# Variable Risk Control via Stochastic Optimization

Scott R. Kuindersma[1,2], Roderic A. Grupen[2], and Andrew G. Barto[2]

[1]Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA USA

[2]Department of Computer Science
University of Massachusetts Amherst
Amherst, MA USA

**Abstract**

We present new global and local policy search algorithms suitable for problems with policy-dependent cost variance (or *risk*), a property present in many robot control tasks. These algorithms exploit new techniques in nonparameteric heteroscedastic regression to directly model the policy-dependent distribution of cost. For local search, the learned cost model can be used as a critic 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 relearning. We describe experiments in dynamic stabilization and manipulation with a mobile manipulator that demonstrate learning of flexible, risk-sensitive policies in very few trials.

## 1 Introduction

Experiments on physical robot systems are typically associated with significant practical costs, such as experimenter time, money, and robot wear and tear. However, such experiments are often necessary to refine controllers that have been hand designed or optimized in simulation. This necessity is a result of the extreme difficulty associated with constructing model systems of sufficiently high fidelity that behaviors translate to hardware without performance loss. For many nonlinear systems, it can even be infeasible to perform simulations or construct a reasonable model (Roberts et al., 2010).

For this reason, model-free policy search methods have become one of the standard tools for constructing controllers for robot systems (Rosenstein and Barto, 2001; Kohl and Stone, 2004; Tedrake et al., 2004; Peters and Schaal, 2006; Kolter and Ng, 2010; Theodorou et al., 2010; Lizotte et al., 2007; Kober and Peters, 2009). These algorithms are designed to minimize the expected value of a noisy cost signal, $\hat{J}(\boldsymbol{\theta})$, by adjusting policy parameters, $\boldsymbol{\theta}$, for a fixed class of policies, $\mathbf{u} = \pi_{\boldsymbol{\theta}}(\mathbf{x}, t)$. By considering only the expected cost of a policy and ignoring cost variance, the solutions found by these algorithms are by definition *risk-neutral*, where *risk* corresponds to a monotonic function of the cost variance. However, for systems that operate in a variety of contexts, it can be advantageous to have a more flexible attitude toward risk.

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 or 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.

Recently there has been increased interest in applying Bayesian optimization algorithms to solve model-free policy search problems (Lizotte et al., 2007; Martinez-Cantin et al., 2007, 2009; Wilson et al., 2011; Tesch et al., 2011; Kuindersma et al., 2011). In contrast to well-studied policy gradient methods (Peters and Schaal, 2006), Bayesian optimization algorithms perform policy search by modeling the distribution of cost in policy parameter space and applying a selection criterion to *globally* select the next policy. 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. It is also straightforward 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.

Previous implementations of Bayesian optimization have assumed that the variance of the cost is the same for all policies in the search space. This is not true in general. In this work, we propose a new type of Bayesian optimization algorithm that relaxes this assumption and efficiently captures both the expected cost and cost variance during the optimization. Specifically, we extend recent work developing a variational Gaussian process model for problems with input-dependent noise (or *heteroscedasticity*) (Lázaro-Gredilla and Titsias, 2011) to the optimization case by deriving an expression for expected improvement (Močkus et al., 1978), a commonly used criterion for selecting the next policy, and incorporating log priors into the optimization to improve numerical performance. We also consider the use of confidence bounds to produce *runtime* changes to risk sensitivity and derive a generalized expected risk improvement criterion that balances exploration and exploitation in risk-sensitive setting. Finally, we consider a simple local search procedure that uses the learned cost model as a critic for performing risk-sensitive stochastic gradient descent. We evaluate these algorithms in dynamic stabilization and manipulation experiments with the uBot-5 mobile manipulator.

# 2 Background

## 2.1 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 (Lizotte et al., 2007; Frean and Boyle, 2008; Brochu et al., 2009; Martinez-Cantin et al., 2009; Wilson et al., 2011; Tesch et al., 2011). In describing these algorithms, we 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 (Brochu et al., 2009).

### 2.1.1 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 (Rasmussen and Williams, 2006). In our case the random variable is the cost, $\hat{J}(\boldsymbol{\theta})$, which is indexed by the set of policy parameters. 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,

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

where $\sigma_f^2$ 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 (Rasmussen and Williams, 2006), but in this work we use the squared exponential kernel (1) exclusively.

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

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

Given the GP prior and data,

$$\begin{aligned} \boldsymbol{\Theta} &= [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N]^\top \in \mathbb{R}^{N \times \dim(\boldsymbol{\theta})}, \\ \mathbf{y} &= [\hat{J}(\boldsymbol{\theta}_1), \hat{J}(\boldsymbol{\theta}_2), \ldots, \hat{J}(\boldsymbol{\theta}_N)]^\top \in \mathbb{R}^N, \end{aligned}$$

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

$$\begin{aligned} \mathbb{E}[\hat{J}_*] &= \mathbf{k}_{f*}^\top (\mathbf{K}_f + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \\ s_*^2 &= k_f(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - \mathbf{k}_{f*}^\top (\mathbf{K}_f + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_{f*}, \end{aligned}$$

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, $[\mathbf{K}_f]_{ij} = k_f(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$.

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 (Lizotte et al., 2007). 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 their values before computing the posterior (Rasmussen and Williams, 2006). The log marginal likelihood and its derivatives can be computed in closed form, so we are free to choose from standard nonlinear optimization methods to maximize the marginal log likelihood for model selection.

### 2.1.2 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) (Močkus et al., 1978; Brochu et al., 2009). 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 $\boldsymbol{\theta}_*$ is defined as

$$I_* = \begin{cases} \mu_{\text{best}} - \hat{J}_* & \text{if } \hat{J}_* < \mu_{\text{best}}, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

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

$$\begin{aligned} \text{EI}(\boldsymbol{\theta}_*) &= \int_0^\infty I_* p(I_*) dI_*, \\ &= s_*(u_* \Phi(u_*) + \phi(u_*)), \end{aligned} \tag{4}$$

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 (4) 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 $\xi$ 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 $\xi$. Lizotte et al. (2011) showed that cost scale invariance can be achieved by multiplying $\xi$ by the signal standard deviation, $\sigma_f$. The Bayesian optimization with expected improvement algorithm is shown in Algorithm 1.

**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** $\boldsymbol{\Theta}, \mathbf{y}$

From a theoretical perspective, Vazquez and Bect (2010) 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 (2011) 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 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 as a criterion for selecting the next input (Kushner, 1964). Confidence bound criteria (discussed in Section 3.2) have been extensively studied in the context of global optimization (Cox and John, 1992; Srinivas et al., 2010) and economic decision making (Levy and Markowitz, 1979). Recent work (Osborne et al., 2009; Garnett et al., 2010) 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. (2009).

## 2.2 Variational Heteroscedastic Gaussian Process Regression

One limitation of the standard regression model (2) 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

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \varepsilon_{\boldsymbol{\theta}}, \quad \varepsilon_{\boldsymbol{\theta}} \sim \mathcal{N}(0, r(\boldsymbol{\theta})^2), \tag{5}$$

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

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

is placed over the unknown log variance function, $g(\boldsymbol{\theta}) \equiv \log r(\boldsymbol{\theta})^2$ (Goldberg et al., 1998; Kersting et al., 2010; Lázaro-Gredilla and Titsias, 2011).[1] This prior, when combined with the cost prior (Section 2.1.1), 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) (Goldberg et al., 1998), 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 (Lázaro-Gredilla and Titsias, 2011), a variational lower bound on the marginal log likelihood is used as a tractable surrogate function for optimizing the hyperparameters.

Let

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

be the vector of unknown 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 (Lázaro-Gredilla and Titsias, 2011),

$$F(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_f + \mathbf{R}) - \frac{1}{4}\mathrm{tr}(\boldsymbol{\Sigma}) - \mathrm{KL}(\mathcal{N}(\mathbf{g}|\boldsymbol{\mu}, \boldsymbol{\Sigma})||\mathcal{N}(\mathbf{g}|\mu_0\mathbf{1}, \mathbf{K}_g)), \tag{6}$$

where $\mathbf{R}$ is a diagonal matrix with elements $[\mathbf{R}]_{ii} = e^{[\boldsymbol{\mu}]_i - [\boldsymbol{\Sigma}]_{ii}/2}$. Intuitively, by maximizing (6) 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,

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{K}_g \left( \boldsymbol{\Lambda} - \frac{1}{2}\mathbf{I} \right) \mathbf{1} + \mu_0 \mathbf{1}, \\ \boldsymbol{\Sigma}^{-1} &= \mathbf{K}_g^{-1} + \boldsymbol{\Lambda}, \end{aligned}$$

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, $\Psi_f$ and $\Psi_g$. If the kernel functions $k_f(\boldsymbol{\theta}, \boldsymbol{\theta}')$ and $k_g(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are squared exponentials (1), 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 (Lizotte et al., 2011). However, a constant hyperparameter, $\mu_0$, 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 (2011) supplementary material), so the maximization problem can be solved using standard nonlinear optimization algorithms such as sequential quadratic programming (SQP).

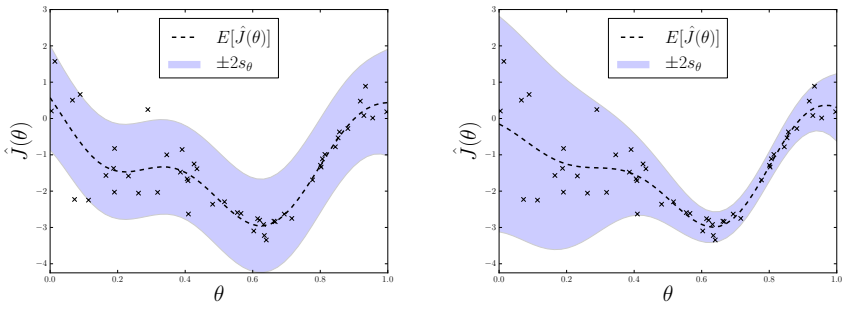
The VHGP model yields a non-Gaussian variational predictive density,

$$q(\hat{J}_*) = \int \mathcal{N}(\hat{J}_*|a_*, c_*^2 + e^{g_*}) \mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*, \tag{7}$$

where

$$\begin{aligned} a_* &= \mathbf{k}_{f*}^\top (\mathbf{K}_f + \mathbf{R})^{-1} \mathbf{y}, \\ c_*^2 &= k_f(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - \mathbf{k}_{f*}^\top (\mathbf{K}_f + \mathbf{R})^{-1} \mathbf{k}_{f*}, \end{aligned}$$

[1] The log variance is used is to ensure positivity of the variance function.




(a) (b)

Figure 1: Comparison of fits for the standard Gaussian process model (a) and the VHGP model (b) on a synthetic heteroscedastic data set.

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

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

$$\begin{aligned} \mathbb{E}_q[\hat{J}_*] &= a_*, \\ \mathbb{V}_q[\hat{J}_*] &= c_*^2 + \exp(\mu_* + \sigma_*^2/2) \equiv s_*^2. \end{aligned}$$

### 2.2.1 Example

Figure 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 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 (6). 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.

## 3 Variational Bayesian Optimization

There are at least two practical motivations for modifying 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, to return more meaningful values for the problem under consideration. For example, the GP model in Figure 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.

We extend the VHGP model to the optimization case by deriving the expression for expected improvement and its gradients and show that both can be efficiently approximated to several decimal places using Gauss-Hermite quadrature (as is the case for the predictive distribution itself (Lázaro-Gredilla and Titsias, 2011)). 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.

### 3.1 Expected Improvement

Recall from Section 2.1.2 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 (7) is

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

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

$$\begin{aligned} \text{EI}(\boldsymbol{\theta}_*) &= \int_0^\infty I_* q(I_*) dI_* \\ &= \int_0^\infty \int I_* \mathcal{N}(I_*|\mu_{\text{best}} - a_*, v_*^2)\mathcal{N}(g_*|\mu_*, \sigma_*^2)dg_* dI_*. \end{aligned} \tag{8}$$

To get (8) into a more convenient form, we can define

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

and rewrite the expression for improvement (3) as,

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

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

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

where $\phi(\cdot)$ is the PDF of the normal distribution. Letting $f(x_*) = v_*(u_* - x_*)$ and integrating $\int_{-\infty}^{u_*} f(x_*)\phi(x_*)dx_*$ by parts, we have

$$\begin{aligned} \int_{-\infty}^{u_*} f(x_*)\phi(x_*)dx_* &= [f(x_*)\Phi(x_*)]_{-\infty}^{u_*} - \int_{-\infty}^{u_*} (-v_*)\Phi(x_*)dx_*, \\ &= v_* \left[x_*\Phi(x_*) + \phi(x_*)\right]_{-\infty}^{u_*}, \\ &= v_*(u_*\Phi(u_*) + \phi(u_*)), \end{aligned}$$

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

$$\text{EI}(\boldsymbol{\theta}_*) = \int v_*(u_*\Phi(u_*) + \phi(u_*))\mathcal{N}(g_*|\mu_*, \sigma_*^2)dg_*. \tag{9}$$

Although this expression is not analytically tractable, it can be efficiently approximated using Gauss-Hermite quadrature (Abramowitz and Stegun, 1972). 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 $u_*$,

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

where $n$ is the number of sample points, $\rho_i$ are the roots of the Hermite polynomial,

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

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 $\rho_i$ for a given $n$. In all of our experiments, $n = 45$.

Similarly, the gradient $\partial \mathrm{EI}(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ can be computed under the integral (9) and the result is of the desired form:

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

where

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

As in the standard Bayesian optimization setting, one can easily incorporate an exploration parameter, $\xi$, 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 our experiments, we used the NLOPT (Johnson, 2011) implementation of SQP with 25 random restarts to optimize EI.

### 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

$$\mathrm{CB}(\boldsymbol{\theta}_*, \kappa) = \mathbb{E}[\hat{J}_*] + b(\mathbb{V}[\hat{J}_*], \kappa), \tag{10}$$

where $b(\cdot, \cdot)$ is a function of the cost variance and a constant risk factor, $\kappa$, that controls the system's sensitivity to risk. Such criteria have been extensively studied in the context of statistical global optimization (Cox and John, 1992; Srinivas et al., 2010) and economic decision making (Levy and Markowitz, 1979). 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 (Srinivas et al., 2010).

Interestingly, CB criteria have a strong connection to the exponential utility functions of risk-sensitive optimal control (Whittle, 1990, 1981). For example, consider the risk-sensitive optimal control objective function,

$$\gamma(\boldsymbol{\theta}_*, \kappa) = -2\kappa^{-1} \log \mathbb{E}[e^{-\frac{1}{2}\kappa \hat{J}_*}]. \tag{11}$$

By taking the second order Taylor expansion of (11) about $\mathbb{E}[\hat{J}_*]$, we have

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

Thus, 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, because the selection is performed with respect to the predictive distribution, *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 risk-sensitive control algorithms that must perform separate optimizations that require significant computation or additional policy executions to produce policies with different risk-sensitivity.

In practice, one typically sets $b(\mathbb{V}[\hat{J}_*], \kappa) = \kappa\sqrt{\mathbb{V}[\hat{J}_*]} = \kappa s_*$ so that terms of the same units are combined and the parameter $\kappa$ 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 (Sharpe, 1966). Since the mean and variance of the VHGP model are analytically computable, extensions that optimize such criteria would be straightforward to implement.

### 3.3 Expected Risk Improvement

The primary advantage CB 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. (1998) considered a generalization of EI where the improvement for $\boldsymbol{\theta}_*$ was defined as

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

where $\rho$ 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 propose an expected risk improvement (ERI) criterion. In this case, the *risk improvement* for the policy parameters $\boldsymbol{\theta}_*$ is defined as

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

where

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

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

$$\begin{aligned} \text{ERI}(\boldsymbol{\theta}_*) &= \int_0^\infty I_*^\kappa q(I_*^\kappa) dI_*^\kappa \\ &= \int v_*(u_*\Phi(u_*) + \phi(u_*))\mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*, \end{aligned} \tag{12}$$

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 $\kappa = 0$.

### 3.4 Coping with Small Sample Sizes

#### 3.4.1 Log Hyperpriors

Numerical precision problems are commonly experienced when performing model selection (which requires kernel matrix inversions and determinant calculations) using small amounts of data. To help improve numerical stability in the VHGP model when $N$ is small, we augment $F(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with independent log-normal priors for each hyperparameter,

$$\hat{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = F(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \sum_{\psi_k \in \Psi} \log \mathcal{N}(\log \psi_k | \mu_k, \sigma_k^2), \tag{13}$$

where $\Psi = \Psi_f \cup \Psi_g$ is the set of all hyperparameters. Lizotte et al. (2011) 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 our experiments with VBO, we 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 our experiments, SQP was used with 10 random restarts to perform model selection.

#### 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 (Jones, 2001). 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 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, $\epsilon$-random EI selection has been shown to yield near-optimal global convergence rates (Bull, 2011).

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 Local Search

Like most standard Bayesian optimization implementations, no general global convergence guarantees exist for VBO. In addition, performing global selection of policy parameters can produce large jumps in policy space between trials, which can be undesirable in some physical systems. A straightforward way to address this latter concern is to restrict the parameter range to the local neighborhood of the nominal policy parameters. However, adding constraints in this way does not improve the convergence properties of the algorithm.

Gradient-based policy search methods make small, incremental changes to the policy parameters and typically have demonstrable local convergence properties under mild assumptions (Bertsekas and Tsitsiklis, 2000). Thus, in addition to using the learned cost model to perform global policy selection, we consider its use as a local critic for performing risk-sensitive stochastic gradient descent (RSSGD). It is straightforward to show that, under certain assumptions, the generalized RSSGD update follows the direction of the gradient of a confidence bound 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 3.3) over the current

**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*: $\kappa$, *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** $\boldsymbol{\Theta}, \mathbf{y}$

policy parameters. This creates the opportunity to flexibly interweave risk-sensitive gradient descent and local VBO to, e.g., select local greedy policies or to change risk sensitivity on-the-fly.

### 4.1 Risk-Sensitive Stochastic Gradient Descent

Stochastic gradient descent methods have had significant practical applicability to solving robot control problems in the expected cost setting (Tedrake et al., 2004; Kohl and Stone, 2004; Roberts and Tedrake, 2009), so we focus on extending this approach to the risk-sensitive case. The stochastic gradient descent algorithm, also called the weight perturbation algorithm (Jabri and Flower, 1992), 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, $\hat{J}(\boldsymbol{\theta}) \equiv \hat{J}_{\boldsymbol{\theta}}$. 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, $\hat{J}(\boldsymbol{\theta}+\mathbf{z}) \equiv \hat{J}_{\boldsymbol{\theta}+\mathbf{z}}$. Finally, update the policy parameters, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, where

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

and $\eta$ 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,

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

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

In contrast, consider the RSSGD update:

$$\Delta\boldsymbol{\theta} = -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} + \kappa\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - b(\boldsymbol{\theta}))\mathbf{z}, \tag{14}$$

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 (Williams, 1992) of the policy parameters.

Substituting (5) into (14) and taking the first order Taylor expansion at $\boldsymbol{\theta}+\mathbf{z}$, we have

$$\begin{aligned}
\Delta\boldsymbol{\theta} &= -\eta\left(J_{\boldsymbol{\theta}+\mathbf{z}} + \varepsilon_{\boldsymbol{\theta}+\mathbf{z}} + \kappa\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - b(\boldsymbol{\theta})\right)\mathbf{z}, \\
&\approx -\eta\left(J_{\boldsymbol{\theta}} + \mathbf{z}^\top\nabla J_{\boldsymbol{\theta}} + \varepsilon_{\boldsymbol{\theta}} + u\mathbf{z}^\top\nabla r_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}} + \kappa\mathbf{z}^\top\nabla\tilde{r}_{\boldsymbol{\theta}} - b(\boldsymbol{\theta})\right)\mathbf{z}, \\
&\equiv \tilde{\Delta}\boldsymbol{\theta},
\end{aligned}$$

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

$$\mathbb{E}[\tilde{\Delta}\boldsymbol{\theta}] = -\eta\sigma^2\left(\nabla J_{\boldsymbol{\theta}} + \kappa\nabla\tilde{r}_{\boldsymbol{\theta}}\right), \tag{15}$$

where the expectation is taken with respect to $\mathbf{z}$, $u$, and $\varepsilon_{\boldsymbol{\theta}}$. Thus, the update equation (14) 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 $\kappa$). If the estimator of the cost standard deviation is unbiased, we have

$$\mathbb{E}[\tilde{\Delta}\boldsymbol{\theta}] = -\eta\sigma^2\nabla\mathrm{CB}(\boldsymbol{\theta}, \kappa), \tag{16}$$

a scaled unbiased estimate of the gradient of the confidence bound objective, $\mathrm{CB}(\boldsymbol{\theta}, \kappa) = J_{\boldsymbol{\theta}} + \kappa r_{\boldsymbol{\theta}}$. 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.

### 4.1.1 Natural Gradient

From (16) 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 (2009), 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,

$$\mathbb{E}[\tilde{\Delta}\boldsymbol{\theta}] = -\eta\boldsymbol{\Sigma}\nabla \mathrm{CB}(\boldsymbol{\theta}, \kappa), \tag{17}$$

but it is still in the direction of the *natural gradient* (Amari, 1998). 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 (Kakade, 2002). 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, (17) is in the direction of the natural gradient.

### 4.1.2 Baseline Selection

The expected update (15) 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. The variance of the update (14) can be written as,

$$\begin{aligned}\mathbb{V}[\tilde{\Delta}\boldsymbol{\theta}] \;=\; & \eta^2\sigma^2(b(\boldsymbol{\theta})^2\mathbf{I} - 2J_{\boldsymbol{\theta}}b(\boldsymbol{\theta})\mathbf{I} - 2\kappa\tilde{r}_{\boldsymbol{\theta}}b(\boldsymbol{\theta})\mathbf{I} + J_{\boldsymbol{\theta}}^2\mathbf{I} + 2\kappa J_{\boldsymbol{\theta}}\tilde{r}_{\boldsymbol{\theta}}\mathbf{I} + \kappa^2\tilde{r}_{\boldsymbol{\theta}}^2\mathbf{I} + r_{\boldsymbol{\theta}}^4\mathbf{I} \\ & + \sigma^2(\nabla J_{\boldsymbol{\theta}}^\top \nabla J_{\boldsymbol{\theta}}\mathbf{I} + \nabla J_{\boldsymbol{\theta}}\nabla J_{\boldsymbol{\theta}}^\top) + \sigma^2\kappa(2\nabla J_{\boldsymbol{\theta}}^\top \nabla \tilde{r}_{\boldsymbol{\theta}}\mathbf{I} + \nabla J_{\boldsymbol{\theta}}\nabla \tilde{r}_{\boldsymbol{\theta}}^\top + \nabla \tilde{r}_{\boldsymbol{\theta}}\nabla J_{\boldsymbol{\theta}}^\top) \\ & + \sigma^2 r_{\boldsymbol{\theta}}^2(\nabla r_{\boldsymbol{\theta}}^\top \nabla r_{\boldsymbol{\theta}}\mathbf{I} + 2\nabla r_{\boldsymbol{\theta}}\nabla r_{\boldsymbol{\theta}}^\top) + \sigma^2\kappa^2(\nabla \tilde{r}_{\boldsymbol{\theta}}^\top \nabla \tilde{r}_{\boldsymbol{\theta}}\mathbf{I} + \nabla \tilde{r}_{\boldsymbol{\theta}}\nabla \tilde{r}_{\boldsymbol{\theta}}^\top)).\end{aligned} \tag{18}$$

It is straightforward to show that the baseline that minimizes (18) is $b(\boldsymbol{\theta}) = J_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}}$. 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

$$\Delta\boldsymbol{\theta} = -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{J}_{\boldsymbol{\theta}} + \kappa(\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{r}_{\boldsymbol{\theta}}))\mathbf{z}. \tag{19}$$

Intuitively, (19) 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 criterion (12). From this point of view, the update can be interpreted as taking steps in the direction of risk improvement over the nominal 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.

As in VBO, 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 $\epsilon$-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 RSSGD algorithm is outlined in Algorithm 3.

**Algorithm 3** Risk-sensitive stochastic gradient descent

**Input**: *Parameters*: $\eta, \sigma, \epsilon$, *Risk factor*: $\kappa$, *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** $\boldsymbol{\Theta}, \mathbf{y}, \boldsymbol{\theta}$

# 5 Experiments

In Sections 5.1 and 5.2 we illustrate the VBO algorithm using simple synthetic domains. In Section 5.3, we apply VBO to a impact recovery task with the uBot-5 mobile manipulator. Finally, in Section 5.4, we apply the RSSGD algorithm in a dynamic heavy lifting task with the uBot-5.

## 5.1 Synthetic Data

As an illustrative example, in Figure 2 we compare the performance of VBO to standard Bayesian optimization in a simple 1-dimensional noisy optimization task. For this task, the true underlying cost distribution (Figure 2(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 2(b), we see that Bayesian optimization succeeds in identifying the regions of low cost, but it cannot capture the policy-dependent variance characteristics.

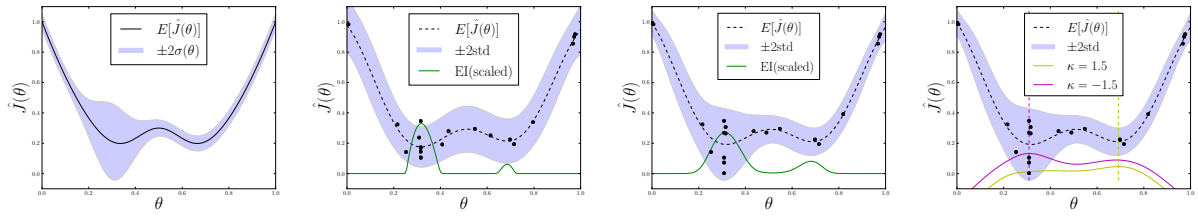
In contrast, VBO reliably identifies the minima *and* approximates the local variance characteristics. Figure 2(d) shows the result of applying two different confidence bound selection criteria to vary risk sensitivity. In this case, $-\text{CB}(\boldsymbol{\theta}_*, \kappa)$ was maximized, where

$$\text{CB}(\boldsymbol{\theta}_*, \kappa) = \mathbb{E}_q[\hat{J}_*] + \kappa s_*. \tag{20}$$

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

## 5.2 Noisy Pendulum

As another simple example, we 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




(a) (b) (c) (d)

Figure 2: (a) An example unknown 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 cases). 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 (indicated by the vertical dotted lines) given the distribution learned using VBO.

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

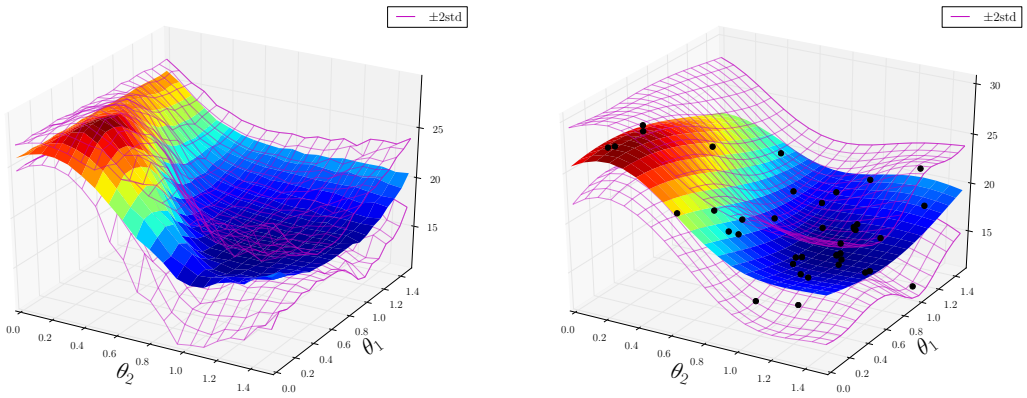
$$J(\boldsymbol{\theta}) = \int_0^T 0.01\alpha(t) + 0.0001u(t)^2 dt, \tag{21}$$

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 $0 \leq t \leq \theta_1$, $u(t) = -\tau_{\max}$ if $\theta_1 < t \leq \theta_1 + \theta_2$, and $u(t) = \tau_{\max}$ if $\theta_1 + \theta_2 < t \leq T$. 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 $\approx 1.0$) perform at least two swings to reach vertical in less than $T$ seconds.

The cost function (21) 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 3(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 3(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 illustrates the behavior of two policies selected by minimizing the CB criterion (20) 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 (roughly $\mathbb{E}[\tau_{\max}] + \sigma[\tau_{\max}]$). 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).

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




(a) (b)

Figure 3: (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.

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.

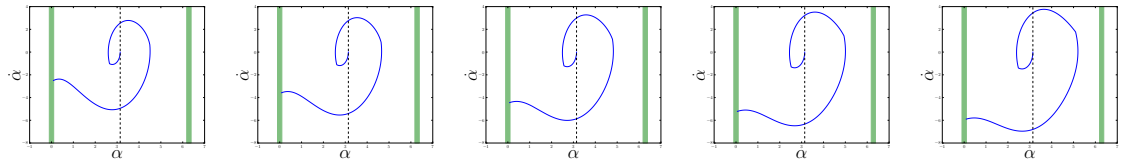
### 5.3 Balance Recovery with the uBot-5

The uBot-5 (Figure 5) is an 11-DoF mobile manipulator developed at the University of Massachusetts Amherst (Deegan, 2010; Kuindersma et al., 2009). 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, it has two wheels attached at the hip. 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 our experiments.

In our previous experiments (Kuindersma et al., 2011), the energetic and stabilizing effects of rapid arm motions on the LQR stabilized system were evaluated in the context of recovery from impact perturbations. One observation 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. We extended these experiments by considering larger impact perturbations, increasing the set of arm initial conditions, and defining a policy space that permits more flexible, asymmetric arm motions (Kuindersma et al., 2012b).

The robot was placed in a balancing configuration with its upper torso aligned with a 3.3 kg mass suspended from the ceiling (Figure 6). 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 $9.9 \pm 0.8$ Ns and the resulting impact force was approximately equal to the robot's total mass in earth gravity. The robot was consistently unable to recover from this perturbation using only the wheel LQR (see the rightmost column of Figure 7). The robot was attached to the ceiling with a loose-fitting safety rig designed to prevent the robot from falling completely to the ground, while not affecting policy performance.

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.



(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: 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.

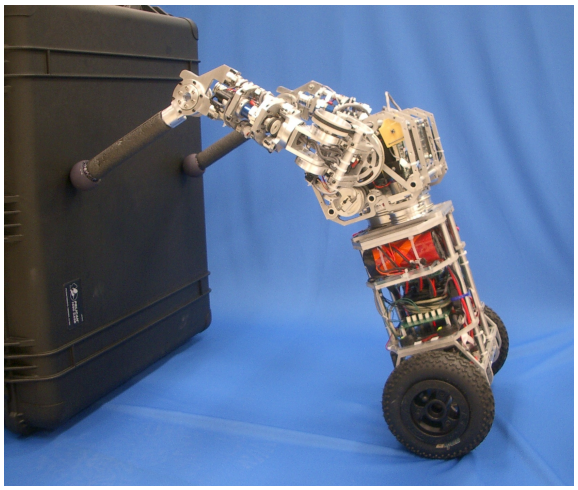


Figure 5: The uBot-5 demonstrating a whole-body pushing behavior.

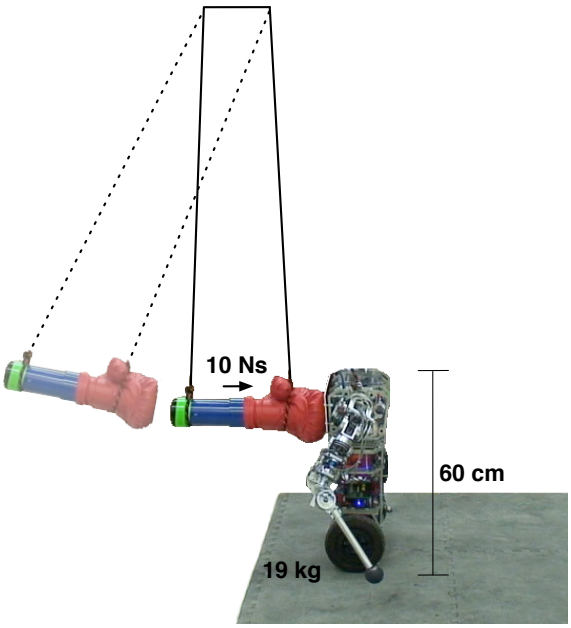



Figure 6: The uBot-5 situated in the impact pendulum apparatus.

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.

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 $\lambda$ 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 $\lambda_i$ 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:

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

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

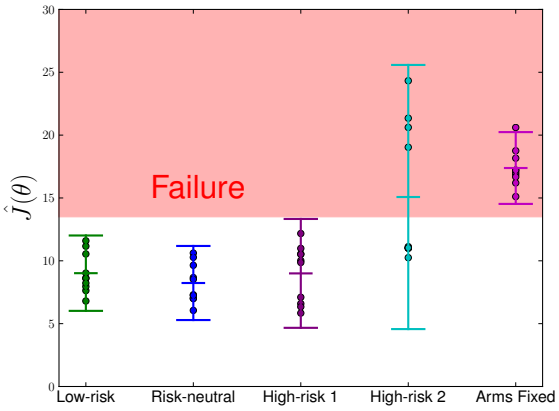



Figure 7: Data collected over 10 trials using policies identified as risk-averse, risk-neutral, and risk-seeking after performing VBO. The policies were selected using confidence bound criteria with $\kappa = 2$, $\kappa = 0$, $\kappa = -1.5$, and $\kappa = -2$, from left to right. The sample means and two times sample standard deviations are shown. The shaded region 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.

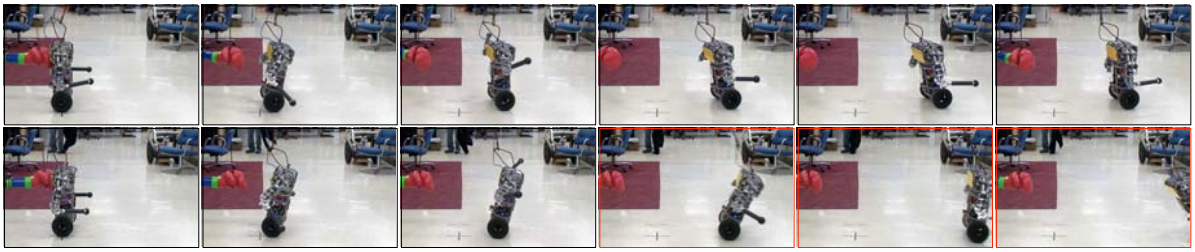
For $\kappa = -2$, the selected policy produced an upward laterally-directed arm motion that failed approximately 50% of the time. In this case, the standard deviation of cost was sufficiently large that the second term in CB objective (20) dominated, producing a policy with high variance and poor average 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 8 shows example runs of the risk-averse and risk-seeking policies.

## 5.4 Dynamic Heavy Lifting

We evaluated the RSSGD algorithm in the dynamic control task of lifting a 1 kg, partially-filled laundry detergent bottle from the ground to a height of 120 cm using the uBot-5 (Kuindersma et al., 2012a). 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 we 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 (Craig, 2005). The initial policy was a hand-crafted smooth and short duration motion to the goal configuration. Our ability to provide a good initial guess for the policy parameters makes local search with RSSGD more attractive. 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) Low-risk policy, $\kappa = 2.0$



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

Figure 8: Time series (time between frames is 0.24 seconds) showing (a) a trial executing the low-risk policy and (b) two trials executing the high-risk policy. Both policies were 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.

The cost function was defined as

$$J(\boldsymbol{\theta}) = \int_0^T \left(\mathbf{x}(t)^\top \mathbf{Q}\mathbf{x}(t) + cI(t)V(t)\right) dt, \tag{22}$$

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 ($\leq 5$ 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 $\theta_i \in [0, 1]$, so the constant $\sigma$ corresponded to different (unscaled) perturbation sizes for each dimension depending on the total parameter range.

### 5.4.1 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 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 9 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 the body. This dynamic interaction would be very difficult to capture in a system model. Incidentally, this

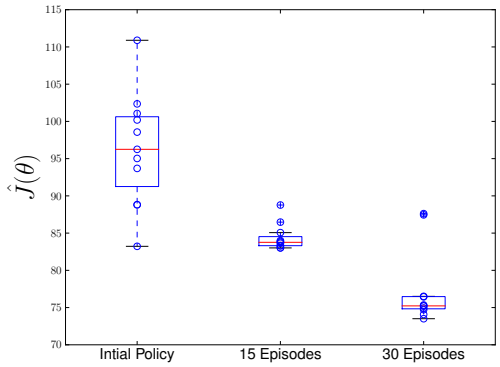



Figure 9: Data collected from 10 test trials executing the initial lifting policy and the policy after 15 and 30 episodes of learning.

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 (Roberts and Tedrake, 2009). Figure 10(a) shows an example run of the learned policy.

### 5.4.2 Variable Risk Control

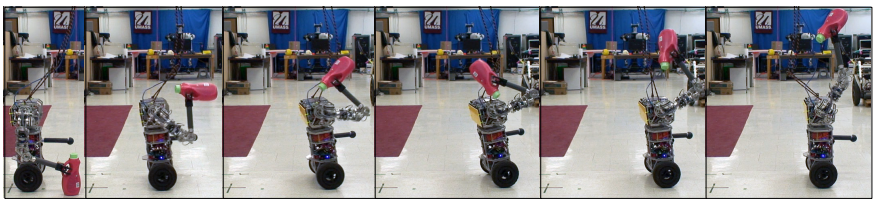
In the process of learning a low average-cost policy, a model of the local cost distribution was repeatedly computed. The next experiments examined the effect of performing offline policy selection using the estimate of the local cost distribution around the learned policy. In particular, we considered two hypothetical changes in operating context: when the 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 confidence bound policy selection and subsequent risk-averse gradient descent were performed for each case and the resulting policies were empirically compared.

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 (23) 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,

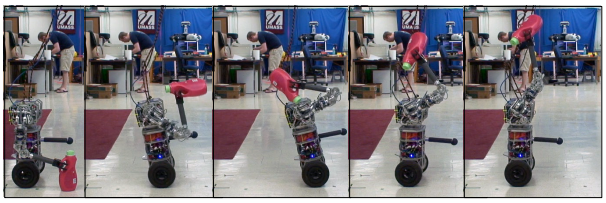
$$J_{en}(\boldsymbol{\theta}) = \int_0^T \left(\mathbf{x}(t)^\top \mathbf{Q}_{en}\mathbf{x}(t) + c_{en}I(t)V(t)\right) dt. \tag{23}$$

The VHGP model was used to approximate the transformed cost distribution, $\hat{J}_{en}(\boldsymbol{\theta})$, around the previously learned policy parameters using the data collected during the 30 learning trials. SQP was used to minimize $\tilde{J}_{en}(\boldsymbol{\theta}) + \kappa\tilde{r}_{en}(\boldsymbol{\theta})$ 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 $\tilde{J}_{tr}(\boldsymbol{\theta}) + \kappa\tilde{r}_{tr}(\boldsymbol{\theta})$ offline.

Both risk-neutral ($\kappa = 0$) and risk-averse ($\kappa = 2$) offline policy selections 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 11(a) shows the results from the translation aversion experiments. The risk-neutral offline policy had significantly lower average (transformed) cost and lower variance 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. An example run of the



(a)



(b)

Figure 10: (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 approximately 3 seconds.

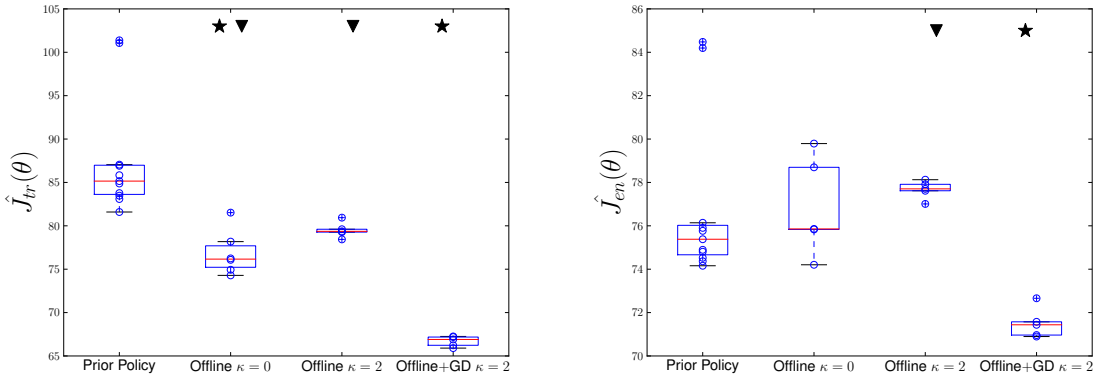
offline risk-averse policy is shown in Figure 10(b). Finally, the policy learned after 5 episodes of risk-averse gradient descent starting from the offline selected policy led to another significant reduction in expected cost while maintaining similarly low variance.

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. 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 11(b)). The statistical significance results given in Figure 11 are strongly in line with our qualitative assessment of the data. However, we should take care to consider these in light of the small sample sizes available, which constrain our ability to verify their underlying assumptions.

# 6 Related Work

Several successful applications of Bayesian optimization to robot control tasks exist in the literature. Lizotte et al. (2007) 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. (2011) used Bayesian optimization to optimize snake robot gaits in several environmental contexts. Martinez-Cantin et al. (2009) describe an application to online sensing and path planning for mobile robots in uncertain environments. Recently, Kormushev and Caldwell (2012) proposed a particle filter approach for performing direct policy search that is closely related to Bayesian optimization techniques.

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. This work included algorithms for solving discrete Markov decision processes (MDPs) (Howard and Matheson, 1972) and linear-quadratic-Gaussian problems (Jacobson, 1973; Whittle, 1981). Borkar derived a variant of the Q-learning algorithm for finite MDPs with exponential utility (Borkar, 2002). Heger (1994) derived a worst-case Q-learning algorithm based on a minimax criterion. For continuous




(a) Translation aversion

(b) Energy aversion

Figure 11: 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, $p < 0.01$) and a triangle signifies a significant reduction in the variance (F-test, $p < 0.03$).

problems, Van den Broek et al. (2010) generalized path integral methods from stochastic optimal control to the risk-sensitive case.

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 (Dearden et al., 1998; Morimura et al., 2010a,b), which can be combined with policy selection criteria that take return variance into account. The algorithms discussed in this paper are related to this line of work, but they are more directly applicable to systems with continuous state and action spaces. The recent work of Tamar et al. (2012) describes likelihood-ratio policy gradient algorithms appropriate for different types of risk-sensitive criteria. The simulation-based algorithm in their work is 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.

Policy gradient approaches that are designed to learn dynamic transition models, such as PILCO (Deisenroth and Rasmussen, 2011), can also be used to capture uncertainty in the cost distribution (Deisenroth, 2010). These approaches are capable of handling high-dimensional policy spaces, whereas the approaches described in this work are only appropriate for low-dimensional policy spaces. However, to achieve this scalability, certain smoothness assumptions must be made about the system dynamics. Furthermore, performing offline optimizations to change risk-sensitivity would be significantly more computationally intensive than the approach presented here.

Mihatsch and Neuneier (2002) 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 (Niv et al., 2012), suggesting that some form of risk-sensitive TD may be present 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 (Preuschoff et al., 2008; Tobler et al., 2007). Recent motor control experiments suggest that humans select motor strategies in a risk-sensitive way (Wu et al., 2009; Nagengast et al., 2011, 2010). For example, Nagengast et al. (2010) 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 (Kacelnik and Bateson, 1996; Bateson, 2002; Niv et al., 2002).

# 7 Discussion and Future Work

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 (Braun et al., 2011; Bateson, 2002).

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 sensitivity. 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.

The 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 many other episodic policy search algorithms, such as Bayesian optimization and finite-difference methods (Kohl and Stone, 2004; Roberts and Tedrake, 2009), 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* (Lizotte et al., 2007).

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. Methods for capturing multimodality of the cost distribution are also important to consider, especially in domains where unobservable differences in initial conditions can lead to qualitatively different outcomes.

One straightforward way to extend VBO 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. (Osborne et al., 2009; Garnett et al., 2010) 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 (Tibshirani and Hastie, 1987; Snelson and Ghahramani, 2006; Kersting et al., 2010; Wilson and Ghahramani, 2011). It would be worthwhile to investigate whether these methods are more appropriate for particular problem domains.

The VBO and RSSGD algorithms are connected by their 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. We considered the possibility of interweaving gradient descent with local offline policy selection in 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 noteworthy that this kind of flexibility is possible after so few trials, especially given the generality of the optimization procedure. However, a 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.

The use of the cost model in the RSSGD algorithm is somewhat restricted and 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 (Roberts and Tedrake, 2009). This idea is related to the covariance matrix adaptation that is done in some cost weighted averaging methods (Stulp and Sigaud, 2012). 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.

# 8 Conclusion

Varying risk sensitivity based on runtime context is a potentially powerful way to generate flexible control in robot systems. We considered this problem in the context of model-free policy search, where risk-sensitive parameterized policies can be selected based on a learned cost distribution. Our experimental results suggest that VBO and RSSGD are efficient and plausible methods for achieving variable risk control.

# 9 Acknowledgments

Scott Kuindersma was supported by a NASA GSRP Fellowship from Johnson Space Center. Roderic Grupen was supported by the ONR MURI award N00014-07-1-0749. Andrew Barto was supported by the AFOSR under grant FA9550-08-1-0418.

# References

Abramowitz, M. and Stegun, I. A., editors (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing*. Dover, New York.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.

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

Bertsekas, D. P. and Tsitsiklis, J. N. (2000). Gradient convergence in gradient methods with errors. *SIAM J. Optim.*, 10(3):627–642.

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

Braun, D. A., Nagengast, A. J., and Wolpert, D. M. (2011). Risk-sensitivity in sensorimotor control. *Frontiers in Human Neuroscience*, 5:1–10.

Brochu, E., Cora, V., and de Freitas, N. (2009). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-023, University of British Columbia, Department of Computer Science.

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904.

Cox, D. D. and John, S. (1992). A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, volume 2, pages 1241–1246.

Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, 3rd edition.

Dearden, R., Friedman, N., and Russell, S. (1998). Bayesian Q-learning. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 761–768.

Deegan, P. (2010). *Whole-Body Strategies for Mobility and Manipulation*. PhD thesis, University of Massachusetts Amherst.

Deisenroth, M. P. (2010). *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, Karlsruhe Institute of Technology.

Deisenroth, M. P. and Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA.

Frean, M. and Boyle, P. (2008). Using Gaussian processes to optimize expensive functions. In *AI 2008: Advances in Artificial Intelligence*, pages 258–267.

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

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

Heger, M. (1994). Consideration of risk in reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ICML)*, pages 105–111.

Howard, R. A. and Matheson, J. E. (1972). Risk-sensitive markov decision processes. *Management Science*, 18(2):356–369.

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

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

Johnson, S. G. (2011). The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt.

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

Kacelnik, A. and Bateson, M. (1996). Risky theories—the effects of variance on foraging decisions. *Amer. Zool.*, 36:402–434.

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

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. (2010). Most likely heteroscedastic Gaussian process regression. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 393–400.

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

Kohl, N. and Stone, P. (2004). Machine learning for fast quadrupedal locomotion. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 611–616.

Kolter, J. Z. and Ng, A. Y. (2010). Policy search via the signed derivative. In *Robotics: Science and Systems V (RSS)*.

Kormushev, P. and Caldwell, D. G. (2012). Direct policy search reinforcement learning based on particle filtering. In *Proceedings of the 10th European Workshop on Reinforcement Learning*.

Kuindersma, S., Grupen, R., and Barto, A. (2011). Learning dynamic arm motions for postural recovery. In *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots*, pages 7–12, Bled, Slovenia.

Kuindersma, S., Grupen, R., and Barto, A. (2012a). Variable risk dynamic mobile manipulation. In *RSS 2012 Workshop on Mobile Manipulation*, Sydney, Australia.

Kuindersma, S., Grupen, R., and Barto, A. (2012b). Variational Bayesian optimization for runtime risk-sensitive control. In *Robotics: Science and Systems VIII (RSS)*, Sydney, Australia.

Kuindersma, S. R., Hannigan, E., Ruiken, D., and Grupen, R. A. (2009). Dexterous mobility with the uBot-5 mobile manipulator. In *Proceedings of the 14th International Conference on Advanced Robotics*, Munich, Germany.

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

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

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

Lizotte, D., Wang, T., Bowling, M., and Schuurmans, D. (2007). Automatic gait optimization with Gaussian process regression. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*.

Lizotte, D. J., Greiner, R., and Schuurmans, D. (2011). An experimental methodology for response surface optimization methods. *J Glob Optim*, 53(4):699–736.

Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J. A., and Doucet, A. (2009). A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27:93–103.

Martinez-Cantin, R., de Freitas, N., Doucet, A., and Castellanos, J. A. (2007). Active policy learning for robot planning and exploration under uncertainty. In *Proceedings of Robotics: Science and Systems*.

Mihatsch, O. and Neuneier, R. (2002). Risk-sensitive reinforcement learning. *Machine Learning*, 49:267–290.

Morimura, T., Sugiyama, M., Kashima, H., and Hachiya, H. (2010a). Nonparametric return distribution approximiation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*.

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., and Tanaka, T. (2010b). Parametric return density estimation for reinforcement learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*.

Močkus, J., Tiesis, V., and Žilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. In *Toward Global Optimization*, volume 2, pages 117–128. Elsevier.

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

Nagengast, A. J., Braun, D. A., and Wolpert, D. M. (2011). Risk-sensitivity and the mean-variance trade-off: decision making in sensorimotor control. *Proc. R. Soc. B*, 278(1716):2325–2332.

Niv, Y., Edlund, J. A., Dayan, P., and O'Doherty, J. P. (2012). Neural prediction errors reveal a risk-sensitive reinforcement-learning process in the human brain. *Journal of Neuroscience*, 32(2):551–562.

Niv, Y., Joel, D., Meilijson, I., and Ruppin, E. (2002). Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior*, 10(1):5–24.

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

Peters, J. and Schaal, S. (2006). Policy gradient methods for robotics. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2219–2225.

Preuschoff, K., Quartz, S. R., and Bossaerts, P. (2008). Human insula activation reflects risk prediction errors as well as risk. *Journal of Neuroscience*, 28(11):2745–2752.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Roberts, J. W., Moret, L., Zhang, J., and Tedrake, R. (2010). Motor learning at intermediate Reynolds number: experiments with policy gradient on the flapping flight of a rigid wing. In Sigaud, O. and Peters, J., editors, *From Motor to Interaction Learning in Robots*, volume 264 of *Studies in Computational Intelligence*, pages 293–309. Springer.

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

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

Schonlau, M., Welch, W. J., and Jones, D. R. (1998). Global versus local search in constrained optimization of computer models. In Flournoy, N., Rosenberger, W. F., and Wong, W. K., editors, *New Developments and Applications in Experimental Design*, volume 34 of *Lecture Notes - Monograph Series*, pages 11–25. IMS.

Sharpe, W. F. (1966). Mutual fund performance. *Journal of Business*, 39(S1):119–138.

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

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*.

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

Tamar, A., Castro, D. D., and Mannor, S. (2012). Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland.

Tedrake, R., Zhang, T. W., and Seung, H. S. (2004). Stochastic policy gradient reinforcement learning on a simple 3D biped. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2849–2854, Sendai, Japan.

Tesch, M., Schneider, J., and Choset, H. (2011). 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.

Theodorou, E., Buchli, J., and Schaal, S. (2010). Reinforcement learning of motor skills in high dimensions: A path integral approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska.

Tibshirani, R. and Hastie, T. (1987). Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567.

Tobler, P. N., O'Doherty, J. P., Dolan, R. J., and Schultz, W. (2007). Reward value coding distinct from risk attitude-related uncertainty coding in human reward systems. *J Neurophysiol*, 97:1621–1632.

van den Broek, B., Wiegerinck, W., and Kappen, B. (2010). Risk sensitive path integral control. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 615–622.

Vazquez, E. and Bect, J. (2010). Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planing and Inference*, 140(11):3088–3095.

Whittle, P. (1981). Risk-sensitive linear/quadratic/Gaussian control. *Advances in Applied Probability*, 13:764–777.

Whittle, P. (1990). *Risk-Sensitive Optimal Control*. John Wiley & Sons.

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

Wilson, A., Fern, A., and Tadepalli, P. (2011). 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.

Wilson, A. and Ghahramani, Z. (2011). Generalized Wishart processes. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, Barcelona, Spain.

Wu, S.-W., Delgado, M. R., and Maloney, L. T. (2009). Economic decision-making compared with an equivalent motor task. *Proc. Natl. Acad. Sci. USA*, 106(15):6088–6093.