VARIABLE RISK POLICY SEARCH FOR DYNAMIC ROBOT CONTROL

A Dissertation Presented

by

SCOTT ROBERT KUINDERSMA

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2012

Department of Computer Science

VARIABLE RISK POLICY SEARCH FOR DYNAMIC ROBOT CONTROL

A Dissertation Presented

by

SCOTT ROBERT KUINDERSMA

Approved as to style and content by:

Roderic A. Grupen, Co-chair

Andrew G. Barto, Co-chair

Sridhar Mahadevan, Member

Brian Umberger, Member

Lori A. Clarke, Department Chair
Department of Computer Science

To my parents, Robert & Jean Kuindersma.

ACKNOWLEDGMENTS

At the end of my Ph.D. journey, I find myself in the extremely fortunate position of having many people to thank. First and foremost, I owe much to my fantastic advisors, Rod Grupen and Andy Barto, who have been constant sources of wisdom and advice over the past six years. Rod has shown great trust in allowing me to pursue strange and sometimes dangerous experiments with his robots. His suggestions often led me down fruitful research paths and saved me from much unneeded work. Under his guidance, I have developed the skills required to do the thing I love: making robots move. Andy's breadth of knowledge and willingness to patiently evaluate ideas make him an indispensable resource for a developing researcher. His scholarship and clarity of thought are attributes I aspire to emulate. Rod and Andy, thanks for showing me the way.

I am grateful to my thesis committee members, Sridhar Mahadevan and Brian Umberger, with whom I have had many stimulating interactions and whose suggestions have improved this document greatly. My undergraduate mentor, Brian Blais, is responsible for sparking my interest in science and giving me early exposure to the joys and pains of research. He taught me to think carefully and to solve problems resourcefully. Thanks, Brian. I look forward to our many future conversations.

I have had the privilege of working in two fantastic research laboratories at UMass: the Laboratory for Perceptual Robotics (LPR) and the Autonomous Learning Laboratory (ALL). I would like to thank the members of these laboratories, past a present, for providing me with many laughs and helpful suggestions along the way. ALL: George Konidaris, Chris Vigorito, Sarah Osentoski, Ashvin Shah, Colin Barringer, Scott Niekum, Philip Thomas, Yariv Levy, Bruno Castro da Silva, Jeff Johns, Kim

Ferguson, Will Dabney, Andrew Stout, Armita Kaboli, Chang Wang, Jonathan Leahey, CJ Carey, Tom Helmuth, and Jie Chen. LPR: Dirk Ruiken, Shiraj Sen, Grant Sherrick, Steve Hart, Ed Hannigan, Patrick Deegan, Bryan Thibodeau, Dan Xie, Yun Lin, Chao Ou, Hee-Tae Jung, Takeshi Takahashi, Tom Billings, Emily Horrell, and Rob Platt. The lab administrators, Priscilla Scott and Gwyn Mitchell, and the graduate program manager, Leeanne Leclerc, are all fantastically knowledgable and hard working people that have made my life at UMass much easier. I owe them many thanks.

I would particularly like to acknowledge two colleagues who have been close collaborators and good friends: George Konidaris and Dirk Ruiken. As the experiments in this thesis suggest, much of my work has involved breaking the robot in creative ways. Dirk, thanks for showing me how to make it work again. George is one of the most supportive and motivated people I've had the pleasure of meeting. Working with him has made me a better researcher. George, thanks for making sure I remembered my slug horn.

For the last three years, I have received generous support from a Graduate Student Researchers Program (GSRP) Fellowship from NASA Johnson Space Center (JSC). I spent 22 weeks at JSC and had the opportunity to work on several exciting projects in the Dexterous Robotics Laboratory and the Robonaut 2 Laboratory. I am grateful to the fantastic engineers and scientists that made my experiences at JSC so rewarding: Bob Savely, Julia Badger, Stephen Hart, Ron Diftler, Bill Bluethmann, Paul Dinh, J.D. Yamokoski, Rob Ambrose, Frank Permenter, Dustin Gooding, Lyndon Bridgwater, Jonathan Rogers, Kody Ensley, Karl Brandt, Kim Hambuchen, Jerry Pratt, Peter Neuhaus, Twan Koolen, Lorraine Williams, and Mike Goza.

I am fortunate to have many talented, creative, and humorous friends that continue to keep me happy and sane. I am quite sure I will leave out several names as I write this two days before the deadline, but I am thankful to Adam and Mallory

Finne, Michael Taft, Chris Daniele and Kara Vicalvi, Matt and Sarah Nicholson, Alex "Gribbons" Corona, Christina Dentici, Colin Amidon, Jon Costanza, Tony DeNucce, Elizabeth Patridge, David and Devon DeNucce, Bill Lynch, Stacy Michael, Chris O'Hara, Sean Dube, Pete McConchie, Justin Kellett, Jeff Castonguay, Mike Mateer, Chris Hubert, Justin and Corinne Pierce, Jared Collins, Chris Rosati, George Konidaris, Chris Vigorito, Sarah Osentoski, Colin Barringer, Ashvin Shah, Scott Niekum, Rachael Singer, Philip Thomas, Yariv Levy, Dirk Ruiken, Shiraj Sen, Grant Sherrick, Charlotte Stanley, Steve Hart and Meg Inners Hart, Marc Cartright, Ilene Magpiong, Sam Huston, and Bobby Simidchieva.

Finally, I attribute all of my accomplishments to the unwavering support of my family. My parents, Robert and Jean Kuindersma, are the most patient and generous people I have ever met. I was not an easy child to raise, but they have always given me the means to pursue my (often rapidly changing) interests. Mom and Dad, thanks for always believing in me. This thesis is for you.

I am forever grateful to my beautiful and thoughtful wife, Jessica, for keeping me balanced and motivated for the past four years. Jess, thank you. I look forward to wandering together. My siblings, Todd Kuindersma and Pamela Melanson, and their families, Michelle and Daniel Kuindersma, and Kevin, Kaydence, and Kole Melanson, have been tremendously encouraging and an absolute joy to be around. Studying close to home has had its benefits. Last but not least, thanks to my very generous and supportive in-laws, William, Michelle, and Cory Baldwin, on whose dining room table I have written significant portions of this thesis.

ABSTRACT

VARIABLE RISK POLICY SEARCH FOR DYNAMIC ROBOT CONTROL

SEPTEMBER 2012

SCOTT ROBERT KUINDERSMA

B.Sc., BRYANT UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Roderic A. Grupen and Professor Andrew G. Barto

A central goal of the robotics community is to develop general optimization algorithms for producing high-performance dynamic behaviors in robot systems. This goal is challenging because many robot control tasks are characterized by significant stochasticity, high-dimensionality, expensive evaluations, and unknown or unreliable system models. Despite these challenges, a range of algorithms exists for performing efficient optimization of parameterized control policies with respect to average cost criteria. However, other statistics of the cost may also be important. In particular, for many stochastic control problems, it can be advantageous to select policies based not only on their average cost, but also their variance (or *risk*).

In this thesis, I present new efficient global and local risk-sensitive stochastic optimization algorithms suitable for performing policy search in a wide variety of problems of interest to robotics researchers. These algorithms exploit new techniques

in nonparameteric heteroscedastic regression to directly model the policy-dependent distribution of cost. For local search, learned cost models can be used as critics for performing risk-sensitive gradient descent. Alternatively, decision-theoretic criteria can be applied to globally select policies to balance exploration and exploitation in a principled way, or to perform greedy minimization with respect to various risk-sensitive criteria. This separation of learning and policy selection permits *variable risk control*, where risk sensitivity can be flexibly adjusted and appropriate policies can be selected at runtime without requiring additional policy executions.

To evaluate these algorithms and highlight the importance of risk in dynamic control tasks, I describe several experiments with the UMass uBot-5 that include learning dynamic arm motions to stabilize after large impacts, lifting heavy objects while balancing, and developing safe fall bracing behaviors. The results of these experiments suggest that the ability to select policies based on risk-sensitive criteria can lead to greater flexibility in dynamic behavior generation.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Motivation

Remarkable and beautiful feats of dynamic control, beyond our current ability to reproduce in robot systems, are ubiquitous in the animal kingdom. For example, consider Coquerel's sifaka, a species of lemur native to the dry deciduous forests of north-western Madagascar. The bodies of these animals are exquisitely specialized to the type of upright arboreal locomotion common to most lemur species. However, partially as a result of this adaptation, members of this species exhibit a remarkable terrestrial locomotion strategy of leaning forward and leaping several meters on their hind legs while using arm motions in flight to regulate the angular momentum of their bodies.

It is clear from such examples that behaviors are often constrained by, if not *guided by*, the physical properties of the embodied system. Sifakas need not cross flat terrain in such a spectacular fashion, but they do so because it is an efficient and reliable method given their bodies and predisposition to leaping behaviors. Likewise, the development of high-performance control policies in robot systems will depend strongly on the kinematic and dynamic properties of the system and the availability of instructive initial policies or suitably constrained behavior spaces. In nature, approximate or partial solutions to control problems are often natively present in infant members of a species. For example, several researchers have reported instances of wildebeest calves struggling to their feet, walking, and running with their herd less than 5 minutes after birth [118]. Native controllers are often improved or replaced

over time with more specialized behaviors that exploit innate dynamics in subtle ways that might be difficult to capture in even a very good system model. It may even be the case that many of these behaviors are discovered without explicit knowledge of the complicated nonlinear dynamics involved.

Sensitivity to *risk* (i.e., variation in performance) is another aspect of animal control that could be pervasive. One reason why this is hard to know for sure is that it is typically very difficult in practice to precisely identify the optimization being performed to produce a behavior (if optimization is, in fact, the correct way to describe such processes). However, there are some instances where the reward or cost associated with particular behaviors is externally measurable. For example, foraging strategies of a variety species have been extensively studied by behavioral ecologists [43, 10]. These studies have repeatedly shown that animals are sensitive to the variance of alternative food sources, where their propensity to be risk-seeking (i.e., preferring higher variance) or risk-averse depends on several factors such as energy reserves and number of available food sources. Other recent work in human motor control and learning has used explicit numerical signals as measures of performance [20, 86]. The results of these experiments suggest that humans may also be sensitive to risk when learning or solving simple control tasks.

The extent to which risk sensitivity plays a part in the optimization of low-level dynamic behaviors in nature is not currently known. However, for many robot systems, it is clear that risk is an important consideration. For example, imagine a humanoid robot that is capable of several dynamic walking gaits that differ based on their efficiency, speed, and predictability. When operating near a large crater, it might be reasonable to select a more predictable, possibly less energy-efficient gait over a less predictable, higher performance gait. Likewise, when far from a power source with low battery charge, it may be necessary to risk a fast and less predictable policy because alternative gaits have comparatively low probability of achieving the

required speed and efficiency. To create flexible systems of this kind, it will be necessary to design optimization processes that produce control policies that differ based on their risk. However, the majority of existing optimization algorithms suitable for solving control tasks in robot systems are designed to be risk-neutral, focusing on average performance and ignoring performance variation.

In this thesis, I consider the problem of learning dynamic behaviors in robot systems using methods that flexibly take risk-sensitivity into account. In particular, I consider the problem of efficiently optimizing parameterized policies, where both the expected cost and cost variance depend on the policy. I present new global and local stochastic optimization algorithms and examine their applicability for solving risk-sensitive policy search problems. By directly modeling the distribution of cost in policy parameter space, these algorithms support *variable risk* policy selection, where risk sensitivity can be flexibly specified and appropriate policies can be selected at runtime without requiring additional policy executions. To evaluate these algorithms and highlight the importance of risk in dynamic control tasks, I describe several experiments with the UMass uBot-5 that include learning dynamic arm motions to stabilize after large impacts, lifting heavy objects while balancing, and developing a safe fall bracing behavior. These experiments suggest that the ability to select policies based on risk-sensitive criteria leads to greater flexibility in dynamic behavior generation.

1.2 Summary of Contributions and Document Outline

The chapters in this work are organized as follows:

- **Chapter 2: Background and Related Work.** This chapter provides the necessary background to understand the contributions of this thesis. In particular, it includes a concise overview of the optimal control and risk-sensitive optimal control frameworks, and an overview of related work in reinforcement

learning on model-free policy search methods. I also provide a detailed introduction to Bayesian optimization algorithms and discuss their application to policy search.

- **Chapter 3: Learning Rapid Stabilizing Arm Motions via Global Policy Search.** This chapter describes experiments on learning arm motion policies for impact recovery with the uBot-5. Parameterized open-loop arm motions were efficiently optimized using Bayesian optimization and a cost function inspired by general observations of arm motion effects on recovery from the biomechanics literature. The learned arm motions, combined with a fixed closed-loop lower body response, significantly increased spatial efficiency, robustness, and energy efficiency. An unexpected result from these experiments was that different arm recovery policies have different sensitivity to initial conditions and hence significantly different cost variance. This policy-dependent variance motivates the development of the algorithm introduced in Chapter 4.

- **Chapter 4: Global Variable Risk Policy Search.** This chapter introduces a new algorithm, called *Variational Bayesian Optimization* (VBO), that extends the standard Bayesian optimization algorithm to the case where cost variance is policy dependent, a property present in many robot control tasks (including the task described in Chapter 3). The VBO algorithm is an extension of standard Bayesian optimization, where the Gaussian process model is replaced with the Variational Heteroscedastic Gaussian Process model [65]. I derive expressions for the expected improvement of a policy under the intractable variational distribution and show that confidence bound policy selection criteria, that have previously been studied in the context of Bayesian optimization, have a direct connection to risk-sensitive optimal control. Finally, I propose a generalized

selection criterion called *expected risk improvement* that balances exploration and exploitation in the risk-sensitive optimization setting.

Experimental results are presented from a simple artificial domain and from large-impact balance recovery experiments with the uBot-5.

- **Chapter 5: Local Variable Risk Policy Search.** This chapter proposes a local variable risk policy search algorithm based on stochastic gradient descent. Global policy search methods, such as Bayesian optimization, lack general convergence guarantees and can produce large policy changes between episodes, which may be undesirable for some systems. The *Risk-Sensitive Stochastic Gradient Descent* (RSSGD) algorithm addresses this shortcoming by using the learned distribution of cost as a local critic for performing gradient descent. Under certain assumptions, the algorithm descends the gradient of the risk-sensitive objective and the minimum variance update equation can be viewed as locally moving in the direction of risk improvement as defined in Chapter 4.

  Experimental results from a dynamic heavy lifting task are presented. The robot efficiently learned a policy for lifting a laundry detergent container that exploited the motion of the liquid in the bottle to cancel out the forward motion produced by the fixed closed-loop balancing controller. These results include a demonstration that, with little or no additional trials, the robot can adjust its lifting policy in a completely model-free way to become translation-averse or energy-averse.

- **Chapter 6: Postural Control and Recovery with the uBot-5.** This chapter discusses the long-term objective of developing a complete postural stability control system for the uBot-5. The controllers developed in this thesis, combined with postural stability controllers developed in our prior work, have contributed to this goal and have greatly improved the deployability of the robot

in unstructured human environments. Results are described from experiments applying risk-sensitive optimization to produce a safe fall bracing behavior and the role of risk-sensitivity in choosing between recovery and bracing behaviors based on inferred impact magnitude is examined.

- **Chapter 7: Conclusions and Future Work.** This chapter summarizes the work presented in this thesis and outlines promising directions for future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter provides a brief overview of optimal control and the variety of algorithms used to find optimal and locally optimal control policies. Particular emphasis is placed on related work on model-free policy search methods to give context to the contributions of this thesis. The possibility of solving optimal control problems using pure stochastic optimization techniques is also discussed, including a more detailed introduction to Bayesian optimization for policy search. Finally, the chapter concludes with a brief overview of risk-sensitive optimal control and a summary of related work in that field.

2.1 Optimal Control

Optimal control theory is a general mathematical framework for deriving control policies that minimize a cost function, possibly subject to several constraints [110, 14]. It has been described as the "computational framework of choice for studying the neural control of movement" [127] and has seen widespread application throughout the robotics community. Furthermore, many algorithms exist in the literature for efficiently finding policies for a wide variety of problems with different stochasticity, nonlinearity, continuity, and dimensionality properties. For these reasons, optimal control is a very attractive framework in which to study problems of dynamic control in robot systems.

Before stating the optimal control problem, a few concepts must be introduced. The first is the notion of a *state space*, $\mathcal{X}$. The system to be controlled is said to be in

the *state* $\mathbf{x}(t) \in \mathcal{X}$ at time t. Typically, $\mathbf{x}(t)$ is defined to be a real vector containing the positions and velocities of all degrees of freedom (DOF) in the system, hence $\mathcal{X} \subseteq \mathbb{R}^{2n}$, where n is the number of DOF. For example, a typical robot arm might have $n = 7$ rotational joints. In practice, it is possible to include other potentially useful measurements in the state vector corresponding to, e.g., motor voltages, locations of visual features, etc.

The *actions* taken by the system are represented by a control vector, $\mathbf{u}(t) \in \mathcal{U} \subseteq \mathbb{R}^k$. Typically, $\mathbf{u}(t)$ is a vector of torque references for a subset of the DOF. Taking an action, $\mathbf{u}(t)$, in state $\mathbf{x}(t)$ produces a change in the state of the system that is captured by a *dynamic equation* or *model*,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.1}$$

where the function $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ is, in general, nonlinear. Finally, to evaluate the system performance, we define a cost function of the form

$$J(\mathbf{x}(0)) = h(\mathbf{x}(T)) + \int_0^T \ell(\mathbf{x}(t), \mathbf{u}(t), t)dt, \tag{2.2}$$

where the term $h(\mathbf{x}(T))$ is the *final cost* for being in state $\mathbf{x}(T)$ at time T, $\ell(\mathbf{x}(t), \mathbf{u}(t), t)$ is the *instantaneous cost* of taking action $\mathbf{u}(t)$ in state $\mathbf{x}(t)$ at time t, and $\mathbf{x}(0)$ is the starting state or *initial conditions*. Cost functions of the form (2.2) are known as a finite-horizon cost functions because of the fixed evaluation time, T. Infinite-horizon cost functions are also possible and are commonly used to describe regulation tasks where, e.g., the system attempts to maintain a particular state indefinitely.

The system generates actions according to a controller, or *policy*, that is a function of state and time, $\mathbf{u}(t) = \pi(\mathbf{x}(t), t)$. Thus, the optimal control problem is to find an optimal policy,

$$\pi^\star \;=\; \arg\min_{\pi} \left[h(\mathbf{x}(T)) + \int_0^T \ell(\mathbf{x}(t), \pi(\mathbf{x}(t), t), t) dt \right], \tag{2.3}$$

subject to

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{x}(0) &= \mathbf{x}_0, \end{aligned}$$

where the last equation defines the fixed starting state. In other words, an optimal control algorithm must find the policy that minimizes cost subject to the system dynamics and initial conditions. In practice, many robot control tasks have the property that the cost incurred by executing a particular policy is not fixed. This commonly arises due to stochasticity in the dynamics,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \tag{2.4}$$

where $\mathbf{w}(t)$ is an uncontrolled disturbance input to the system that is drawn from some noise process. In this case, we can consider the cost to be a random variable drawn from a probability distribution that depends on the policy and initial conditions, $\hat{J}(\pi) \sim P(J|\pi, \mathbf{x}_0)$. To define the optimization problem, one must then specify a minimization objective that is a functional of the cost distribution. A straightforward and widely used criterion is the average or *expected cost*, $\mathbb{E}[\hat{J}(\pi)]$. However, as will be discussed in Section 2.4, more general criteria are also possible.

Analytical approaches to solving optimal control problems are primarily based on a result called the *Hamilton-Jacobi-Bellman* (HJB) equation, which gives a sufficient but not a necessary condition for optimality. This result exploits the recursive structure of the optimal cost-to-go function, $J^\star(\mathbf{x}(t), t)$, that was famously described by Bellman [12] in his *principle of optimality*,

$$
\begin{aligned}
J^\star(\mathbf{x}(T), T) &= h(\mathbf{x}(T)), \\
J^\star(\mathbf{x}(t), t) &= \lim_{dt \to 0} \min_{\mathbf{u}} \left[\ell(\mathbf{x}(t), \mathbf{u}, t) dt + J^\star(\mathbf{x}(t + dt), t + dt) \right]. \qquad (2.5)
\end{aligned}
$$

Intuitively, these equations capture the obvious fact that the cost of an optimal policy starting in state $\mathbf{x}(t)$ at time t is equal to the the instantaneous cost of the best possible action plus the cost of following an optimal policy thereafter. Equation (2.5) can be approximated by a first-order Taylor expansion to yield the HJB equation [120],

$$
0 = \min_{\mathbf{u}} \left[\ell(\mathbf{x}(t), \mathbf{u}, t) + \frac{\partial J^\star}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}) + \frac{\partial J^\star}{\partial t} \right] \quad \forall \mathbf{x} \in \mathcal{X}, t \in [0, T]. \qquad (2.6)
$$

The above expression is for the deterministic case, however the HJB equation can also be derived for stochastic systems with expected cost criteria [127]. The reason that this equation is not also a necessary condition for optimality is the requirement that $\frac{\partial J^\star}{\partial \mathbf{x}}$ exist for all states, which is not true for even some very simple problems.

One way to avoid this difficulty is to instead attempt to solve (2.6) *locally* along a single trajectory [14]. Leaving the details aside, the result, known as *Pontryagin's minimum principle* [94], provides a necessary but not a sufficient condition for optimality in deterministic systems. The important practical implication of this result is that the gradient $\frac{\partial J^\star}{\partial \mathbf{x}}$ need only be calculated along a single trajectory, rather than over the entire state space, making it applicable to problems with discontinuous optimal cost-to-go functions. However, as a penalty for this convenience, it only guarantees local optimality, whereas solutions to the global HJB equation are guaranteed to be optimal (if they exist).

Unfortunately, direct derivations of optimal policies using these analytical insights are only possible in very simple problems, e.g., those involving systems with linear dynamics. However, these results have laid the foundation for a wide variety of numerical and sample-based algorithms that have much broader ranges of applicability. These algorithms can be similarly distinguished based on whether they attempt to

find global or local solutions to control problems. For example, the discrete-time formulation of (2.5), called the *Bellman equation*, serves as the basis for dynamic programming (DP) algorithms [12, 14]. DP algorithms work by iteratively improving an estimate of the cost-to-go, or *value function*, by repeatedly updating the value of each state using the immediate cost and the current estimate of the remaining cost-to-go. If an optimal value function is found, the optimal policy can be derived via the principle of optimality with a one-step lookahead search over actions. This search becomes costly as the number of actions grows, and in the limiting case of continuous actions, one must settle for approximate solutions found by performing line search or resort to specialized techniques for representing the cost-to-go function [8]. For finite state and action spaces, DP algorithms are guaranteed to converge to the optimal value function in a finite number of iterations. However, when the number of states and actions is large, the time required for DP to converge can be prohibitively long, a well-known problem referred to as the *curse of dimensionality* [12].

Work in reinforcement learning (RL) [115, 15] has focused on developing a variety of sample-based algorithms for solving discrete-time stochastic optimal control problems called *Markov decision processes* (MDPs). Central to this field are several efficient algorithms based on temporal-difference (TD) methods [114, 132, 104, 19]. TD methods can be viewed as a middle ground between DP and Monte Carlo methods that update predictions of the cost-to-go using samples from trajectories obtained from policy executions. Unlike DP methods, many of these algorithms do not require knowledge of the system dynamics (i.e., the are *model-free*). However, as is the case with DP methods, these algorithms do not scale well to high-dimensional state and action spaces, so successful applications to robot control tasks can require considerable ingenuity. Fortunately, recent advances in basis function methods for approximating value functions in continuous spaces have begun to narrow this gap [70, 39, 49, 55].

Rather than attempting to compute optimal cost-to-go functions from which optimal policies can be derived, local *policy search* algorithms consider parameterized policies, $\mathbf{u}(t) = \pi_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$, and attempt to minimize cost by directly searching in the space of policy parameters. Here, the parameter vector, $\boldsymbol{\theta}$, might contain the gains of a linear feedback policy, $\pi_{\boldsymbol{\theta}}(\mathbf{x}(t), t) = \text{diag}(\boldsymbol{\theta}) \cdot \mathbf{x}(t)$, or waypoint positions and times used to generate an open-loop trajectory. In the optimal control literature, several model-based algorithms have been developed that employ nonlinear programming to perform trajectory optimization [17]. In particular, for deterministic systems with fixed initial conditions, general second-order nonlinear optimization methods can be applied since the gradients of the cost with respect to the policy parameters can be efficiently computed via techniques such as backpropagation through time [133].

A variety of efficient model-free policy search algorithms have been developed by the RL community. Many of these algorithms attempt to estimate and descend the gradient of the expected cost by exploiting the underlying Markov structure of the discrete-time dynamics. This class of algorithms is particularly relevant for robot applications due to their ability to cope with the properties commonly present in these types of control problems, such as stochasticity and high-dimensional continuous state and action spaces. The model-free attribute is also attractive because the form of the dynamic equation for real robot systems is often only approximately known, so relying on knowledge of the dynamics to derive solutions can lead to poor performance. In fact, by virtue of ignoring the model, the algorithms are *insensitive* to the complexity of the dynamics [100], allowing them to potentially produce behaviors that exploit subtle dynamic properties of the physical system that would be very difficult to capture in a model.

The contributions of this thesis lie within the general class of model-free policy search algorithms. Thus, to provide sufficient context for the work that follows, an overview of these methods is given in the next section.

2.2 Model-Free Policy Search

As was previously described, the approach taken by policy search algorithms is direct. First, a parametric representation of the policy is defined, $\mathbf{u}(t) = \pi_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$, then the policy parameters, $\boldsymbol{\theta}$, are incrementally adjusted to minimize expected cost. In the RL literature, policy search algorithms often attempt to estimate the gradient of the expected cost, $\frac{\partial \mathbb{E}[\hat{J}(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}}$, using sample trajectories and subsequently make small changes to the policy parameters,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \frac{\mathbb{E}[\hat{J}(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}}, \tag{2.7}$$

where η_k is a step size parameter that is typically set to be a constant or decreasing function of the update iteration, k.

The simplest type of policy gradient methods are *finite difference* methods, which attempt to estimate the gradient by 1) generating perturbations to the policy parameters, 2) executing the resulting policies to generate unbiased samples of the expected return, and 3) using these data to produce a gradient estimate by, e.g., performing a least squares fit. These methods have the advantage of typically being very easy to implement because the update rules are simple and the algorithm parameters can, in some cases, be easily tuned. Not surprisingly, these approaches have been successfully applied to several robot control tasks [103, 47, 121, 79, 99]. However, in practice these approaches can have high *update variance*, which is to say that for systems with significant stochasticity and many policy parameters, the number of samples required to obtain a reliable gradient estimate can be large. Roberts and Tedrake [100] provide an insightful analysis of this general class of algorithms that shows how performance is related to policy parameter dimensionality, noise magnitude, and the perturbation distribution.

Another well-studied class of algorithms are *likelihood ratio methods*, such as REINFORCE [136] and the related GPOMDP algorithm [11], that exploit a mathemat-

ical trick to compute the gradient of the expected cost-to-go using only derivatives of the policy with respect to its parameters. Rather than perturbing policy parameters directly, these methods rely on a probabilistic policy representation where actions are drawn from a distribution conditioned on the policy parameters,

$$\mathbf{u}(t) \sim \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}(t), \boldsymbol{\theta}, t). \tag{2.8}$$

These algorithms have faster converge rates than finite difference methods, however for deterministic policies, a system model is required [90].

Actor-critic algorithms [9, 51] are designed to combine the sample-efficiency of TD methods with the advantages of policy gradient methods (i.e., local convergence guarantees and the ability to cope with continuous action spaces). By learning an approximate cost-to-go function and using it to make incremental changes to the policy parameters, lower update variance can be achieved. In addition, local convergence guarantees exist as long as the policy parameter updates are gradient based and meet the conditions described by Bertsekas and Tsitsiklis [16]. Sutton et al. [116] proved that by representing the expected cost-to-go with a *compatible function approximator*, the true policy gradient could be calculated and, under certain assumptions, convergence to a locally optimal policy is guaranteed.

More recently, building on the work of Amari [3] and Kakade [44], Peters and Schaal developed the natural actor-critic (NAC) algorithm [91]. The major insight that inspired this work was that the policy parameter space has Riemannian structure, i.e., it forms a manifold. Thus, the Euclidean distance metric implied by the standard gradient update (2.7) is not generally correct, and performance therefore depends on the policy parameterization. To remedy this, it is suggested that the parameters be updated in the direction of the *natural gradient*,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \mathbf{G}_\theta^{-1} \frac{\partial \mathbb{E}[\hat{J}(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}}, \tag{2.9}$$

where $\mathbf{G}_\theta$ is the Riemannian metric tensor that captures the intrinsic curvature of the parameter space manifold. For stochastic policies, the policy parameters specify a probability distribution and it can be shown that $\mathbf{G}_\theta = \mathbf{F}_\theta$ is the Fisher information matrix [44, 7, 91]. This led to the critical insight that the natural gradient update (2.9) can be simplified further in the RL setting by observing that $\frac{\partial \mathbb{E}[\hat{J}(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \mathbf{F}_\theta \mathbf{w}$ where $\mathbf{w}$ are the learned value function parameters using a compatible function approximator [116]. Thus, (2.9) becomes,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \mathbf{w}. \tag{2.10}$$

This surprisingly simple update rule forms the basis for NAC algorithms which are widely regarded as the state of the art in policy gradient methods.

Another approach to policy gradient is to use sample trajectories to learn a dynamic model using techniques from regression, and then use the learned model to analytically compute the policy gradient. Naive implementations of this approach are unlikely to succeed because of the bias of the model estimator. However, a better approach is taken by the PILCO [28] algorithm where a probabilistic dynamics model is constructed using Gaussian process regression (Section 2.3.1), which explicitly takes model uncertainty into account. Although this approach is computationally intensive, remarkably sample-efficient learning has been reported in simple nonlinear control tasks.

Cost-weighted averaging approaches, such as cross entropy [71], PoWER [46], PI^2 [123], and the recent PI^2-CMA [113], have become popular for solving policy search problems in robotics with fixed initial states. Rather than performing gradient estimation, these methods use Monte Carlo cost samples from randomly perturbed policies to perform a weighted average to compute new parameters. Theodorou et al. [123] showed how such an algorithm can be derived from first principles of stochas-

tic optimal control. Their experiments with PI^2 demonstrated an order-of-magnitude performance increase over episodic natural actor-critic.

Finally, it is also possible to use pure stochastic optimization approaches, such as response surface methods [41], to perform policy search. In this case, Monte Carlo costs are used to fit a model of the cost as a function of the policy parameters. This model is then used to perform offline optimization to select the next policy parameter setting. These approaches tend to be very sample-efficient, but their performance degrades as the dimensionality of the policy parameterization grows. Another distinguishing characteristic is that they perform global policy search, however convergence to a global optimum can only be guaranteed in certain cases [22]. A detailed description of one such approach, called Bayesian optimization [21], is given in the next section.

Policy search methods are considered to be the most appropriate RL algorithms for many robotics applications because they provide a natural way for a designer to incorporate prior knowledge in the form of a parameterized policy while maintaining theoretically attractive properties in continuous, stochastic state and action spaces [92]. Indeed, numerous applications of policy search methods to robot control tasks exist in the literature [32, 13, 102, 47, 121, 90, 46, 99, 123, 50]. However, successful application of these algorithms still requires several important experimenter decisions. In particular, it is often desirable to find a task-appropriate policy representation that is both expressive and low-dimensional. Another challenge is finding suitable values for the algorithm parameters and initial policy parameters. For the former, data can be collected from the robot to help perform parameter fits. Alternatively, a task simulator could be constructed in some cases to perform a more complete parameter search. In general, algorithms with fewer parameters are preferred. Finding good initial policy parameters is particularly important for local methods such as policy gradient. It is common to harness expert operator knowledge or learn from

human demonstration [5]. However, good initial policies are somewhat less important for global policy search methods such Bayesian optimization, which is described in the next section.

2.3 Bayesian Optimization

Bayesian optimization algorithms are a family of global optimization techniques that are well suited to problems where noisy samples of an objective function are expensive to obtain [67, 29, 21, 73, 137, 122]. In describing these algorithms, I use the language of policy search where the inputs are policy parameters and outputs are costs. However, these algorithms are applicable to general stochastic nonlinear optimization problems not related to control.

In contrast to the policy gradient methods highlighted in the previous section, Bayesian optimization algorithms perform policy search by modeling the distribution of cost in policy parameter space and applying a policy *selection criterion* to this distribution to globally select the next policy parameters. Selection criteria are typically designed to balance exploration and exploitation with the intention of minimizing the total number of policy evaluations. These properties make Bayesian optimization attractive for robotics since cost functions often have multiple local minima and policy evaluations are typically expensive. Other attractive features of Bayesian optimization algorithms include the ability to incorporate approximate prior knowledge about the distribution of cost (such as could be obtained from simulation) and enforce hard constraints on the policy parameters.

2.3.1 Gaussian Processes

Most Bayesian optimization implementations represent the prior over cost functions as a *Gaussian process* (GP). A GP is defined as a (possibly infinite) set of random variables, any finite subset of which is jointly Gaussian distributed [97]. It

is useful to think about Gaussian processes as a prior distribution over continuous functions of the input variables. The GP prior, $J(\boldsymbol{\theta}) \sim \mathcal{GP}(m(\boldsymbol{\theta}), k_f(\boldsymbol{\theta}, \boldsymbol{\theta}'))$, is fully specified by its mean function and covariance (or *kernel*) function,

$$\begin{aligned} m(\boldsymbol{\theta}) &= \mathbb{E}[J(\boldsymbol{\theta})], \\ k_f(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \mathbb{E}[(J(\boldsymbol{\theta}) - m(\boldsymbol{\theta}'))(J(\boldsymbol{\theta}) - m(\boldsymbol{\theta}'))]. \end{aligned}$$

Typically, we set $m(\boldsymbol{\theta}) = 0$ and let $k_f(\boldsymbol{\theta}, \boldsymbol{\theta}')$ take on one of several standard forms. A common choice is the anisotropic squared exponential kernel,

$$k_f(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2 \exp(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}')^\top M(\boldsymbol{\theta} - \boldsymbol{\theta}')), \tag{2.11}$$

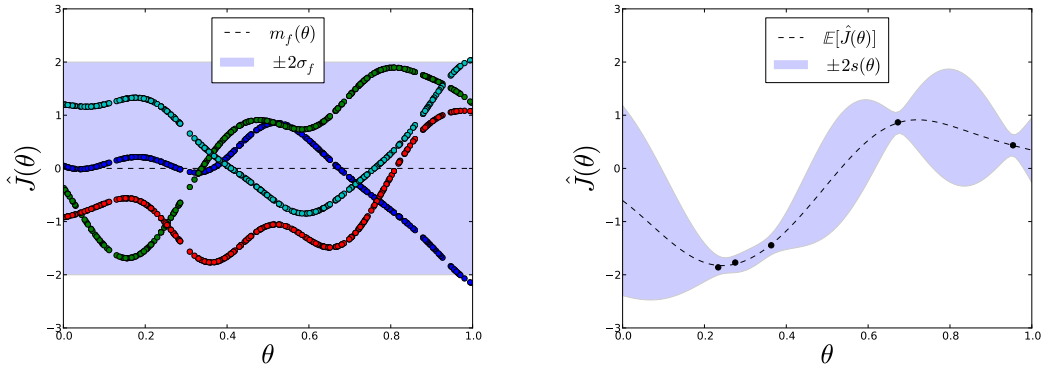
where σ_f^2 is the signal variance and $M = \text{diag}(\boldsymbol{\ell}_f^{-2})$ is a diagonal matrix of length-scale hyperparameters. Intuitively, the signal variance hyperparameter captures the overall magnitude of the cost function variation and the length-scales capture the sensitivity of the cost with respect to changes in each policy parameter. The squared exponential kernel is *stationary* since it is a function of $\boldsymbol{\theta} - \boldsymbol{\theta}'$, i.e., it is invariant to translations in parameter space. In some applications, the target function will be non-stationary: flat in some regions, with large changes in others. There are kernel functions appropriate for this case [97], but the work described in this thesis uses the squared exponential kernel exclusively.

Samples of the latent cost function are typically assumed to have additive independent and identically-distributed (i.i.d.) noise,

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{2.12}$$

Given the GP prior and data,

$$\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N]^\top \in \mathbb{R}^{N \times \dim(\boldsymbol{\theta})},$$



(a) Prior (b) Posterior

Figure 2.1. (a) Four functions drawn randomly from the GP prior. (b) The corresponding posterior distribution computed using (2.13) and (2.14) after $N = 5$ samples.

$$\mathbf{y} \quad = \quad [\hat{J}(\boldsymbol{\theta}_1), \hat{J}(\boldsymbol{\theta}_2), \ldots, \hat{J}(\boldsymbol{\theta}_N)]^\top \in \mathbb{R}^N,$$

the posterior (predictive), cost distribution can be computed for a policy parameterized by $\boldsymbol{\theta}_*$ as, $\hat{J}_* \equiv \hat{J}(\boldsymbol{\theta}_*) \sim \mathcal{N}(\mathbb{E}[\hat{J}_*], s_*^2)$,

$$\mathbb{E}[\hat{J}_*] \quad = \quad \mathbf{k}_{f*}^\top (\mathbf{K}_f + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \tag{2.13}$$

$$s_*^2 \quad = \quad k_f(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - \mathbf{k}_{f*}^\top (\mathbf{K}_f + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_{f*}, \tag{2.14}$$

where $\mathbf{k}_{f*} = [k_f(\boldsymbol{\theta}_1, \boldsymbol{\theta}_*), k_f(\boldsymbol{\theta}_2, \boldsymbol{\theta}_*), \ldots, k_f(\boldsymbol{\theta}_N, \boldsymbol{\theta}_*)]^\top$ and $\mathbf{K}_f$ is the positive-definite kernel matrix, $[\mathbf{K}_f]_{ij} = k_f(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$.

Figure 2.1 shows a simple 1-dimensional example of the result of randomly drawing from a GP prior (see [97], Appendix A.2) with $m(\boldsymbol{\theta}) = 0$ and a squared exponential kernel with $\sigma_f = 1.0$ and $\ell = 0.15$. The corresponding posterior distribution for $N = 5$ sample points is computed using (2.13) and (2.14) and assuming $\sigma_n = 0.1$. Notice that this posterior distribution captures uncertainty about the cost for parameters that are not near the samples. I will discuss in the next section how this property can be exploited to perform principled exploration to select new parameters to evaluate.

If prior information regarding the shape of the cost distribution is available, e.g., from simulation experiments, the mean function and kernel hyperparameters can be set accordingly [67]. However, in many cases such information is not available and *model selection* must be performed. Typically, when the hyperparameters, $\Psi_f = \{\sigma_f, \boldsymbol{\ell}_f, \sigma_n\}$, are unknown, the log marginal likelihood, $\log p(\mathbf{y}|\boldsymbol{\Theta}, \Psi_f)$, is used to optimize the hyperparameters before computing the posterior [97]. The log marginal likelihood and its derivatives can be computed in closed form,

$$\log p(\mathbf{y}|\boldsymbol{\Theta}, \Psi_f) = -\frac{1}{2}\left(\mathbf{y}^\top \boldsymbol{\alpha} + \log|\mathbf{K}_{f,n}| + N\log 2\pi\right), \tag{2.15}$$

$$\frac{\partial \log p(\mathbf{y}|\boldsymbol{\Theta}, \Psi_f)}{\partial \psi_i} = \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}_{f,n}^{-1})\frac{\partial \mathbf{K}_{f,n}}{\partial \psi_i}\right) \quad \text{for } \psi_i \in \{\sigma_f, \sigma_n, \ell_1, \ldots, \ell_{\dim(\boldsymbol{\theta})}\}, \tag{2.16}$$

where $\mathbf{K}_{f,n} = \mathbf{K}_f + \sigma_n^2\mathbf{I}$, $\boldsymbol{\alpha} = \mathbf{K}_{f,n}^{-1}\mathbf{y}$, and $|\mathbf{K}_{f,n}|$ is the determinant of the matrix $\mathbf{K}_{f,n}$. Thus, we are free to choose from standard nonlinear optimization methods, such as Newton's method or conjugate gradient, to maximize the marginal log likelihood to perform model selection.

2.3.2 Expected Improvement

To select the $(N+1)^{\text{th}}$ policy parameters, an offline optimization of a selection criterion is performed with respect to the posterior cost distribution. A commonly used criterion is *expected improvement* (EI) [82, 21]. Expected improvement is defined as the expected reduction in cost, or *improvement*, over the the best policy previously evaluated. The improvement of a policy parameter setting, $\boldsymbol{\theta}_*$, is defined as

$$I_* = \begin{cases} \mu_{\text{best}} - \hat{J}_* & \text{if } \hat{J}_* < \mu_{\text{best}}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.17}$$

where $\mu_{\text{best}} = \min_{i=1,\ldots,N} \mathbb{E}[\hat{J}(\boldsymbol{\theta}_i)]$. Since the predictive distribution under the GP model is Gaussian, the expected value of I_* is

$$\begin{aligned} \text{EI}(\boldsymbol{\theta}_*, \mu_{\text{best}}) &= \int_0^\infty I_* p(I_*) dI_*, \\ &= \int_0^\infty I_* \mathcal{N}(I_* | \mu_{\text{best}} - \mathbb{E}[\hat{J}_*], s_*^2) dI_*, \\ &= s_*(u_* \Phi(u_*) + \phi(u_*)), \end{aligned} \tag{2.18}$$

where $u_* = (\mu_{\text{best}} - \mathbb{E}[\hat{J}_*])/s_*$, and $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and PDF of the normal distribution, respectively. If $s_* = 0$, the expected improvement is defined to be 0. Both (2.18) and its gradient, $\partial \text{EI}(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$, are efficiently computable, so we can apply standard nonlinear optimization methods to maximize EI to select the next policy. In practice, a parameter ξ is often used to adjust the balance of exploration and exploitation, $u_* = (\mu_{\text{best}} - \mathbb{E}[\hat{J}_*] + \xi)/s_*$, where $\xi > 0$ leads to an optimistic estimate of improvement and tends to encourage exploration. Setting $\xi > 0$ can be interpreted as increasing the expected cost of $\boldsymbol{\theta}_{\text{best}}$ by ξ. Lizotte [68] showed that cost scale invariance can be achieved by multiplying ξ by the signal standard deviation, σ_f.

The Bayesian optimization with expected improvement algorithm is shown in Algorithm 1.

From a theoretical perspective, Vazquez and Bect [131] proved that using EI selection for Bayesian optimization converges for all cost functions in the reproducing kernel Hilbert space of the GP covariance function and almost surely for all functions drawn from the GP prior. However, these results rest on the assumption that the GP hyperparameters remain fixed throughout the optimization. Recently, Bull [22] proved convergence rates for EI selection with fixed hyperparameters and the case where model selection is performed according to a modified maximum marginal likelihood procedure. The general case of applying Bayesian optimization with maximum

Algorithm 1 Bayesian Optimization with Expected Improvement

Input: *Previous experience*: $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N], \mathbf{y} = [\hat{J}(\boldsymbol{\theta}_1), \ldots, \hat{J}(\boldsymbol{\theta}_N)]$, *Iterations*: n

1. **for** $i := 1 : n$
   (a) *Perform model selection by optimizing hyperparameters*:
   $\Psi_f^+ := \arg\max_{\Psi_f} \log p(\mathbf{y}|\boldsymbol{\Theta}, \Psi_f)$
   (b) *Maximize expected improvement w.r.t. optimized model*:
   $\mu_{\text{best}} := \min_{j=1,\ldots,|\mathbf{y}|} \mathbb{E}[\hat{J}(\boldsymbol{\theta}_j)]$
   $\boldsymbol{\theta}' := \arg\min_{\boldsymbol{\theta}} \text{EI}(\boldsymbol{\theta}, \mu_{\text{best}})$
   (c) *Execute* $\boldsymbol{\theta}'$, *observe cost*, $\hat{J}(\boldsymbol{\theta}')$
   (d) *Append* $\boldsymbol{\Theta} := [\boldsymbol{\Theta}; \boldsymbol{\theta}']$, $\mathbf{y} := [\mathbf{y}; \hat{J}(\boldsymbol{\theta}')]$
2. **Return** $\boldsymbol{\Theta}, \mathbf{y}$

marginal likelihood model selection and EI policy selection is not guaranteed to converge to the global optimum.

Although EI is a commonly used selection criterion, a variety of other criteria have been studied. For example, early work by Kushner considered the probability of improvement [64] as a criterion for selecting the next input. Confidence bound criteria (introduced in Chapter 4) have been extensively studied in the context of global optimization [23, 109] and economic decision making [66]. Recently, work from Osborne et al. [88, 30] has considered multi-step lookahead criteria that are less myopic than methods that only consider the next best input. For an excellent tutorial on Bayesian optimization, see Brochu et al. [21].

One might reasonably expect that using Bayesian optimization for policy search would be inefficient since it ignores the Markov structure of the problem, relying instead on Monte Carlo rollouts to perform the optimization. However, the idea of constructing models of the cost distribution directly in policy parameter space is a powerful one, especially when the number of policy parameters is small and cost smoothness properties can be exploited to quickly identify regions of policy space that

have low expected cost. Indeed, several applications of Bayesian optimization to robot control tasks exist in the literature. Lizotte et al. [67] applied Bayesian optimization to discover an Aibo gait that surpassed the state-of-the-art in a comparatively small number of trials. Tesch et al. [122] used Bayesian optimization to optimize snake robot gaits in several environmental contexts. Martinez-Cantin et al. [73] describe an application to online sensing and path planning for mobile robots in uncertain environments. In Chapter 3, I describe experiments using Bayesian optimization with a humanoid robot to search for rapid open-loop arm responses that improve stabilization after impact perturbations.

2.4 Risk-Sensitive Optimal Control

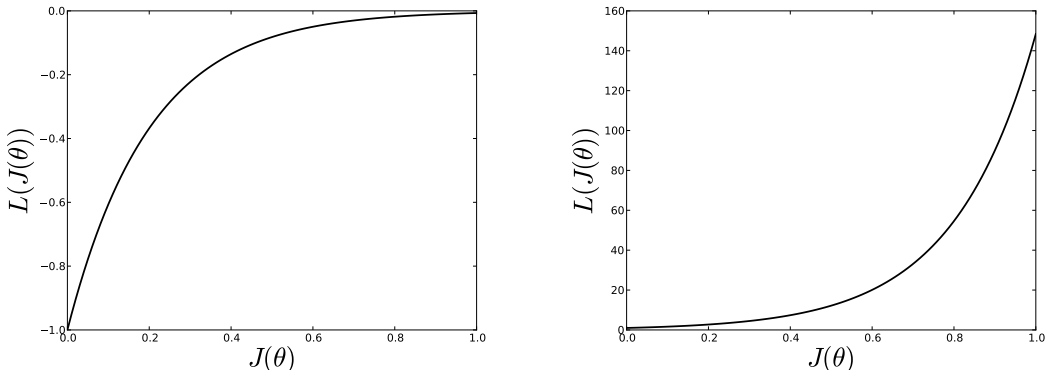
Most stochastic optimal control algorithms, including all algorithms described thus far, are concerned with minimizing the expected cost, $\mathbb{E}[\hat{J}(\boldsymbol{\theta})]$. However, it can be advantageous to consider more general loss functions of the noisy cost signal, $L(\hat{J}(\boldsymbol{\theta}))$, as the minimization objective. For example, consider the monotonically increasing loss function,

$$L(\hat{J}_{\boldsymbol{\theta}}) \quad = \quad -\text{sgn}(\kappa) e^{-\frac{1}{2}\kappa \hat{J}_{\boldsymbol{\theta}}}, \tag{2.19}$$

where $\hat{J}_{\boldsymbol{\theta}} \equiv \hat{J}(\boldsymbol{\theta})$. As is shown in Figure 2.2, depending on the value of the parameter $\kappa \neq 0$, $L(\hat{J}_{\boldsymbol{\theta}})$ is either concave ($\kappa > 0$) or convex ($\kappa < 0$). In the deterministic case, this amounts to a simple reshaping of the relative weight assigned to increasing cost. However, for stochastic cost signals, minimizing $\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]$ has a more interesting effect.

To see this, recall that Jensen's inequality states that for any convex function, $Y = \psi(X)$, of a random variable, X, the following inequality holds,

$$\mathbb{E}[Y] \geq \psi(\mathbb{E}[X]). \tag{2.20}$$



(a) $\kappa = 10$ (b) $\kappa = -10$

Figure 2.2. Comparison of an exponential loss functions for two settings of the risk-sensitivity parameter κ.

Thus, for $\kappa < 0$, we have $\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})] \geq L(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}])$. Intuitively, what this says is that for a fixed value of $\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]$, policies with wider cost distribution will be avoided since they will map to larger values of $\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]$. A system optimizing such an objective can be viewed as being *risk-averse* since it explicitly avoids uncertainty. Likewise, for $\kappa > 0$, we have $\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})] \leq L(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}])$ and minimizing $\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]$ would lead to *risk-seeking* behavior, where higher variance policies are preferred for fixed values of $\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]$. More generally, we say that systems that select policies according to such criteria are *risk-sensitive*.

It is typical to define the minimization criterion so that it has the same units as the original cost function. Thus, the risk-sensitive objective function is

$$\begin{aligned} \gamma(\boldsymbol{\theta}, \kappa) &= L^{-1}\left(\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]\right), & (2.21) \\ &= -2\kappa^{-1}\log\left(-\text{sgn}(\kappa)\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]\right), & (2.22) \\ &= -2\kappa^{-1}\log\left(\mathbb{E}\left[e^{-\frac{1}{2}\kappa\hat{J}_{\boldsymbol{\theta}}}\right]\right), & (2.23) \end{aligned}$$

where $\gamma(\boldsymbol{\theta}, 0) = \mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]$.

Another way to see how this criterion leads to risk-sensitivity is by taking the second order Taylor expansion of $L(\hat{J}_{\boldsymbol{\theta}})$ about $\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]$,

$$L(\hat{J}_{\boldsymbol{\theta}}) \approx L(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]) + (\hat{J}_{\boldsymbol{\theta}} - \mathbb{E}[\hat{J}_{\boldsymbol{\theta}}])L'(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]) + \frac{1}{2}(\hat{J}_{\boldsymbol{\theta}}) - \mathbb{E}[\hat{J}_{\boldsymbol{\theta}}])^2 L''(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]), \quad (2.24)$$

which implies

$$\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})] \approx L(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]) + \frac{1}{2}\mathbb{V}[\hat{J}_{\boldsymbol{\theta}}]L''(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}]), \quad (2.25)$$

$$L^{-1}(\mathbb{E}[L(\hat{J}_{\boldsymbol{\theta}})]) \approx \mathbb{E}[\hat{J}_{\boldsymbol{\theta}}] + \frac{1}{2}\mathbb{V}[\hat{J}_{\boldsymbol{\theta}}]\frac{L''(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}])}{L'(\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}])}. \quad (2.26)$$

Thus, for the exponential loss function (2.19), we have

$$\gamma(\boldsymbol{\theta}, \kappa) \approx \mathbb{E}[\hat{J}_{\boldsymbol{\theta}}] - \frac{1}{4}\kappa\mathbb{V}[\hat{J}_{\boldsymbol{\theta}}]. \quad (2.27)$$

This approximation exposes the role of the parameter κ in determining the risk sensitivity of the system: $\kappa < 0$ is risk-averse, $\kappa > 0$ is risk-seeking, and $\kappa = 0$ is risk-neutral [134]. Figure 2.3 shows an example application of the risk-sensitive optimal control objective (2.23) to a synthetic cost distribution. Two global minima (in the expected cost sense) are distinguished by their variance using risk-sensitive objectives, preferring or avoiding high variance policies depending on the value of κ.

A variety of algorithms have been designed to find optimal policies with respect to risk-sensitive criteria. Early work in risk-sensitive control was aimed at extending dynamic programming methods to optimize exponential objective functions of the form (2.23). This work included algorithms for solving discrete Markov decision processes (MDPs) [36] and linear-quadratic-Gaussian problems [38, 134]. Borkar derived a variant of the Q-learning algorithm for finite MDPs with exponential utility [18]. In earlier work, Heger [35] derived a worst-case Q-learning algorithm based on a minimax criterion. For continuous problems, Van den Broek et al. [130] generalized path integral methods from stochastic optimal control to the risk-sensitive case.

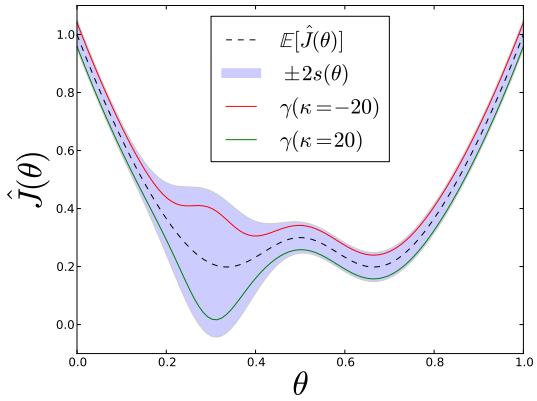


Figure 2.3. Example cost distribution with two global expected cost minima that have different cost variance. By changing the value of the risk sensitivity parameter κ, different objective functions arise that differentiate the two solutions based on their cost variance by preferring either low variance or high variance solutions.

Other work has approached the problem of risk-sensitive control with methods other than exponential objective functions. For example, several authors have developed algorithms in discrete model-free RL setting for learning conditional return distributions [25, 80, 81], which can be combined with policy selection criteria that take return variance into account. The algorithms presented in this thesis are related to this line of work, but they are more directly applicable to systems with continuous state and action spaces. Most recently, Tamar et al. [119] derived an expression for the variance of the cost-to-go in episodic tasks and used it to derive various risk-sensitive policy gradient algorithms. The simulation-based algorithm described by these authors is closely related to the algorithm described in Chapter 5.

Mihatsch and Neuneier [76] developed risk-sensitive variants of TD(0) and Q-learning by allowing the step size in the value function update to be a function of the sign of the temporal difference error. For example, by making the step size for positive errors slightly larger than the step size for negative errors, the value of a particular state and action will tend to be optimistic, yielding a risk-seeking system. Recently, this algorithm was found to be consistent with behavioral and

neurological measurements taken while humans learned a decision task involving risky outcomes [86], suggesting that some form of risk-sensitive TD may be used in the brain.

The connection between these types of methods and biological learning and control processes is an active area of research in the biological sciences. For example, some neuroscience researchers have identified separate neural encodings for expected cost and cost variance that appear to be involved in risk-sensitive decision making [96, 126]. Recent motor control experiments suggest that humans select motor strategies in a risk-sensitive way [139, 84, 83]. For example, Nagengast et al. [83] show that control gains selected by human subjects in a noisy control task are consistent with risk-averse optimal control solutions. There is also an extensive literature on risk-sensitive foraging behaviors in a wide variety of species [43, 10, 87].

2.5 Discussion

Solving dynamic control tasks on robot systems is generally a hard problem. However, the various challenges that arise in such problems can be understood and addressed within the framework of optimal control. A variety of algorithms exist for finding approximate, local, or global optimal policies. However the suitability of any particular algorithm is strongly determined by the properties of the problem at hand. For example, many robot control tasks can be characterized as having significant stochasticity, continuous state and action spaces, and limited or inaccurate model information.

One class of algorithms appropriate for solving such tasks are policy search methods that directly attempt to minimize expected cost by adjusting the parameters of a control policy. Policy gradient methods are a particularly well-studied type of policy search algorithms. Typically, these algorithms exploit some structure of the problem to efficiently compute sample-based estimates of the gradient of the expected cost.

Carefully performing gradient descent with these algorithms produces locally optimal solutions. Another approach is to treat the optimal control problem as a general stochastic optimization problem and apply a method such as Bayesian optimization. Bayesian optimization algorithms work by nonparametrically estimating the cost distribution conditioned on the policy parameters and using this distribution to select the next policy parameters in a principled fashion. These methods assume very little about the structure of the problem and have been successfully applied to perform efficient global policy search. However, their applicability is limited to problems with low-dimensional policy parameterizations.

The vast majority of stochastic optimal control and RL algorithms are designed to minimize expected cost. However, more general optimization criteria, such as those that consider performance variation, may play an important role in generating flexible dynamic control in robot systems. Risk-sensitive optimal control is broadly concerned with optimization criteria that capture not only the expected cost, but also the variance of the cost. A variety of algorithms exist for solving risk-sensitive control problems, but few examples exist of model-free risk-sensitive policy search methods. The algorithms described in Chapters 4 and 5 are contributions to this general class of methods.

In the next chapter, I describe an application of Bayesian optimization to the dynamic control task of using rapid open-loop arm motions to help stabilize a dynamically balancing robot after impact perturbations. These experiments produced learned policies with measurable performance benefits in very few trials, but also lead to observations of significant policy-dependent cost variance that motivate the need for more general risk-sensitive policy search methods.

CHAPTER 3

LEARNING RAPID STABILIZING ARM MOTIONS VIA GLOBAL POLICY SEARCH

3.1 Introduction

The successful deployment of mobile humanoid robots in dynamic environments will require solutions to many challenging hardware, perception, and control problems. One particularly challenging control problem is that of maintaining stability in the face of postural perturbations caused by impacts or unpredicted terrain changes. The best solutions to these problems will exhibit a high degree of resourcefulness, exploiting many actuators and innate dynamics to achieve rapid, robust, and efficient stabilization. Indeed, a typical adult human exhibits a remarkable ability to generate whole-body recovery strategies that frequently involve rapid arm movements that co-occur with the lower body response [75, 72]. Biomechanics researchers have made significant progress toward understanding the functional contributions of these movements under different experimental conditions [78, 101, 128, 93]. However, relatively little work has focused on controlling arm recovery responses in robot systems.

In this chapter, I provide an overview of previous research on upper body recovery motions and present experimental results involving a dynamically balancing mobile manipulator, the uBot-5, that efficiently learns rapid open-loop arm responses to impact perturbations [59]. In these experiments, I apply Bayesian optimization (Section 2.3) to perform global model-free policy search to minimize the expected value of a simple cost function inspired by observations of arm motion effects in the biomechanics literature. The resulting policies exhibit decreased total energy expenditure,

decreased recovery footprint, and increased ability to stabilize after large impacts. An unexpected result of these experiments was that, for larger impacts, some policies stabilized a fraction of the time, leading to very high cost variance, while others had low variance (either failing or stabilizing predictably). This policy-dependent cost variance motivated the extensions to the Bayesian optimization algorithm and subsequent experiments that are described in Chapter 4.

3.2 Background

3.2.1 Arm Recovery Motions in Humans

McIlroy and Maki [75] were perhaps the first to specifically consider arm responses to external disturbances. In this study, subjects stood upon a platform that delivered translational perturbations while shoulder and lower leg muscle responses were measured. They observed that the magnitude of the shoulder response was correlated with the magnitude and direction of the perturbation. Furthermore, the authors concluded that these movements are unlikely to be startle responses because no apparent habituation was present over multiple trials. Together, these observations suggested a possible *functional role* of arm movements in the recovery behavior.

Researchers have since begun to uncover more about the functional contributions of the upper extremities during balance recovery. Marigold et al. [72] observed rapid elevation of the arms during slip recovery in young adults. The authors noted a marked change in responses after repeated exposure to the same perturbation, suggesting that whole-body recovery strategies can be short-term adaptive. Troy et al. [128] observed a similar rapid elevation behavior in slipping experiments performed on both young and old adults. Using a simplified sagittal plane model, the authors concluded that arm responses served to reduce trunk rotational velocity immediately following the slip while repositioning the upper body center of mass away from the rear support boundary.

Similar arm response characteristics have been observed for tripping perturbations [101, 93] and hip disturbances [77, 78]. Misiaszek and Krauss [78] observed that recovery responses of leg musculature *increased in magnitude* when arm motions were voluntarily suppressed. Several studies have demonstrated significant differences between the responses of young and old subjects [101, 128, 2]. Generally speaking, younger adults who were capable of faster movements and reduced reaction times tended to produce fast motions that affected the body angular momentum, while older subjects tended to resort to more protective strategies such as grasping and bracing.

Perhaps the most complete functional analysis to date is from Pijnapples et al. [93]. Using a 3D physical model, the authors analyzed the contribution of arm responses in tripping experiments by calculating what the body angular velocity *would have been* had the arms not been present between the perturbation onset and recovery step. The results of this analysis suggest that, for tripping perturbations during normal walking, arm recovery motions contribute most significantly to controlling rotation in the transverse (yaw) plane which helps position the body to successfully take a recovery step [93]. However, because tripping perturbations induce a rotation in the transverse plane toward the tripped foot that must be counteracted, it is possible that similar analyses for a different perturbation modality would produce different results.

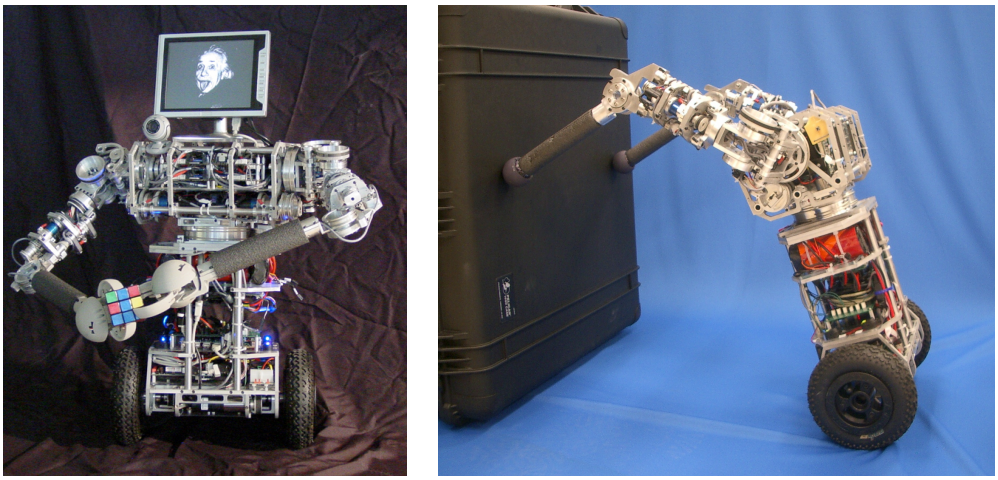
3.2.2 Arm Recovery Motions in Artificial Systems

There is a very rich literature devoted to robust humanoid locomotion and recovery from perturbation. However, relatively little work exists which aims to create postural stability controllers that exploit articulated upper bodies, especially in the context of rapid balance recovery. This is not to imply a lack of empirical success. Indeed, for the case of bipedal postural stability, the coordination of ankle, hip, and stepping recovery

strategies has yielded impressive results on real systems (e.g. [111]). However, given our increasing understanding of human balance recovery, there is reason to suspect that coordination of the arms may offer significant advantages.

Several researchers have studied model systems that have provided valuable insights. Pratt et al. [95] introduced the Linear Inverted Pendulum Plus Flywheel model that abstractly models the angular momentum induced by upper body motions as a flywheel about the body center of mass. Atkeson and Stephens [6] used a multi-link pendulum model to show that different impact recovery strategies can arise from a single quadratic optimization criterion, suggesting that whole-body responses in humans may similarly be the product of a unified control scheme. A recent paper from Nakada et al. [85] described an increase in balance recovery of a simulated biped using a learned arm rotation strategy. Other related work has considered quasistatic contributions of free arm movements in real systems [141, 58].

In the character animation literature, several researchers have produced controllers for generating whole-body recovery responses. Kudoh et al. [56, 57] formulated a quadratic programming problem to produce arm swinging motions that stabilized the system after impacts. Shiratori et al. [107] used human motion capture data during tripping experiments to create controllers that produced human-like responses in characters that were tripped under different initial conditions. Macchietto et al. [69] described a method for directly controlling linear and angular momenta that produced realistic whole-body balance recovery strategies for standing characters. These results are among the most impressive in the literature, but it remains unclear how they will translate to robotic systems with imprecise sensors and models, constrained actuators, and lower bandwidth control.



(a) (b)

Figure 3.1. The uBot-5 (a) using prototype hands to grasp a Rubik's cube and (b) demonstrating a whole-body pushing behavior.

3.3 Experiments

I performed two experiments to quantify the advantages of whole-body recovery strategies using a dynamically balancing mobile manipulator, the uBot-5, and an apparatus designed to impart controlled impact perturbations to the upper torso of the robot.

3.3.1 The uBot-5

The uBot-5 (Figure 3.1) is an 11-DoF mobile manipulator developed at the University of Massachusetts Amherst [26, 63]. The uBot-5 has two 4-DoF arms, a rotating trunk, and two wheels in a differential drive configuration. The robot stands approximately 60 cm from the ground and has a total mass of 19 kg. The robot's torso is roughly similar to an adult human in terms of geometry and scale, but instead of legs, the uBot has two wheels attached at the hip. The robot has three interchangeable heads: a touch LCD screen for human-robot interaction, a stereo camera system mounted on a pan-tilt unit, and a 1-DOF tilt unit with an ASUS Xtion PRO® for 3D point cloud sensing. In Figure 3.1(a), prototype 1-DOF hands are shown, but unactu-

ated spherical endpoint contacts (Figure 3.1(b)) are used in all recovery experiments in this thesis. The dynamic manipulation experiments in Chapter 5 involve a simple 1-DoF claw gripper with a servo driven thumb.

All 11 joints are driven by DC motors with planetary gear heads. Joint position and velocity proportional-integral-derivative (PID) controllers run on an onboard field-programmable gate array (FPGA) at approximately 2000 Hz. An onboard PC104 is used to run control software that streams PID references or raw motor commands over ethernet to the FPGA at approximately 500 Hz. The uBot has no dedicated force sensors, although some work has been done to control endpoint forces using motor current measurements [33].

The robot balances using a linear-quadratic regulator (LQR) with feedback from an onboard inertial measurement unit (IMU) to stabilize around the vertical fixed point. The LQR controller has proved to be very robust throughout five years of frequent usage and it remains fixed in the experiments. The robot's wheeled base permits a fast and energy efficient solution to upright stability that is achieved using well-understood techniques from optimal control. This makes the uBot a unique and attractive experimental platform for studying the problem of upper-body recovery because it allows one to assess the influence of arm motions on the stabilized system without first solving the difficult legged recovery problem.

3.3.2 Impact Pendulum

The robot was placed in a balancing configuration with the dorsal side of its torso aligned with a 3.3 kg mass suspended from the ceiling (Figure 3.2). The mass was pulled away from the robot to a fixed angle and released, producing a controlled impact between the swinging mass and the robot's upper torso. This device is similar that used by Hasson et al. [34] in a human study aimed at developing predictive models for step recovery after impact perturbations. The robot was attached to

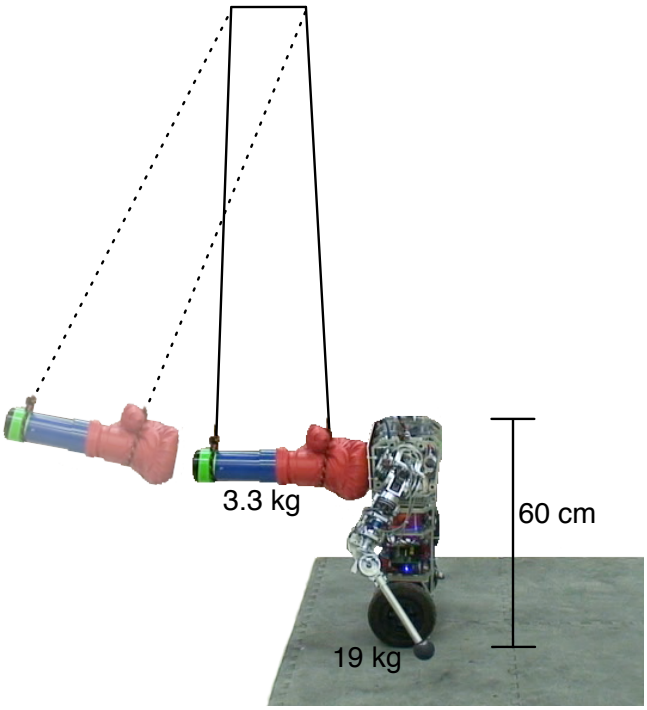


Figure 3.2. The uBot-5 situated in the impact pendulum apparatus.

the ceiling with a loose-fitting safety rig designed to prevent the robot from falling completely, while not affecting the performance of the controlled response. Impacts were detected using the robot's onboard IMU and arm responses were initiated within approximately 50 ms. The arm initial conditions were fixed across trials.

Two learning experiments were performed using different impact magnitudes. I aimed to evaluate the hypotheses that the robot could learn to exploit dynamic interactions between its arms and the LQR to

1. reduce the spatial footprint of the recovery,
2. reduce the total energy expenditure, and
3. increase robustness to large perturbations.

In the first experiment, the robot was situated at the base of the impact pendulum, and the release angle was chosen such that the robot could reliably recover balance

using only the wheel LQR controller. The momentum of the pendulum mass prior to impact was estimated to be 5.6 Ns with a measurement error of ± 0.8 Ns by analyzing video footage of the experiment. The impact duration could not be accurately inferred from the video, but it appeared to be between 1 and 2 video frames, or 1/25 to 2/25 s. In the second experiment, the impact magnitude was increased so that a fixed arm policy would fail to stabilize the system a significant fraction of the time. The perturbation in this case was approximately 6.7 ± 1.0 Ns.

3.3.3 Optimal Control Formulation

This problem is well suited for model-free policy optimization since there are several physical properties, such as joint friction, wheel backlash, and tire slippage, that make the system difficult to model accurately. In addition, although the underlying state and action spaces are high dimensional (22 and 8, respectively), low-dimensional policy spaces that contain high-quality solutions are relatively straightforward to identify.

Shoulder and elbow pitch motion trajectories were generated using the cubic spline method [24]. In this approach, given a sequential list of joint positions, $\{\alpha_0, \ldots, \alpha_k\}$, velocities, $\{\dot{\alpha}_0, \ldots, \dot{\alpha}_k\}$, and relative times, $\{t_1, \ldots, t_k\}$, each trajectory segment is computed by first solving the set of polynomial equations,

$$\alpha_i = a, \tag{3.1}$$

$$\alpha_{i+1} = a + bt_{i+1} + ct_{i+1}^2 + dt_{i+1}^3, \tag{3.2}$$

$$\dot{\alpha_i} = b, \tag{3.3}$$

$$\dot{\alpha_{i+1}} = b + 2ct_{i+1} + 3dt_{i+1}^2, \tag{3.4}$$

then using these coefficients to define the trajectory generating function between waypoints i and $i+1$,

$$\alpha_{i+1}^{i}(t) = a + b(t - t_i) + c(t - t_i)^2 + d(t - t_i)^3, \tag{3.5}$$

where $t_0 = 0$. In the experiments, arm motions were constrained to be symmetric in the sagittal plane, so a single cubic spline parameterization described the motion for both arms. The spline parameters were $\boldsymbol{\theta} = [\alpha_{\text{shoulder}}, \alpha_{\text{elbow}}, t_{\text{wp}}, t_{\text{f}}]$, where α_{shoulder} and α_{elbow} are the shoulder and elbow waypoint positions, respectively. The remaining two time parameters describe the desired time to reach the waypoint positions and the time to return to the starting configuration. The waypoint velocity parameters were set to 0 and the trajectory was followed using fixed PD controllers. Using prior knowledge about what policies are feasible, these parameters were conservatively constrained:

$$1.5 \text{ rad} \geq \quad \alpha_{\text{shoulder}} \quad \geq -1.5 \text{ rad}, \tag{3.6}$$

$$1.0 \text{ rad} \geq \quad \alpha_{\text{elbow}} \quad \geq -1.0 \text{ rad}, \tag{3.7}$$

$$1.0 \geq \quad t_{\text{wp}} \quad \geq d(\alpha_{\text{shoulder}}, \alpha_{\text{elbow}}), \tag{3.8}$$

$$1.5 \geq \quad t_{\text{f}} \quad \geq d(\alpha_{\text{shoulder}}, \alpha_{\text{elbow}}) + t_{\text{wp}}, \tag{3.9}$$

where the function $d(\alpha_{\text{shoulder}}, \alpha_{\text{elbow}})$ returns the minimum time required to move to the waypoint positions given the uBot's maximum joint velocity: $5\pi/4$ rad/s.

A simple cost function was defined to encourage spatially and energetically efficient solutions:

$$J(\boldsymbol{\theta}) = \int_0^T \left(x_{wheel}^2(t) + \dot{x}_{wheel}^2(t) + g(\mathbf{x}(t))I(t)V \right) dt, \tag{3.10}$$

where $x_{wheel}(t)$ and $\dot{x}_{wheel}(t)$ are the wheel position and velocity at time t, respectively, $I(t)$ is the total absolute current being drawn by all motors, and $V = 13.1$ volts is the system voltage. The state vector $\mathbf{x}(t)$ contains the IMU readings, a failure bit,

and positions, velocities, and motor currents for all joints at time t. The function $g(\mathbf{x}(t))$ captures the additional energetic cost associated with a failure to recover. If $\mathbf{x}(t) \in FailureStates$, then $g(\mathbf{x}(t)) = 0.005$. Otherwise, $g(\mathbf{x}(t)) = 0.001$. A state $\mathbf{x}(t) \in FailureStates$ if and only if the state $\mathbf{x}(t)$ is detected as a failure or $\exists t' < t$ such that $\mathbf{x}(t') \in FailureStates$. Failure states were detected reliably as large spikes in the IMU data. In all experiments, $T = 3.5$ s and the sampling frequency was 100 Hz.

I applied Bayesian optimization (Section 2.3) to optimize the policy parameters with respect to the expected cost, $\mathbb{E}[\hat{J}(\boldsymbol{\theta})]$. An anisotropic squared exponential kernel was used in the GP prior and the hyperparameters were optimized after each trial with respect to a maximum a posteriori (MAP) criterion. To achieve cost scale invariance, the maximum likelihood mean was computed analytically after each trial and used in the log likelihood computation [68]. A prior was placed over the logarithm of the length-scale hyperparameters: $\log(\boldsymbol{\ell}) \sim \mathcal{N}(\mathbf{0}, 3^2 I)$. Although this prior is quite broad for this problem[1], it provides a flexible way to constrain the optimization process in the early stages of learning [68].

The gradients of the log likelihood and log prior terms were computed analytically and the optimization of hyperparameters was performed using the NLOPT [40] implementation of the Method of Moving Asymptotes (MMA) [117]. After each trial, the hyperparameters were optimized starting from the MAP estimate from the previous trial, and 30 random restarts were performed to decrease the chance of arriving at a low-quality local optimum. Policy parameters were selected using EI ($\xi = 0.1 \cdot \sigma_f$), where EI maximization was performed using MMA under the inequality constraints (3.6)–(3.9). Forty random restarts were performed during EI maximization and the best among these was used to select the next data point.

[1]The maximum parameter range is 3 units while the prior states that there is about a 95% chance that the length-scales are between 403 and 0.002.

3.4 Results

A total of 35 trials was performed in the high impact case and 30 in the low impact case. After the learning trials, a greedy policy was selected by maximizing the probability of improvement [64] with respect to the posterior distribution,

$$P(\hat{J}(\boldsymbol{\theta}) \leq \mu_{\text{best}}) = \Phi\left(\frac{\mu_{\text{best}} - \mathbb{E}[\hat{J}(\boldsymbol{\theta})]}{s(\boldsymbol{\theta})}\right), \tag{3.11}$$

where $\mu_{\text{best}} = \min_{i=1,\ldots,N} \mathbb{E}[\hat{J}(\boldsymbol{\theta}_i)]$ and $\Phi(\cdot)$ is the CDF of the normal distribution. The greedy policies were

$$\begin{aligned} \boldsymbol{\theta}_{\text{low}} &= [-0.681, 0.681, 0.174, 1.5] \quad \text{and} \\ \boldsymbol{\theta}_{\text{high}} &= [-0.562, -0.562, 0.143, 1.478] \end{aligned}$$

for the low and high impact cases, respectively. The symmetry in the shoulder and elbow displacements appeared to be a consequence of the constraints, (3.8) and (3.9), and the desire to maximize joint movements over a short initial response time. This symmetry was not strictly observed during the learning process. Interestingly, the rotations of the shoulder and elbow joints are opposite in the low impact policy. This produces a contracted backward arm motion as opposed to the extended backward arm motion in the high impact policy. A 25% higher peak shoulder torque (inferred from motor current data) was observed 0.1 seconds post-impact for the high impact policy.

To evaluate the three hypotheses regarding spatial footprint, total energy, and robustness, 10 trials using the learned greedy policy and a control (fixed arm) policy were performed for each impact magnitude. The learned policies exhibited a 17.1% reduction in average cost (1554.59 to 1288.34) in the low impact case and a 61.6% reduction in average cost (4507.36 to 1728.64) in the high impact case. The fixed

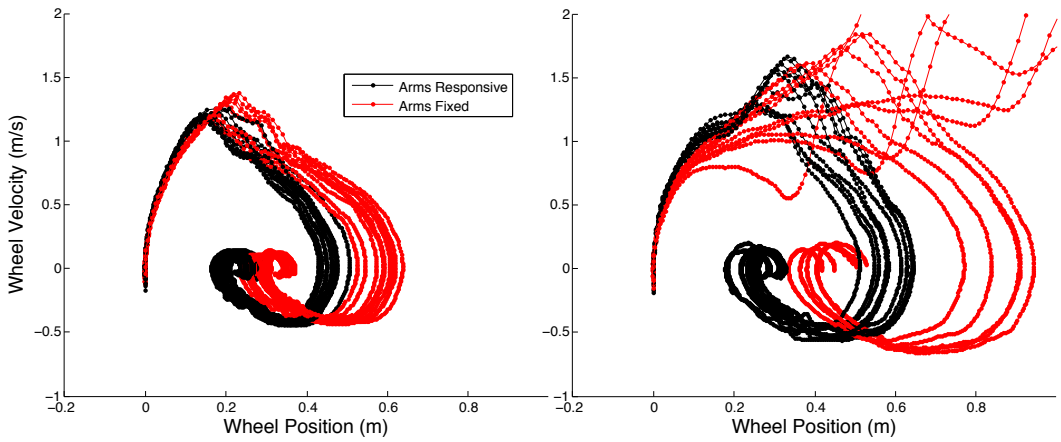


Figure 3.3. Wheel position and velocity trajectories for the learned and fixed arm policies in both the low impact (left) and high impact (right) cases.

arm policy failed to stabilize in 5 out of the 10 high impact trials. Excluding these failure trials, the reduction in cost in the high impact case was still 29.7% (2458.98 to 1728.64).

3.4.1 Efficiency Gains

A statistically significant decrease in the recovery footprint was observed when using the learned arm motions for both impact magnitudes. The wheel trajectories in Figure 3.3 illustrate this difference. Interestingly, there was also a statistically significant reduction in *total energy expenditure* when using the learned arm recovery motions. The total energy was calculated as $E = \int_0^T I(t)V dt$, where $I(t)$ is the total absolute current through all motors at time t, and $V = 13.1$ volts. Table 3.1 summarizes the reduction in average energy expenditure. Since we could not quantify the true energetic requirements of recovering from a failure, we only included the successful fixed arm trials in these statistics. Thus, the energy savings reported for the high impact case is very conservative. These data suggest that the reduction in wheel motor energy consumption more than compensates for the additional energy consumed by the shoulder and elbow motors in the learned policies.

Table 3.1. Comparison of mean energy expenditure averaged over 10 trials. The 5 fixed arm failure trials were excluded from the high impact data because the actual energy required to recover from a failure was not measured. Thus, we expect the true energetic gain of the learned policy to be much larger than reported (marked with an asterisk).

	Fixed Arms	Learned Response	Behrens-Fisher
Low impact	194.03 joules	176.37 joules	$p < 0.0001$
High impact	242.16* joules	215.67 joules	$p = 0.0046$

3.4.2 Stability Gains

During the evaluation of the learned high impact policy, the robot successfully recovered in 10 out of 10 trials. In contrast, the control (fixed arm) policy only resulted in recovery in 5 out of 10 trials. Figure 3.4 compares an example run of the learned response to a failure control trial.

It is interesting that a fixed policy and impact magnitude can produce different stabilization results. Careful analysis of the experiment video showed that the pendulum motion varied very little between trials. However, the state of the robot's slight back-and-forth balancing motion at the time of impact was loosely correlated with the trial outcome. Thus, the system performance under some policies seems to be sensitive to the initial conditions. This result suggests that it may be necessary to capture the policy-dependent cost variance during the optimization. Such an approach would, for example, allow the robot to learn the variance of policies for different impacts and explicitly favor more predictable recovery strategies. Extending Bayesian optimization to capture this type of policy-dependent cost variance is the subject of the next chapter.

3.4.3 Uncontrolled Impacts

The learned policies were successfully used to respond to uncontrolled impacts in the laboratory environment. Using data from the learning trials, a simple impact magnitude classifier was constructed using low-pass filtered IMU data. The robot

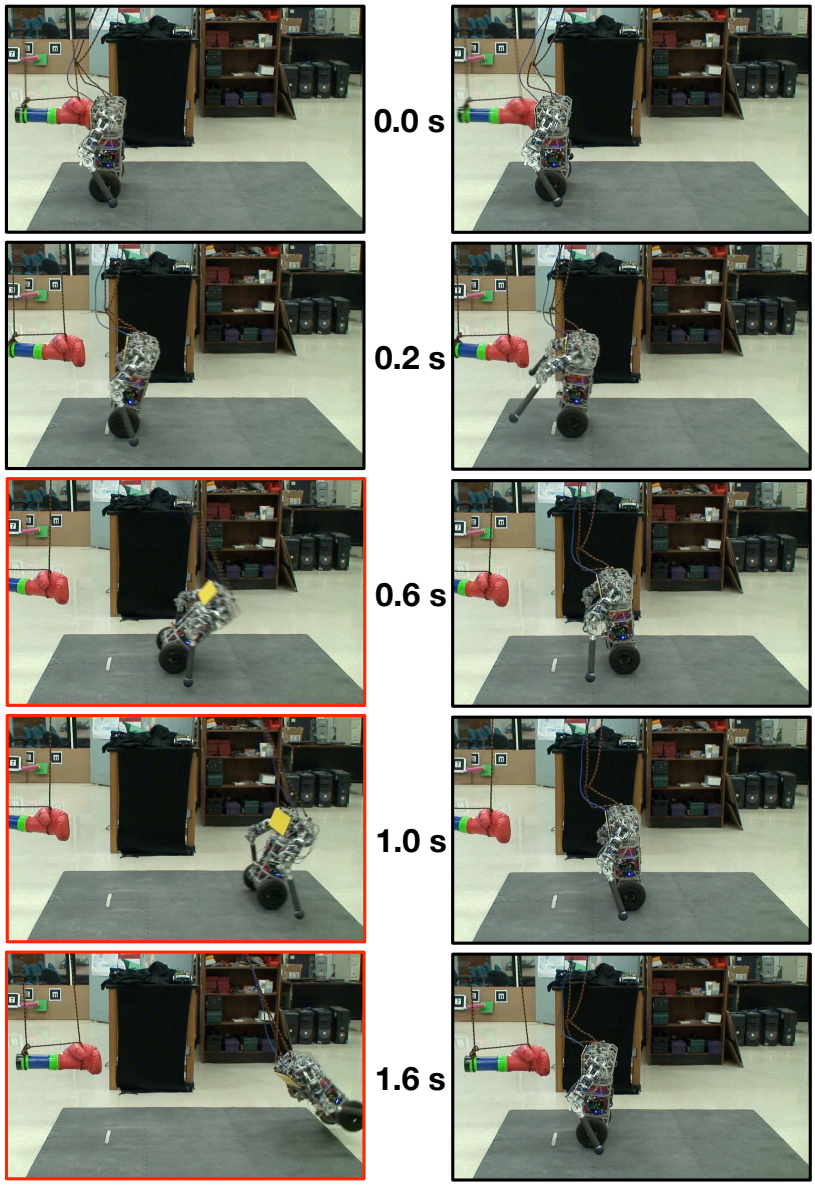


Figure 3.4. Comparison of the recovery behavior without (left) and with (right) learned arm motions after a large impact perturbation. The bottom three panels on the left outlined in red indicate the point of failure when the safety rig was engaged.

successfully responded to various uncontrolled impact perturbations: small bumps caused by a person walking into it (no arm response), pushing the robot (low impact arm response), bouncing a dodgeball off of the robot (low impact arm response, kicking the robot (high impact response), and throwing a large exercise ball at the

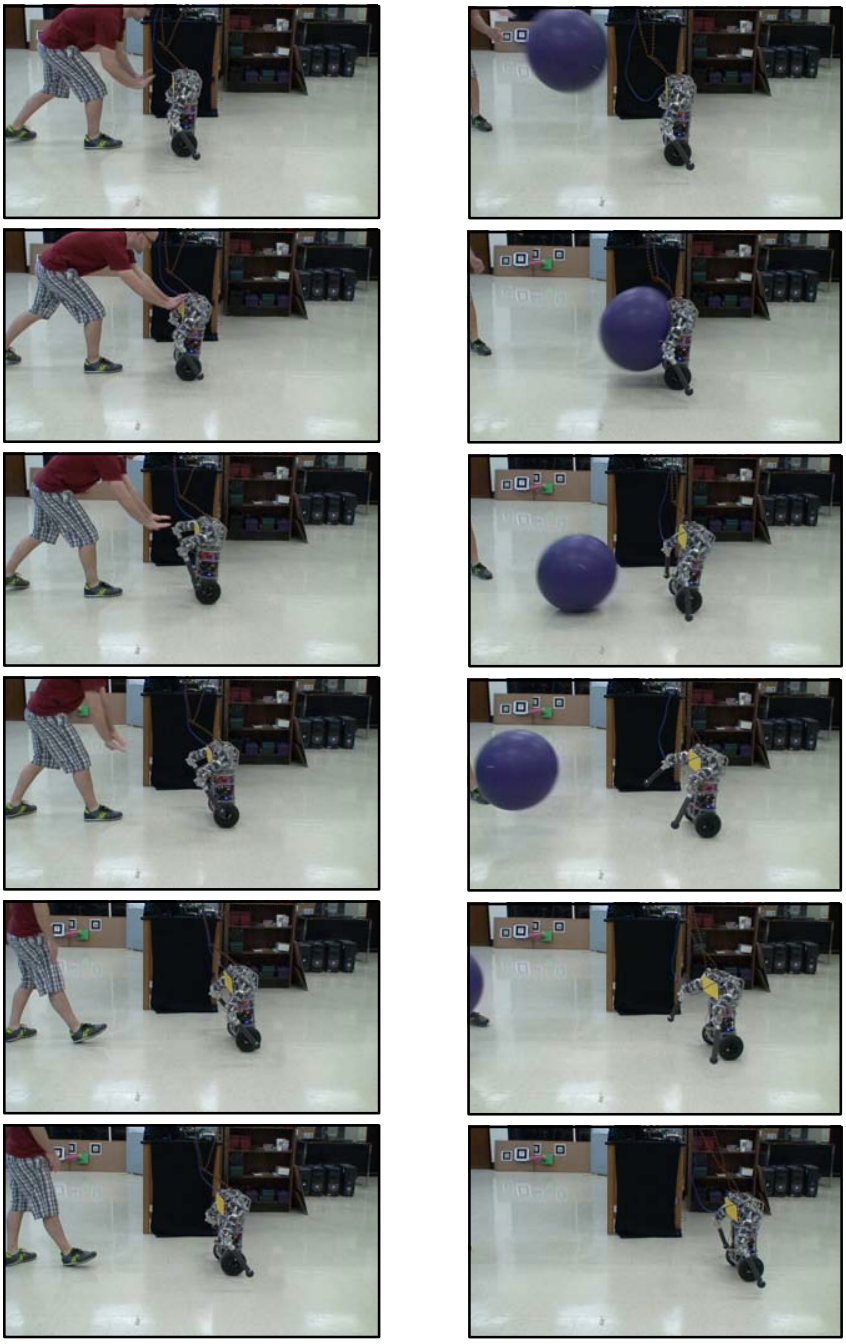
robot (high impact response). Example recovery sequences for push recovery and exercise ball impact recovery are shown in Figure 3.5.

3.5 Discussion

These results suggest that the integration of arm motions in balance recovery can reduce the recovery footprint and total energy expenditure, and increase the robot's ability to stabilize after large perturbations. Although the uBot's wheeled base is very different from that of a bipedal humanoid, there is considerable practical value in being able to experimentally determine the dynamic effects of upper body responses using this simpler system. In addition to having direct practical implications for wheeled mobile manipulators [4, 112, 74], we expect that the observed benefits could translate to other morphologies by the simple fact that using all available control resources is better than using only a subset. Indeed, our results are compatible with previous observations that the magnitude of human lower body recovery responses increased when arm motions were suppressed [78].

This general problem has several attributes that make it interesting from a control learning perspective: expensive evaluations, nonlinearity, underactuation, stochasticity, and high-dimensionality. Given a simple cost function and a low-dimensional policy representation, the Bayesian optimization algorithm was able to discover effective policies in a small number of trials. The two learned policies produced measurable efficiency and robustness gains over the wheels-only LQR response. Interestingly, although learning was done with fixed impact perturbations, the policies appear to be effective against more general, uncontrolled impacts. This suggests that, in practice, it might be only necessary to construct a small set of recovery policies and select among them based on, e.g., the perceived impact direction and magnitude.

One benefit of the Bayesian optimization approach that was not emphasized is the ability to use the learned cost model to interpret the robot's state of knowledge



(a) Low impact arm response (b) High impact arm response

Figure 3.5. Example trials of the learned high and low impact arm responses being selected executed for uncontrolled impact perturbations. (a) The robot uses the low impact policy in response to a human pushing. (b) The high impact response is selected to recover from a significantly larger impact. In both cases, impact magnitude is inferred using a simple classifier on IMU data.

about the problem during learning. For example, by examining the MAP length-scale hyperparameters, we can learn something about the relative sensitivity of the cost with respect to changes in each policy parameter. The length-scales after learning in the high impact experiment suggested that the cost is most sensitive to changes in initial response time and shoulder angle, with total movement time and elbow angle having considerably lower sensitivity. This information could, for example, be used to identify lower-dimensional policy representations by fixing parameters that have little effect on the cost.

One of the key observations from these experiments is that different policies can have different cost variance. Input-dependent variance leads to practical issues in applying Bayesian optimization since the cost variance in regions of high and low variance will tend to be underestimated and overestimated, respectively. For tasks such as impact stabilization, it might also be important to capture the cost variance of policies while learning to, e.g., select policies that have low cost *and* low variance. In the next chapter, I present an extension to the Bayesian optimization algorithm that supports this kind of policy selection.

CHAPTER 4

GLOBAL VARIABLE RISK POLICY SEARCH

4.1 Introduction

Model-free policy search methods (Section 2.2) are typically designed to minimize the expected value of a noisy cost signal by adjusting the parameters of a policy. By considering only the expected cost of a policy and ignoring cost variance, the solutions found by these algorithms are by definition *risk-neutral*. However, in many systems it can be advantageous to have a more flexible attitude toward risk. For example, a subsystem at a nuclear power plant might reasonably be risk-averse since even rare high cost events could have significant practical impact. On the other hand, a robot attached to a safety apparatus in a laboratory might seek out low probability, low cost trials to, e.g., attempt to identify the subset of initial conditions that led to such events. Studies in human motor control and animal behavior suggest that variable risk sensitivity may also be pervasive in nature [20, 10].

In the previous chapter, I described an application of a particular type of policy search algorithm, called Bayesian optimization, to the problem of learning arm motions that help stabilize the uBot after impact perturbations. By virtue of modeling the distribution of cost using a Gaussian process, Bayesian optimization algorithms make the assumption that the variance of the cost is the same for all policies in the search space. In general, this is not true. Indeed, in the experiments described, some impact recovery policies exhibited high variance, stabilizing in a fraction of the trials, while other policies had much lower variance. By capturing this policy-dependent

variance while learning, more flexible policy selection criteria can be applied to, e.g., explicitly favor predictable recovery strategies over those with higher risk.

In this chapter, I propose a new type of Bayesian optimization algorithm designed to handle problems with policy-dependent cost variance. The algorithm, called *Variational Bayesian Optimization* (VBO), is constructed by replacing the Gaussian process model with the Variational Heteroscedastic Gaussian Process model [65] designed for problems with input-dependent noise. I derive expressions for the expected improvement of a policy under the intractable variational predictive distribution that results from the VHGP model. I also show that confidence bound policy selection criteria that have been studied in the context of Bayesian optimization have a direct connection in this setting to risk-sensitive optimal control. Finally, I propose a generalized selection criterion called *expected risk improvement* that balances exploration and exploitation in the risk-sensitive optimization setting [61]. Results are presented from high-magnitude impact recovery experiments with the uBot-5.

4.2 Background

4.2.1 Variational Heteroscedastic Gaussian Process Regression

One limitation of the standard regression model (2.12) is the assumption of independent and identically distributed noise over the input space. Many data do not adhere to this simplification and models capable of capturing input-dependent noise (or *heteroscedasticity*) are required. The heteroscedastic regression model takes the form

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta}), \tag{4.1}$$

$$\varepsilon(\boldsymbol{\theta}) \sim \mathcal{N}(0, r(\boldsymbol{\theta})^2), \tag{4.2}$$

where the noise variance, $r(\boldsymbol{\theta})^2$, is dependent on the input, $\boldsymbol{\theta}$. In the Bayesian setting, a second GP prior,

$$g(\boldsymbol{\theta}) \sim \mathcal{GP}(\mu_0, k_g(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{4.3}$$

is placed over the unknown log variance function, $g(\boldsymbol{\theta}) \equiv \log r(\boldsymbol{\theta})^2$ [31, 45, 65]. The log variance is used is to ensure positivity of the variance function. This prior, when combined with the GP prior over cost functions (Section 2.3), forms the heteroscedastic Gaussian process (HGP) model. Unfortunately, the HGP model has property that the computations of the posterior distribution and the marginal log likelihood are intractable, thus making model selection and prediction difficult.

Stochastic techniques, such as Markov chain Monte Carlo (MCMC) [31], offer a principled way to deal with intractable probabilistic models. However, these methods tend to be computational demanding. An alternative approach is to analytically define the marginal probability in terms of a *variational* density, $q(\cdot)$. By restricting the class of variational densities by, e.g., assuming $q(\cdot)$ is Gaussian or factored in some way, it is often possible to define tractable bounds on the quantity of interest. In the Variational Heteroscedastic Gaussian Process (VHGP) model [65], a variational lower bound on the marginal log likelihood is used as a tractable surrogate function for optimizing the hyperparameters.

Let

$$\mathbf{g} = [g(\boldsymbol{\theta}_1), g(\boldsymbol{\theta}_2), \ldots, g(\boldsymbol{\theta}_N)]^\top \tag{4.4}$$

be the vector of latent log noise variances for the N data points. By defining a normal variational density, $q(\mathbf{g}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the following marginal variational bound can be derived [65],

$$F(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad = \quad \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_f + \mathbf{R}) - \tfrac{1}{4}\mathrm{tr}(\boldsymbol{\Sigma})$$

$$- \mathrm{KL}(\mathcal{N}(\mathbf{g}|\boldsymbol{\mu}, \boldsymbol{\Sigma})||\mathcal{N}(\mathbf{g}|\mu_0\mathbf{1}, \mathbf{K}_g)), \tag{4.5}$$

where $\mathbf{R}$ is a diagonal matrix with elements $[\mathbf{R}]_{ii} = e^{[\boldsymbol{\mu}]_i - [\boldsymbol{\Sigma}]_{ii}/2}$. Intuitively, by maximizing (4.5) with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we maximize the log marginal likelihood under the variational approximation while minimizing the distance (in the Kullback-Leibler sense) between the variational distribution and the distribution implied by the GP prior. By exploiting properties of $F(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ at its maximum, it is possible to write $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in terms of just N variational parameters,

$$\boldsymbol{\mu} = \mathbf{K}_g \left(\boldsymbol{\Lambda} - \frac{1}{2}\mathbf{I} \right) \mathbf{1} + \mu_0 \mathbf{1}, \tag{4.6}$$

$$\boldsymbol{\Sigma}^{-1} = \mathbf{K}_g^{-1} + \boldsymbol{\Lambda}, \tag{4.7}$$

where $\boldsymbol{\Lambda}$ is a positive semidefinite diagonal matrix of variational parameters. $F(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be simultaneously maximized with respect to the variational parameters and the HGP model hyperparameters, Ψ_f and Ψ_g. If the kernel functions $k_f(\boldsymbol{\theta}, \boldsymbol{\theta}')$ and $k_g(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are squared exponentials (2.11), then $\Psi_f = \{\sigma_f, \boldsymbol{\ell}_f\}$ and $\Psi_g = \{\mu_0, \sigma_g, \boldsymbol{\ell}_g\}$. Notice that the mean function of the cost GP prior is typically set to 0 since the data can be standardized or the maximum likelihood mean can be calculated and used when performing model selection [68]. However, a constant hyperparameter, μ_0, is included to capture the mean log variance since setting this value to 0 would be an arbitrary choice that would generally be incorrect. The gradients of $F(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to the parameters can be computed analytically in $\mathcal{O}(N^3)$ time (see Lázaro-Gredilla and Titsias [65] supplementary material), so the maximization problem can be solved using standard nonlinear optimization algorithms such as sequential quadratic programming (SQP).

The VHGP model yields a non-Gaussian variational predictive density,

$$q(\hat{J}_*) = \int \mathcal{N}(\hat{J}_*|a_*, c_*^2 + e^{g_*}) \mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*, \tag{4.8}$$

where

$$
\begin{aligned}
a_* &= \mathbf{k}_{f*}^\top (\mathbf{K}_f + \mathbf{R})^{-1}\mathbf{y}, \\
c_*^2 &= k_f(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - \mathbf{k}_{f*}^\top (\mathbf{K}_f + \mathbf{R})^{-1}\mathbf{k}_{f*}, \\
\mu_* &= \mathbf{k}_{g*}^\top (\boldsymbol{\Lambda} - \frac{1}{2}\mathbf{I})\mathbf{1} + \mu_0, \\
\sigma_*^2 &= k_g(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - \mathbf{k}_{g*}^\top (\mathbf{K}_g + \boldsymbol{\Lambda}^{-1})^{-1}\mathbf{k}_{g*}.
\end{aligned}
$$

Although this predictive density is intractable, its mean and variance can be calculated in closed form [65]:

$$\mathbb{E}_q[\hat{J}_*] = a_*, \tag{4.9}$$

$$\mathbb{V}_q[\hat{J}_*] = c_*^2 + \exp(\mu_* + \sigma_*^2/2) \equiv s_*^2. \tag{4.10}$$

4.2.1.1 Example

Figure 4.1(a) shows the result of performing model selection given a GP prior with a squared exponential kernel and unknown constant noise variance on a synthetic heteroscedastic data set. Figure 4.1(b) shows the result of optimizing the VHGP model on the same data. Model selection was performed using SQP to maximize the marginal log likelihood or, in the case of the VHGP model, the marginal variational bound. Due to the constant noise assumption, the GP model overestimates the cost variance in regions of low variance and underestimates in regions of high variance. In contrast, the VHGP model captures the input-dependent noise structure.

4.3 Variational Bayesian Optimization

There are at least two practical motivations for extending Bayesian optimization to capture policy-dependent cost variance. The first reason is to enable metrics computed on the predictive distribution, such as EI or probability of improvement,

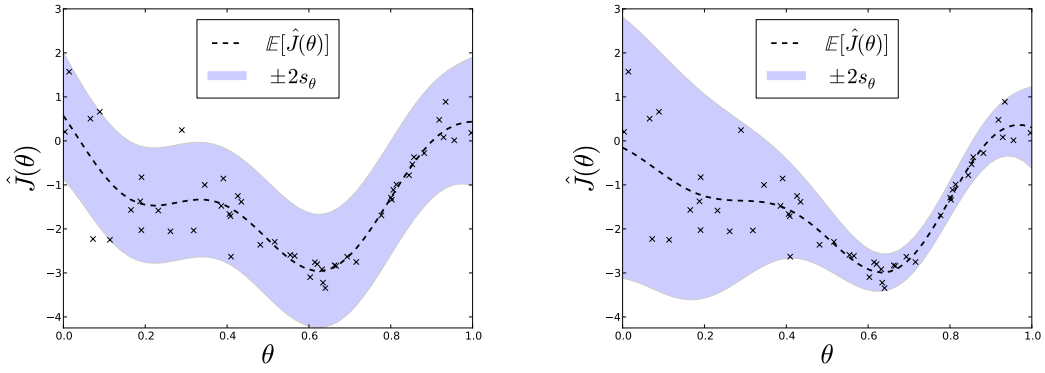


Figure 4.1. Comparison of fits for the standard Gaussian process model (a) and the VHGP model (b) on a synthetic heteroscedastic data set.

to return more meaningful values for the problem under consideration. For example, the GP model in Figure 4.1 would overestimate the expected improvement for $\theta = 0.6$ and underestimate the expected improvement of $\theta = 0.2$. The second reason is that it creates the opportunity to employ policy selection criteria that take cost variance into account, i.e., that are risk-sensitive.

I extend the VHGP model to the optimization case by deriving the expression for expected improvement and its gradient and show that both can be efficiently approximated to several decimal places using Gauss-Hermite quadrature [1] (as is the case for the predictive distribution itself [65]). Efficiently computable confidence bound selection criteria are also considered for selecting greedy risk-sensitive policies. A generalization of EI, called *expected risk improvement*, is derived that balances exploration and exploitation in the risk-sensitive case. Finally, to address numerical issues that arise when N is small (i.e., in the early stages of optimization), independent log priors are added to the marginal variational bound and heuristic sampling strategies are identified.

4.3.1 Expected Improvement

Recall from Section 2.3.2 that the expected improvement is defined as the expected reduction in cost, or *improvement*, over the the average cost of the best policy previously evaluated. The probability of the policy parameters, $\boldsymbol{\theta}_*$, having improvement, I_*, under the variational predictive distribution (4.8) is

$$q(I_*) = \int \mathcal{N}(I_*|\mu_{\text{best}} - a_*, v_*^2)\mathcal{N}(g_*|\mu_*, \sigma_*^2)dg_*, \tag{4.11}$$

where $v_*^2 = c_*^2 + e^{g_*}$. The expression for expected improvement then becomes

$$\begin{aligned} \text{EI}(\boldsymbol{\theta}_*, \mu_{\text{best}}) &= \int_0^\infty I_* q(I_*) dI_* & (4.12) \\ &= \int_0^\infty I_* dI_* \int \mathcal{N}(I_*|\mu_{\text{best}} - a_*, v_*^2)\mathcal{N}(g_*|\mu_*, \sigma_*^2)dg_*. & (4.13) \end{aligned}$$

To get (4.13) into a more convenient form, one can define

$$u_* = \frac{\mu_{\text{best}} - a_*}{v_*}, \quad x_* = \frac{\hat{J}_* - a_*}{v_*}, \tag{4.14}$$

and rewrite the expression for improvement (2.17) as,

$$I_* = \begin{cases} v_*(u_* - x_*) & \text{if } x_* < u_*, \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

By using this alternative form of improvement and changing the order of integration, we have

$$\text{EI}(\boldsymbol{\theta}_*, \mu_{\text{best}}) = \int \int_{-\infty}^{u_*} v_*(u_* - x_*)\phi(x_*)dx_* \mathcal{N}(g_*|\mu_*, \sigma_*^2)dg_*. \tag{4.16}$$

Letting $f(x_*) = v_*(u_* - x_*)$ and integrating $\int_{-\infty}^{u_*} f(x_*)\phi(x_*)dx_*$ by parts, we have

$$\int_{-\infty}^{u_*} f(x_*)\phi(x_*)dx_* = [f(x_*)\Phi(x_*)]_{-\infty}^{u_*} - \int_{-\infty}^{u_*} (-v_*)\Phi(x_*)dx_*, \tag{4.17}$$

$$= 0 + v_* \left[x_* \Phi(x_*) + \phi(x_*) \right]_{-\infty}^{u_*}, \tag{4.18}$$

$$= v_*(u_* \Phi(u_*) + \phi(u_*)), \tag{4.19}$$

where we have used the facts that $\lim_{x_* \to -\infty} \phi(x_*) = 0$ and $\lim_{x_* \to -\infty} C x_* \Phi(x_*) = 0$, where C is an arbitrary constant. Thus, the expression for expected improvement is

$$\mathrm{EI}(\boldsymbol{\theta}_*, \mu_{\mathrm{best}}) = \int v_*(u_* \Phi(u_*) + \phi(u_*)) \mathcal{N}(g_* | \mu_*, \sigma_*^2) dg_*. \tag{4.20}$$

Although this expression is not analytically tractable, it can be efficiently approximated using Gauss-Hermite quadrature [1]. This can be made clear by setting $\rho = (g_* - \mu_*)/\sqrt{2}\sigma_*$ and replacing all occurrences of g_* in the expressions for v_* and u_*,

$$\begin{aligned} \mathrm{EI}(\boldsymbol{\theta}_*, \mu_{\mathrm{best}}) &= \int e^{-\rho^2} \frac{v_*}{\sqrt{2\pi}\sigma_*} \left(u_* \Phi(u_*) + \phi(u_*) \right) d\rho, \\ &\equiv \int e^{-\rho^2} h(\rho) d\rho \approx \sum_{i=1}^{k} w_i h(\rho_i), \end{aligned} \tag{4.21}$$

where n is the number of sample points, ρ_i are the roots of the Hermite polynomial,

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n e^{-x^2}}{dx^n} \qquad i \in \{1, 2, \ldots, n\}, \tag{4.22}$$

and the weights are computed as $w_i = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 H_{n-1}(\rho_i)^2}$. In practice, a variety of tools are available for efficiently computing both w_i and ρ_i for a given n. In all of my experiments, $n = 45$.

Similarly, the gradient $\partial \mathrm{EI}(\boldsymbol{\theta}, \mu_{\mathrm{best}})/\partial \boldsymbol{\theta}$ can be computed under the integral (4.20) and the result is of the desired form:

$$\frac{\partial \mathrm{EI}(\boldsymbol{\theta}_*, \mu_{\mathrm{best}})}{\partial \boldsymbol{\theta}} = \int e^{-\rho^2} z(\rho) d\rho, \tag{4.23}$$

where

$$
\begin{aligned}
z(\rho) &= \frac{1}{\sqrt{2\pi}\sigma_*}\Big[\frac{1}{\sigma_*}v_*\left(u_*\Phi(u_*)+\phi(u_*)\right) \\
&\times \left(-\frac{\partial\sigma_*}{\partial\boldsymbol{\theta}}+2\rho^2\frac{\partial\sigma_*}{\partial\boldsymbol{\theta}}+\sqrt{2}\rho\frac{\partial\mu_*}{\partial\boldsymbol{\theta}}\right) \\
&+ \frac{\partial v_*}{\partial\boldsymbol{\theta}}(u_*\Phi(u_*)+\phi(u_*))+v_*\frac{\partial u_*}{\partial\boldsymbol{\theta}}\Phi(u_*)\Big].
\end{aligned}
$$

For the squared exponential kernel (2.11), the remaining gradients are

$$
\begin{aligned}
\frac{\partial\sigma_*}{\partial\boldsymbol{\theta}} &= -\frac{1}{\sigma_*}\mathbf{k}_{g*}^\top(\mathbf{K}_g-\boldsymbol{\Lambda}^{-1})^{-1}\frac{\partial\mathbf{k}_{g*}}{\partial\boldsymbol{\theta}}, && (4.24)\\
\frac{\partial\mu_*}{\partial\boldsymbol{\theta}} &= \mathbf{1}^\top\left(\boldsymbol{\Lambda}-\frac{1}{2}\mathbf{I}\right)\frac{\partial\mathbf{k}_{g*}}{\partial\boldsymbol{\theta}}, && (4.25)\\
\frac{\partial u_*}{\partial\boldsymbol{\theta}} &= -\frac{1}{v_*^2}\left(v_*\frac{\partial a_*}{\partial\boldsymbol{\theta}}+(\mu_{\text{best}}-a_*)\frac{\partial v_*}{\partial\boldsymbol{\theta}}\right), && (4.26)\\
\frac{\partial a_*}{\partial\boldsymbol{\theta}} &= \mathbf{y}^\top(\mathbf{K}_f+\mathbf{R})^{-1}\frac{\partial\mathbf{k}_{f*}}{\partial\boldsymbol{\theta}}, && (4.27)\\
\frac{\partial v_*}{\partial\boldsymbol{\theta}} &= -\frac{1}{v_*}\mathbf{k}_{f*}^\top(\mathbf{K}_f+\mathbf{R})^{-1}\frac{\partial\mathbf{k}_{f*}}{\partial\boldsymbol{\theta}}, && (4.28)\\
\left[\frac{\partial\mathbf{k}_{f*}}{\partial\boldsymbol{\theta}}\right]_i &= k_f(\boldsymbol{\theta}_i,\boldsymbol{\theta}_*)(\boldsymbol{\theta}_i-\boldsymbol{\theta}_*)^\top\mathbf{M}_f, \quad \text{and} && (4.29)\\
\left[\frac{\partial\mathbf{k}_{g*}}{\partial\boldsymbol{\theta}}\right]_i &= k_g(\boldsymbol{\theta}_i,\boldsymbol{\theta}_*)(\boldsymbol{\theta}_i-\boldsymbol{\theta}_*)^\top\mathbf{M}_g. && (4.30)
\end{aligned}
$$

As in the standard Bayesian optimization setting, one can easily incorporate an exploration parameter, ξ, by setting $u_* = (\mu_{\text{best}} - a_* + \xi)/v_*$, and maximize EI using standard nonlinear optimization algorithms. Since flat regions and multiple local maxima may be present, it is common practice to perform random restarts during EI optimization to avoid low-quality solutions. In my experiments, I used the NLOPT [40] implementation of SQP with 25 random restarts to optimize EI.

4.3.2 Confidence Bound Selection

In order to exploit cost variance information for policy selection, we must consider selection criteria that flexibly take cost variance into account. Although EI performs

well during learning by balancing exploration and exploitation, it falls short in this regard since it always favors high variance (or uncertainty) among solutions with equivalent expected cost. In contrast, *confidence bound* (CB) selection criteria allow one to directly specify the sensitivity to cost variance.

The family of confidence bound selection criteria have the general form

$$\mathrm{CB}(\boldsymbol{\theta}_*, \kappa) = \mathbb{E}[\hat{J}_*] + b(\mathbb{V}[\hat{J}_*], \kappa), \tag{4.31}$$

where $b(\cdot, \cdot)$ is a function of the cost variance and a constant risk factor, κ, that controls the system's sensitivity to risk. Such criteria have been extensively studied in the context of statistical global optimization [23, 109] and economic decision making [66]. Favorable regret bounds for sampling with CB criteria with $b(\mathbb{V}[J_*], \kappa) = \kappa\sqrt{\mathbb{V}[J_*]} \equiv \kappa s_*$ have also been derived for certain types of Bayesian optimization problems [109].

Interestingly, CB criteria have a strong connection to the exponential utility functions of risk-sensitive optimal control [135, 134]. By considering the risk-sensitive optimal control objective function introduced in Section 2.4,

$$\begin{aligned} \gamma(\boldsymbol{\theta}_*, \kappa) &= -2\kappa^{-1} \log \mathbb{E}[e^{-\frac{1}{2}\kappa \hat{J}_*}], & (4.32) \\ &\approx \mathbb{E}[\hat{J}_*] - \frac{1}{4}\kappa \mathbb{V}[\hat{J}_*], & (4.33) \end{aligned}$$

it is clear that policies selected according to a CB criterion with $b(\mathbb{V}[\hat{J}_*], \kappa) = -\frac{1}{4}\kappa\mathbb{V}[\hat{J}_*]$ can be viewed as approximate risk-sensitive optimal control solutions. Furthermore, since the selection is performed with resect to the predictive distribution (4.8), *policies with different risk characteristics can be selected on-the-fly*, without having to perform additional policy executions. This is a distinguishing property of this approach compared to other sample-based risk-sensitive optimal control algorithms that must perform separate optimizations that require policy executions to produce policies with different risk-sensitivity.

In practice, one typically sets $b(\mathbb{V}[\hat{J}_*], \kappa) = \kappa\sqrt{\mathbb{V}[\hat{J}_*]}$ so that terms of the same units are combined and the parameter κ has a straightforward interpretation. It is noteworthy that other functions of the mean and variance can also be used to form useful risk-sensitive criteria. For example, the Sharpe Ratio, $\text{SR} = \mathbb{E}[\hat{J}_*]/s_*$, is a commonly used metric in financial analysis [106]. Since the mean and variance of the VHGP model are analytically computable, extensions that optimize such criteria would be straightforward to implement.

4.3.3 Expected Risk Improvement

The primary advantage CB selection criteria offer is the ability to flexibly specify sensitivity to risk. However, CB criteria are greedy with respect to risk-sensitive objectives and therefore do not have the same exploratory quality as EI does for expected cost minimization. It is therefore natural to consider whether the EI criterion could be extended to perform risk-sensitive policy selection in a way that balances exploration and exploitation.

Schonlau et al. [105] considered a generalization of EI where the improvement for $\boldsymbol{\theta}_*$ was defined as

$$I_*^\rho = \max\{0, (\mu_{\text{best}} - \hat{J}_*)^\rho\}, \tag{4.34}$$

where ρ is an integer-valued parameter that affects the relative importance of large, low probability improvements and small, high probability improvements. Interestingly, the authors showed that for $\rho = 2$, $\text{EI}(\boldsymbol{\theta}_*, \rho) = \mathbb{E}[\hat{J}_*]^2 + \mathbb{V}[\hat{J}_*]$, which can be interpreted as a risk-seeking policy selection strategy. However, to perform balanced exploration in systems with more general risk sensitivity, a different generalization of EI is needed.

To address this problem, we define the *expected risk improvement* (ERI) criterion. In this case, the *risk improvement* for the policy parameters $\boldsymbol{\theta}_*$ is defined as

$$I_*^\kappa = \begin{cases} \mu_{\text{best}} + \kappa s_{\text{best}} - \hat{J}_* - \kappa s_* & \text{if } \hat{J}_* + \kappa s_* < \mu_{\text{best}} + \kappa s_{\text{best}}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.35}$$

where

$$\begin{aligned} i &= \arg \min_{j=1,\ldots,N} \mathbb{E}[\hat{J}(\boldsymbol{\theta}_j)] + \kappa s(\boldsymbol{\theta}_j), & (4.36) \\ \mu_{\text{best}} &= \mathbb{E}[\hat{J}(\boldsymbol{\theta}_i)], & (4.37) \\ s_{\text{best}} &= s(\boldsymbol{\theta}_i). & (4.38) \end{aligned}$$

Intuitively, the risk improvement captures the reduction in the value of the risk-sensitive objective, $\mathbb{E}[\hat{J}] + \kappa s$, over the best policy previously evaluated. Following a similar derivation as for EI, the expected risk improvement under the variational distribution is

$$\begin{aligned} \text{ERI}(\boldsymbol{\theta}_*, \kappa, \mu_{\text{best}}, s_{\text{best}}) &= \int_0^\infty I_*^\kappa q(I_*^\kappa) dI_*^\kappa \\ &= \int v_*(u_*\Phi(u_*) + \phi(u_*))\mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*, \end{aligned} \tag{4.39}$$

where $u_* = (\mu_{\text{best}} - a_* + \kappa(s_{\text{best}} - s_*))/v_*$. Thus, ERI can be viewed as a straightforward generalization of EI, where ERI = EI if $\kappa = 0$. Figure 4.2 shows how the ERI metric differs from EI in two simple examples with synthetic cost distributions.

4.3.4 Coping with Small Sample Sizes

4.3.4.1 Log Hyperpriors

Numerical precision problems are commonly experienced when performing model selection (which requires kernel matrix inversions and determinant calculations) using

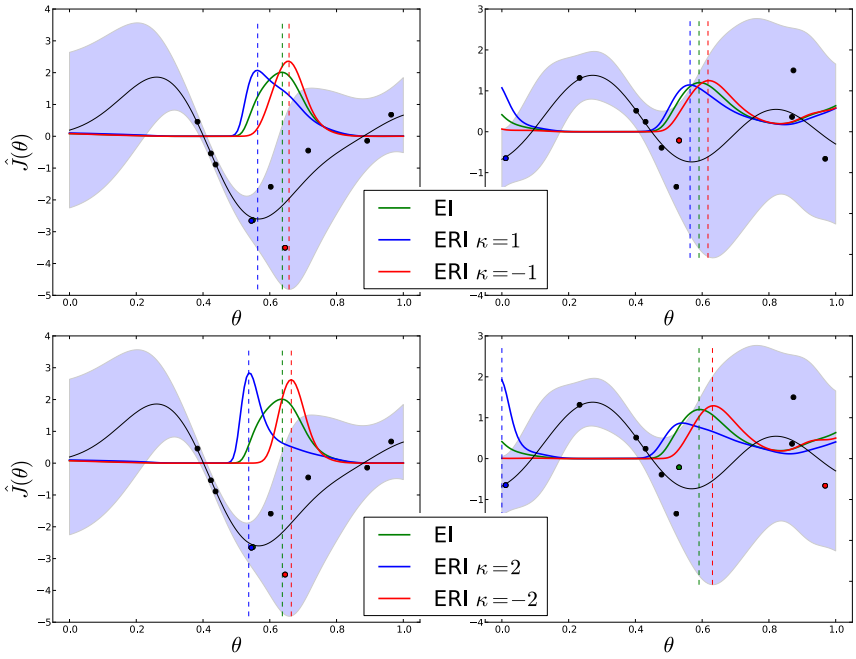


Figure 4.2. Qualitative comparison of ERI and EI for two simple synthetic cost distributions. The θ_{best} point for each criterion colored in correspondence with the lines. The EI and ERI are scaled in each plot for illustration purposes.

small amounts of data. To help avoid such numerical instability in the VHGP model when N is small, we augment $F(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with independent log-normal priors for each hyperparameter,

$$\hat{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = F(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \sum_{\psi_k \in \Psi} \log \mathcal{N}(\log \psi_k | \mu_k, \sigma_k^2), \tag{4.40}$$

where $\Psi = \Psi_f \cup \Psi_g$ is the set of all hyperparameters. Lizotte [68] showed that empirical performance can be improved in the standard Bayesian optimization setting by incorporating log-normal hyperpriors into the model selection procedure. In practice, these priors can be quite vague and thus do not require significant experimenter insight. For example, in the experiments described in this chapter, I set the log prior on length-scales so that the width of the 95% confidence region is at least 20 times the actual policy parameter ranges.

As is the case with standard marginal likelihood maximization, $\hat{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ may have several local optima. In practice, performing random restarts helps avoid low-quality solutions (especially when N is small). In all experiments, SQP was used with 10 random restarts to perform model selection.

4.3.4.2 Sampling

It is well known that selecting policies based on distributions fit using very little data can lead to myopic sampling and premature convergence [41]. For example, if one were unlucky enough to sample only the peaks of a periodic cost function, there would be good reason to infer that all policies have approximately equivalent (high) cost. Incorporating external randomization is one way to help alleviate this problem. For example, it is common to obtain a random sample of N_0 initial policies prior to performing optimization. Sampling according to EI with probability $1 - \epsilon$ and randomly otherwise can also perform well empirically. In the standard Bayesian optimization setting with model selection, ϵ-random EI selection has been shown to yield near-optimal global convergence rates [22].

Randomized CB selection with, e.g., $\kappa \sim \mathcal{N}(0, 1)$ can also be applied when the policy search is aimed at identifying a spectrum of policies with different risk sensitivities. However, since this technique relies completely on the estimated cost distribution, it is most appropriate to apply after a reasonable initial estimate of the cost distribution has been obtained.

The Variational Bayesian Optimization (VBO) algorithm is shown in Algorithm 2.

4.4 Experiments

4.4.1 Synthetic Data

As an illustrative example, in Figure 4.3 we compare the performance of the VBO to standard Bayesian optimization in a simple 1-dimensional noisy optimization task.

Algorithm 2 Variational Bayesian Optimization

Input: *Previous experience*: $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N], \mathbf{y} = [\hat{J}(\boldsymbol{\theta}_1), \ldots, \hat{J}(\boldsymbol{\theta}_N)]$, *Risk factor*: κ, *Iterations*: n

1. **for** $i := 1 : n$

   (a) *Perform model selection by optimizing hyperparameters and variational parameters using, e.g., SQP with random restarts*:
   $\Psi_f^+, \Psi_g^+, \boldsymbol{\Lambda}^+ := \arg\max \hat{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

   (b) *Maximize policy selection criterion w.r.t. optimized model*:

   - *Confidence Bound*:
     $\boldsymbol{\theta}' := \arg\min_{\boldsymbol{\theta}} \mathbb{E}_q[\hat{J}(\boldsymbol{\theta})] + \kappa\sqrt{\mathbb{V}_q[\hat{J}(\boldsymbol{\theta})]}$
   - *Expected Improvement*:
     $\mu_{\text{best}} := \min_{j=1,\ldots,|\mathbf{y}|} \mathbb{E}_q[\hat{J}(\boldsymbol{\theta}_j)]$
     $\boldsymbol{\theta}' := \arg\min_{\boldsymbol{\theta}} \text{EI}(\boldsymbol{\theta}, \mu_{\text{best}})$
   - *Expected Risk Improvement*:
     $b := \arg\min_{j=1,\ldots,|\mathbf{y}|} \mathbb{E}_q[\hat{J}(\boldsymbol{\theta}_j)] + \kappa\sqrt{\mathbb{V}_q[\hat{J}(\boldsymbol{\theta}_j)]}$
     $\mu_{\text{best}} := \mathbb{E}_q[\hat{J}(\boldsymbol{\theta}_b)]$
     $s_{\text{best}} := \sqrt{\mathbb{V}_q[\hat{J}(\boldsymbol{\theta}_b)]}$
     $\boldsymbol{\theta}' := \arg\min_{\boldsymbol{\theta}} \text{ERI}(\boldsymbol{\theta}, \kappa, \mu_{\text{best}}, s_{\text{best}})$

   (c) *Execute* $\boldsymbol{\theta}'$, *observe cost*, $\hat{J}(\boldsymbol{\theta}')$

   (d) *Append* $\boldsymbol{\Theta} := [\boldsymbol{\Theta}; \boldsymbol{\theta}']$, $\mathbf{y} := [\mathbf{y}; \hat{J}(\boldsymbol{\theta}')]$

2. **Return** $\boldsymbol{\Theta}, \mathbf{y}$

For this task, the true underlying cost distribution (Figure 4.3(a)) has two global minima (in the expected cost sense) with different cost variances. Both algorithms begin with the same $N_0 = 10$ random samples and perform 10 iterations of EI selection ($\xi = 1.0$, $\epsilon = 0.25$). In Figure 4.3(b), we see that Bayesian optimization succeeds in identifying the regions of low cost, but it cannot capture the policy-dependent variance characteristics.

In contrast, VBO reliably identifies the minima *and* approximates the local variance characteristics. Figure 4.3(d) shows the result of applying two different confidence bound selection criteria to vary risk sensitivity. In this case, $-\mathrm{CB}(\boldsymbol{\theta}_*, \kappa)$ was maximized, where

$$\mathrm{CB}(\boldsymbol{\theta}_*, \kappa) = \mathbb{E}_q[\hat{J}_*] + \kappa s_*. \tag{4.41}$$

Risk factors $\kappa = -1.5$ and $\kappa = 1.5$ were used to select a risk-seeking and risk-averse policy parameters, respectively.

4.4.2 Noisy Pendulum

As another simple example, I considered a swing-up task for a noisy pendulum system. In this task, the maximum torque output of the pendulum actuator is unknown and is drawn from a normal distribution at the beginning of each episode. As a rough physical analogy, this might be understood as fluctuations in motor performance that are caused by unmeasured changes in temperature. The policy space consisted of "bang-bang" policies in which the maximum torque is applied in the positive or negative direction, with switching times specified by two parameters, $0 \leq t_1, t_2 \leq 1.5$ sec. Thus, $\boldsymbol{\theta} = [t_1, t_2]$. The cost function was defined as

$$J(\boldsymbol{\theta}) = \int_0^T 0.01\alpha(t) + 0.0001u(t)^2 dt, \tag{4.42}$$

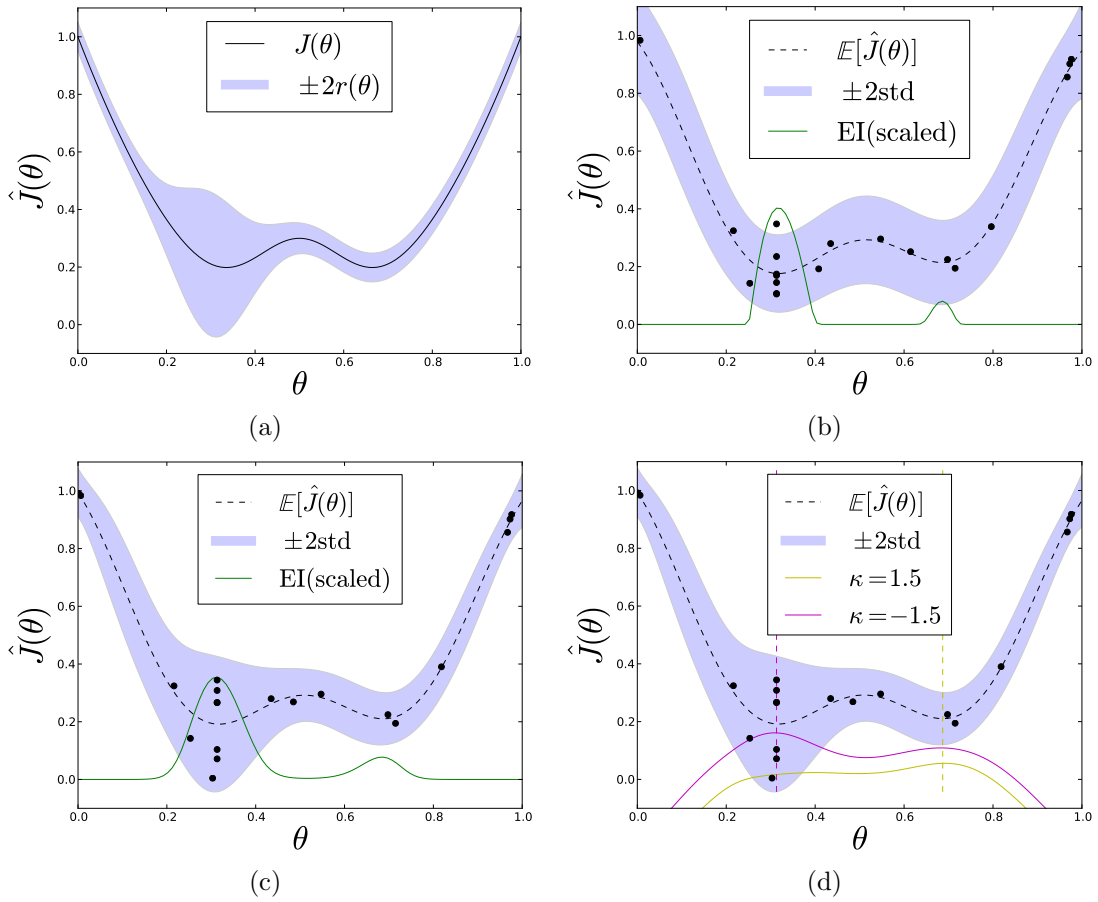
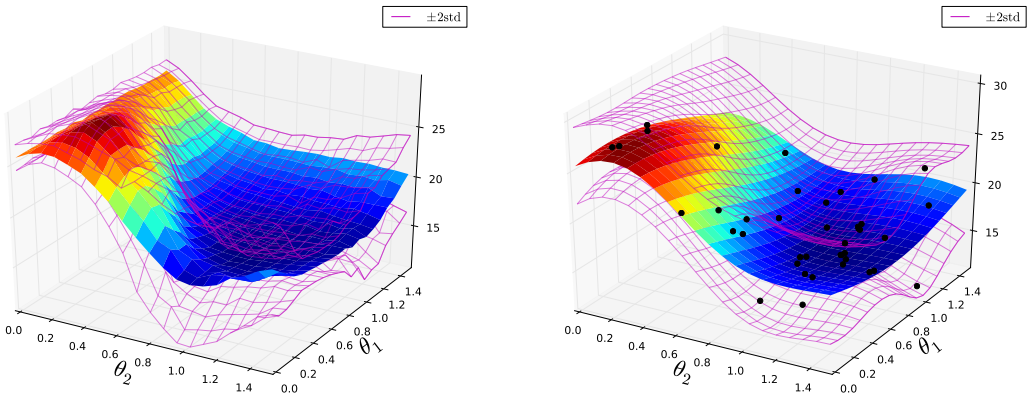


Figure 4.3. (a) An example latent noise distribution with two equivalent expected cost minima with different cost variance. (b) The distribution learned after 10 iterations of Bayesian optimization with EI selection and (c) after 10 iterations of VBO with EI selection (using the same initial $N_0 = 10$ random samples for both experiments). Bayesian optimization succeeded in identifying the minima, but it cannot distinguish between high and low variance solutions. (d) Confidence bound selection criteria are applied to select risk-seeking and risk-averse policy parameters given the distribution learned using VBO.

where $0 \leq \alpha(t) \leq \pi$ is the pendulum angle measured from upright vertical, $T = 3.5$ sec, and $u(t) = \tau_{\max}$ if $0 \leq t \leq t_1$, $u(t) = -\tau_{\max}$ if $t_1 < t \leq t_1 + t_2$, and $u(t) = \tau_{\max}$ if $t_1 + t_2 < t \leq T$. The system always started in the downward vertical position with zero initial velocity and the episode terminated if the pendulum came within 0.1 radians of the upright vertical position. The parameters of the system were $l = 1.0$ m, $m = 1.0$ kg, and $\tau_{\max} \sim \mathcal{N}(4, 0.3^2)$ Nm. With these physical parameters, the pendulum must (with probability ≈ 1.0) perform at least two swings to reach vertical in less than T seconds.

The cost function (4.42) suggests that policies that reach vertical as quickly as possible (i.e., using the fewest swings) are preferred. However, the success of an aggressive policy depends on the torque generating capability of the pendulum. With a noisy actuator, it is reasonable to expect aggressive policies to have higher variance. An approximation of the cost distribution obtained via discretization ($N = 40000$) is shown in Figure 4.4(a). It is clear from this figure that regions around policies that attempt two-swing solutions ($\boldsymbol{\theta} = [0.0, 1.0]$, $\boldsymbol{\theta} = [1.0, 1.5]$) have low expected cost, but high cost variance.

Figure 4.4(b) shows the results of 25 iterations of VBO using EI selection ($N_0 = 15, \xi = 1.0, \epsilon = 0.2$) in the noisy pendulum task. After $N = 40$ total evaluations, the expected cost and cost variance are sensibly represented in regions of low cost. Figure 4.5 illustrates the behavior of two policies selected by minimizing the CB criterion (4.41) on the learned distribution with $\kappa = \pm 2.0$. The risk-seeking policy ($\boldsymbol{\theta} = [1.03, 1.5]$) makes a large initial swing, attempting to reach the vertical position in two swings. In doing so, it only succeeds in reaching the goal configuration when the unobserved maximum actuator torque is large (greater than $\mathbb{E}[\tau_{\max}] + \sigma[\tau_{\max}]$). The risk-averse policy ($\boldsymbol{\theta} = [0.63, 1.14]$) always produces three swings and exhibits low cost variance, though it has higher cost than the risk-seeking policy when the maximum torque is large (15.93 versus 13.03).



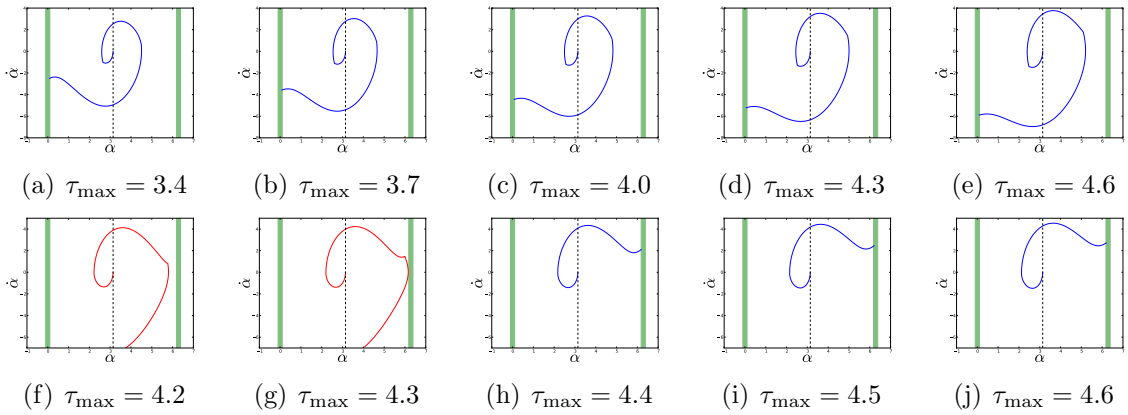
(a) (b)

Figure 4.4. (a) The cost distribution for the simulated noisy pendulum system obtained by a 20x20 discretization of the policy space. Each policy was evaluated 100 times to estimate the mean and variance ($N = 40000$). (b) Estimated cost distribution after 25 iterations of VBO with 15 initial random samples ($N = 40$). Because of the sample bias that results from EI selection, the optimization algorithm tends to focus modeling effort in regions of low cost.

It is often easy to understand the utility of risk-averse and risk-neutral policies, but the motivation for selecting risk-seeking policies might be less clear. The above result suggests one possibility: the acquisition of specialized, high-performance policies. For example, in some cases risk-seeking policies could be chosen in an attempt to identify observable initial conditions that lead to rare low-cost events. Subsequent optimizations might then be performed to direct the system to these initial conditions. One could also imagine situations when the context demands performance that lower risk policies are very unlikely to generate. For example, if the minimum time to goal was reduced so that only two swing policies had a reasonable chance of succeeding. In such instances it may be desirable to select higher risk policies, even if the probability of succeeding is quite low.

4.4.3 Variable Risk Balance Recovery with the uBot-5

In the experiments described in the previous chapter, the energetic and stabilizing effects of rapid arm motions on the LQR stabilized system were evaluated in the con-



(a) $\tau_{\max} = 3.4$ (b) $\tau_{\max} = 3.7$ (c) $\tau_{\max} = 4.0$ (d) $\tau_{\max} = 4.3$ (e) $\tau_{\max} = 4.6$

(f) $\tau_{\max} = 4.2$ (g) $\tau_{\max} = 4.3$ (h) $\tau_{\max} = 4.4$ (i) $\tau_{\max} = 4.5$ (j) $\tau_{\max} = 4.6$

Figure 4.5. Performance of risk-averse (a)-(e) and risk-seeking (f)-(j) policies as the maximum pendulum torque is varied. Shown are phase plots with the goal regions shaded in green. The risk-averse policy always used three swings and consistently reached the vertical position before the end of the episode. The risk-seeking policy used longer swing durations, attempting to reach the vertical position in only two swings. However, this strategy only pays off when the unobserved maximum actuator torque is large.

text of recovery from impact perturbations. One observation we made was that high energy impacts caused a subset of possible recovery policies to have high cost variance: successfully stabilizing in some trials, while failing to stabilize in others. In this section, I discuss subsequent impact recovery experiments where VBO was applied select risk-sensitive policies under more general conditions involving larger impact perturbations, an increased set of arm initial conditions, and a policy representation that permitted more flexible, asymmetric arm motions [60].

The robot was placed in a balancing configuration with its upper torso aligned with a 3.3 kg mass suspended from the ceiling (Figure 3.2). The mass was pulled away from the robot to a fixed angle and released, producing a controlled impact between the swinging mass and the robot. The pendulum momentum prior to impact was 9.9 ± 0.8 Ns and the resulting impact force was approximately equal to the robot's weight. The robot was consistently unable to recover from this perturbation using only the wheel LQR (see the rightmost column of Figure 4.6).

The parameterized policy controlled each arm joint according to an exponential trajectory, $\tau_i(t) = e^{-\lambda_i t}$, where $0 \leq \tau_i(t) \leq 1$ is the commanded DC motor power for joint i at time t. The λ parameters were paired for the shoulder/elbow pitch and the shoulder roll/yaw joints. This pairing allowed the magnitude of dorsal and lateral arm motions to be independently specified. The pitch (dorsal) motions were specified separately for each arm and the lateral motions were mirrored, which reduced the number of policy parameters to 3. The range of each λ_i was constrained: $1 \leq \lambda_i \leq 15$. At time t, if $\forall_i\ \tau_i(t) < 0.25$, the arms were retracted to a nominal configuration (the mean of the initial configurations) using a fixed, low-gain linear position controller.

The cost function was designed to encourage energy efficient solutions that successfully stabilized the system:

$$J(\boldsymbol{\theta}) = h(\mathbf{x}(T)) + \int_0^T \frac{1}{10} I(t)V(t)dt, \tag{4.43}$$

where $I(t)$ and $V(t)$ are the total absolute motor current and voltage at time t, respectively, $T = 3.5$ s, and $h(\mathbf{x}(T)) = 5$ if $\mathbf{x}(T) \in FailureStates$, otherwise $h(\mathbf{x}(T)) = 0$. After 15 random initial trials, we applied VBO with EI selection ($\xi = 1.0, \epsilon = 0.2$) for 15 episodes and randomized CB selection ($\kappa \sim \mathcal{N}(0, 1)$) for 15 episodes resulting in a total of $N = 45$ policy evaluations (approximately 2.5 minutes of total experience). Since the left and right pitch parameters are symmetric with respect to cost, we imposed an arbitrary ordering constraint, $\lambda_{\text{left}} \geq \lambda_{\text{right}}$, during policy selection.

After training, we evaluated four policies with different risk sensitivities selected by minimizing the CB criterion (4.41) with $\kappa = 2$, $\kappa = 0$, $\kappa = -1.5$, and $\kappa = -2$. Each selected policy was evaluated 10 times and the results are shown in Figure 4.6. The sample statistics confirm the algorithmic predictions about the relative riskiness of each policy. In this case, the risk-averse and risk-neutral policies were very similar (no statistically significant difference between the mean or variance), while the two

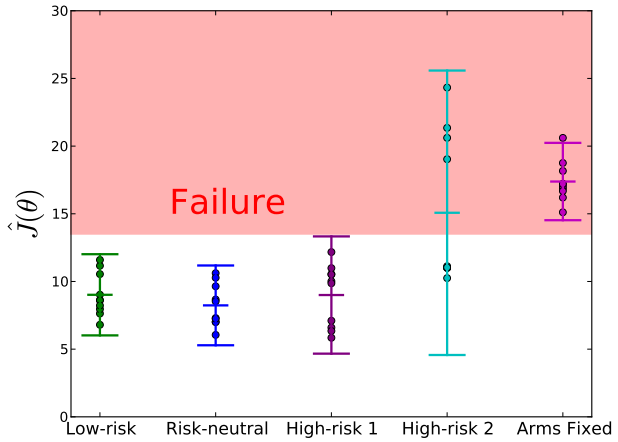


Figure 4.6. Data collected over 10 trials using policies identified as risk-averse, risk-neutral, and risk-seeking after performing VBO. The policies were selected using confidence bound criteria with $\kappa = 2$, $\kappa = 0$, $\kappa = -1.5$, and $\kappa = -2$, from left to right. The sample means and two times sample standard deviations are shown. The shaded region on the top part of the plot contains all trials that resulted in failure to stabilize. Ten trials with a fixed-arm policy are plotted on the far right to serve as a baseline level of performance for this impact magnitude.

risk-seeking policies had higher variance (for $\kappa = -2$, the differences in both the sample mean and variance were statistically significant).

For $\kappa = -2$, the selected policy produced an upward laterally-directed arm motion that failed approximately 50% of the time. In this case, the standard deviation of cost was sufficiently large that the second term in CB objective (4.41) dominated, producing a policy with high variance and poor average performance. A slightly less risk-seeking selection ($\kappa = -1.5$) yielded a policy with conservative low-energy arm movements that was more sensitive to initial conditions than the lower risk policies. This exertion of minimal effort could be viewed as a kind of gamble on initial conditions. Figure 4.7 gives a qualitative comparison of two successful trials executing the risk-averse and risk-seeking policies.

(a) Low risk policy, $\kappa = 2.0$ (b) High risk policy, $\kappa = -2.0$

Figure 4.7. Time series (duration: 1 second) showing two successful trials executing low-risk (a) and high-risk (b) policies selected using confidence bound criteria on the learned cost distribution. The low-risk policy produced an asymmetric dorsally-directed arm motion with reliable recovery performance. The high-risk policy produced an upward laterally-directed arm motion that failed approximately 50% of the time.

4.5 Discussion

In many real-world control problems, it can be advantageous to adjust risk sensitivity based on runtime context. For example, systems whose environments change in ways that make failures more or less costly (such as operating around catastrophic obstacles or in a safety harness) or when the context demands that the system seek low-probability high-performance events. Perhaps not surprisingly, this variable risk property has been observed in a variety of animal species, from simple motor tasks in humans to foraging birds and bees [20, 10].

However, most methods for learning policies by interaction focus on the risk-neutral minimization of expected cost. Extending Bayesian optimization methods to capture policy-dependent cost variance creates the opportunity to select policies with different risk sensitivities. Furthermore, the ability to efficiently vary risk sensitivity offers an advantage over existing model-free risk-sensitive control techniques that require separate optimizations and additional policy executions to produce policies with different risk.

This variable risk property was illustrated in experiments applying VBO to the problem of impact stabilization. After a short period of learning, an empirical comparison of policies selected with different confidence bound criteria confirmed the algorithmic predictions about the relative riskiness of each policy. However, how to set the system's risk sensitivity for a particular task remains an important open problem. In particular, we saw that when variance is very large for some policies, risk-seeking optimizations must be done carefully to avoid selecting policies with high variance and poor average performance. Other risk-sensitive policy selection criteria may be less susceptible to such phenomena.

Several properties of VBO should be considered when determining its suitability for a particular problem. First, although the computational complexity is the same as Bayesian optimization, $\mathcal{O}(N^3)$, the greater flexibility of the VHGP model means

that VBO tends to require more initial policy evaluations than standard Bayesian optimization. In addition, like several other model-free policy search algorithms, such as Bayesian optimization and finite-difference methods [100], VBO is sensitive to the number of policy parameters—high-dimensional policies can require many trials to optimize. These algorithms are therefore most effective in problems where low-dimensional policy representations are available, but accurate system models are not. However, there is evidence that policy spaces at least up to 15 dimensions can be efficiently explored with Bayesian optimization if estimates of the GP hyperparameters can be obtained *a priori* [67].

Another important consideration is the choice of kernel functions in the GP priors. In this work, we used the anisotropic squared exponential kernel to encode our prior assumptions regarding the smoothness and regularity of the underlying cost function. However, for many problems the underlying cost function is not smooth or regular; it contains flat regions and sharp discontinuities that can be difficult to represent. An interesting direction for future work is the use kernel functions with *local support*. Kernels that are not invariant to shifts in policy space will be necessary to capture cost surfaces that, e.g., contain both flat regions and regions with large changes in cost. Other methods for modeling the heteroscedastic cost distribution would also be interesting to investigate [125, 108, 45, 138].

In contrast to local methods, such as policy gradient, Bayesian optimization and VBO can produce large changes in policy parameters between episodes, which could be undesirable in some situations. One approach to alleviating this potential problem (other than simply limiting the range of the parameter search) is to combine VBO with local gradient methods. In the next chapter, I present an algorithm that uses a local approximation to the cost distribution as a critic structure for performing incremental, gradient-based updates to the policy parameters. This leads to some

attractive properties, such as local convergence, and the opportunity to construct hybrid approaches that combine gradient descent with local offline policy selection.

CHAPTER 5

LOCAL VARIABLE RISK POLICY SEARCH

5.1 Introduction

The VBO algorithm presented in the previous chapter performs risk-sensitive policy search by learning a heteroscedastic cost model and using it to perform global policy selection using one of several selection criteria. This approach has several attractive properties, including sample efficiency and the ability to change risk sensitivity without relearning. However, like most other algorithms of this kind, no general global convergence guarantees exist.

In contrast, gradient-based policy search methods typically have demonstrable local convergence properties [16]. In this chapter, I propose a simple risk-sensitive policy search algorithm based on stochastic gradient descent. Instead of using a global cost model to perform policy selection, the Risk-sensitive Stochastic Gradient Descent (RSSGD) algorithm uses a local cost model as a critic structure to make small, incremental changes to the policy parameters. It is straightforward to show that, under certain assumptions, the general RSSGD update follows the direction of the gradient of the risk-sensitive objective. Additionally, when a minimum variance baseline is used, the algorithm can be viewed as taking local steps in the direction of the risk improvement (Section 4.3.3) over the current policy parameters.

The possibility of interweaving online and offline local policy optimization is also considered. Offline optimizations, such as those discussed in the previous chapter, can be used to select local greedy policies or to change risk sensitivity on-the-fly. Ex-

periments with the uBot-5 learning to lift a heavy, liquid-filled bottle while balancing are discussed.

5.2 Risk-Sensitive Stochastic Gradient Descent

As in the previous chapter, the simple heteroscedastic regression model is used to describe the noisy cost signal,

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta}) \equiv J_{\boldsymbol{\theta}} + \varepsilon_{\boldsymbol{\theta}}, \tag{5.1}$$

where $\varepsilon_{\boldsymbol{\theta}} \sim \mathcal{N}(0, r_{\boldsymbol{\theta}}^2)$. The requirement that the noise term, $\varepsilon_{\boldsymbol{\theta}}$, be normally distributed is not strictly necessary to derive the expected performance results below (any mean 0 distribution with variance $r_{\boldsymbol{\theta}}^2$ would suffice). However, properties of the normal distribution are used to calculate the update variance (5.12). We define the risk-sensitive policy search problem as minimizing a confidence bound objective,

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \kappa), \quad \text{where} \tag{5.2}$$

$$F(\boldsymbol{\theta}, \kappa) = J_{\boldsymbol{\theta}} + \kappa r_{\boldsymbol{\theta}}, \tag{5.3}$$

and the risk factor, κ, specifies the system's sensitivity to risk.

Stochastic gradient descent methods have had significant practical applicability to solving robot control problems in the expected cost setting [121, 47, 103, 100], so I focus on extending this approach to the risk-sensitive case. The stochastic gradient descent algorithm, also called the weight perturbation algorithm [37], is a simple method for descending the gradient of a noisy objective function. The algorithm proceeds as follows. Starting with parameters, $\boldsymbol{\theta}$, execute the policy, $\pi_{\boldsymbol{\theta}}$, and observe the cost, $\hat{J}_{\boldsymbol{\theta}}$. Next, randomly sample a parameter perturbation, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$,

execute the perturbed policy, $\pi_{\boldsymbol{\theta}+\mathbf{z}}$, and observe the cost, $\hat{J}_{\boldsymbol{\theta}+\mathbf{z}}$. Finally, update the policy parameters, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, where

$$\Delta\boldsymbol{\theta} = -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} - \hat{J}_{\boldsymbol{\theta}})\mathbf{z}, \tag{5.4}$$

and η is a step size parameter. Intuitively, this rule updates the parameters in the direction of $\mathbf{z}$ if $\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} < \hat{J}_{\boldsymbol{\theta}}$, and in the direction of $-\mathbf{z}$ if $\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} > \hat{J}_{\boldsymbol{\theta}}$. It can be shown that, in expectation, this update follows the true (scaled) gradient of the expected cost,

$$\mathbb{E}[\Delta\boldsymbol{\theta}] = -\eta\sigma^2\nabla\mathbb{E}[\hat{J}_{\boldsymbol{\theta}}], \tag{5.5}$$

where $\nabla f_{\boldsymbol{\theta}} \equiv \frac{\partial f}{\partial \boldsymbol{\theta}}\big|_{\boldsymbol{\theta}}$.

In contrast, consider the risk-sensitive stochastic gradient descent (RSSGD) update:

$$\Delta\boldsymbol{\theta} \;=\; -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} + \kappa\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - b(\boldsymbol{\theta}))\mathbf{z}, \tag{5.6}$$

where $\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}}$ is an estimate of the cost standard deviation of $\pi_{\boldsymbol{\theta}+\mathbf{z}}$ and $b(\boldsymbol{\theta})$ is an arbitrary *baseline* function [136] of the policy parameters.

Substituting (5.1) into (5.6) and taking the first order Taylor expansion at $\boldsymbol{\theta} + \mathbf{z}$, we have

$$\Delta\boldsymbol{\theta} \;=\; -\eta\left(J_{\boldsymbol{\theta}+\mathbf{z}} + \varepsilon_{\boldsymbol{\theta}+\mathbf{z}} + \kappa\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - b(\boldsymbol{\theta})\right)\mathbf{z}, \tag{5.7}$$

$$\approx\; -\eta\left(J_{\boldsymbol{\theta}} + \mathbf{z}^\top\nabla J_{\boldsymbol{\theta}} + \varepsilon_{\boldsymbol{\theta}} + u\mathbf{z}^\top\nabla r_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}} + \kappa\mathbf{z}^\top\nabla\tilde{r}_{\boldsymbol{\theta}} - b(\boldsymbol{\theta})\right)\mathbf{z}, \tag{5.8}$$

$$\equiv \quad \tilde{\Delta}\boldsymbol{\theta},$$

where $u \sim \mathcal{N}(0,1)$. In expectation, this becomes

$$\mathbb{E}[\tilde{\Delta}\boldsymbol{\theta}] \quad = \quad -\eta\sigma^2 \left(\nabla J_{\boldsymbol{\theta}} + \kappa \nabla \tilde{r}_{\boldsymbol{\theta}}\right), \tag{5.9}$$

where the expectation is taken with respect to $\mathbf{z}$, u, and $\varepsilon_{\boldsymbol{\theta}}$. Thus, the update equation (5.6) is an estimator of the gradient of expected cost that is biased in the direction of the estimated gradient of the standard deviation to a degree specified by the risk factor, κ. If the estimator of the cost standard deviation is unbiased, we have

$$\mathbb{E}[\tilde{\Delta}\boldsymbol{\theta}] = -\eta\sigma^2 \nabla F(\boldsymbol{\theta}, \kappa), \tag{5.10}$$

a scaled unbiased estimate of the gradient of the risk-sensitive objective. Using a nonparameteric model, such as VHGP, as a local critic will not, in general, lead to unbiased estimates of the mean and variance of the cost. However, by introducing bias, these methods can potentially produce useful approximations of the local cost distribution after only a small number of policy evaluations.

5.2.1 Natural Gradient

From (5.10) it is clear that the unbiasedness of the update is also dependent on the isotropy of the sampling distribution, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. However, as was shown by Roberts and Tedrake [100], learning performance can be improved in some cases by optimizing the sampling distribution variance independently for each policy parameter, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. In this case, the expected update becomes biased,

$$\mathbb{E}[\tilde{\Delta}\boldsymbol{\theta}] = -\eta\boldsymbol{\Sigma} \nabla F(\boldsymbol{\theta}, \kappa), \tag{5.11}$$

but it is still in the direction of the *natural gradient* [3]. To see this, recall that for probabilistically sampled policies, the natural gradient is defined as $\mathbf{F}^{-1}\nabla f(\boldsymbol{\theta})$, where $\mathbf{F}^{-1}$ is the inverse Fisher information matrix [44]. When the policy sampling distribution is mean-zero Gaussian with covariance $\boldsymbol{\Sigma}$, the inverse Fisher information matrix is $\mathbf{F}^{-1} = \boldsymbol{\Sigma}$. Thus, (5.11) is in the direction of the natural gradient.

5.2.2 Baseline Selection

The expected update (5.9) is unaffected by the choice of the baseline function, $b(\boldsymbol{\theta})$, given that it depends only on $\boldsymbol{\theta}$. However, the choice of baseline does affect the *variance* of the update. As a rather trivial example that illustrates this point, consider the difference in performance that would result from setting $b(\boldsymbol{\theta}) = 0$ versus $b(\boldsymbol{\theta}) \sim \mathcal{N}(0, 100^2)$, where 100 is large relative to the cost.

The variance of the update (5.6) can be written as,

$$\begin{aligned}
\mathbb{V}[\tilde{\Delta}\boldsymbol{\theta}, b(\boldsymbol{\theta})] \;=\; & \eta^2\sigma^2 \left(b(\boldsymbol{\theta})^2\mathbf{I} - 2J_{\boldsymbol{\theta}} b(\boldsymbol{\theta})\mathbf{I} - 2\kappa\tilde{r}_{\boldsymbol{\theta}} b(\boldsymbol{\theta})\mathbf{I} + J_{\boldsymbol{\theta}}^2\mathbf{I} + 2\kappa J_{\boldsymbol{\theta}}\tilde{r}_{\boldsymbol{\theta}}\mathbf{I} \right. \\
& + \kappa^2\tilde{r}_{\boldsymbol{\theta}}^2\mathbf{I} + r_{\boldsymbol{\theta}}^4\mathbf{I} + \sigma^2(\nabla J_{\boldsymbol{\theta}}^\top \nabla J_{\boldsymbol{\theta}}\mathbf{I} + \nabla J_{\boldsymbol{\theta}} \nabla J_{\boldsymbol{\theta}}^\top) \\
& + \sigma^2\kappa(2\nabla J_{\boldsymbol{\theta}}^\top \nabla \tilde{r}_{\boldsymbol{\theta}}\mathbf{I} + \nabla J_{\boldsymbol{\theta}} \nabla \tilde{r}_{\boldsymbol{\theta}}^\top + \nabla \tilde{r}_{\boldsymbol{\theta}} \nabla J_{\boldsymbol{\theta}}^\top) \\
& + \sigma^2 r_{\boldsymbol{\theta}}^2(\nabla r_{\boldsymbol{\theta}}^\top \nabla r_{\boldsymbol{\theta}}\mathbf{I} + 2\nabla r_{\boldsymbol{\theta}} \nabla r_{\boldsymbol{\theta}}^\top) \\
& \left. + \sigma^2\kappa^2(\nabla \tilde{r}_{\boldsymbol{\theta}}^\top \nabla \tilde{r}_{\boldsymbol{\theta}}\mathbf{I} + \nabla \tilde{r}_{\boldsymbol{\theta}} \nabla \tilde{r}_{\boldsymbol{\theta}}^\top) \right) . \qquad (5.12)
\end{aligned}$$

It is straightforward to show that the baseline that minimizes (5.12) is $b(\boldsymbol{\theta}) = J_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}}$, which yields

$$\begin{aligned}
\mathbb{V}[\tilde{\Delta}\boldsymbol{\theta}, J_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}}] \;=\; & \eta^2\sigma^2 \left(r_{\boldsymbol{\theta}}^4\mathbf{I} + \sigma^2(\nabla J_{\boldsymbol{\theta}}^\top \nabla J_{\boldsymbol{\theta}}\mathbf{I} + \nabla J_{\boldsymbol{\theta}} \nabla J_{\boldsymbol{\theta}}^\top) \right. \\
& + \sigma^2\kappa(2\nabla J_{\boldsymbol{\theta}}^\top \nabla \tilde{r}_{\boldsymbol{\theta}}\mathbf{I} + \nabla J_{\boldsymbol{\theta}} \nabla \tilde{r}_{\boldsymbol{\theta}}^\top + \nabla \tilde{r}_{\boldsymbol{\theta}} \nabla J_{\boldsymbol{\theta}}^\top) \\
& \left. + \sigma^2 r_{\boldsymbol{\theta}}^2(\nabla r_{\boldsymbol{\theta}}^\top \nabla r_{\boldsymbol{\theta}}\mathbf{I} + 2\nabla r_{\boldsymbol{\theta}} \nabla r_{\boldsymbol{\theta}}^\top) \right)
\end{aligned}$$

$$+\sigma^2\kappa^2(\nabla\tilde{r}_{\boldsymbol{\theta}}^\top\nabla\tilde{r}_{\boldsymbol{\theta}}\mathbf{I}+\nabla\tilde{r}_{\boldsymbol{\theta}}\nabla\tilde{r}_{\boldsymbol{\theta}}^\top)\big). \tag{5.13}$$

However, since $J_{\boldsymbol{\theta}}$ is unknown, we define the baseline using an estimate of the expected cost, $\tilde{J}_{\boldsymbol{\theta}}$. The resulting increase in variance over the optimal baseline is proportional to the squared error of the expected cost estimate: $\eta^2\sigma^2(J_{\boldsymbol{\theta}}-\tilde{J}_{\boldsymbol{\theta}})^2$. The RSSGD update then becomes

$$\Delta\boldsymbol{\theta} \;=\; -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}}-\tilde{J}_{\boldsymbol{\theta}}+\kappa(\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}}-\tilde{r}_{\boldsymbol{\theta}}))\mathbf{z}. \tag{5.14}$$

Intuitively, (5.14) reduces to the classical stochastic gradient descent update when either the system has a neutral attitude toward risk ($\kappa=0$) or when the estimate of the cost standard deviation is locally constant: $\nabla\tilde{r}_{\boldsymbol{\theta}}=0\Rightarrow\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}}-\tilde{r}_{\boldsymbol{\theta}}=0$, for small $\mathbf{z}$ such that the linearization holds. Note the relationship between the RSSGD update and the expected risk improvement (ERI) criterion (4.39) from the previous chapter. From this point of view, the update can be interpreted as taking steps in the direction of risk improvement over the current policy parameter setting.

In implementation, it can be helpful to divide the step size by $\tilde{r}_{\boldsymbol{\theta}}$ so the update maintains scale invariance to changing noise magnitude (see Algorithm 3). This way, samples are weighted by the local cost variance estimate so, e.g., large differences in cost in high variance regions do not cause large fluctuations in the policy parameter values. On the other hand, large fluctuations in the cost variance estimate could produce undesirably large or small step sizes. We therefore also constrain the scaled step size to stay in some reasonable range, e.g., $\eta/\tilde{r}_{\boldsymbol{\theta}}\in[0.01,0.9]$. Although this approach is heuristic, it does have practical advantages such as weighting updates according to their perceived reliability.

5.2.3 Critic Representation

The RSSGD algorithm requires a local model of the cost distribution in the neighborhood of $\boldsymbol{\theta}$. This model can be viewed as a kind of *critic* because its role is similar to that played by the critic structure in actor-critic algorithms [9, 51]: it reduces the variance of the gradient descent update by constructing long-term cost statistics. One possible approach to constructing a local critic is to apply the same method for learning heteroscedastic cost models used by the VBO algorithm. In my experiments, the VHGP [65] model was used to construct the local critic based on noisy observations of cost, although other algorithms could also be used [45, 125, 108, 138].

As in the VBO algorithm, the critic is updated after each policy evaluation by recomputing the predictive cost distribution. However, in this case model selection and prediction are performed using only observations near the current parameterization, $\boldsymbol{\theta}$. A nearest neighbor selection can be performed efficiently around the current policy parameters by storing observations in a KD-tree data structure and using, e.g., a k-nearest neighbors or an ϵ-ball criterion. However, because the number of samples is typically small in the types of robot control tasks under consideration, the actual computational effort required to find nearest neighbors and perform model selection is quite modest. Thus, the primary advantage of constructing a local, rather than a global, model is that cost distributions that are nonstationary with respect to their optimal hyperparameter values can be handled more easily.

The risk-sensitive stochastic gradient descent (RSSGD) algorithm is outlined in Algorithm 3.

The relationship of the RSSGD algorithm to VBO leads to the straightforward insight that the local critic can also be used to perform offline optimizations, e.g.,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}^*} \tilde{F}(\boldsymbol{\theta}^*, \kappa) = \tilde{J}_{\boldsymbol{\theta}^*} + \kappa \tilde{r}_{\boldsymbol{\theta}^*}. \tag{5.15}$$

Algorithm 3 Risk-sensitive stochastic gradient descent

Input: *Parameters*: η, σ, ϵ, *Risk factor*: κ, *Initial policy*: $\boldsymbol{\theta}$

1. *Initialize* $\boldsymbol{\Theta} = [\,], \mathbf{y} = [\,]$,
2. **while** not converged:
   (a) *Sample perturbation:* $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$
   (b) *Execute* $\boldsymbol{\theta} + \mathbf{z}$, *record cost* $\hat{J}_{\boldsymbol{\theta}+\mathbf{z}}$
   (c) *Update data:*
   $\boldsymbol{\Theta}, \mathbf{y} = [\boldsymbol{\Theta}; \boldsymbol{\theta} + \mathbf{z}], [\mathbf{y}; \hat{J}_{\boldsymbol{\theta}+\mathbf{z}}]$
   $\boldsymbol{\Theta}_{\text{loc}}, \mathbf{y}_{\text{loc}} = NearestNeighbors(\boldsymbol{\Theta}, \mathbf{y}, \boldsymbol{\theta}, \epsilon)$
   (d) *Compute posterior mean and variance:*
   $\tilde{J}_{\boldsymbol{\theta}} = \mathbb{E}[\hat{J}_{\boldsymbol{\theta}} \mid \boldsymbol{\Theta}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
   $\tilde{r}^2_{\boldsymbol{\theta}} = \mathbb{V}[\hat{J}_{\boldsymbol{\theta}} \mid \boldsymbol{\Theta}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
   $\tilde{r}^2_{\boldsymbol{\theta}+\mathbf{z}} = \mathbb{V}[\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} \mid \boldsymbol{\Theta}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
   (e) *Update policy parameters:*
   $\Delta\boldsymbol{\theta} := -\frac{\eta}{\tilde{r}_{\boldsymbol{\theta}}} \left(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{J}_{\boldsymbol{\theta}} + \kappa(\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{r}_{\boldsymbol{\theta}}) \right) \mathbf{z}$
   $\boldsymbol{\theta} := \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
3. **Return** $\boldsymbol{\Theta}, \mathbf{y}, \boldsymbol{\theta}$

This is essentially the same as the VBO algorithm from the previous chapter, except that policy selection is performed in the local neighborhood of $\boldsymbol{\theta}$. This is particularly useful when κ is varied online to adjust risk based on the current operating context. This simple procedure is given in Algorithm 4.

More generally, it is possible to imagine a spectrum of risk-sensitive policy search algorithms where, on one end, are algorithms like VBO that model the entire cost distribution and perform offline global policy selection, and, on the other end, are algorithms like RSSGD that construct local models of the cost distribution and make small incremental changes to the policy parameters. In between these approaches are algorithms that interweave gradient descent an offline policy selection to, e.g., speed up gradient descent or quickly change risk-sensitivity. The experimental results described in Section 5.3 show how local offline policy selection can be used to make runtime changes to a dynamic lifting policy that led to significant performance improvements under changing optimization criteria.

Algorithm 4 Offline local policy optimization

Input: *Neighbor threshold*: ϵ, *Risk factor*: κ, *Initial policy*: $\boldsymbol{\theta}$, *Data*: $\boldsymbol{\Theta}, \mathbf{y}$

1. *Compute local neighborhood:*
   $\boldsymbol{\Theta}_{\text{loc}}, \mathbf{y}_{\text{loc}} = NearestNeighbors(\boldsymbol{\Theta}, \mathbf{y}, \boldsymbol{\theta}, \epsilon)$
2. *Optimize $\boldsymbol{\theta}$ locally using, e.g., SQP:*
   Return $\arg\min_{\boldsymbol{\theta}} \tilde{F}(\boldsymbol{\theta}, \kappa)$

5.2.4 Example

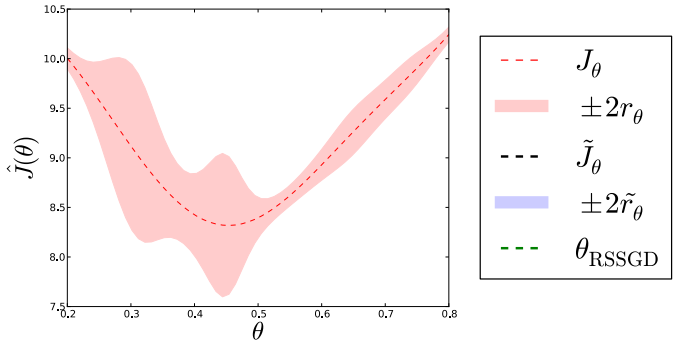
Figure 5.1 illustrates example runs of the above algorithms using the synthetic cost distribution in Figure 5.1(a). Figure 5.1(b) shows the result of applying the RSSGD algorithm with a risk-averse objective, $\kappa = 2$. The algorithm descends the gradient of the upper confidence bound to a local minimum while maintaining a reasonable local approximation of the cost distribution.

Figure 5.1(c) shows the result of applying offline local policy optimization using the local estimate of the cost distribution obtained during a risk-neutral gradient descent ($N = 50$). By performing offline local optimization using a risk-neutral objective, the algorithm selects a near-optimal average cost policy. Changing the value of the risk factor in the offline optimization objective leads to selection of local risk-averse ($\kappa = 2$) and risk-seeking ($\kappa = -2$) policies.

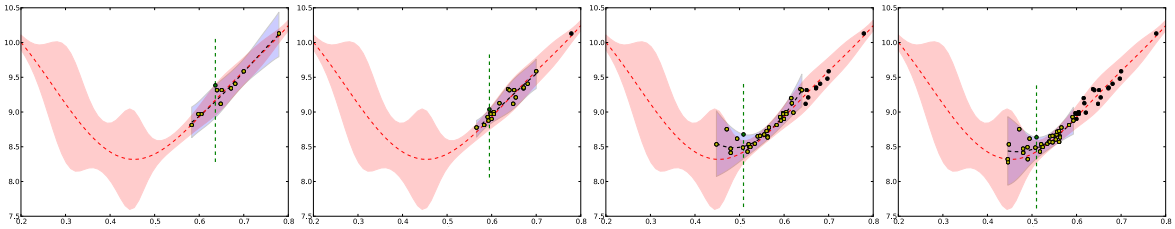
5.3 Experiments in Dynamic Heavy Lifting

To evaluate the performance of the RSSGD algorithm in a dynamic robot control task, we considered the problem of using the uBot-5 to lift a 1 kg, partially-filled laundry detergent bottle from the ground to a height of 120 cm. This problem is challenging for several reasons. First, the bottle is heavy, so most arm trajectories from the starting configuration to the goal will not succeed because of the limited torque generating capabilities of the arm motors. Second, the upper body motions act as disturbances to the LQR. Thus, violent lifting trajectories will cause the robot to destabilize and fall. Finally, the bottle itself has significant dynamics because the heavy liquid sloshes as the bottle moves. Since the robot had only a simple claw gripper and I made no modifications to the bottle, the bottle moved freely in the hand, which had a significant effect on the stabilized system.

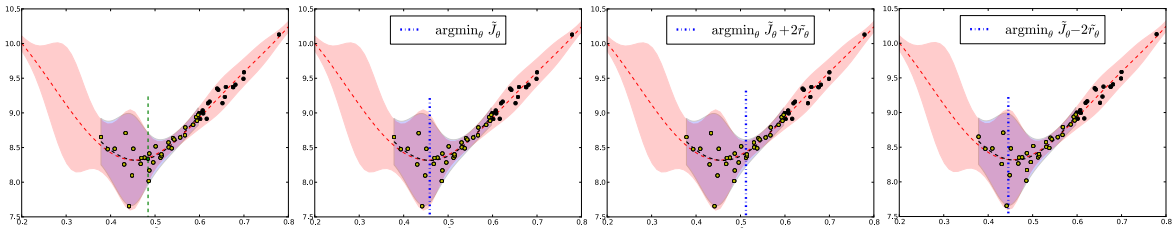
The policy was represented as a cubic spline trajectory in the right arm joint space with 7 open parameters to be optimized by the algorithm. The parameters included 4 shoulder and elbow waypoint positions and 3 time parameters. The start and end configurations were fixed. Joint velocities at the waypoints were computed using the tangent method [24]. The initial policy was a hand-crafted smooth and short duration motion to the goal configuration. However, with the bottle in hand, this policy succeeded only a small fraction of the time, with most trials resulting in a failure to lift the bottle above the shoulder.



(a) Example latent cost distribution.



(b) Risk-averse stochastic gradient decent



(c) Different risk-sensitive policies can are selected offline using the local distribution learned during risk-neutral gradient descent.

Figure 5.1. (a) A synthetic latent cost distribution with input-dependent variance. (b) Risk-averse stochastic gradient descent descends the upper confidence bound of the latent cost distribution while maintaining a reasonable approximation of the cost distribution around the nominal parameter value. (c) Offline local optimization is performed using different risk-sensitive objectives given the local distribution learned during risk-neutral gradient descent.

The cost function was defined as

$$J(\boldsymbol{\theta}) = \int_0^T \left(\mathbf{x}(t)^\top \mathbf{Q}\mathbf{x}(t) + cI(t)V(t)\right) dt, \tag{5.16}$$

where $\mathbf{x} = [x_{wheel}, \dot{x}_{wheel}, \alpha_{body}, \dot{\alpha}_{body}, h_{error}]^\top$, $I(t)$ and $V(t)$ are total motor current and voltage for all motors at time t, $\mathbf{Q} = \text{diag}([0.001, 0.001, 0.5, 0.5, 0.05])$, and $c = 0.01$. The components of the state vector are the wheel position and velocity, body angle and angular velocity, and vertical error between the desired and actual bottle position, respectively. Intuitively, this cost function encourages fast and energy efficient solutions that do not violently perturb the LQR. In each trial, the sampling rate was 100 Hz and $T = 6$ s. A trial ended when either $t > T$ or the robot reached the goal configuration with maintained low translational velocity (≤ 5 cm/s). The algorithm parameter values in all experiments were $\eta = 0.5, \sigma = 0.075, \epsilon = 3.5\sigma$, and $\eta/\tilde{r}_{\boldsymbol{\theta}} \in [0.01, 0.5]$. Each policy parameter range was scaled to be $\theta_i \in [0, 1]$, thus the constant σ corresponded to different (unscaled) perturbation sizes for each dimension depending on the total parameter range.

5.3.1 Risk-Neutral Learning

In the first experiment, we ran RSSGD with $\kappa = 0$ to perform a risk-neutral gradient descent. The VHGP model was used to locally construct the critic and model selection was performed using the NLOPT [40] implementation of SQP. A total of 30 trials (less than 2.5 minutes of total experience) were performed and a reliable, low-cost policy was learned. The robot failed to recover balance in 3 of the 30 trials. In these cases, the emergency stop was activated and the robot was manually reset. Figure 5.2 illustrates the reduction in cost via empirical measurements taken at fixed intervals during learning. Interestingly, the learned policy exploits the dynamics of the liquid in the bottle by timing the motion such that the shifting bottle contents coordinate with the LQR controller to correct the angular displacement of

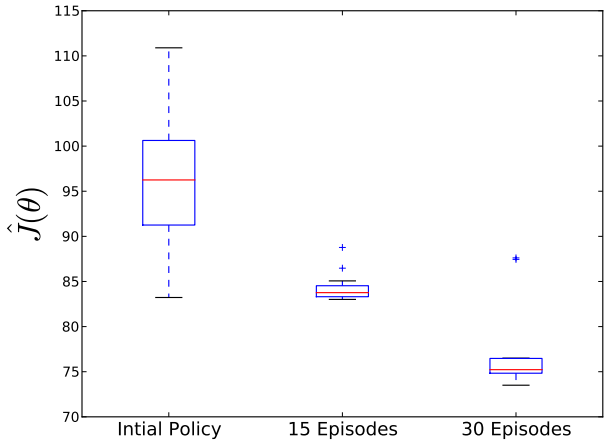
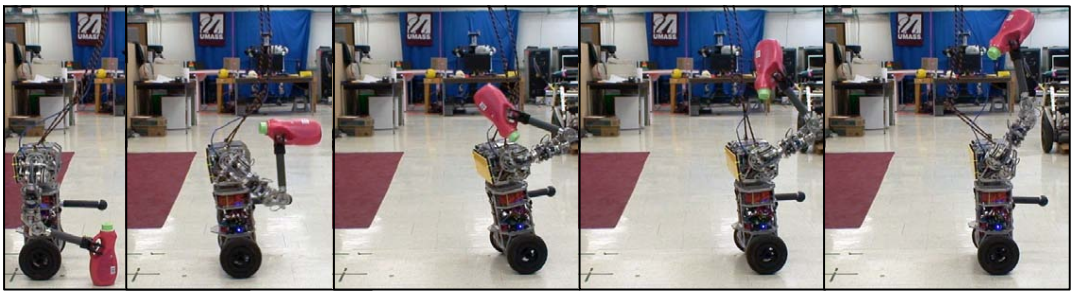


Figure 5.2. Data collected from 10 test trials executing the initial lifting policy, the policy after 15 episodes of learning, and the final policy after 30 episodes of learning.

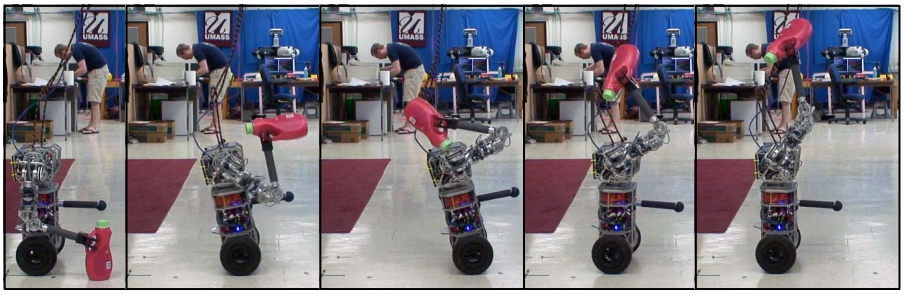
the body. This dynamic interaction would be very difficult to capture in a system model. Incidentally, this serves as a good example of the value of policy search techniques: by virtue of ignoring the dynamics, they are in some sense insensitive to the complexity of the dynamics [100]. Figure 5.3(a) shows an example run of the learned policy.

5.3.2 Variable Risk Control

In the process of learning a low average-cost policy, a model of the local cost distribution was repeatedly computed. The next experiments examined the effect of performing offline policy selection using the estimate of the local cost distribution around the learned policy. In particular, I considered two hypothetical changes in operating context: when robot's workspace is reduced, requiring that the policy have a small footprint with high certainty, and when the battery charge is very low, requiring that the policy uses very little energy with high certainty. Offline policy selection and subsequent risk-averse gradient descent was performed for each case and the resulting policies were empirically compared.



(a)



(b)

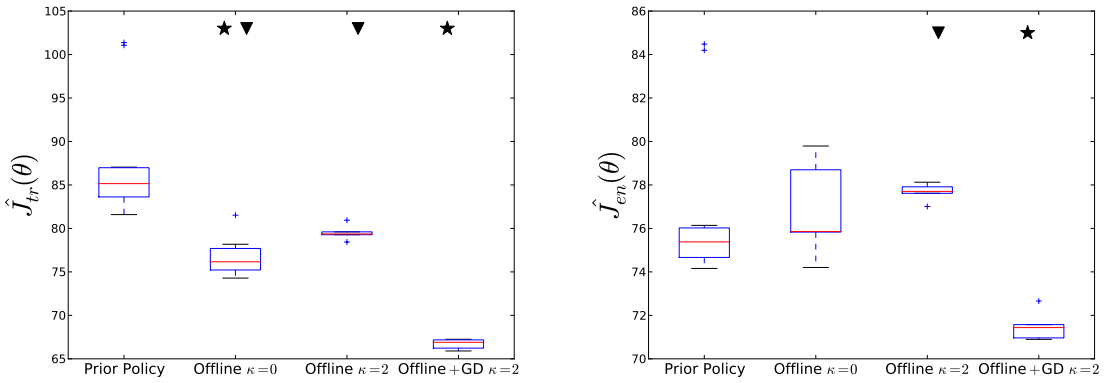
Figure 5.3. (a) The learned risk-neutral policy exploits the dynamics of the container to reliably perform the lifting task. (b) With no additional learning trials, a risk-averse policy is selected offline that reliably reduces translation. The total time duration of each of the above sequences is about 3 seconds.

Context changes were represented by a reweighting of cost function terms. For example, to capture the low battery charge context, the relative weight of the motor power term in (5.17) was increased: $\mathbf{Q}_{en} = \text{diag}([0.0005, 0.0005, 0.25, 0.25, 0.05])$ and $c_{en} = 0.1$. The cost of previous trajectories was then computed using the transformed cost function,

$$J_{en}(\boldsymbol{\theta}) = \int_0^T \left(\mathbf{x}(t)^\top \mathbf{Q}_{en}\mathbf{x}(t) + c_{en}I(t)V(t)\right) dt. \tag{5.17}$$

The VHGP model was used to approximate the transformed cost distribution, $\hat{J}_{en}(\boldsymbol{\theta})$, around the previously learned policy parameters. SQP was used to minimize $\tilde{F}_{en}(\boldsymbol{\theta}, \kappa)$ offline. Likewise, to represent the translation-averse case, the relative weight assigned to wheel translation was increased, $\mathbf{Q}_{tr} = \text{diag}([0.002, 0.001, 0.5, 0.5, 0.05])$ and $c_{tr} = 0.001$, and the resulting transformed local model was used to minimize $\tilde{F}_{tr}(\boldsymbol{\theta}, \kappa)$ offline.

Both risk-neutral ($\kappa = 0$) and risk-averse ($\kappa = 2$) offline policy selection were performed for each case. Additionally, 5 episodes of risk-averse ($\kappa = 2$) gradient descent was performed starting from the offline selected risk-averse policy. Each policy was executed 5 times and the results were empirically compared. Figure 5.4(a) shows the results from the translation aversion experiments. The risk-neutral offline policy had statistically significantly lower average (transformed) cost (Behrens-Fisher, $p < 0.05$) and lower variance (F-test, $p < 0.05$) than the original learned policy. The risk-averse offline policy also has significantly lower average cost than the prior learned policy, but its average cost was slightly (not statistically significantly) higher than the offline risk-neutral policy. However, the offline risk-averse policy had significantly lower variance than the risk-neutral offline policy (F-test, $p < 0.05$). An example run of the offline risk-averse policy is shown in Figure 5.3(b). Finally, the policy learned after 5 episodes of risk-averse gradient descent starting from the offline selected policy



(a) Translation aversion

(b) Energy aversion

Figure 5.4. Data from test runs of the prior learned policy, the offline selected risk-neutral and risk-averse policies, and the policy after 5 episodes of risk-averse gradient descent starting from the risk-averse offline policy. A star at the top of a column signifies a statistically significant reduction in the mean compared with the previous column (Behrens-Fisher, $p < 0.05$) and a triangle signifies a significant reduction in the variance (F-test, $p < 0.05$).

led to another significant reduction in expected cost while maintaining similarly low variance.

For the energy-averse case, the offline risk-neutral policy had no statistically significant difference in sample average or variance compared with the prior learned policy. The risk-averse policy had slightly (not statistically significantly) higher average cost than both the original learned policy and the offline risk-neutral policy, but it had significantly lower variance (F-test, $p < 0.05$). The policy learned after 5 episodes of risk-averse gradient descent had significantly lower average cost than the offline risk-averse while maintaining similar variance (see Figure 5.4(b)).

5.4 Discussion

The VBO and RSSGD algorithms are connected by their shared use of a learned heteroscedastic cost model to perform policy search. VBO uses this model to globally select policies, whereas RSSGD uses it as a local critic to descend the gradient of a

risk-sensitive objective. Both algorithms have the advantage of being independent of the dynamics, dimensionality, and cost function structure, and the disadvantage of their performance being dependent on the dimensionality of the policy parameter space. Thus, as is the case with other parameter perturbation methods [100, 90], the expressiveness of policy parameterizations should be balanced with their parsimony to ensure that the number of trials needed to find a suitable policy remains small.

Policy gradient approaches that are designed to learn dynamic models, such as PILCO [28], can also be used to capture uncertainty in the cost distribution for different policies. Such approaches are capable of handling high-dimensional policy spaces, however certain smoothness assumptions must be made about the system dynamics. Furthermore, performing offline optimizations to change risk-sensitivity would be much more computationally intensive than the approach presented here.

The very recent work of Tamar et al. [119] describes likelihood-ratio policy gradient algorithms appropriate for different types of risk-sensitive criteria. The simulation-based algorithm in their work is the most closely related to the RSSGD update rule. However, rather than learning a nonparameteric cost model, their algorithm uses a two-timescale approach to obtain incremental unbiased estimates of the cost mean and variance. In some cases, this unbiasedness might be more important than the sample efficiency that cost-model-based approaches can offer.

Roberts and Tedrake [100] showed that adjusting the covariance of the perturbation distribution based on a signal-to-noise optimization can lead to better performance. This idea is related to the covariance matrix adaptation that is done in some cost weighted averaging methods [113]. An interesting direction for future work would be to use the learned local model to adjust the sampling distribution by, e.g., scaling the perturbation covariance by the optimized length-scale hyperparameters. In this way, the perturbation magnitude for each parameter could be scaled by the inferred sensitivity of the cost to changes in that parameter. Methods for using gra-

dient estimates from the local critic to update the policy parameters or, conversely, using gradient observations to update the critic could also be explored.

Local offline optimization can be performed by applying the VBO algorithm with constraints on the parameter search space. This leads to the possibility to interweave gradient descent with local offline policy selection to select local greedy polices to speed up gradient descent or quickly change risk-sensitivity. This approach was used in the dynamic lifting experiments with the uBot-5. First, a policy was learned that exploited the system dynamics to produce an efficient and reliable lifting strategy. Then, starting from this learned policy, new local cost models were fit and used to select translation-averse and energy-averse policies. It is interesting that this kind of flexibility is possible after so few trials, especially given the generality of the optimization procedure. However, a notable limitation of the implementation described is that generalization to different objects or lifting scenarios would require separate optimizations. The extent to which more sophisticated closed-loop or model-based policy representations could support generalization is an interesting open question.

CHAPTER 6

POSTURAL CONTROL AND RECOVERY WITH THE UBOT-5

6.1 Introduction

In the previous two chapters, new tools for performing efficient risk-sensitive stochastic optimization were presented and applied to various policy search problems. In particular, two of these experiments involved the uBot-5 mobile manipulator (Section 3.3.1). As one of the primary experimental platforms in the Laboratory for Perceptual Robotics, a long-term research goal is to develop a complete postural stability control system that increases the robustness and deployability of the robot into unconstrained human environments. In this chapter, I describe how risk-sensitive optimization has played a role in the development of postural stability and recovery controllers that support this general goal.

6.2 Postural Modes and Dynamic Transition Events

The uBot is a versatile research platform that has supported a variety of experiments in mobile manipulation [124, 52, 62, 53, 54] and human-robot interaction [27, 89, 42, 140]. As a mid-sized humanoid that balances on two wheels, it is a unique platform for studying the advantages and limitations of dynamically stable mobile manipulators. In particular, dynamic stability leads to a coupling of effectors that can, for example, be exploited to increase pushing and pulling forces [124, 48] or, as we saw in Chapter 3, increase stabilization performance after impacts. Another

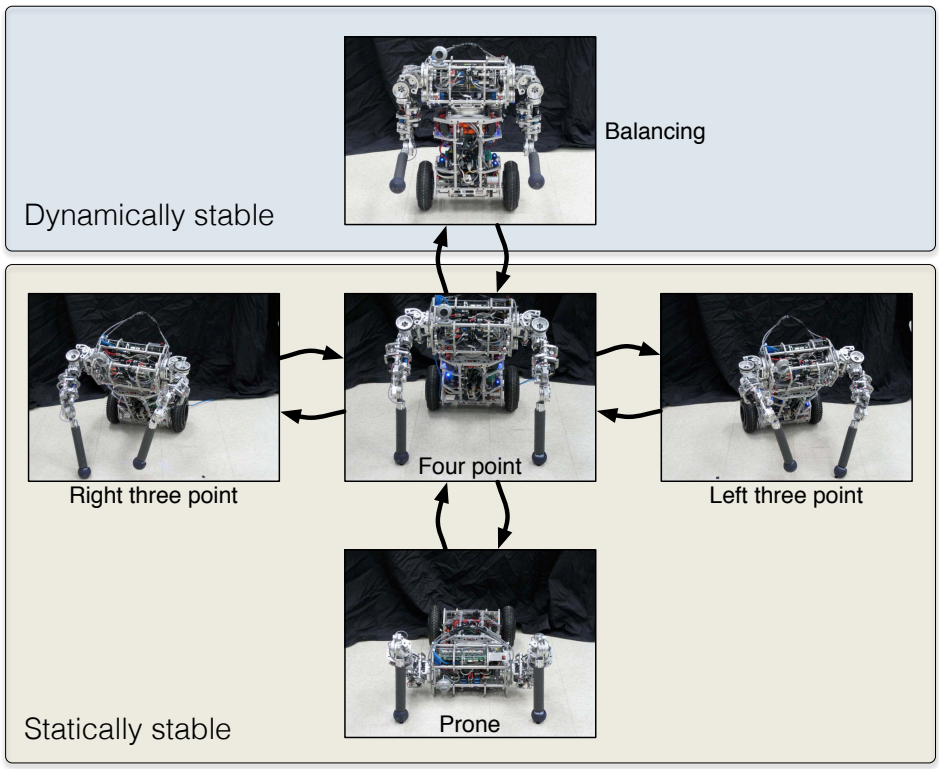


Figure 6.1. Examples illustrating the five basic postures of the uBot-5.

interesting question is the extent to which the dynamic response of balancing systems can be used to measure manipulation forces [74].

When the uBot is not balancing in an upright configuration, it can be in one of several statically stable poses (Figure 6.1). Due to arm redundancy, each postural mode actually contains many feasible configurations, so the configurations shown in Figure 6.1 should be viewed as representative examples. Simple quasistatic controllers for transitioning between the statically stable postures, and simple gaits that arise out of sequences of these transitions, are described in our prior work [63]. Essentially, by moving sufficiently slowly and ensuring that the robot's center of mass stays within the ground support polygon, reliable transitions between postures can be performed.

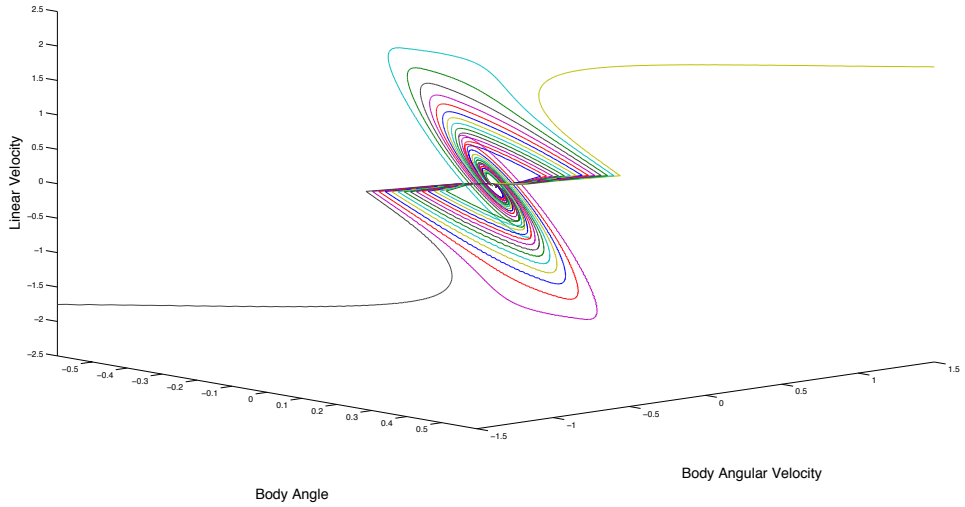
However, transitions to and from the balancing configuration cannot be handled in a quasistatic way. This is not to imply that simple controllers are precluded

as a result. In disturbance-free environments, simply moving the arms to specific configurations and turning off the LQR controller will produce reliable transitions to the 4-point posture. Likewise, transitioning to the balancing posture from a 4-point configuration can be achieved by doing a "push up" [63] until the robot's body angle is near vertical and then activating the LQR. However, controlling dynamic transitions and maintaining stability in the face of environmental perturbations is more difficult.

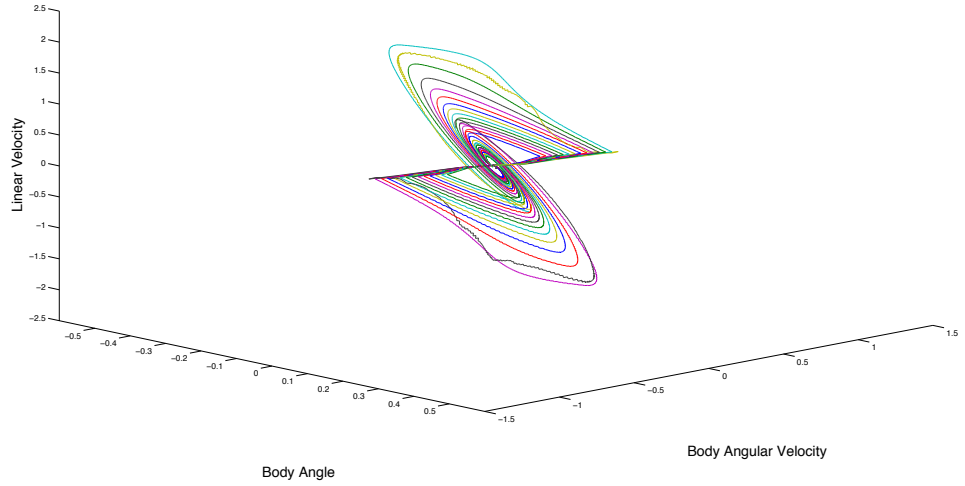
For example, consider the task of maintaining stability in the balancing posture under unknown perturbations. If we define the balancing posture as the set of states that can be stabilized by the LQR, i.e., all states in its *basin of attraction*, then one can imagine dynamic transition events where external disturbances cause the system to leave the set of balancing states. In this case, control actions must be taken to either return the system to the set of balancing states, or to transition to another stable posture in a way that protects the hardware and supports subsequent recovery.

The experiments described in Chapters 3 and 4 considered the effects of combining learned open-loop arm motions with the LQR response after impact perturbations. In particular, the risk-averse and risk-neutral policies learned using VBO significantly increased the robot's ability to recover from very large impact forces roughly equivalent to the robot's total mass in earth gravity. At these large impact magnitudes, the LQR consistently fails to recover. Thus, the arm responses help return the robot from an unstable state to the set of balancing states. Another way to say this is that the arm motions increase the basin of attraction for the balancing posture. A graphical example of this obtained from a simple simulated uBot-5 is shown in Figure 6.2.

When balance recovery is not possible, such as after a very large impact perturbation, actions must be taken to safely bring the system to rest. Transitions directly to a prone posture typically produce very large body accelerations upon ground impact, so these transitions are to be avoided. Likewise, falling on top of the arms in an uncontrolled way would likely damage the hardware and possibly produce electrical



(a) LQR responses of a simulated uBot after various impact magnitudes. At the largest impact magnitude considered, the robot fails to stabilize and return to the fixed point at the origin.



(b) Optimized arm motions increase the basin of attraction.

Figure 6.2. Example phase plots from a simple 2D dynamic simulation of the uBot-5. Impulse forces of increasing magnitude were generated and symmetric arm responses for the largest impact were learned via a direct trajectory optimization.

shorts due to the robot's open chassis. Thus, bracing strategies that facilitate safe transitions to the 4-point posture are considered in the next section.

6.3 Bracing for Falls

In the face of very large impact perturbations, the uBot must perform a bracing behavior to transition to the statically stable 4-point posture in a way that minimizes body acceleration and hardware strain. Because the stakes are high in this case (i.e., there is a significant chance of hardware damage), the robot must aim to achieve good performance with high certainty. In other words, the system should optimize its bracing strategy with respect to a risk-averse criterion.

To develop the fall bracing controller, controlled impact perturbations to the torso were generated using the same pendulum apparatus from the arm recovery experiments (Chapter 3). The drop height was varied randomly in a small range, so the momentum prior to impact was approximately 14 ± 2 Ns. This is a significantly larger perturbation than was considered in the previous experiments and under no circumstances has the robot been able to recover balance from these large impacts.

The class of feasible bracing policies was strongly constrained by the physical limitations of the robot. The time between impact onset and arm endpoint contact with the ground was approximately 1/4 second. Given this short time duration, arm initial conditions, and the robot's actuator velocity limitations, the range of configurations of the arms for endpoint ground contact was very limited. Additionally, torque had to be minimized for a subset of the arm joints that are driven with rubber belts, since these can slip and fail to absorb the impact. The problem of selecting arm configurations for bracing was therefore effectively solved by the physical constraints. However, the optimal arm stiffnesses remained unknown. Thus, the bracing problem involved selecting the joint stiffnesses for ground impact given the bracing arm configuration that satisfied the constraints of the system.

The joint stiffnesses were governed by a parameter $\theta \in [0, 1]$, where the value of θ was linearly related to the proportional gains of the low-level joint PD controllers and the maximum joint motor PWM signal (effectively a maximum torque threshold). The stiffness parameter was optimized with respect to the cost function,

$$J(\theta) = h(\mathbf{x}(T)) + \int_0^T (0.1\ddot{\alpha}(t)^2 + 5I(t)V(t))dt, \tag{6.1}$$

where $T = 2.0$ sec, $\ddot{\alpha}(t)$ is the body acceleration at time t, and $I(t)$ and $V(t)$ are the motor currents and voltages for all arm joints, respectively. If hardware was damaged as a result of the bracing trial, $h(\mathbf{x}(T)) = 10$ and $h(\mathbf{x}(T)) = 0$ otherwise. All observed hardware failures were broken steel pulley cables at the elbow joints. In principle, this failure could be detected by the robot with a simple elbow flexion routine, but for simplicity the presence or absence of hardware failures was manually identified after each trial. Risk-averse ($\kappa = 2$) gradient descent using the RSSGD algorithm was performed with $\eta = 0.7, \sigma = 0.05, \epsilon = 4\sigma$, and $\eta/\tilde{r}_{\boldsymbol{\theta}} \in [0.01, 0.5]$. Although the problem is a simple 1-dimensional optimization task, the high relative noise magnitude throughout the search space makes it challenging to perform gradient descent efficiently.

Snapshots of the learning sequence are shown in Figure 6.3. Initially, the robot started with a low-stiffness policy and gradually adjusted the policy to increase the bracing stiffness. Although high-stiffness policies have low average cost since they tend to produce lower body accelerations, they are more likely to causing hardware damage due to increased strain on the arm joints. Thus, high-stiffness policies have high risk and the risk-averse optimization settled on a slightly higher expected cost, but lower risk policy. I collected 52 additional samples of randomly selected policies to verify that the learned policy is near-optimal for the $k = 2$ criterion (Figure 6.4). An example run of the resulting bracing policy is shown in Figure 6.5.

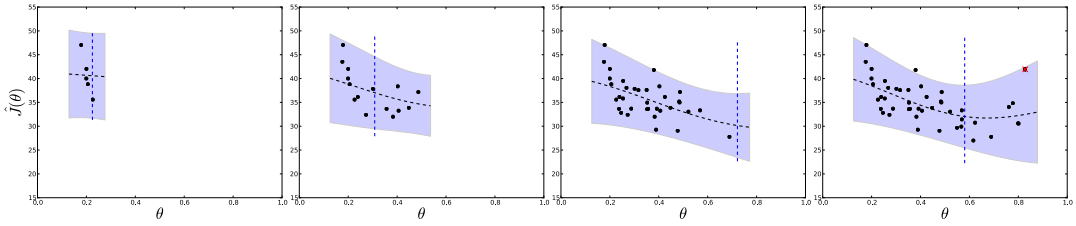


Figure 6.3. Snapshots of the learning sequence for risk-averse bracing. From left to right, $N = 5, 15, 35$, and 45. The vertical blue line indicates the nominal policy and the red data point indicates a hardware failure.

A complete bracing and recovery sequence is shown in Figure 6.6, where the bracing behavior is used to respond to a kicking perturbation and the push-up controller is used to return to the balancing configuration.

6.4 Recovery Policy Switching

The development of whole-body balance recovery and bracing policies raises an obvious question: when should each of these policies be used? Ideally, the robot should always try to recover balance except in those cases where it is unable to do so. However, because of sensor limitations and the non-negligible performance variation of both policies, this line is not clearly defined. Thus, the policy for switching between bracing and recovery will depend strongly on the risk-sensitivity of the system.

To illustrate this point, I performed a set of impact perturbation experiments, where the robot selected between the learned (risk-averse) arm recovery policy or the bracing policy based on inferred impact magnitude. Impacts were generated randomly and ranged from moderate (arm recovery succeeds) to very large (arm recovery fails). The robot sensed the impact magnitude using a simple low-pass filter on gyroscope data. The filtered body angular velocity was computed as $\dot{\alpha}_k^{\text{filt}} = (1-\beta)\dot{\alpha}_{k-1}^{\text{filt}} + \beta\dot{\alpha}_k$, where $\beta = 0.3$. If at time step k, the absolute filtered body angular velocity decreased, $|\dot{\alpha}_{k-1}^{\text{filt}}| > |\dot{\alpha}_k^{\text{filt}}|$, and had magnitude greater than 1.0 rad/s, an impact of magnitude

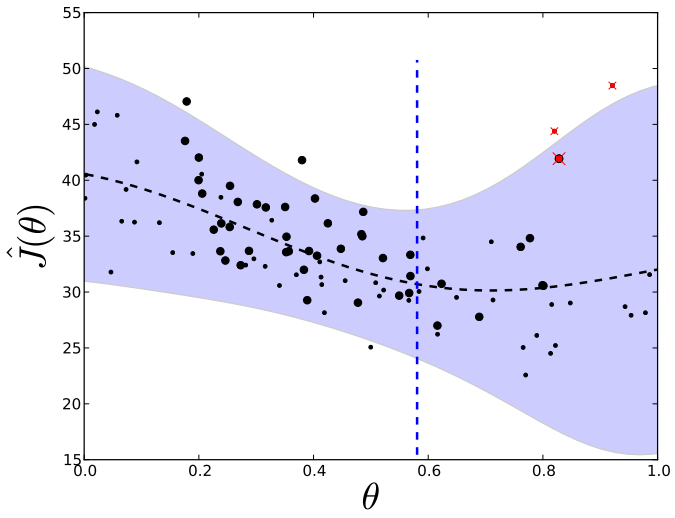


Figure 6.4. The cost distribution for bracing fit using 97 data points: 45 from the learning sequence (bold) and 52 from randomly selected policies. The vertical blue line indicates the final policy after 45 episodes of risk-averse ($\kappa = 2$) gradient descent. The red points indicate hardware failures.

Figure 6.5. Bracing policy execution after a large impact perturbation. Total duration of the above sequence is 0.7 seconds.

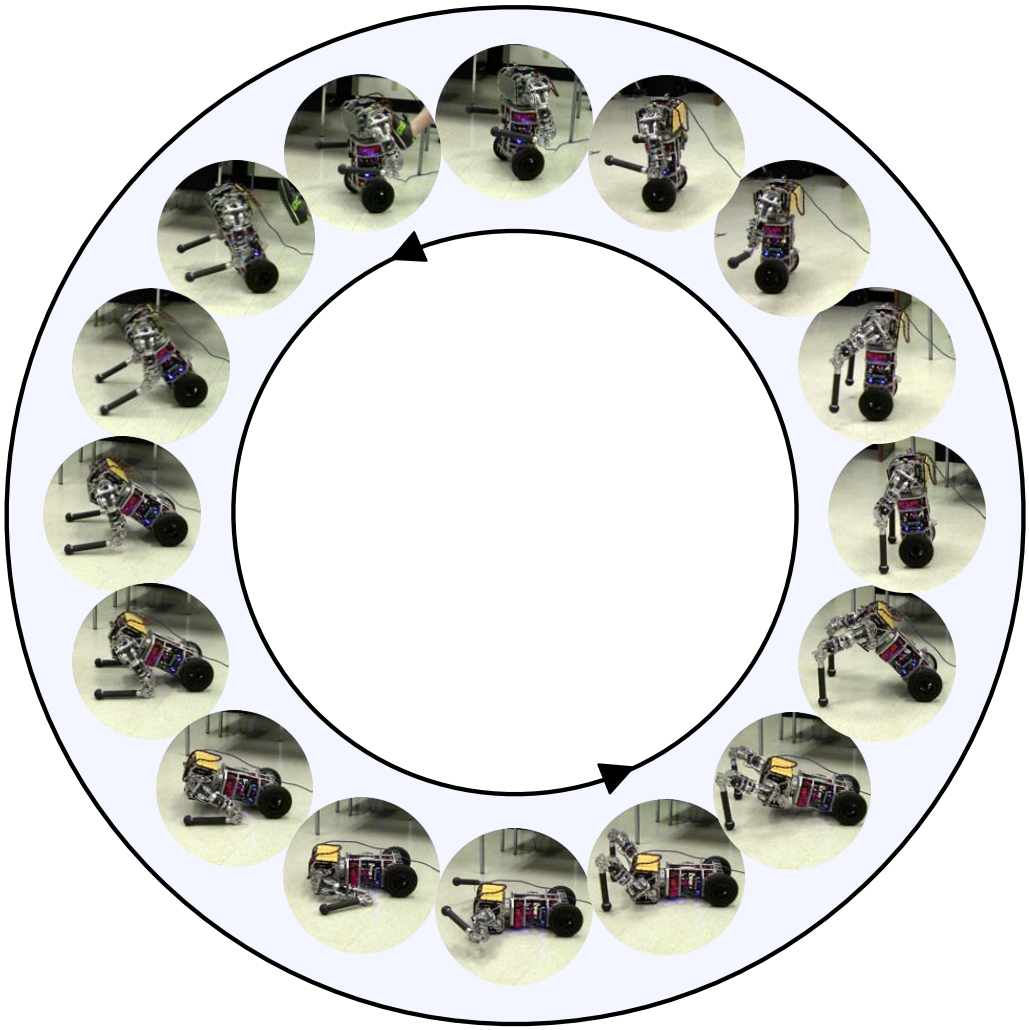


Figure 6.6. The recovery sequence executed in response to a human kicking the robot. The uBot detects the large impact and initiates the bracing controller. When the robot comes to rest, the arms are repositioned and a closed-loop push-up controller developed in our prior work [63] is used to return the robot to the near vertical position. From this position, the LQR controller is engaged and the arms are repositioned.

$\mathrm{I} = |\dot{\alpha}_{k-1}^{\mathrm{filt}}|$ was inferred. The inferred magnitudes ranged from 1.24 to 1.97 in the trials performed.

The probability of selecting the bracing policy was defined to be

$$p(\text{brace} \mid \mathrm{I}, \theta) = \frac{1}{1 + e^{-50(\mathrm{I}-\theta)}}, \tag{6.2}$$

where the parameter θ effectively defines the threshold impact magnitude for bracing. If θ is set to a low value, the robot will brace after most impacts. Alternatively, if θ is set to a high value, the robot will almost always try to recover balance, which may or may not succeed depending on the particular impact. The cost was computed for each trial as,

$$J(\theta) = h(\mathbf{x}(T)) + \int_0^T (0.005\ddot{\alpha}(t)^2 + 5I(t)V(t))dt, \tag{6.3}$$

where $T = 3.5$ sec and $h(\mathbf{x}(T))$ captured the cost of having to perform a subsequent push-up: $h(\mathbf{x}(T) = 10$ if the robot braced or failed to recover and $h(\mathbf{x}(T)) = 0$ otherwise. Under this cost function, balance recovery events using the learned arm motions yielded the lowest average cost because they produced low body accelerations, used little energy, and did not require a subsequent push-up to recover balance. Bracing had comparatively higher cost because significant energy was used by the arms to reduce the body acceleration when coming into contact with the ground. Failing to recover yielded the highest average cost because very high body accelerations were recorded and, like the bracing maneuver, a push-up was required to return to the balancing configuration. Note that once balance recovery was selected, successful bracing was no longer possible due to arm actuator velocity limitations (although failures to recover were reliably detected and bracing was attempted in each case).

Figure 6.7 shows data collected from 50 trials where θ was selected uniformly at random. The VHGP model was fit to the data and confidence bound selection

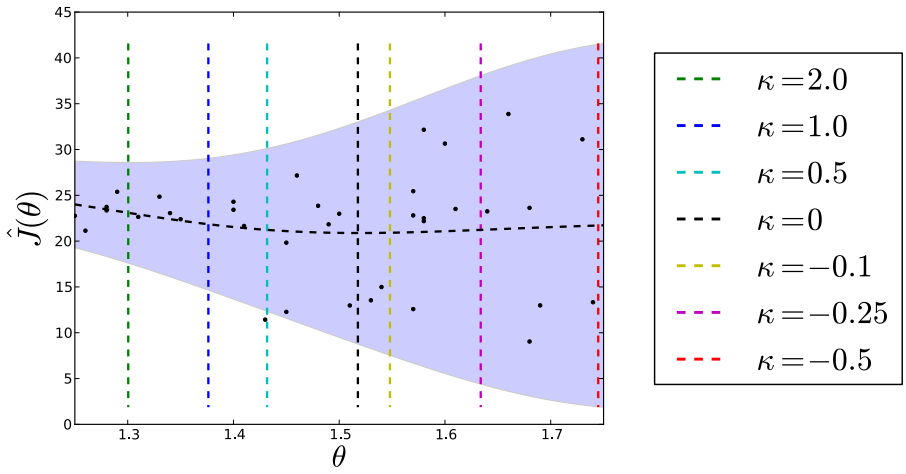


Figure 6.7. Data collected in policy switching experiments are used to construct a cost model and perform subsequent risk-sensitive selection. As κ is increased, the robot becomes increasingly risk-averse by bracing for most impacts. As κ is decreased, the robot becomes increasingly risk-seeking by attempting to recover balance for most impacts.

was performed using a range of risk factors. The results correspond strongly with intuition. When $\kappa = 2$, the robot tends toward risk-aversion by bracing for even small impacts. This strategy is quite predictable, but it is very conservative since bracing is performed in some cases where it would otherwise be able to recover balance. On the other hand, when $\kappa = -0.5$ the robot attempts arm recovery in most cases. This leads to stabilization in all cases where stabilization is possible, but it also produces dangerous failures when the robot is unable to recover balance. The risk-neutral policy ($\kappa = 0$) is near what appears to be the recovery limit.

In this case, the switching threshold was positively related to risk: increasing κ decreased θ (nonlinearly), and vice versa. Thus, the learned cost model is used to reparameterize the switching behavior from a somewhat obscure threshold on the magnitude of body angular velocity, θ, to a risk factor, κ, that specifies the system's sensitivity to the standard deviation of a known cost function. In this simple example,

the positive relationship between these quantities could have probably been guessed by a programmer with sufficient experience with the robot under these experimental conditions. But the mapping between risk and the parameters of the arm recovery or dynamic lifting policies learned in Chapters 4 and 5 would have been much more difficult to predict.

6.5 Discussion

The ability to perform safe, reliable bracing maneuvers and use arm motions to help stabilize after large impact perturbations significantly improves the robustness and deployability of the uBot-5. The approach taken to develop these behaviors in this work was to optimize the responses of parameterized policies to particular impact disturbances. This is, of course, not the only, or even the most general, way to solve these problems. For example, a dynamic model of the system could have been learned via system identification and model-based techniques could have been applied to produce arm motions that, e.g., attempt to control the body angular momentum in a particular way. However, it is interesting that these rather complicated dynamic control tasks can be solved using a very general process of formulating and solving stochastic optimization problems. From a practical perspective, it is also interesting that a small set of learned policies can be used to respond to a wide range of impact perturbations and that the learned solutions can be adapted to reflect different sensitivities toward risk.

Nevertheless, the bracing and recovery controllers developed in this work are necessarily limited. In particular, they are designed to recover from unpredicted rear impact perturbation on flat terrain. Extending this set of controllers to enable the robot to respond to front impacts would be straightforward, but the behaviors would likely differ qualitatively because the arm initial conditions are not symmetric across the coronal plane. Addressing other common types of perturbations, such as tripping

and slipping, is an interesting direction for future work. The bracing strategy developed in this work would likely translate to these cases, however whole-body balance recovery strategies may be more difficult since the perturbation directly affects the motion of the wheels. Preliminary experiments suggest that maintaining stability using only the arms is probably infeasible, so in these cases bracing may be frequently used.

Policies that directly adjust the impedance of the robot will be useful to respond to long-duration or anticipated contact perturbations. It is likely that anticipatory actions, such as leaning into an impact, will improve recovery performance significantly. Another interesting possibility is exploiting noncoplanar environmental surfaces for bracing. This problem would likely require some knowledge of the robot dynamics to compute suitable bracing configurations for a given surface. Risk sensitivity might play an interesting role in this problem since, e.g., strategies will depend on uncertain estimates of state, surface orientation, and friction coefficients. Developing methods for autonomously setting the system's runtime risk sensitivity for different recovery scenarios is an important direction for future work.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

The goal of this thesis was to develop new tools for performing risk-sensitive optimization and evaluate their utility in a range of dynamic robot control tasks. In pursuit of this goal, two new stochastic optimization algorithms were derived. The first algorithm, called Variational Bayesian Optimization (VBO), is an extension of Bayesian optimization methods that can be used to perform global policy search with respect to a variety of risk-sensitive optimization criteria. The second algorithm, called Risk-Sensitive Stochastic Gradient Descent (RSSGD), is related to VBO through its use of a learned cost model, but instead of performing global policy selection, RSSGD uses the cost model as a local critic to perform risk-sensitive gradient descent.

Several experiments with the uBot-5 were described in which dynamic stability, recovery, and manipulation controllers were learned using the algorithms presented in this thesis. In addition to providing examples of efficiently learned dynamic behaviors, these experiments highlighted the important role that risk can play in dynamic robot control. However, the algorithms and experiments presented in this thesis are necessarily limited and there is much that remains for future work.

7.1 Future Work

There are several exciting and promising opportunities for future work both extending the methods presented in this thesis, and developing new kinds of risk-sensitive policy search algorithms. The VBO and RSSGD algorithms are general

risk-sensitive stochastic optimization methods that can be applied to the problem of policy search. However, more domain-driven policy search algorithms are also conceivable. For example, Mihatsch and Neuneier's risk-sensitive TD approach [76] could potentially be applied to build new types of actor-critic algorithms that use biased value function estimates to perform local risk-sensitive policy search. Such approaches would likely be able to handle larger dimensional policy spaces, but the ability to rapidly change risk-sensitivity would be limited. The method by which risk-sensitivity would be specified in this case (effectively a ratio of step size parameters) is perhaps not as intuitive as the confidence bound criteria considered in this work, but the possibility of extending methods such as natural actor-critic to the risk-sensitive case is very exciting.

One straightforward way to extend the VBO algorithm would be to consider different policy selection criteria. In particular, multi-step methods that select a sequence of n policy parameters could be valuable in systems with fixed experimental budgets. Osborne et al. [88, 30] have proposed a multi-step criterion in the standard Bayesian optimization setting that has produced promising results. Other risk-sensitive global optimization algorithms could also be conceived by using other methods to build the heteroscedastic cost model [125, 108, 45, 138]. It would be interesting to see if different properties arise that make certain methods more appropriate for particular problem domains. Methods for capturing multimodality of the cost distribution would also be interesting to consider, especially in domains where unobservable differences in initial conditions can lead to qualitatively different outcomes.

The way in which the local cost model was used as a critic in the RSSGD algorithm was somewhat limited. There are several possibilities for improvements. For example, some work has shown that adjusting the covariance of the perturbation distribution while learning can produce better performance [100]. This idea is related to the covariance matrix adaptation that is done in some cost weighted averaging meth-

ods [113]. An interesting direction for future work would be to use the learned local model to adjust the sampling distribution by, e.g., scaling the perturbation covariance by the optimized length-scale hyperparameters of the VHGP model. In this way, parameters would be perturbed based on the inferred relative sensitivity of the cost to changes in each parameter value. Methods for using gradient estimates from the local critic to update the policy parameters or, conversely, using gradient observations to update the critic could also be explored.

Two open problems that were not explicitly addressed in this thesis are methods for setting the risk factor, κ, and selecting the policy representation, $\pi_{\boldsymbol{\theta}}$. These are, of course, extremely important problems that strongly effect the outcome of the optimization. Methods for selecting κ are likely to be context-specific, where the system's risk level might depend on fast-changing quantities such as battery charge, motor temperature, the presence of dangerous obstacles, etc. The VBO algorithm is can act on changes in risk sensitivity, but equivalently responsive methods for specifying risk factors based on state are also needed. In particular, it will be necessary to devise ways to map features of the robot's state and environment to a common cost currency to determine appropriate dispositions toward risk. In natural systems, mechanisms for doing this clearly exist, but the rules by which they operate are often elusive [10].

The policy search experiments described in this thesis have involved optimization of simple open-loop policies. The reason for this twofold. First, it is often easy for the robot programmer to predict the types of motions that are likely to succeed and hence identify suitable trajectory-based representations, such as cubic splines. The second reason is that closed-loop representations can often be difficult to apply successfully in policy search, especially in weakly-stable systems, because small changes in policy parameter values can give way to large changes in cost for some regions of parameter space. This problem was investigated in detail by Roberts et al. [98]. It would be

interesting to see if global methods like VBO fair better than gradient-based methods in these cases, although implementations that can handle nonstationarity in the cost distribution would likely be needed. A very interesting open problem is the automatic identification of suitable policy representations based on demonstrated or planned solutions. Dimensionality reduction techniques developed by the machine learning community may play a significant role in solving this problem.

7.2 Conclusions

It is well known that risk plays a central role in a wide variety of decision processes, from portfolio investments [66, 129, 86] to food source selection [10]. Recent studies suggest that risk-sensitivity may also be a fundamental component of human motor control [20, 83, 139]. However, unlike the decision sciences, work in stochastic optimal control and reinforcement learning has placed less emphasis on risk, focusing instead on developing methods for maximizing average performance.

In this thesis, I presented new approaches for performing efficient risk-sensitive optimization of noisy cost functions. These algorithms are quite general in that they assume little about the structure of the optimization problem. When applied to policy search, they are capable of handling high-dimensional continuous state and action spaces with unknown dynamics, significant stochasticity, and non-additive cost functions. However, as a consequence of this generality, these approaches require low-dimensional policy representations and careful consideration of the properties of the cost distribution. Nevertheless, these methods are relevant to state-of-the-art control development in robotics because low-dimensional policies can exist for even very challenging control problems.

I evaluated the algorithms in several dynamic control tasks with the uBot-5. These experiments involved learning rapid arm responses for stabilizing after large impact perturbations, learning dynamic heavy-lifting strategies while balancing, and devel-

oping safe and reliable fall bracing behaviors to respond to destabilizing impacts. In addition to serving as unique contributions to the robot control literature, these results provide initial evidence that variable risk control may be important for developing high-performance and reliable robot systems. However, these results constitute only a very small step toward the greater goal of developing general methods for autonomous dynamic behavior generation in robot systems.

The role that risk-sensitive optimization will ultimately play in the development of robots capable of control feats like those we observe in nature is still unclear. At this point, there is good reason to suspect that risk will be important, but there is much work that lies ahead. I hope that the tools and experiments described in this thesis offer some value to those researchers that will inevitably develop new insights that lead us closer to the goal toward which we strive.

BIBLIOGRAPHY

[1] Abramowitz, M., and Stegun, I. A., Eds. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing.* Dover, New York, 1972.

[2] Allum, J. H. J., Carpenter, M. G., Honegger, F., Adkin, A. L., and Bloem, B. R. Age-dependent variations in the directional sensitivity of balance corrections and compensatory arm movements in man. *The Journal of Physiology 542*, 2 (2002), 643–663.

[3] Amari, Shun-ichi. Natural gradient works efficiently in learning. *Neural Computation 10*, 2 (1998), 251–276.

[4] Ambrose, Robert O., Savely, Robert T., Goza, S. Michael, Strawser, Philip, Diftler, Myron A., Spain, Ivan, and Radford, Nicolaus. Mobile manipulation using NASA's robonaut. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (2004), pp. 2104–2109.

[5] Argall, Brenna D., Chernova, Sonia, Veloso, Manuela, and Browning, Brett. A survey of robot learning from demonstration. *Robotics and Autonomous Systems 57*, 5 (May 2009), 469–483.

[6] Atkeson, Christopher G., and Stephens, Benjamin. Multiple balance strategies from one optimization criterion. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (December 2007), pp. 57–64.

[7] Bagnell, J. Andrew, and Schneider, Jeff. Covariant policy search. Tech. Rep. 81, Robotics Institute, 2003.

[8] Baird, Leemon C., and Klopf, Harry. Reinforcement learning with high-dimensional continuous actions. Tech. Rep. WL-TR-93-1147, Wright Laboratory, Wright-Patterson Air Force Base, OH 45433-7301, 1993.

[9] Barto, Andrew G., Sutton, Richard S., and Anderson, Charles W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics 13*, 5 (1983), 835–846.

[10] Bateson, Melissa. Recent advances in our understanding of risk-sensitive foraging preferences. *Proceedings of the Nutrition Society 61* (2002), 1–8.

[11] Baxter, Jonathan, and Bartlett, Peter L. Reinforcement learning in pomdps via direct gradient ascent. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)* (2000).

[12] Bellman, Richard. *Dynamic Programming*. Dover, 1957.

[13] Benbrahim, Hamid, and Franklin, Judy A. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems 22*, 3-4 (Dec. 1997), 283–302.

[14] Bertsekas, Dimitri P. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[15] Bertsekas, Dimitri P., and Tsitsiklis, John N. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

[16] Bertsekas, Dimitri P., and Tsitsiklis, John N. Gradient convergence in gradient methods with errors. *SIAM J. Optim. 10*, 3 (2000), 627–642.

[17] Betts, John T. *Practical methods for optimal control using nonlinear programming*, vol. 3 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[18] Borkar, V. S. Q-learning for risk-sensitive control. *Mathematics of Operations Research 27*, 2 (May 2002), 294–311.

[19] Boyan, Justin A. Least-squares temporal difference learning. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)* (June 1999), Ivan Bratko and Saso Dzeroski, Eds., Morgan Kaufmann.

[20] Braun, Daniel A., Nagengast, Arne J., and Wolpert, Daniel M. Risk-sensitivity in sensorimotor control. *Frontiers in Human Neuroscience 5* (January 2011), 1–10.

[21] Brochu, Eric, Cora, Vlad, and de Freitas, Nando. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Tech. Rep. TR-2009-023, University of British Columbia, Department of Computer Science, 2009.

[22] Bull, Adam D. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research 12* (October 2011), 2879–2904.

[23] Cox, Dennis D., and John, Susan. A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on* (1992), vol. 2, pp. 1241–1246.

[24] Craig, John J. *Introduction to Robotics: Mechanics and Control*, 3rd ed. Pearson Prentice Hall, 2005.

[25] Dearden, Richard, Friedman, Nir, and Russell, Stuart. Bayesian Q-learning. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (1998), pp. 761–768.

[26] Deegan, Patrick. *Whole-Body Strategies for Mobility and Manipulation*. PhD thesis, University of Massachusetts Amherst, 2010.

[27] Deegan, Patrick, Grupen, Roderic, Hanson, Allen, Horrell, Emily, Ou, Shichao, Riseman, Edward, Sen, Shiraj, Thibodeau, Bryan, Williams, Adam, and Xie, Dan. Mobile manipulators for assisted living in residential settings. *Autonomous Robots 24*, 2 (2007), 179–192.

[28] Deisenroth, Marc Peter, and Rasmussen, Carl Edward. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning* (Bellevue, WA, 2011).

[29] Frean, Marcus, and Boyle, Phillip. Using Gaussian processes to optimize expensive functions. In *AI 2008: Advances in Artificial Intelligence* (2008), pp. 258–267.

[30] Garnett, R., Osborne, M., and Roberts, S. Bayesian optimization for sensor set selection. In *In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks* (2010), ACM, pp. 209–219.

[31] Goldberg, Paul W., Williams, Christopher K. I., and Bishop, Christopher M. Regression with input-dependent noise: A Gaussian process treatment. In *Advances in Neural Information Processing Systems 10 (NIPS)* (1998), pp. 493–499.

[32] Gullapalli, V., Franklin, J.A., and Benbrahim, H. Acquiring robot skills via reinforcement learning. *Control Systems Magazine, IEEE 14*, 1 (February 1994), 13–24.

[33] Hannigan, Edward. Endpoint force sensing for mobile manipulators. Master's thesis, UMass Amherst, Sept. 2008.

[34] Hasson, Christopher J., Emmerik, Richard E.A. Van, and Caldwell, Graham E. Predicting dynamic postural instability using center of mass time-to-contact information. *Journal of Biomechanics 41*, 10 (2008), 2121–2129.

[35] Heger, Matthias. Consideration of risk in reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ICML)* (1994), pp. 105–111.

[36] Howard, Ronald A., and Matheson, James E. Risk-sensitive markov decision processes. *Management Science 18*, 2 (March 1972), 356–369.

[37] Jabri, Marwan, and Flower, Barry. Weight perturbation: An optimal architecture and learning technique for analog vlsi feedforward and recurrent multi-layer networks. *IEEE Transactions on Neural Networks 3* (1992), 154–157.

[38] Jacobson, David. Optimal stochastic linear systems with exponential performance criteria and their relationship to deterministic differential games. *IEEE Transactions on Automatic Control 18*, 2 (April 1973), 124–131.

[39] Johns, Jeff. *Basis Construction and Utilization for Markov Decision Processes using Graphs*. PhD thesis, University of Massachusetts Amherst, 2010.

[40] Johnson, Steven G. The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt, 2011.

[41] Jones, Donald R. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization 21* (2001), 345–383.

[42] Jung, Hee-Tae, Baird, Jennifer, Choe, Yu-Kyong, and Grupen, Roderic A. Upper extremity physical therapy for stroke patients using a general purpose robot. In *Proceedings of the 20th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (Atlanta, GA, August 2011).

[43] Kacelnik, Alex, and Bateson, Melissa. Risky theories—the effects of variance on foraging decisions. *Amer. Zool. 36* (1996), 402–434.

[44] Kakade, Sham. A natural policy gradient. In *Advances in Neural Information Processing Systems 14 (NIPS)* (2002).

[45] Kersting, Kristian, Plagemann, Christian, Pfaff, Patrick, and Burgard, Wolfram. Most likely heteroscedastic Gaussian process regression. In *Proceedings of the International Conference on Machine Learning (ICML)* (2010), pp. 393–400.

[46] Kober, Jens, and Peters, Jan. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems 21* (2009), MIT Press.

[47] Kohl, Nate, and Stone, Peter. Machine learning for fast quadrupedal locomotion. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence* (July 2004), pp. 611–616.

[48] Kolhe, Pushkar, Dantam, Neil, and Stilman, Mike. Dynamic pushing strategies for dynamically stable mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation* (May 2010).

[49] Kolter, J. Zico, and Ng, Andrew Y. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)* (2009).

[50] Kolter, J. Zico, and Ng, Andrew Y. Policy search via the signed derivative. In *Robotics: Science and Systems V (RSS)* (2010).

[51] Konda, Vijay R., and Tsitsiklis, John N. On actor-critic algorithms. *SIAM J. Control Optim. 42*, 4 (2003), 1143–1166.

[52] Konidaris, George, Kuindersma, Scott, Barto, Andrew, and Grupen, Roderic. Constructing skill trees for reinforcement learning agents from demonstration trajectories. In *Advances in Neural Information Processing Systems 23* (December 2010), pp. 1162–1170.

[53] Konidaris, George, Kuindersma, Scott, Grupen, Roderic, and Barto, Andrew. Autonomous skill acquisition on a mobile manipulator. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11)* (San Francisco, CA, August 2011), pp. 1468–1473.

[54] Konidaris, George, Kuindersma, Scott, Grupen, Roderic, and Barto, Andrew. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research 31*, 3 (March 2012), 360–375.

[55] Konidaris, George D., Osentoski, Sarah, and Thomas, Philip. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI '11)* (August 2011).

[56] Kudoh, Shunsuke, Komura, Taku, and Ikeuchi, Katsushi. The dynamic postural adjustment with the quadratic programming method. In *International Conference on Intelligent Robots and Systems (IROS)* (October 2002), pp. 2563–2568.

[57] Kudoh, Shunsuke, Komura, Taku, and Ikeuchi, Katsushi. Stepping motion for a human-like character to maintain balance against large perturbations. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (Orlando, FL, May 2006).

[58] Kuindersma, Scott. Control model learning for whole-body mobile manipulation. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI-10)* (Atlanta, GA, July 2010), pp. 1939–1940.

[59] Kuindersma, Scott, Grupen, Roderic, and Barto, Andrew. Learning dynamic arm motions for postural recovery. In *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots* (Bled, Slovenia, October 2011), pp. 7–12.

[60] Kuindersma, Scott, Grupen, Roderic, and Barto, Andrew. Variable risk dynamic mobile manipulation. In *RSS 2012 Workshop on Mobile Manipulation* (Sydney, Australia, 2012).

[61] Kuindersma, Scott, Grupen, Roderic, and Barto, Andrew. Variational Bayesian optimization for runtime risk-sensitive control. In *Robotics: Science and Systems VIII (RSS)* (Sydney, Australia, July 2012).

[62] Kuindersma, Scott, Konidaris, George, Grupen, Roderic, and Barto, Andrew. Learning from a single demonstration: Motion planning with skill segmentation. In *NIPS Workshop on Learning and Planning from Batch Time Series Data* (Vancouver, BC, December 2010).

[63] Kuindersma, Scott R., Hannigan, Edward, Ruiken, Dirk, and Grupen, Roderic A. Dexterous mobility with the uBot-5 mobile manipulator. In *Proceedings of the 14th International Conference on Advanced Robotics* (Munich, Germany, June 2009).

[64] Kushner, Harold J. A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. *J. Basic Engineering 86* (1964), 97–106.

[65] Lázaro-Gredilla, Miguel, and Titsias, Michalis K. Variational heteroscedastic Gaussian process regression. In *Proceedings of the International Conference on Machine Learning (ICML)* (2011).

[66] Levy, H., and Markowitz, H. M. Approximating expected utility by a function of mean and variance. *The American Economic Review 69*, 3 (June 1979), 308–317.

[67] Lizotte, Daniel, Wang, Tao, Bowling, Michael, and Schuurmans, Dale. Automatic gait optimization with Gaussian process regression. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)* (2007).

[68] Lizotte, Daniel James. *Practical Bayesian Optimization*. PhD thesis, University of Alberta, Edmonton, Alberta, 2008.

[69] Macchietto, Adriano, Zordan, Victor, and Shelton, Christian R. Momentum control for balance. In *Transactions on Graphics/ACM SIGGRAPH* (2009).

[70] Mahadevan, Sridhar, Maggioni, Mauro, Ferguson, Kimberly, and Osentoski, Sarah. Learning representation and control in continuous markov decision processes. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)* (July 2006), AAAI Press.

[71] Mannor, Shie, Rubinstein, Reuven, and Gat, Yohai. The cross entropy method for fast policy search. In *Proceedings of the 20th International Conference on Machine Learning (ICML)* (Washington, DC, 2003).

[72] Marigold, Daniel S., Bethune, Allison J., and Patla, Aftab E. Role of the unperturbed limb and arms in the reactive recovery response to an unexpected slip during locomotion. *J Neurophysiol 89* (2003), 1727–1737.

[73] Martinez-Cantin, Ruben, de Freitas, Nando, Brochu, Eric, Castellanos, José A., and Doucet, Arnaud. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots 27* (2009), 93–103.

[74] McClung, III, Arthur J., Zheng, Ying, and Morrell, John B. Contact feature extraction on a balancing manipulation platform. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (Anchorage, Alaska, May 2010).

[75] McIlroy, William E., and Maki, Brian E. Early activation of arm muscles follows external perturbation of upright stance. *Neuroscience Letters 184*, 3 (1995), 177–180.

[76] Mihatsch, Oliver, and Neuneier, Ralph. Risk-sensitive reinforcement learning. *Machine Learning 49* (2002), 267–290.

[77] Misiaszek, John. Early activation of arm and leg muscles following pulls to the waist during walking. *Experimental Brain Research 151*, 3 (2003), 318–329.

[78] Misiaszek, John E., and Krauss, Emily M. Restricting arm use enhances compensatory reactions of leg muscles during walking. *Experimental Brain Research 161*, 4 (2005), 474–485.

[79] Mitsunaga, Noriaki, Smith, Christian, Kanda, Takayuki, Ishiguro, Hiroshi, and Hagita, Norihiro. Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2005), pp. 1594–1601.

[80] Morimura, Tetsuro, Sugiyama, Masashi, Kashima, Hisashi, and Hachiya, Hirotaka. Nonparametric return distribution approximiation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML)* (2010).

[81] Morimura, Tetsuro, Sugiyama, Masashi, Kashima, Hisashi, Hachiya, Hirotaka, and Tanaka, Toshiyuki. Parametric return density estimation for reinforcement learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)* (2010).

[82] Močkus, J., Tiesis, V., and Žilinskas, A. The application of Bayesian methods for seeking the extremum. In *Toward Global Optimization*, vol. 2. Elsevier, 1978, pp. 117–128.

[83] Nagengast, Arne J., Braun, Daniel A., and Wolpert, Daniel M. Risk-sensitive optimal feedback control accounts for sensorimotor behavior under uncertainty. *PLos Comput Biol 6*, 7 (2010), 1–15.

[84] Nagengast, Arne J., Braun, Daniel A., and Wolpert, Daniel M. Risk-sensitivity and the mean-variance trade-off: decision making in sensorimotor control. *Proc. R. Soc. B* (2010).

[85] Nakada, Masaki, Allen, Brian F., Morishima, Shigeo, and Terzopoulos, Demetri. Learning arm motion strategies to recover balance in bipedal robots. In *Intl Symposium on Learning and Adaptive Behavior in Robotic Systems* (Canterbury, UK, September 2010).

[86] Niv, Yael, Edlund, Jeffrey A., Dayan, Peter, and O'Doherty, John P. Neural prediction errors reveal a risk-sensitive reinforcement-learning process in the human brain. *Journal of Neuroscience 32*, 2 (January 2012), 551–562.

[87] Niv, Yael, Joel, Daphna, Meilijson, Isaac, and Ruppin, Eytan. Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior 10*, 1 (2002), 5–24.

[88] Osborne, M. A., Garnett, R., and Roberts, S. J. Gaussian processes for global optimization. In *Third International Conference on Learning and Intelligent Optimization (LION3)* (Trento, Italy, 2009).

[89] Ou, Shichao. *A Behavioral Approach to Human-Robot Communication*. PhD thesis, University of Massachusetts Amherst, February 2010.

[90] Peters, Jan, and Schaal, Stefan. Policy gradient methods for robotics. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (2006), pp. 2219–2225.

[91] Peters, Jan, and Schaal, Stefan. Natural actor-critic. *Neurocomputing 71*, 7-9 (2008), 1180–1190.

[92] Peters, Jan, and Schaal, Stefan. Reinforcement learning of motor skills with policy gradients. *Neural Networks 21*, 4 (2008), 682–697.

[93] Pijnappels, Mirjam, Kingma, Idsart, Wezenberg, Daphne, Reurink, Guus, and van Dieën, Jaap H. Armed against falls: the contribution of arm movements to balance recovery after tripping. *Experimental Brain Research 201* (2010).

[94] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mischenko, E. F. The mathematical theory of optimal processes. In *International Conference on Interaction Sciences* (1962).

[95] Pratt, Jerry, Carff, John, Drakunov, Sergey, and Goswami, Ambarish. Capture point: A step toward humanoid push recovery. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (2006), pp. 200–207.

[96] Preuschoff, Kerstin, Quartz, Steven R., and Bossaerts, Peter. Human insula activation reflects risk prediction errors as well as risk. *Journal of Neuroscience 28*, 11 (March 2008), 2745–2752.

[97] Rasmussen, Carl Edward, and Williams, Christopher K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[98] Roberts, John, Manchester, Ian, and Tedrake, Russ. Feedback controller parameterizations for reinforcement learning. In *Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)* (2011).

[99] Roberts, John W., Moret, Lionel, Zhang, Jun, and Tedrake, Russ. Motor learning at intermediate Reynolds number: experiments with policy gradient on the flapping flight of a rigid wing. In *From Motor to Interaction Learning in Robots*, Olivier Sigaud and Jan Peters, Eds., vol. 264 of *Studies in Computational Intelligence*. Springer, 2010, pp. 293–309.

[100] Roberts, John W., and Tedrake, Russ. Signal-to-noise ratio analysis of policy gradient algorithms. In *Advances of Neural Information Processing Systems 21 (NIPS)* (2009).

[101] Roos, Paulien E., McGuigan, M. Polly, Kerwin, David G., and Trewartha, Grant. The role of arm movement in early trip recovery in younger and older adults. *Gait & Posture 27* (2008), 352–356.

[102] Rosenstein, Michael T. *Learning to Exploit Dynamics for Robot Motor Coordination*. PhD thesis, University of Massachusetts Amherst, 2003.

[103] Rosenstein, Michael T., and Barto, Andrew G. Robot weightlifting by direct policy search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (2001).

[104] Rummery, G. A., and Niranjan, M. On-line Q-learning using connectionist systems. Tech. rep., Cambridge University Engineering Department, 1994.

[105] Schonlau, Matthias, Welch, William J., and Jones, Donald R. Global versus local search in constrained optimization of computer models. In *New Developments and Applications in Experimental Design*, Nancy Flournoy, William F. Rosenberger, and Weng Kee Wong, Eds., vol. 34 of *Lecture Notes - Monograph Series*. IMS, 1998, pp. 11–25.

[106] Sharpe, William F. Mutual fund performance. *Journal of Business 39*, S1 (1966), 119–138.

[107] Shiratori, Takaaki, Coley, Brooke, Cham, Rakié, and Hodgins, Jessica K. Simulating balance recovery responses to trips based on biomechanical principles. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (August 2009).

[108] Snelson, Edward, and Ghahramani, Zoubin. Variable noise and dimensionality reduction for sparse Gaussian processes. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence* (Cambridge, MA, 2006).

[109] Srinivas, Niranjan, Krause, Andreas, Kakade, Sham, and Seeger, Matthias. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)* (2010).

[110] Stengel, Robert F. *Optimal Control and Estimation*. Dover, 1994.

[111] Stephens, Benjamin, and Atkeson, Christopher. Push recovery by stepping for humanoid robots with force controlled joints. In *Proceedings of the International Conference on Humanoid Robots* (Nashville, TN, 2010).

[112] Stilman, Mike, Olson, Jon, and Gloss, William. Golem krang: Dynamically stable humanoid robot for mobile manipulation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (2010).

[113] Stulp, Freek, and Sigaud, Olivier. Path integral policy improvement with covariance matrix adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)* (Edinburgh, Scotland, 2012).

[114] Sutton, Richard S. Learning to predict by the methods of temporal differences. *Machine Learning 3* (1988), 9–44.

[115] Sutton, Richard S., and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[116] Sutton, Richard S., McAllester, David, Singh, Satinder, and Mansour, Yishay. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)* (2000), vol. 12, pp. 1057–1063.

[117] Svanberg, Krister. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J. Optim 12*, 2 (2002), 555–573.

[118] Talbot, Lee M., and Talbot, Martha H. The wildebeest in Western Masailand, east africa. *Wildlife Monographs 12* (September 1963), 3–88.

[119] Tamar, Aviv, Castro, Dotan Di, and Mannor, Shie. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning (ICML)* (Edinburgh, Scotland, June 2012).

[120] Tedrake, Russ. *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines: Course Notes for MIT 6.832*, working draft ed. 2010.

[121] Tedrake, Russ, Zhang, Teresa Weirui, and Seung, H. Sebastian. Stochastic policy gradient reinforcement learning on a simple 3D biped. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (Sendai, Japan, September 2004), vol. 3, pp. 2849–2854.

[122] Tesch, Matthew, Schneider, Jeff, and Choset, Howie. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA, 2011).

[123] Theodorou, Evangelos A., Buchli, Jonas, and Schaal, Stefan. A generalized path integral control approach to reinforcement learning. *Journal of Machine Learning Research* (2010), 3137–3181.

[124] Thibodeau, Bryan J., Deegan, Patrick, and Grupen, Roderic A. Static analysis of contact forces with a mobile manipulator. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (2006), IEEE, pp. 4007–4012.

[125] Tibshirani, Robert, and Hastie, Trevor. Local likelihood estimation. *Journal of the American Statistical Association 82*, 398 (June 1987), 559–567.

[126] Tobler, Philippe N., O'Doherty, John P., Dolan, Raymond J., and Schultz, Wolfram. Reward value coding distinct from risk attitude-related uncertainty coding in human reward systems. *J Neurophysiol 97* (2007), 1621–1632.

[127] Todorov, Emanuel. Optimal control theory. In *Bayesian Brain*, Kenji Doya, Ed. MIT Press, 2006.

[128] Troy, Karen L., Donovan, Stephanie J., and Grabiner, Mark D. Theoretical contribution of the upper extremities to reducing trunk extension following a laboratory-induced slip. *Journal of Biomechanics 42* (2009), 1339–1344.

[129] Tversky, A., and Kahneman, D. Advances in prospect theory–cumulative representation of uncertainty. *J. Risk Uncertain 5* (1992), 297–323.

[130] van den Broek, Bart, Wiegerinck, Wim, and Kappen, Bert. Risk sensitive path integral control. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)* (2010), pp. 615–622.

[131] Vazquez, Emmanuel, and Bect, Julien. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planing and Inference 140*, 11 (2010), 3088–3095.

[132] Watkins, C. J. C. H. *Learning from Delayed Rewards*. PhD thesis, King's College, 1989.

[133] Werbos, P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[134] Whittle, Peter. Risk-sensitive linear/quadratic/Gaussian control. *Advances in Applied Probability 13* (1981), 764–777.

[135] Whittle, Peter. *Risk-Sensitive Optimal Control*. John Wiley & Sons, 1990.

[136] Williams, Ronald J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning 8* (1992), 229–256.

[137] Wilson, Aaron, Fern, Alan, and Tadepalli, Prasad. A behavior based kernel for policy search via Bayesian optimization. In *Proceedings of the ICML 2011 Workshop: Planning and Acting with Uncertain Model* (Bellevue, WA, 2011).

[138] Wilson, Andrew, and Ghahramani, Zoubin. Generalized Wishart processes. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)* (Barcelona, Spain, July 2011).

[139] Wu, Shih-Wei, Delgado, Mauricio R., and Maloney, Laurence T. Economic decision-making compared with an equivalent motor task. *Proc. Natl. Acad. Sci. USA 106*, 15 (April 2009), 6088–6093.

[140] Xie, Dan, Lin, Yun, Grupen, Roderic, and Hanson, Allen. Intention-based coordination and interface design for human-robot cooperative search. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA, September 2011).

[141] Yoshida, Eiichi, and Laumond, Jean-Paul. Motion planning for humanoid robots: Highlights with HRP-2. *Journées Nationales de la Recherche en Robotique* (October 2007), 9–12.