, \overline{\mathbf{p}^g}).$$

3 Ramping limit penalties

$$\text{Pen}\left(\underline{\mathbf{p}_r^g}, \frac{\mathbf{p}^g(t+1) - \mathbf{p}^g(t)}{\delta}, \overline{\mathbf{p}_r^g}\right).$$

4 Power flow thermal limit penalties

$$\sum_{i \to j \in \mathcal{E}} \text{Pen}(\mathbf{p}_{i \to j}(t), -\overline{\mathbf{p}}_{i \to j}, \overline{\mathbf{p}}).$$

5 Frequency deviation penalties

$$\text{Pen}(\omega(t), -0.01, 0.01)$$

6 Integral frequency deviation penalties

$$\text{Pen}(\Omega(t), -0.01, 0.01).$$

7 An integral deviation penalty on generation:

$$\text{Pen}(\mathbf{p}^I(T), 0.95\overline{E^{\mathbf{go}}}, 1.05\overline{E^{\mathbf{go}}})$$

Cost 1 simply represents the financial cost of energy from different generators. Costs 2-4 are normal power system constraints converted to costs using the penalty function defined above. Cost 5 is an additional penalty designed to constrain the system frequency to within a 10 mHz band, and Cost 6 is designed to constrain the deviation of the integral of the frequency

deviation so that the frequency is not allowed to be low or high for extended periods of time. Finally, Cost 7 is designed to keep the total energy delivered by each controllable generator over the control period within a $\pm 5\%$ band around a $\mathbf{p}^I(T)$ mimicking constraint the would occur in either a run-of-river hydro system or an energy storage device.

Ensemble Optimal Control

The evolution equations listed in (6.10) are functions of a given scenario χ, therefore, we can think of the state as a function of the scenario χ and the control parameters $\alpha = \{\alpha^P, \alpha^I, \alpha^F, \mathbf{p}_0^{\mathbf{go}}(t) : 0 \leq t \leq T\}$: $\mathbf{x}(\alpha, \chi, t)$. The overall optimization problem can then be written

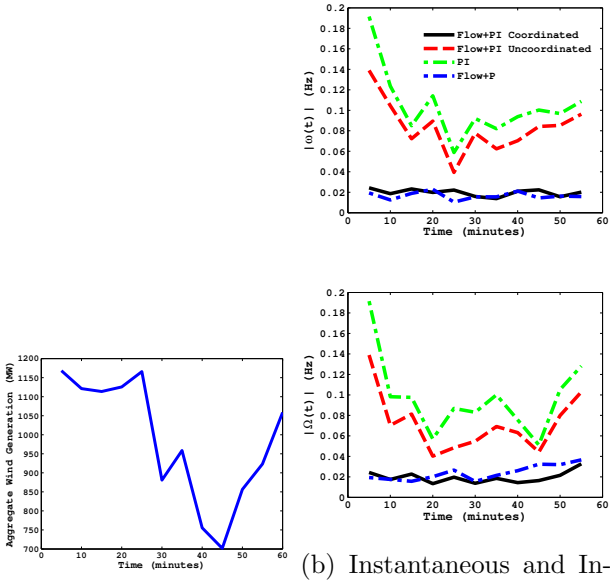
$$\min_{\alpha} \sum_{\chi} \text{Prob}(\chi) \left(\sum_{\tau=0}^{T-1} \text{Cost}(\mathbf{x}(\alpha, \chi, t), \mathbf{x}(\alpha, \chi, t+1), t) \right)$$
$$\text{Subject to (6.10).} \tag{6.11}$$

We optimize this objective using a standard numerical optimization algorithm (LBFGS Schmidt). The gradients of the objective function can be computed efficiently using a forward propagation algorithm that uses the chain rule to propagate gradients in time. This computation can be easily vectorized over all the scenarios, leading to significant speedup if run on a cluster or on GPUs.

6.8.3 Numerical Results

Description of Test System

We test our algorithm using publicly available historical data for hydro and thermal generation, wind generation, and load from the Bonneville Power Administration (BPA) websiteBPA [a]. We use a model of the BPA transmission system (shown in Fig. 6.6) that has 2209 buses and 2866 transmission lines. By identifying major hydroelectric stations on the transmission system and overlaying this onto a publicly available BPA wind site map BPA [d], we located the existing wind farms on the BPA transmission system (as of January 2010). We located the meteorological stations where BPA collects wind data BPA [b] in a similar manner. Using the same overlay, we used a simple incompressible air-flow model



(a) Aggregate Wind Generation

(b) Instantaneous and Integral Frequency Deviations

Figure 6.7: Comparison of control schemes. a) Aggregate wind generation from a period with significant ramping events. b) *Worst-case* frequency deviations over the control period for 18 validation scenarios not used in the control design.

to infer hub height wind speeds at the wind farms. The resulting wind speeds were passed through the power curve of a standard 1.5-MW GE Wind Turbine which was scaled to the wind farm nameplate capacity to estimate the power output $\mathbf{p}^r(t)$ (in MW) at each wind farm as a function of time. When we aggregate our wind farm-specific estimates of wind generation, we typically over estimate the BPA aggregate data by 20%, which may caused by several factors including: spilling of wind by BPA, under performance of wind farms relative to single-turbine estimates, or shortcomings in our model of interpolating wind speeds. BPA also provides aggregate load dataBPA [a] that we divide among the nodes in the network according to population densities. BPA also makes publicly available aggregate interchange flows BPA [a], which we apportion to different tie lines in a similar manner.

To test our control algorithm on difficult conditions, we select a control period of one hour from 10:35 AM to 11:35 AM on February 12, 2010, when the wind generation was ramping significantly (shown in Fig. 6.7a). We then create 26 scenarios (site-specific wind profiles) for this period by adding random time-varying Gaussian noise to the wind speeds at each meteorological station (from which we infer site-specific wind generation as outlined above). We set the magnitude of the noise so as to match, on average, the aggregate wind generation hour-ahead forecast errors reported by BPA BPA [c]. All the time series data used in our study was available at a 5-minute resolution.

Unit commitment data is missing from our model, therefore, we assume that all hydro generators larger than 300 MW are online and are all participating in frequency regulation. From inspection of the BPA historical generation data BPA [a], we infer that the thermal generation dispatch is fixed over time. In our model, we replicate this dispatch by dividing the total thermal generation among the online thermal generators (randomly chosen).

6.8.4 Comparison of Various Control Schemes

For difficult wind ramping conditions, we illustrate the value of feedback based on local flows by comparing four control schemes. We use P to designate proportional control (to

frequency deviations $\omega(t)$) and I designates integral control (to integral frequency deviations $\Omega(t)$). The control schemes we consider using are:

1 *PI:* Joint optimization of the time-varying dispatch $\mathbf{p}_0^{\mathbf{go}}(t)$ and the local feedback parameters for $\omega(t)$ and $\Omega(t)$.

2 *Flow+PI Uncoordinated:* Time-varying dispatch $\mathbf{p}_0^{\mathbf{go}}(t)$ plus feedback on $\omega(t), \Omega(t)$ and local flows $\mathbf{p}_{g\to i}$ at each generator. The optimization in 1 is performed first followed by a second optimization over the flow feedback parameters.

3 *Flow+PI Coordinated:* Same as 2, but the optimization is performed jointly.

4 *Flow+P:* Same as 3, but without feedback on $\Omega(t)$.

The experimental protocol is as follows. We setup each of the four optimization problems according to Eqs. 6.11 with the scenarios described in Section 6.8.3 and determine a single set of feedback parameters for each of the four feedback schemes. We use 8 of the 26 created scenarios as input to the optimization algorithm. The remaining 18 unseen scenarios are reserved for validation of the control policy discovered by the optimization algorithm. We note that all four control strategies are able to achieve similar generation costs while maintaining all the other constraints (line thermal capacities, ramping limits, and integral energy constraints), however, there are significant differences in the quality of the frequency regulation. Figure 6.7b shows the worst-case frequency deviations over the 18 validation scenarios. The frequency deviations are at an unacceptable level (.1-.2 Hz) when using just PI feedback (scheme 1). If the flow feedback is included but optimized separately (scheme 2), there is little improvement. However, the if the PI and flow feedback are coordinated via joint optimization (scheme 3), the frequency deviations are reduced to an acceptable level. Interestingly, removing the feedback on the integral of the frequency deviations (scheme 4) does not impact the frequency deviations significantly relative to scheme 3.

Discussion of the Results

The distributed frequency control method we have presented benefits greatly from the incorporation of local power flows as demonstrated in Fig. 6.7b. There are several possible reasons for this improved performance. First, power flows make the local generation-load imbalances visible to the generators so that the closest generators respond, effectively screening the more distant generators from the need to respond. When compared to feedback based on frequency deviation, which is a global measure of the imbalance, feedback on local power flows confines imbalances to shorter spatial scales with a corresponding decrease in the time scale of the response. An alternative explanation is that the optimization over the ensemble of possible futures in Eq. 6.11 is acting as a sort of machine learning that encodes correlations between the wind prediction errors and the resulting local power flows into the flow feedback parameters. When wind prediction error occurs, the change in power flows drives the feedback to nearly compensate for the error without a frequency deviation existing for any significant length of time. More numerical experiments are required to distinguish between these two (and other) possibilities. In both of the possibilities discussed above, variations in the local power flows appear to be acting as "pseudo-communication" channels between the renewable and controllable generators. Such a communication analogy may help explain why the independent optimizations in scheme 2 does not yield significant improvement in control performance. The first optimization over frequency deviations may effectively washout the important local information in the power flows such that it is not available when optimizing over power flows.

6.8.5 Conclusions and Future Work

We introduced a control architecture based on off-line centralized optimization that can occur on a slow time scale coupled that sets the feedback parameters for fast distributed control of generation. The control scheme takes into account explicitly the variability in renewable generation using ensemble control. We showed that local feedback based on line flows and frequency deviations is sufficient to maintain all operational constraints and limit frequency deviations to an acceptable level even when the system is experiencing significant

ramps in wind generation. Our method exploits the hour-scale predictability of wind energy while using the off-line optimization to re-adjust control policies over longer timescales where wind predictability suffers. Our hybrid approach has the potential enable even higher levels time-intermittent renewable generation than presented here, and it can do so without real-time computation or communication.

These results are quite exciting and promising, however, they are preliminary and much work needs to be done to ensure the viability of this scheme in practice.

- Dynamical simulations are needed to check the dynamical stability of a grid with flow feedback. If these simulations show that the scheme is unstable, we believe that this can be rectified by appropriate exciter control at the generators to damp the fast electro-mechanical transients.

- The scenario approach can be extended to include the (N-1) security criterion, so that the optimized control strategy can deal with contingencies arising from the failure of a grid component.

- It is possible that flow feedback acts as a pseudo-communication channel between generators in the absence of a dedicated communication channel. It would be interesting to investigate this from an information theoretic point of view and investigate how much of information can be encoded in the flows.

- We have used the simplest possible algorithmic approach by defining a smooth version of the optimization problem using penalty functions solving it using a generic LBFGS algorithm Schmidt. Second-order algorithms such as Stagewise NewtonPantoja [1983] or Differential Dynamic Programming (DDP)Jacobson [1968] efficiently exploit the problem structure of deterministic optimal control problems. These can be leveraged in our ensemble control context by noting that when the feedback parameters $\alpha^I, \alpha^P, \alpha^F$ are fixed, we have a deterministic optimal control problem in $\mathbf{p}_0^{\mathbf{go}}(t)$ for each scenario. We have also been working on a Gauss-Newton algorithm for optimiz-

ing the fixed feedback$\alpha^I, \alpha^P, \alpha^F$ efficiently. One can perform alternate minimization of $\mathbf{p}_0^{\mathbf{go}}(t)$ and $\alpha^I, \alpha^P, \alpha^F$ to get an efficient algorithm for optimizing both. Further, we note that when feedback does not include the integral term $\Omega(t)$, the ensemble control problem is a convex programming problem, and the global optimum can be found efficiently using specialized convex optimization techniques.

- We plan to incorporate more accurate AC modeling of power flows taking advantage of most recent advances in analysis and algorithms related to optimizations of nonlinear power flows, e.g. Lavaei and Low [2012], Kraning et al. [2012].

- The integral energy constraint we introduced can also model energy storage, and our algorithm can easily be extended to incorporate distributed control of energy storage.

Chapter 7

A VISION FOR AUTOMATED STOCHASTIC CONTROL

This chapter presents preliminary ideas for combining the ideas of inverse optimal control (chapter 3) and convex policy optimization (chapter 5) into an integrated cost-policy shaping framework. We believe that this is critical to realizing the long-term vision of research presented in this thesis: An automated framework for stochastic optimal control that takes as input a set of plausible high level costs, a model of the system dynamics and demonstrations of the control task (figure 7.1). This is an important for the following reasons:

1 Even though costs may be simple to specify for many control problems, these costs are often uninformative (for example, one suffers a fixed cost unless the system ends up in a particular goal state). It has been known that faster convergence can be achieved for stochastic optimal control algorithms by designing appropriate costs Ng et al. [1999]. How does one go about automating this process of reward or cost shaping?

2 Further, for many problems, there is a fixed cost that one needs to minimize to accomplish the control task. However, one needs to add additional costs in order to produce

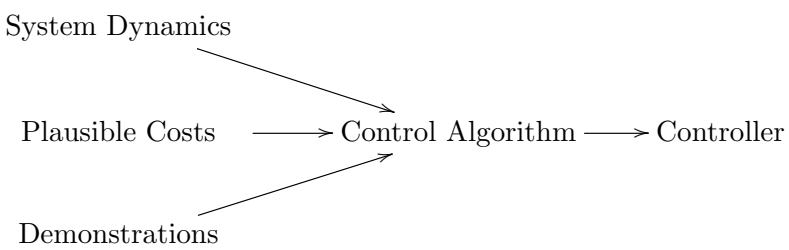


Figure 7.1: Data-Driven Cost-Shaping Controller Design

desirable behavior. For example, for robot locomotion, a cost penalizing the deviation of the com velocity from a target forward velocity should suffice. However, many times, in addition to this, one wants to achieve a certain walking style, or penalize the use of some joints (avoid excessive arm swinging) etc. Again, we try to address the question of how one automate the process of augmenting cost functions in order to achieve desired behavior while accomplishing the control task.

3 Iterative processes that alternate between cost shaping and control design given costs are often too cumbersome and may involve solving more difficult optimal control problems than required. Hence, it would be desirable to have an algorithm to learn both costs and policies give data, a model and a family of plausible costs, as outlined in figure 7.1.

In section 7.1, we describe algorithms that take advantage of the properties of LMDP-sto develop convex optimization-based methods for automatically shaping costs in order to match some notion of "prior" costs and fit data coming from demonstrations of the control task being accomplished. However, the disadvantage is that this approach still assumes that LMDPs can be solved efficiently, which requires approximations and heuristics in practice. However, the approach does allow us to deal with general prior costs and incorporate information from both positive (successful) and negative (unsuccessful) demonstrations in cost shaping and control design.

In section 7.2, we build on ideas from 5.2 and develop a framework that can jointly optimize over both cost function parameters and control policy parameters in a unified convex approach to control design and cost shaping.

7.1 Convex Data-Driven Cost-Shaping for LMDPs

In this work, we consider problems where one has a prior cost function $\mathcal{Q}(\mathbf{X})$ (we allow this to be a general cost on trajectories: It can include state and control costs, with controls implicitly determined through state transitions). This cost can be something abstract and high-level. An example is a cost that prevents a robot from falling while walking, by

penalizing the inverse of the distance of the robot's head from the ground. However, this cost does not completely determine the task specification (for example, not falling is necessary for walking but not sufficient). Thus, one needs to infer additional information about the cost from other data: In this work, we look at inferring this cost from demonstrations of the task being performed. Given these, we seek an LMDP cost $\ell(\mathbf{X};\theta)$ whose corresponding optimal policy (given by (7.1)) maximizes the likelihood of the observed demonstrations, while minimizing the prior cost $\mathcal{Q}(\mathbf{X})$ in expectation. We allow for both desirable demonstrations (denote $\{\mathbf{X}_+^{(i)}\}_{i=1}^{\mathrm{N}_+}$) and undesirable demonstrations ($\{\mathbf{X}_-^{(i)}\}_{i=1}^{\mathrm{N}_-}$).

Within this general framework, we consider two different parameterizations of the cost function, which both result in convex formulations for the cost-shaping problem.

7.1.1 Optimally Controlled Trajectory Distribution

Given a state cost function $\ell(\mathbf{x})$, the distribution of trajectories ($\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots]$) under the distribution of trajectories under the optimal control policy for an LMDP (in the finite horizon or first-exit formulations) is given by:

$$\Pi^*(\mathbf{X}) = \frac{\Pi^0(\mathbf{X}) \exp\left(-\sum_t \ell(\mathbf{x}_t)\right)}{\sum_{\mathbf{X}} \Pi^0(\mathbf{X}) \exp\left(-\sum_t \ell(\mathbf{x}_t)\right)} = \frac{\Pi^0(\mathbf{X}) \exp\left(-\ell(\mathbf{X})\right)}{\sum_{\mathbf{X}} \Pi^0(\mathbf{X}) \exp\left(-\ell(\mathbf{X})\right)}.$$

In this paper, we will work with parameterized cost functions $\ell(\mathbf{X};\theta)$, where $\theta \in \mathbf{R}^n$ is a vector of real-valued parameters. We denote by $\Pi^\theta(\mathbf{X})$ the optimal distribution of trajectories for cost $\ell(\mathbf{X};\theta)$:

$$\Pi^\theta(\mathbf{X}) = \frac{\Pi^0(\mathbf{X}) \exp\left(-\ell(\mathbf{X};\theta)\right)}{\sum_{\mathbf{X}'} \Pi^0(\mathbf{X}') \exp\left(-\ell(\mathbf{X}';\theta)\right)}. \tag{7.1}$$

7.1.2 Parameterizing costs linearly

In this formulation, we parameterize costs linearly

$$\ell(\mathbf{X};\theta) = \theta^T f(\mathbf{X}) = \sum_{t=1}^{T} \theta^T f(\mathbf{x}_t).$$

The objective is then formulated as a combination of two terms: The first term is the average negative log-likelihood of observations of the system. The second term is the expected abstract cost. Mathematically, we solve

$$\underset{\theta}{\text{Minimize}}\ \text{CCS}_1(\theta) = \underbrace{-\frac{1}{\text{N}_+}\sum_{i=1}^{\text{N}_+}\log\left(\Pi^\theta\left(\mathbf{X}_+^{(i)}\right)\right)}_{\text{Negative Log-Likelihood}} + \lambda\underbrace{\log\left(\underset{\Pi^\theta(\mathbf{X})}{\text{E}}\left[\mathcal{Q}(\mathbf{X})\right]\right)}_{\text{log expected prior cost}} \tag{7.2}$$

$$\Pi^\theta(\mathbf{X}) = \frac{\Pi^0(\mathbf{X})\exp\left(-\ell\left(\mathbf{X};\theta\right)\right)}{\sum_{\mathbf{X}'}\Pi^0(\mathbf{X}')\exp\left(-\ell\left(\mathbf{X}';\theta\right)\right)}.$$

$$\ell(\mathbf{X};\theta) = \theta^T\mathbf{X}$$

Theorem 7.1.1. *If $\lambda < 1$, the problem* (7.2) *is a convex optimization problem.*

Without the second term, this parameterization is equivalent to maximum entropy IRL Ziebart et al. [2008b] (with a uniform $\Pi^0(\mathbf{X})$) or to the OptQ algorithm in Dvijotham and Todorov [2010]. However, we show here that adding a regularizer that minimizes a prior abstract cost also preserves convexity, as long as the weight on the regularizer is smaller than the weight on the average data likelihood term.

7.1.3 Parameterizing costs in log-space

We also consider an alternative formulation where one parameterizes $-\log\left(\ell\left(\mathbf{X};\theta\right)\right)$ linearly. minimizes the maximum of the negative log-likelihood over the data points $\{\mathbf{X}_+^{((i))}\}$ (this formulation says that any of the observed trajectories ought to have a certain minimum likelihood):

$$\min_{\theta}\ \text{CCS}_2(\theta) = \underbrace{\max_i -\log\left(\Pi^\theta\left(\mathbf{X}_+^{((i))}\right)\right)}_{\text{Negative Log-Likelihood}} + \lambda\underbrace{\log\left(\underset{\Pi^\theta\left(\mathbf{X}_+^{((i))}\right)}{\text{E}}\left[\mathcal{Q}(\mathbf{X})\right]\right)}_{\text{log expected prior cost}} \tag{7.3}$$

$$\Pi^\theta(\mathbf{X}) = \frac{\Pi^0(\mathbf{X})\exp\left(-\ell\left(\mathbf{X};\theta\right)\right)}{\sum_{\mathbf{X}'}\Pi^0(\mathbf{X}')\exp\left(-\ell\left(\mathbf{X}';\theta\right)\right)}.$$

$$\ell(\mathbf{X};\theta) = -\log\left(\theta^T f(\mathbf{X})\right)$$

Theorem 7.1.2. *The problem* (7.3) *can be solved using quasi-convex optimization in* θ *(a sequence of convex-feasibility problems combined with search in a 2-D grid).*

We also allow for undesirable demonstrations $\{\mathbf{X}_-^{((i))}\}$ whose likelihood we want to minimize. This allows us to define the following objective:

$$\min_{\theta} \mathrm{CCS}_2(\theta) = \underbrace{\max_i -\log\left(\Pi^\theta\left(\mathbf{X}_+^{((i))}\right)\right)}_{\text{Negative Log-Likelihood}} + \lambda \underbrace{\log\left(\mathop{\mathrm{E}}_{\Pi^\theta\left(\mathbf{X}_+^{((i))}\right)}[\mathcal{Q}(\mathbf{X})]\right)}_{\text{log expected prior cost}} \tag{7.4}$$

$$\text{Subject to } \max_i \log\left(\Pi^\theta\left(\mathbf{X}_-^{((i))}\right)\right) \leq \mu \tag{7.5}$$

where μ is a threshold for the likelihood of undesirable trajectories.

Theorem 7.1.3. *When* $\ell(\mathbf{X};\theta) = -\log\left(\theta^T f(\mathbf{X})\right)$ *for arbitrary positive-valued features* $f(\mathbf{X})$*, the problem* (7.4) *can be solved using quasi-convex optimization in* θ *(a sequence of convex-feasibility problems combined with search in a 2-D grid).*

7.1.4 Appplications

Combining trajectory tracking costs

For a linear dynamical system, the passive dynamics $\Pi^0(\mathbf{X})$ are jointly Gaussian. Further, if $\ell(\mathbf{X};\theta) = -\log\left(\theta^T f(\mathbf{X})\right)$ where $f(\mathbf{X})$ is chosen to be a mixture of gaussians x polyomials in $\mathbf{X}$, with $\ell(\mathbf{X})$ having the same form, all the expectations in (7.3) can be evaluated analytically and the convex cost-shaping problem can be solved exactly in closed form. A particularly useful parameterization is of the form:

$$-\log\left(\sum_i \left(\frac{\mathbf{X}^T W_i \mathbf{X}}{2} + {w_i}^T \mathbf{X} + c_i\right) \exp\left(-\left\|\mathbf{X} - \mathbf{X}^{(i)}\right\|^2\right)\right)$$

which allows one to parameterize costs as a combination of trajectory tracking costs (here W_i, w_i, c_i are the cost parameters). This is of the form specified required in (7.3) or (7.4). More specifically, we can choose W_i, w_i, c_i such that

$$\begin{aligned}
&\frac{\mathbf{X}^T W_i \mathbf{X}}{2} + {w_i}^T \mathbf{X} + c_i \\
&= \sum_{t=1}^{T} \frac{{\mathbf{x}_t}^T W_{it} \mathbf{x}_t}{2} + {\mathbf{x}_t}^T w_{it} + c_{it} \\
&= \sum_{t=1}^{T} \frac{\left(\mathbf{x}_t + {W_{it}}^{-1} w_{it}\right)^T W_{it} \left(\mathbf{x}_t + {W_{it}}^{-1} w_{it}\right)}{2} + \left(c_{it} - \frac{{w_{it}}^T {W_{it}}^{-1} w_{it}}{2}\right)
\end{aligned}$$

Thus, this can be used to encode a quadratic trajectory tracking cost which decays to 0 as one moves away from the trajectory $\mathbf{X}^{(i)}$.

This formulation has the natural interpretation of trying to combine local trajectory tracking costs (which are usually more amenable to numerical optimization) in order to produce a controller that optimizes the real cost $\mathcal{Q}(\mathbf{X})$. Note that this application makes sense even in the absence of the likelihood term: One can simply pose it as a problem of finding a shaping cost (combination of trajectory tracking costs) that produces a controller that minimizes a master cost $\mathcal{Q}(\mathbf{X})$ that one really cares about.

This can be extended further to dynamics expressed as a mixture of Gaussian distributions (perhaps arising out of different time-varying linearizations of true nonlinear dynamics starting at different initial states).

This idea of optimally combining trajectory costs applies to arbitrary nonlinear dynamical systems as well. However, the integrals cannot be evaluated analytically in these case and sampling-based approximations will have to be used.

Learning Final Costs

This application looks at learning final costs for a finite-horizon or first exit control problem optimally. Consider a problem with a final cost $\ell_f(\mathbf{x}) = -\log\left(\sum_i w_i \exp\left(-f_i(\mathbf{x})\right)\right)$ and some fixed running cost $\ell_r(\mathbf{X}) = \sum_{t=1}^{T-1} \ell_t(\mathbf{x}_t)$. Then, the overall cost can be written as

$$\ell(\mathbf{X}) = -\log\left(\sum_i w_i f_i(\mathbf{x}_T) \exp\left(-\ell_r(\mathbf{X})\right)\right)$$

Defining $f_i(\mathbf{X}) = f_i(\mathbf{x}_T) \exp\left(-\ell_r(\mathbf{X})\right)$, this falls into the form required in (7.3) and (7.4). For model-predictive control, often the final cost can be taken as a proxy for a value function

(which is required to make the horizon short enough to enable real-time implementation). Thus, the final cost can be optimized so that the resulting controller minimizes a "true cost", matches desirable demonstrations and stays away from undesirable demonstrations.

Again, for linear dynamical systems with costs that are mixtures of Gaussians x polynomials, the resulting integrals can be evaluated analytically leading to a tractable formulation that finds the optimal final cost in polynomial time.

7.1.5 Relationship to Prior Work

Relationship to PI^2

In (7.2), when $\lambda \to \infty$, this algorithm can be seen as performing policy gradients (where the policy is parameterized implicitly by the cost parameters $\ell(\mathbf{X}; \theta)$). When we restrict ourselves to control-affine systems with an inverse relationship between noise and control costs, the policy gradient updates derived from this formulation coincide with the PI^2 algorithm Theodorou et al. [2010b]. Of course, for this large λ, the problem is no longer convex and is susceptible to local optima. An interesting takeaway is that if one has reliable demonstrations, one can regularize the policy optimization problem to get rid of local minima in the policy optimization problem.

Relationship to Policy Gradients in MDPs

In Todorov [2010b], the authors propose parameterizing the one-step controlled dynamics $Pol\mathbf{x}'\mathbf{x}$ in an LMDP and derive a policy gradient theorem for the infinite horizon formulation. However, their approach requires computing value function corresponding to a given policy (as in an actor-critic architecture), requiring approximate evaluation based on Temporal-Difference learning and other reinforcement learning algorithms. Further, there are no guarantees that the policy optimization problem is convex. The advantage of their formulation, however, is that they directly learn a control policy that can be implemented on the system. In our approach, we learn a shaping cost that needs to be optimized (for example using model-predictive control) to produce a successful controller for the system. We do not have an explicit control policy except in the special cases where analytical integration

is possible.

Relationship to MaxEnt IRL/Inverse Optimal Control in LMDPs

The formulation (7.2) is very close to MaxEnt IRL and the equivalent OptQ algorithm presented in Dvijotham and Todorov [2010]. These can be recovered as special cases with $\lambda \to 0$ (when the prior cost term is dropped, effectively).

Relationship to Compositionality Properties in LMDPs

The compositionality properties Todorov [2009d] allow one to compose solutions to individual control problems. However, the work here goes one-step further: It answers the question: How do I use the composable costs to generate a policy to minimize a true cost (which may not be in the particular form that allows easy composability)?

7.1.6 Appendix

Proof of Theorem 1

Proof. Writing out the objective (7.2), we get

$$\begin{aligned}
&\frac{1}{N}\left(\sum_{i=1}^{N}\theta^T f(\mathbf{X})\right) + \log\left(\sum_{\mathbf{X}}\Pi^0(\mathbf{X})\exp\left(-\theta^T f(\mathbf{X})\right)\right) \\
&+ \lambda\log\left(\sum_{\mathbf{X}}\Pi^0(\mathbf{X})\exp\left(-\theta^T f(\mathbf{X})\right)\mathcal{Q}(\mathbf{X})\right) - \lambda\log\left(\sum_{\mathbf{X}}\Pi^0(\mathbf{X})\exp\left(-\theta^T f(\mathbf{X})\right)\right) \\
&= \frac{1}{N}\left(\sum_{i=1}^{N}\theta^T f(\mathbf{X})\right) + \lambda\log\left(\sum_{\mathbf{X}}\Pi^0(\mathbf{X})\exp\left(-\theta^T f(\mathbf{X})\right)\mathcal{Q}(\mathbf{X})\right) \\
&\quad + (1-\lambda)\log\left(\sum_{\mathbf{X}}\Pi^0(\mathbf{X})\exp\left(-\theta^T f(\mathbf{X})\right)\right)
\end{aligned}$$

When, $0 \leq \lambda \leq 1$, all three terms are convex in θ (the first term is linear and the other two are of the log-sum-exp form).

□

Proof of Theorem 2

Proof. We will peform binary search on t, at each step solving the feasibility problem

$$\text{Find } \theta \text{ such that } \ \mathrm{CCS}_2(\theta) \leq t.$$

We show that each such feasibility problem can again be solved as a sequence of convex feasibility problems.

For each t, the constraint $\mathrm{CCS}_2(\theta) \leq t$ can be written as

$$\max_i -\log\left(\frac{\theta^T f(\mathbf{X}^{(i)})}{\theta^T \mathrm{E}_{\Pi^0(\mathbf{X})}[f(\mathbf{X})]}\right) + \lambda \log\left(\frac{\theta^T \mathrm{E}_{\Pi^0(\mathbf{X})}[f(\mathbf{X})\,\mathcal{Q}(\mathbf{X})]}{\theta^T \mathrm{E}_{\Pi^0(\mathbf{X})}[f(\mathbf{X})]}\right) \leq t$$

Suppose we assume that the second term takes value s. The problem then becomes the convex (LP) feasibility problem:

$$\theta^T f(\mathbf{X}^{(i)}) \leq \exp(t - \lambda s)\left(\theta^T \mathop{\mathrm{E}}_{\Pi^0(\mathbf{X})}[f(\mathbf{X})]\right) \quad \forall i$$
$$\theta^T \mathop{\mathrm{E}}_{\Pi^0(\mathbf{X})}[f(\mathbf{X})\,\mathcal{Q}(\mathbf{X})] = \exp(s)\left(\theta^T \mathop{\mathrm{E}}_{\Pi^0(\mathbf{X})}[f(\mathbf{X})]\right)$$

Thus, for every value of s, t, we have a convex (in fact LP) feasibility problem in θ. Performing binary search on t and grid search on s for each t, we can find the global optimum by solving a sequence of LP feasibility problems. This requires having bounds on s, t. Bounds on s are easily specifiable based on expected values of observations.

□

7.1.7 Proof of Theorem 3

Proof. Follows from theorem 2 and the quasi-convexity of the log-likelihood function. □

7.2 Convex Data-Driven Policy Optimization with Cost Shaping

In this section, we outline some ideas for combining the ideas from 5.2 with the ideas of cost-shaping presented in this chapter. Suppose that there is no external noise and that the

discrete-time dynamics is affine in the controls:

$$\mathbf{x}_{t+1} = \mathrm{a}\left(\mathbf{x}_t\right) + \mathrm{B}\left(\mathbf{x}_t\right)\left(u_t + \omega_t\right), \omega_t \sim \mathcal{N}\left(0, \Sigma_t\right).$$

Further, suppose that the policy is stationary: $u_t = K\phi\left(\mathbf{x}_t\right) \forall t$. In the absence of external noise, the convex stochastic policy optimization problem (5.17) can be written as

$$\begin{aligned}
\underset{K}{\text{Minimize}} \quad & \underset{\omega_t \sim \mathcal{N}(0,\Sigma_t)}{\mathrm{E}} \left[\exp\left(\alpha\left(\sum_{t=1}^{N} \ell_{\mathrm{t}}(\mathbf{x}_t) + \sum_{t=1}^{N-1} \frac{{u_t}^T R_t u_t}{2}\right)\right)\right] \\
\text{Subject to} \quad & \mathbf{x}_{t+1} = \mathrm{a}\left(\mathbf{x}_t\right) + \mathrm{B}\left(\mathbf{x}_t\right)\mathbf{y}_t \quad t = 1, 2, \ldots, N-1 \\
& \mathbf{y}_t = \mathrm{K}\,\phi\left(\mathbf{x}_t\right) + \omega_t, \omega_t \sim \mathcal{N}\left(0, \Sigma_t\right) \quad t = 1, 2, \ldots, N-1 \\
& K \in \mathcal{C}
\end{aligned}$$

Suppose that we are in the cost shaping setting with parameterized costs $\ell\left(\mathbf{X}; \theta\right)$. Further suppose that we have demonstrations from which we can extract pairs $\{\mathbf{x}^{(i)}, u^{(i)}\}_{i=1}^{m}$ (say using inverse dynamics). One can define a combined cost-shaping and policy optimization problem:

$$\begin{aligned}
\underset{K,\theta}{\text{Minimize}} \quad & \underbrace{\frac{1}{\alpha}\log\left(\underset{\omega_t \sim \mathcal{N}(0,\Sigma_t)}{\mathrm{E}} \left[\exp\left(\alpha\left(\ell\left(\mathbf{X};\theta\right) + \sum_{t=1}^{N-1} \frac{{u_t}^T R u_t}{2}\right)\right)\right]\right)}_{\text{Minimize Parameterized Cost}} \\
& \qquad + \underbrace{\mu\left(\sum_{i=1}^{m} \left\| u^{(i)} - K\phi\left(\mathbf{x}^{(i)}\right)\right\|_{\Sigma^{-1}}^2\right)}_{\text{Fit Data}} \\
\text{Subject to} \quad & \mathbf{x}_{t+1} = \mathrm{a}\left(\mathbf{x}_t\right) + \mathrm{B}\left(\mathbf{x}_t\right)\mathbf{y}_t \quad t = 1, 2, \ldots, N-1 \\
& \mathbf{y}_t = \mathrm{K}\,\phi\left(\mathbf{x}_t\right) + \omega_t, \omega_t \sim \mathcal{N}\left(0, \Sigma\right) \quad t = 1, 2, \ldots, N-1 \\
& K \in \mathcal{C}
\end{aligned} \tag{7.6}$$

Theorem 7.2.1. *If $\ell\left(\mathbf{X}; \theta\right)$ depends in an affine (or more generally convex) way on θ, the overall resulting optimization problem* (7.6) *is convex.*

Proof. The proof follows from the convexity proof of (5.17): The term inside the exponent is a convex function of K, and a convex function of θ is added to it, so the result is jointly convex in K, θ. The second term is simply a least-squares fit and is hence convex. □

Chapter 8

CONCLUSIONS & FUTURE WORK

In this thesis, we have presented theoretical and algorithmic developments for solving problems of inverse optimal control, risk-sensitive control, policy optimization and cost shaping, as well adaptations and applications of these ideas to problems in power and energy systems. We have studied a diverse set of problems and issues, but we believe that all the pieces come together nicely in the vision outlined in chapter 7. As computing capabilities increase, we are at a stage where it is feasible to process huge amounts of data in a scalable manner using parallel distributed setups. The "big-data" revolution has gained momentum in several areas: Science, Healthcare, E-commerce, Genetics etc. All of these fields are now adopting automated data-driven approaches successfully and are able to go beyond traditional limitations of human intuition and analyses based on first-principle models. Several businesses now use data-driven approaches to making business decisions: However, these are still in the realm of policy level decisions and not real-time reactive control of the form typically studied in control theory. We believe that the time is ripe for stochastic control to take advantage of these developments and that such a paradigm shift will help scale stochastic control to new applications that have so far been infeasible. This would be particularly useful for large and complex systems like the electric power grid, where data-driven model building and control design would be key to enabling new demand-side devices that can provide useful regulation services (demand response, distributed generation) to the grid and help reduce the dependence on non-renewable resources while supporting the increased penetration of fluctuating resources like wind. The theoretical and algorithmic developments in this thesis, especially as they come together in chapter 7(section 7.2), present a promising framework for realizing this vision.

From the perspective of the user, the control design process involves:

1 Collecting demonstrations of the control task being performed successfully. This can

be done in simulation, by performing trajectory optimization Tassa et al. [2012] starting and different initial conditions and under different noise sequences.

2 Defining a set of plausible costs. This can simply be a list of various features of the state that the cost can depend on.

3 Solving the combined cost-shaping and control-design problem (7.6) using stochastic gradient methods.

This is fairly automated from the point of view of the end-user and does for control what algorithms like Support Vector Machines (SVMs) have done for machine learning: Allow the user to work at the level of features and demonstrations (training data) that can be collected easily and not worry about details of the particular system being studied or the control design process.

Of course, the ideas presented here are preliminary and numerical studies need to be done to validate this approach. In particular, the following are immediate directions for future work:

1 Applications of stochastic gradient methods have mainly focused on applications in supervised learning which typically lead to well-conditioned optimization problems. Control problems tend to be more sensitive and badly conditioned, so we would need to investigate the use of second order methods. There has been recent progress along these lines Byrd et al. [2014] and we hope to make use of these developments.

2 Further, we presented a model-free algorithm for control design (algorithm 2). However, these methods may exhibit slow convergence (our preliminary numerical experiments indicate this too). Thus, for difficult-to-model systems, we would need to first perform system identification and use the model-based variant (algorithm 1). However, this approach can be extended to deal with minimizing costs over an ensemble of possible models and can be extended to an adaptive control.

BIBLIOGRAPHY

BPA wind data. http://transmission.bpa.gov/business/operations/wind/, a.

BPA meteorological site map. http://transmission.bpa.gov/business/operations/wind/MetData.aspx, b.

BPA wind forecast data. http://transmission.bpa.gov/Business/Operations/Wind/forecast/forecast.aspx, c.

BPA wind site map. http://transmission.bpa.gov/PlanProj/Wind/documents/BPA_wind_map_2011.pdf, d.

P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *International Conference on Machine Learning*, 21, 2004.

Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

S.I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Martin S Andersen, Joachim Dahl, and Lieven Vandenberghe. Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones. *Mathematical Programming Computation*, 2(3):167–201, 2010.

Pierre Apkarian, Dominikus Noll, and Aude Rondepierre. Mixed h-2/h-inf control via nonsmooth optimization. *SIAM Journal on Control and Optimization*, 47(3):1516–1546, 2008.

C. Baker, J. Tenenbaum, and R. Saxe. Goal inference as inverse planning. *Annual Conference of the Cognitive Science Society*, 29, 2007.

T. Başar and P. Bernhard. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Birkhauser, 1995. ISBN 0817638148.

Tamer Başar and Pierre Bernhard. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer, 2008.

J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Robert Becker. The variance drain and jensen's inequality. *CAEPR Working Paper*, 2012.

R. Bellman. Dynamic Programming, Princeton. *NJ: Princeton UP*, 1957.

D. Bienstock, M. Chertkov, and S. Harnett. Chance Constrained Optimal Power Flow: Risk-Aware Network Control under Uncertainty. *ArXiv e-prints*, September 2012.

Vincent Blondel and John N Tsitsiklis. Np-hardness of some linear control design problems. *SIAM Journal on Control and Optimization*, 35(6):2118–2127, 1997.

Vincent D Blondel and John N Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

S. Bose, D.F. Gayme, U. Topcu, and K.M. Chandy. Optimal placement of energy storage in the grid. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5605–5612, 2012. doi: 10.1109/CDC.2012.6426113.

F. Bouffard and F.D. Galiana. Stochastic security for operations planning with significant wind power generation. In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, pages 1 –11, july 2008. doi: 10.1109/PES.2008.4596307.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambrdige, UK, 2004.

Stephen P Boyd. *Linear matrix inequalities in system and control theory*, volume 15. Siam, 1994.

B. Van Den Broek, W. Wiegerinck, and B. Kappen. Risk sensitive path integral control. In *Uncertainty in AI, 2010. Proceedings of the 2010*, 2010.

JV Burke, D Henrion, AS Lewis, and ML Overton. Hifoo-a matlab package for fixed-order controller design and h optimization. In *Fifth IFAC Symposium on Robust Control Design, Toulouse*, 2006.

RH Byrd, SL Hansen, Jorge Nocedal, and Y Singer. A stochastic quasi-newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.

Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex algebraic geometry of linear inverse problems. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 699–703. IEEE, 2010.

A. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, 2005.

M. Da Silva, F. Durand, and J. Popović. Linear Bellman combination for control of character animation. *ACM Transactions on Graphics (TOG)*, 28(3):1–10, 2009. ISSN 0730-0301.

H. Deng and M. Krstic. Stochastic nonlinear stabilization - II: Inverse optimality. *Systems and Control Letters*, 32:151–159, 1997.

Geir E Dullerud and Fernando Paganini. *A course in robust control theory*, volume 6. Springer New York, 2000.

K. Dvijotham, M. Chertkov, and S. Backhaus. Operations-based planning for placement and sizing of energy storage in a grid with a high penetration of renewables. *Arxiv preprint arXiv:1107.1382*, 2011.

K. Dvijotham, S. Backhaus, and M. Chertkov. Distributed control of generation in a transmission grid with a high penetration of renewables. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pages 635–640, 2012. doi: 10.1109/SmartGridComm.2012.6486057.

Krishnamurthy Dvijotham and Emanuel Todorov. Inverse optimal control with linearly-solvable mdps. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 335–342, 2010.

J. Eto, J. Undrill, P. Mackin, H. Illian, A. Martinez, M. O'Malley, and K. Coughlin. Use of frequency response metrics to assess the planning and operating requirements for reliable integration of variable renewable generation. LBNL-4142E; `http://transmission.bpa.gov/business/operations/wind/`, 2010.

Chih-Hai Fan, Jason L Speyer, and Christian R Jaensch. Centralized and decentralized solutions of the linear-exponential-gaussian problem. *IEEE Transactions on Automatic Control*, 39(10):1986–2003, 1994.

D. Gayme and U. Topcu. Optimal power flow with large-scale storage integration. *Power Systems, IEEE Transactions on*, 28(2):709–717, 2013. ISSN 0885-8950. doi: 10.1109/TPWRS.2012.2212286.

Madeleine Gibescu, Arno J. Brand, and Wil L. Kling. Estimation of variability and predictability of large-scale wind energy in the netherlands. *Wind Energy*, 12(3):241–260, 2009. ISSN 1099-1824. doi: 10.1002/we.291. URL `http://dx.doi.org/10.1002/we.291`.

C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh. The ieee reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee. *Power Systems, IEEE Transactions on*, 14(3):1010 –1020, aug 1999. ISSN 0885-8950. doi: 10.1109/59.780914.

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2014. URL `http://www.gurobi.com`.

E. Hirst. Integrating wind energy with the bpa power system: Preliminary study. Technical report, Consulting in Electric-Industry Restructuring, 2002.

Eric Hirst and Brendan Kirby. Separating and measuring the regulation and load-following ancillary services. *Utilities Policy*, 8(2):75–81, June 1999. URL `http://ideas.repec.org/a/eee/juipol/v8y1999i2p75-81.html`.

Pablo A Iglesias. Tradeoffs in linear time-varying systems: an analogue of bode's sensitivity integral. *Automatica*, 37(10):1541–1550, 2001.

WIND GENERATION & Total Load in The BPA Balancing Authority. `http://transmission.bpa.gov/business/operations/wind/`.

Peters J. and Schaal S. Natural actor-critic. *Neurocomputing*, 71(7-9):1180 – 1190, 2008. ISSN 0925-2312. doi: 10.1016/j.neucom.2007.11.026. URL `http://www.sciencedirect.com/science/article/pii/S0925231208000532`.

D.H. Jacobson. Second-order and second-variation methods for determining optimal control: A comparative study using differential dynamic programming. *International Journal of Control*, 7(2):175–196, 1968.

R. Kalman. When is a linear control system optimal? *Trans AMSE J Basic Eng, Ser D*, 86:51–60, 1964.

Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

H.J. Kappen. Linear theory for control of nonlinear stochastic systems. *Physical Review Letters*, 95(20):200201, 2005.

K. Kording and D. Wolpert. The loss function of sensorimotor learning. *Proceedings of the National Academy of Sciences*, 101:9839–9842, 2004.

M. Kraning, E. Chu, J. Lavaei, and S. Boyd. Message passing for dynamic network energy management. `http://www.stanford.edu/~boyd/papers/decen_dyn_opt.html`, 2012.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3, 2012.

P. Kundur. *Power system stability and control*. McGraw-Hill, 1994.

Andrew Lamperski and Laurent Lessard. Optimal decentralized state-feedback control with sparsity and delays. *arXiv preprint arXiv:1306.0036*, 2013.

J. Lavaei and S. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, 2012.

Javad Lavaei. Optimal decentralized control problem as a rank-constrained optimization. In *Proceedings of the Allerton Conference on Control and Computing*, 2013.

Adrian S Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis*, 2(1):173–183, 1995.

Adrian S Lewis and Michael L Overton. Nonsmooth optimization via quasi-newton methods. *Mathematical Programming*, pages 1–29, 2012.

F. Lin, M. Fardad, and M. R. Jovanović. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Trans. Automat. Control*, 58(9): 2426–2431, September 2013.

S. Mahadevan and M. Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.

S.I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernandez, S. Coraluppi, and P. Fard. Risk sensitive Markov decision processes. *Systems and Control in the Twenty-First Century*, 29, 1997.

P. Meibom, R. Barth, B. Hasche, H. Brand, C. Weber, and M. O'Malley. Stochastic optimization model to study the operational impacts of high wind penetrations in ireland. *Power Systems, IEEE Transactions on*, PP(99):1 –12, 2010. ISSN 0885-8950. doi: 10.1109/TPWRS.2010.2070848.

Andrew Mills and Ryan Wiser. Implications of wide-area geographic diversity for short-term variability of solar power. Technical report, LBNL-3884E, 2010. URL `http://eetd.lbl.gov/ea/emp/reports/lbnl-3884e.pdf`.

Jun Morimoto, Garth Zeglin, and Christopher G Atkeson. Minimax differential dynamic programming: Application to a biped walking robot. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1927–1932. IEEE, 2003.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

A.Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann Publishers Inc., 2000.

Michael Overton. Hanso http://www.cs.nyu.edu/overton/software/hanso/, 2013.

J.F.A.D.O. Pantoja. Algorithms for constrained optimization problems. *Differential dynamic programming and Newton's method. International Journal of Control*, 47:1539–1553, 1983.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

D. Precup, R.S. Sutton, and S. Singh. Multi-time models for temporally abstract planning. In *Advances in Neural Information Processing Systems 11*, 1998.

Xin Qi, Murti V Salapaka, Petros G Voulgaris, and Mustafa Khammash. Structured optimal

and robust control with multiple criteria: A convex solution. *IEEE Transactions on Automatic Control*, 49(10):1623–1640, 2004.

: Rotkowitz, Michael and Sanjay Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, 2006.

Michael Rotkowitz and Sanjay Lall. Decentralized control information structures preserved under feedback. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 1, pages 569–575, 2002.

A. Sandryhaila and J. M. F. Moura. Eigendecomposition of Block Tridiagonal Matrices. *ArXiv e-prints Arxiv:1306.0217*, June 2013.

C. W. Scherer. Structured Hinf-Optimal Control for Nested Interconnections: A State-Space Solution. *ArXiv e-prints: arXiv:1305.1746*, May 2013.

M. Schmidt. Matlab *minfunc*. URL `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`.

M Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab, 2012.

P. Shah. H2-optimal decentralized control over posets: A state-space solution for state-feedback. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2013. ISSN 0018-9286. doi: 10.1109/TAC.2013.2281881.

Lloyd S Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, 39(10):1095, 1953.

J Speyer, John Deyst, and D Jacobson. Optimization of stochastic linear systems with additive measurement and process noise using exponential performance criteria. *Automatic Control, IEEE Transactions on*, 19(4):358–366, 1974.

M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:7072–7075, 2006.

J Michael Steele. *Stochastic calculus and financial applications*, volume 45. Springer, 2001.

R. Sutton, D. Mcallester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge MA, 1998a.

R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*. The MIT press, 1998b. ISBN 0262193981.

John Swigart and Sanjay Lall. An explicit dynamic programming solution for a decentralized two-player optimal linear-quadratic regulator. In *Proceedings of mathematical theory of networks and systems*, 2010.

U. Syed, M. Bowling, and R.E. Schapire. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*, pages 1032–1039. ACM, 2008.

Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, 2010.

Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE, 2012.

E. Theodorou, J. Buchli, and S. Schaal. Reinforcement learning of motor skills in high dimensions: A path integral approach. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2397–2403. IEEE, 2010a.

Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 9999: 3137–3181, 2010b.

E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9): 907–915, 2004a.

E. Todorov. Linearly-solvable Markov decision problems. *Advances in neural information processing systems*, 19:1369, 2007.

E. Todorov. Compositionality of optimal control laws. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1856–1864, 2009a.

E. Todorov. Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences*, 106(28):11478, 2009b.

E. Todorov. Eigen-function approximation methods for linearly-solvable optimal control problems. *IEEE ADPRL*, 2009c.

E. Todorov. Policy gradients in linearly-solvable mdps. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2298–2306, 2010a.

Emanuel Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7 (9):907–915, 2004b.

Emanuel Todorov. Compositionality of optimal control laws. In *Advances in Neural Information Processing Systems*, pages 1856–1864, 2009d.

Emanuel Todorov. Policy gradients in linearly-solvable mdps. *Advances in Neural Information Processing Systems*, 23:2298–2306, 2010b.

K. Tomsovic, D.E. Bakken, V. Venkatasubramanian, and A. Bose. Designing the next generation of real-time control, communication, and computations for large power systems. *Proceedings of the IEEE*, 93(5):965–979, 2005.

A. Treuille, Y. Lee, and Z. Popović. Near-optimal character animation with continuous control. In *ACM SIGGRAPH 2007 papers*, page 7. ACM, 2007.

A. Tuohy, E. Denny, and M. O'Malley. Rolling unit commitment for systems with significant installed wind capacity. In *Power Tech, 2007 IEEE Lausanne*, pages 1380 –1385, july 2007. doi: 10.1109/PCT.2007.4538517.

R. Williams. Simple statistical gradient following algorithms for connectionist reinforcement learning. *Machine Learning*, pages 229–256, 1992.

Hans S Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):131–147, 1968.

Guisheng Zhai, Masao Ikeda, and Yasumasa Fujisaki. Decentralized h-2/h-inf controller design: a matrix inequality approach using a homotopy method. *Automatica*, 37(4): 565–572, 2001.

Mingyuan Zhong. *Value Function Approximation Methods for Linearly-solvable Markov Decision Process*. PhD thesis, University of Washington, Seattle, May 2013.

B.D. Ziebart, A. Maas, J.A. Bagnell, and A.K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAAI*, pages 1433–1438, 2008a.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438, 2008b.

Ray D Zimmerman, Carlos E Murillo-Sánchez, and Deqiang Gan. A matlab power system simulation package, 2005.