Episodic Risk-Sensitive Actor-Critic

Scott Kuindersma
Department of Computer Science
University of Massachusetts Amherst
scottk@cs.umass.edu

Roderic Grupen
Department of Computer Science
University of Massachusetts Amherst
grupen@cs.umass.edu

Andrew Barto
Department of Computer Science
University of Massachusetts Amherst
barto@cs.umass.edu

Abstract

We present an episodic risk-sensitive actor-critic algorithm that is suitable for stochastic, continuous, and high-dimensional systems with policy-dependent cost variance. We generalize the simple stochastic gradient descent update to the risk-sensitive case, derive the minimum variance baseline, and show that, under certain conditions, it leads to an unbiased estimate of the gradient of the risk-sensitive objective. We show that the local critic structure used in the update can be exploited to interweave offline and online search to select local greedy policies or quickly change risk sensitivity. Our experiments include learning to lift a heavy, liquid-filled bottle with a dynamically balancing mobile manipulator and online learning of stiffnesses for fall bracing after very large impacts.

1 Introduction

Suppose we have a humanoid robot that executes a gait specified by a set of parameters that we wish to optimize with respect to some cost function that, e.g., accounts for total energy expenditure, stability, and deviation from the desired walking trajectory. However, because of sensor noise, actuator noise, and variations in terrain and initial conditions, the policy evaluations are stochastic and we are unable to assign a single cost to a policy. How should we proceed? A reasonable thing to do is simply minimize the expected value of the noisy cost signal. Indeed, the vast majority of optimal control and reinforcement algorithms are designed to do exactly this.

Now suppose we intend to deploy our robot to the moon. Is average cost still an appropriate criterion? In this case, it might be preferable to select predictable policies over policies with higher cost variance, even if the average cost is slightly higher. A system that performs policy selection by taking variance into account in this way is referred to as *risk-sensitive*, where the term *risk* is equivalent to cost variance. A system that prefers low-variance policies, such as our hypothetical moon robot, is therefore risk-averse. Of course, risk-sensitive systems could also be designed select higher variance policies to, e.g., seek out rare high performance events. In general, a system need not even have a fixed disposition toward risk for a particular task, but may vary risk sensitivity based on context. Perhaps not surprisingly, this variable risk property has been observed in a variety of animal species [1, 2].

We present an efficient risk-sensitive policy search algorithm based on stochastic gradient descent. The algorithm shares several properties (such as scalability, local convergence, and sample efficiency) with existing risk-neutral policy gradient algorithms that have been shown to perform well in robot learning tasks [3, 4, 5]. We also show that the local critic used in the gradient descent update

also supports efficient offline optimization to select policies consistent with different risk-sensitive objectives *on-the-fly* without relearning. We describe experimental results on learning stiffness parameters for fall bracing and learning a dynamic heavy lifting behavior, both performed on a real humanoid mobile manipulator. We also show how a learned policy can be adjusted at runtime to produce policies with different spatial and energetic risk sensitivity.

2 Related Work

Early work in risk-sensitive control was aimed at finding solutions to discrete Markov decision processes (MDPs) [6] and linear-quadratic-Gaussian problems [7, 8] with exponential utility functions. More recent work from Borkar relaxes the assumption of a system model by deriving a variant of the Q-learning algorithm for finite MDPs with exponential utility [9]. For continuous problems, Van den Broek et al. [10] generalized path integral methods to risk-sensitive stochastic optimal control. Recently, Kuindersma et al. [11] extended Bayesian optimization techniques for global model-free policy search to the risk-sensitive case.

Other work in the discrete model-free RL setting has focused on algorithms for learning conditional return distributions [12, 13, 14], which can be combined with policy selection criteria that take return variance into account. Heger [15] derived a worst-case Q-learning algorithm based on a minimax criterion. Mihatsch and Neuneier [16] developed risk-sensitive variants of TD(0) and Q-learning by allowing the learning rate to be a function of the sign of the temporal difference error. Recently, this algorithm was found to be consistent with behavioral and neurological measurements of humans learning a decision task that involving risky outcomes [17].

Other related work in neuroscience has identified separate neural encodings for expected cost and cost variance that are involved in risk-sensitive decision making [18, 19]. Recent motor control experiments suggest that humans select motor strategies in a risk-sensitive way [20, 21, 22]. For example, Nagengast et al. [22] 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 behavior in a wide variety of species [1, 2].

3 Problem Statement

We assume the system executes a (possibly stochastic) policy, $\pi_{\boldsymbol{\theta}}$, that is parameterized by a vector, $\boldsymbol{\theta}$. Executions of $\pi_{\boldsymbol{\theta}}$ yield a noisy signal of cost,

$$\hat{J}_{\boldsymbol{\theta}} = J_{\boldsymbol{\theta}} + \varepsilon_{\boldsymbol{\theta}}, \tag{1}$$

where $J_{\boldsymbol{\theta}}$ is the expected cost of the policy, $\pi_{\boldsymbol{\theta}}$, and the noise term, $\varepsilon_{\boldsymbol{\theta}} \sim \mathcal{N}(0, r^2_{\boldsymbol{\theta}})$, is a function of the policy parameters. This policy-dependent variance is critical since, in general, the variance of the cost signal will not be constant across the policy space. For example, for problems where $\pi_{\boldsymbol{\theta}}$ is performing some type of stabilization (e.g., grasping, balancing), some settings of $\boldsymbol{\theta}$ may only succeed for a subset of the initial conditions, leading to high cost variance.

The optimal policy in the risk-sensitive setting is defined as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \kappa), \text{ where} \tag{2}$$

$$F(\boldsymbol{\theta}, \kappa) = J_{\boldsymbol{\theta}} + \kappa r_{\boldsymbol{\theta}}, \tag{3}$$

and κ is a parameter that controls the systems sensitivity to risk: $\kappa = 0$ is *risk-neutral*, $\kappa > 0$ is *risk-averse*, and $\kappa < 0$ is *risk-seeking*. For example, a subsystem at a nuclear power plant might require $\kappa > 0$ since even rare high cost events could have significant practical impact. On the other hand, a robot attached to a safety apparatus in the lab might set $\kappa < 0$ to seek out rare low cost trials to, e.g., attempt to identify the initial conditions that lead to such events.

4 Episodic Risk-Sensitive Actor-Critic

Our goal is to perform the minimization in eq. (2) under the implicit constraint that observations are costly to obtain. Stochastic gradient descent methods have been shown to be very efficient in

solving episodic control tasks in the average cost setting [3, 23, 4], so we focus on extending this approach to the risk-sensitive case.

We consider the following risk-sensitive stochastic gradient descent update:

$$\Delta\boldsymbol{\theta} \quad = \quad -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} + \kappa\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - b(\boldsymbol{\theta}))\mathbf{z}, \tag{4}$$

where η is a learning rate parameter, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is a perturbation to the current policy parameters, $\boldsymbol{\theta}$, and $\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}}$ is an estimate of the cost standard deviation of $\pi_{\boldsymbol{\theta}+\mathbf{z}}$. The function $b(\boldsymbol{\theta})$ is an arbitrary *baseline* function [24] of the policy parameters.

Substituting eq. (1) into eq. (4) and taking the first order Taylor expansion, we have

$$\begin{aligned}\Delta\boldsymbol{\theta} \quad &= \quad -\eta\left(J_{\boldsymbol{\theta}+\mathbf{z}} + \varepsilon_{\boldsymbol{\theta}+\mathbf{z}} + \kappa\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - b(\boldsymbol{\theta})\right)\mathbf{z} \\ &\approx \quad -\eta\left(J_{\boldsymbol{\theta}} + \mathbf{z}^\top\nabla J_{\boldsymbol{\theta}} + \varepsilon_{\boldsymbol{\theta}} + \mathbf{z}^\top\nabla\varepsilon_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}} + \kappa\mathbf{z}^\top\nabla\tilde{r}_{\boldsymbol{\theta}} - b(\boldsymbol{\theta})\right)\mathbf{z},\end{aligned} \tag{5}$$

where $\nabla f_{\boldsymbol{\theta}} \equiv \left.\frac{\partial f}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}}$. In expectation, this update becomes

$$\mathbb{E}[\Delta\boldsymbol{\theta}] \quad = \quad -\eta\sigma^2\left(\nabla J_{\boldsymbol{\theta}} + \kappa\nabla\tilde{r}_{\boldsymbol{\theta}}\right). \tag{6}$$

Thus, eq. (4) is an estimator of the gradient of expected cost that is biased in the direction of the estimated gradient of standard deviation to a degree specified by the risk sensitivity parameter, κ. If the estimator of the cost standard deviation is unbiased, we have

$$\mathbb{E}[\Delta\boldsymbol{\theta}] = -\eta\sigma^2\nabla F(\boldsymbol{\theta}, \kappa), \tag{7}$$

a scaled unbiased estimate of the gradient of the risk-sensitive objective.

4.1 Natural Gradient

From eq. (7) it is clear that the unbiasedness of the update is 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 [4], 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, our expected update becomes biased:

$$\mathbb{E}[\Delta\boldsymbol{\theta}] = -\eta\boldsymbol{\Sigma}\nabla F(\boldsymbol{\theta}, \kappa). \tag{8}$$

However, eq. (8) is in the direction of the *natural gradient* [25]. To see this, recall that for probabilistically sampled policies, the natural gradient is defined as $\widetilde{\nabla} f(\boldsymbol{\theta}) = \mathbf{G}^{-1}\nabla f(\boldsymbol{\theta})$, where $\mathbf{G}^{-1}$ is the inverse Fisher information matrix [25]. When the policy sampling distribution is mean-zero Gaussian with covariance $\boldsymbol{\Sigma}$, the inverse Fisher information matrix is $\mathbf{G}^{-1} = \boldsymbol{\Sigma}$.

4.2 Baseline Selection

The result given in eq. (6) is unaffected by the choice of the baseline function, $b(\boldsymbol{\theta})$, given that it depends only on $\boldsymbol{\theta}$. However, the baseline does affect the *variance* of the update:

$$\begin{aligned}\mathrm{Var}[\Delta\boldsymbol{\theta}] \quad = \quad & \eta^2\sigma^2\left(b(\boldsymbol{\theta})^2 - 2J_{\boldsymbol{\theta}}b(\boldsymbol{\theta}) - 2\kappa\tilde{r}_{\boldsymbol{\theta}}b(\boldsymbol{\theta}) + J_{\boldsymbol{\theta}}^2 + 2\kappa J_{\boldsymbol{\theta}}\tilde{r}_{\boldsymbol{\theta}} + \kappa^2\tilde{r}_{\boldsymbol{\theta}}^2 + r_{\boldsymbol{\theta}}^4\right. \\ & +\sigma^2(\nabla J_{\boldsymbol{\theta}}^\top\nabla J_{\boldsymbol{\theta}} + \nabla J_{\boldsymbol{\theta}}\nabla J_{\boldsymbol{\theta}}^\top) + \sigma^2\kappa(2\nabla J_{\boldsymbol{\theta}}^\top\nabla\tilde{r}_{\boldsymbol{\theta}} + \nabla J_{\boldsymbol{\theta}}\nabla\tilde{r}_{\boldsymbol{\theta}}^\top + \nabla\tilde{r}_{\boldsymbol{\theta}}\nabla J_{\boldsymbol{\theta}}^\top) \\ & \left.+\sigma^2 r_{\boldsymbol{\theta}}^2(\nabla r_{\boldsymbol{\theta}}^\top\nabla r_{\boldsymbol{\theta}} + 2\nabla r_{\boldsymbol{\theta}}\nabla r_{\boldsymbol{\theta}}^\top) + \sigma^2\kappa^2(\nabla\tilde{r}_{\boldsymbol{\theta}}^\top\nabla\tilde{r}_{\boldsymbol{\theta}} + \nabla\tilde{r}_{\boldsymbol{\theta}}\nabla\tilde{r}_{\boldsymbol{\theta}}^\top)\right).\end{aligned} \tag{9}$$

It is straightforward to show that the baseline that minimizes eq. (9) is $b(\boldsymbol{\theta}) = J_{\boldsymbol{\theta}} + \kappa\tilde{r}_\theta$. However, since $J_{\boldsymbol{\theta}}$ is unknown, we set $b(\boldsymbol{\theta}) = \tilde{J}_{\boldsymbol{\theta}} + \kappa\tilde{r}_\theta$. The resulting increase in variance over the optimal baseline is proportional to the squared error of the expected cost estimate. The update then becomes

$$\Delta\boldsymbol{\theta} \quad = \quad -\eta(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{J}_{\boldsymbol{\theta}} + \kappa(\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{r}_{\boldsymbol{\theta}}))\mathbf{z}. \tag{10}$$

Intuitively, eq. (10) 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. In implementation, we typically divide the learning rate by $\tilde{r}_{\boldsymbol{\theta}}$ so the update maintains scale invariance to changing noise magnitude.

4.3 Critic Representation

The update equation (10) requires a local model of the cost distribution in the neighborhood of $\boldsymbol{\theta}$. We refer to this model as a *critic* because its role is similar to that played by the critic structure in actor-critic algorithms [26, 27]. The problem of constructing the critic in this setting can be viewed as a regression problem with input-dependent noise. There are many algorithms suitable for solving such problems [28, 29, 30, 31, 32]. In our experiments, we used the Variational Heteroscedastic Gaussian Process (VHGP) model [33], which extends the standard Gaussian process model to capture input-dependent noise (or *heteroscedasticity*) in a way that maintains tractability of the mean and variance of the predictive distribution. In general, the hyperparameters of the model are not known exactly, so model selection is performed efficiently by maximizing a tractable lower bound on the marginal log-likelihood. For details regarding the VHGP model, we direct the reader to the original paper [33].

The critic is updated after each policy evaluation by recomputing the predictive cost distribution using previous observations near the current parameterization, $\boldsymbol{\theta}$. The nearest neighbor selection can be performed efficiently by storing observations in a KD-tree data structure and using, e.g., a k-nearest neighbors or an ϵ-ball criterion. The episodic risk-sensitive actor-critic (ERSAC) algorithm is outlined in Algorithm 1.

The local critic can also be used to perform efficient offline optimization of $\tilde{F}(\boldsymbol{\theta}, \kappa) = \tilde{J}_{\boldsymbol{\theta}} + \kappa \tilde{r}_{\boldsymbol{\theta}}$ using standard nonlinear optimization algorithms, such as sequential quadratic programming (SQP). This is particularly useful when κ is varied online to adjust risk based on the current operating context. In our experiments in Section 5, we show that this optimization can be used to make runtime changes to the policy parameters that lead to significant performance improvements under changing optimization criteria.

Algorithm 1 Episodic risk-sensitive actor-critic

1. **Input**: $\eta, \kappa, \sigma, M, \epsilon, \boldsymbol{\theta}, \mathbf{X}, \mathbf{y}$
   (a) **for** $i := 1 : M$
      i. *Sample perturbation:* $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$
      ii. *Execute* $\boldsymbol{\theta} + \mathbf{z}$, *record cost* $\hat{J}_{\boldsymbol{\theta}+\mathbf{z}}$
      iii. *Update data:*
      $\mathbf{X}, \mathbf{y} = [\mathbf{X}; \boldsymbol{\theta} + \mathbf{z}], [\mathbf{y}; \hat{J}_{\boldsymbol{\theta}+\mathbf{z}}]$
      $\mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}} = NearestNeighbors(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}, \epsilon)$
      iv. *Compute posterior mean and variance:*
      $\tilde{J}_{\boldsymbol{\theta}} = \mathbb{E}[\hat{J}_{\boldsymbol{\theta}} \mid \mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
      $\tilde{r}^2_{\boldsymbol{\theta}} = \text{Var}[\hat{J}_{\boldsymbol{\theta}} \mid \mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
      $\tilde{r}^2_{\boldsymbol{\theta}+\mathbf{z}} = \text{Var}[\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} \mid \mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
      v. *Update policy parameters:*
      $\Delta\boldsymbol{\theta} := -\frac{\eta}{\tilde{r}_{\boldsymbol{\theta}}} \left(\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{J}_{\boldsymbol{\theta}} + \kappa(\tilde{r}_{\boldsymbol{\theta}+\mathbf{z}} - \tilde{r}_{\boldsymbol{\theta}}) \right) \mathbf{z}$
      $\boldsymbol{\theta} := \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
   (b) **Return** $\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}$

5 Experiments

5.1 Bracing for Falls

We performed learning experiments with the uBot-5, an 11-DOF humanoid robot that balances on two wheels attached at the hip in a differential drive configuration. It has a mass of 19 kg and stands approximately 60 cm from the ground to its shoulder. Balancing is achieved using a linear-quadratic regulator (LQR) with feedback from an onboard inertial measurement unit (IMU).

In the first experiment, we consider the problem of bracing for a fall in response to very large impact perturbations. Impacts were generated by a swinging 3.3 kg mass affixed to the ceiling with rope. The drop height was varied randomly so the momentum prior to impact was approximately 14 ± 2 Ns. This is a significantly larger perturbation than was considered by Kuindersma et al. in

balance recovery experiments with the same apparatus [34, 11] and under no circumstances have we observed successful recovery from these high impacts.

The class of feasible bracing policies was strongly constrained by the physical limitations of the robot. For example, time between impact onset and arm contact with the ground was approximately $1/4$ second. Given this short time duration and the robot's actuator velocity limitations, the range of kinematic configurations of the arms for ground contact was very limited. Additionally, torque must be minimized for a subset of the arm joints that are driven with rubber belts, since these can slip and fail to absorb the ground impact. Thus, the optimization problem we designed involved selecting the joint stiffnesses for ground contact given the bracing arm configuration that satisfied the constraints of the system.

The joint stiffnesses were governed by a parameter $\theta \in [0, 1]$ that was optimized with respect to the cost function

$$J(\theta) = h(\mathbf{x}(T)) \int_{t=0}^{T} (0.1\ddot{\alpha}(t)^2 + 5I(t)V(t))dt, \tag{11}$$

where $T = 2.0$ sec, $\ddot{\alpha}(t)$ is the body acceleration at time t, and $I(t)$ and $V(t)$ are the motor currents and voltages for all arm joints, respectively. We set $h(\mathbf{x}(T)) = 10$ if hardware is damaged as a result of the bracing trial, and $h(\mathbf{x}(T)) = 0$ otherwise. In our experiments, all observed hardware failures were broken steel pulley cables at the elbow joints.

We performed risk-averse $\kappa = 2$ gradient descent using the ERSAC algorithm with $\eta = 0.7, \sigma = 0.05, \epsilon = 4\sigma, \eta/\tilde{r}_{\boldsymbol{\theta}} \in [0.01, 0.5]$. The VHGP algorithm with fixed hyperparameters was used to capture the cost distribution. Snapshots of the learning sequence are shown in Figure 1. Initially, the robot started with a low-stiffness policy and gradually adjusted the policy to increase the bracing stiffness. Although high-stiffness policies have low average cost since they often produce lower body accelerations, they run the risk of causing hardware damage due to increased strain on the arm joints. Thus, high-stiffness policies have high risk and the risk-averse optimization settled on a slightly higher expected cost, but lower risk policy. We collected 52 additional samples of randomly selected policies to verify that the learned policy is near-optimal for the $k = 2$ criterion (Figure 2). An example run of the learned policy is shown in Figure 3.

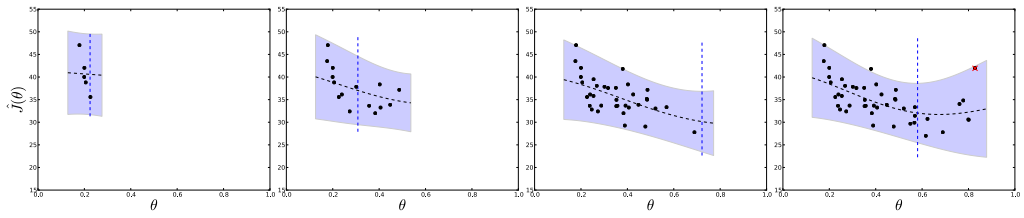


Figure 1: Snapshots of the learning sequence for risk-averse bracing. From left to right, $N = 5, 15, 35$, and 45. The vertical blue line indicates the nominal policy and the red data point indicates a hardware failure.

5.2 Dynamic Heavy Lifting

The next task we considered was lifting a 1 kg, partially-filled laundry detergent bottle from the ground to a height of about 120 cm. This problem is challenging for several reasons. First, the bottle is heavy, so most arm trajectories from the starting configuration to the goal will not succeed because of the limited torque generating capabilities of the arm motors. Second, the upper body motions act as disturbances to the LQR and violent lifting trajectories will cause the robot to destabilize. Finally, the bottle itself has significant dynamics because the heavy liquid sloshes as the bottle moves. Since the robot has only a simple claw gripper and we made no modifications to the bottle, the bottle moves freely in the hand, which we observed to have a significant effect on the stabilized system.

The policy was represented as a cubic spline trajectory in arm joint space with 7 open parameters that were learned by the algorithm. The parameters included 4 shoulder and elbow waypoint positions

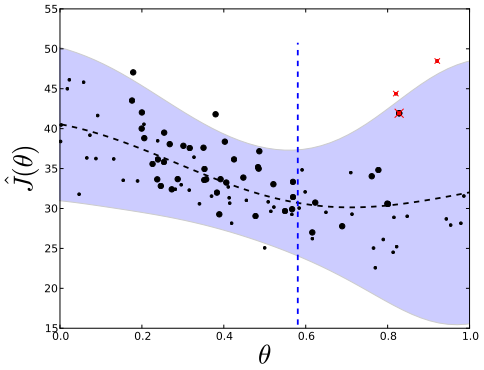


Figure 2: The cost distribution for bracing fit using 97 data points: 45 from the learning sequence (bold) and 52 from randomly selected policies. The vertical blue line indicates the final policy after 45 episodes of risk-averse ($\kappa = 2$) gradient descent. The red points indicate hardware failures.

Figure 3: Bracing policy execution after a large impact perturbation. Total duration of the above sequence is 0.7 seconds.

and 3 time parameters. Joint velocities at the waypoints were computed using the tangent method. The initial policy was a hand selected smooth and short duration motion to the goal configuration, such as a motion planner without detailed knowledge of the bottle might have produced. The initial policy succeeded only a fraction of the time, with most trials resulting in a failure to lift the bottle above the shoulder. The cost function was defined as

$$J(\boldsymbol{\theta}) = \int_{t=0}^{T} \left(\mathbf{x}(t)^\top \mathbf{Q}\mathbf{x}(t) + 0.01 I(t) V(t)\right) dt, \tag{12}$$

where $T = 6$ sec, $\mathbf{x} = [x_{wheel}, \dot{x}_{wheel}, \alpha_{body}, \dot{\alpha}_{body}, h_{error}]^\top$, $I(t)$ and $V(t)$ are total motor current and voltage for all motors at time t, and $\mathbf{Q} = diag([0.001, 0.001, 0.5, 0.5, 0.05])$. A trial ended when either $t > T$ or the robot reached the goal configuration with maintained low wheel velocity. The ERSAC parameter values in our experiments were $\eta = 0.5$, $\sigma = 0.075$, $\epsilon = 3.5\sigma$, and $\eta/\tilde{r}_{\boldsymbol{\theta}} \in [0.01, 0.5]$.

5.2.1 Risk-Neutral Learning

In the first experiment, we ran ERSAC with $\kappa = 0$. The VHGP model was used to locally construct the critic and model selection was performed using the NLOPT [35] implementation of SQP. A total of 30 trials were performed and a reliable, low expected cost policy was learned. Figure 4 illustrates the reduction in expected cost via empirical measurements taken at discrete times 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 response to correct the angular displacement of the body. Figure 5(a) shows an example run of the learned policy.

5.2.2 Variable Risk Control

Given that we can learn a low average cost policy in a small number of trials, we next examined the extent to which the policy could be adjusted on-the-fly to reflect different risk sensitivity. In particular, our experiments were aimed at generating translation-averse and energy-averse policies.

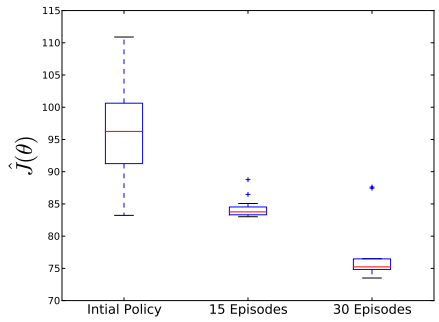
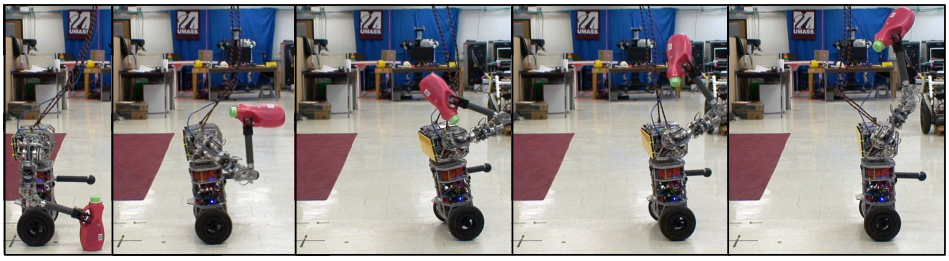
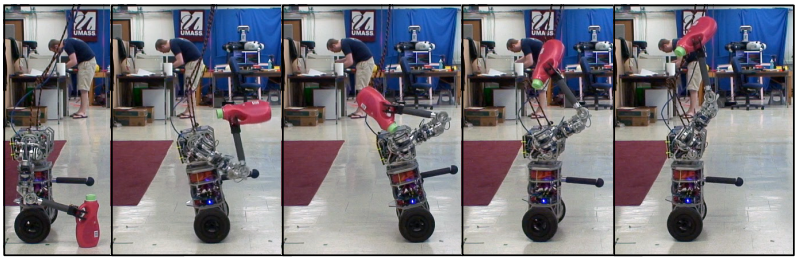


Figure 4: Data collected from 10 test trials executing the initial lifting policy, the policy after 15 episodes of learning, and the final policy after 30 episodes of learning.



(a)



(b)

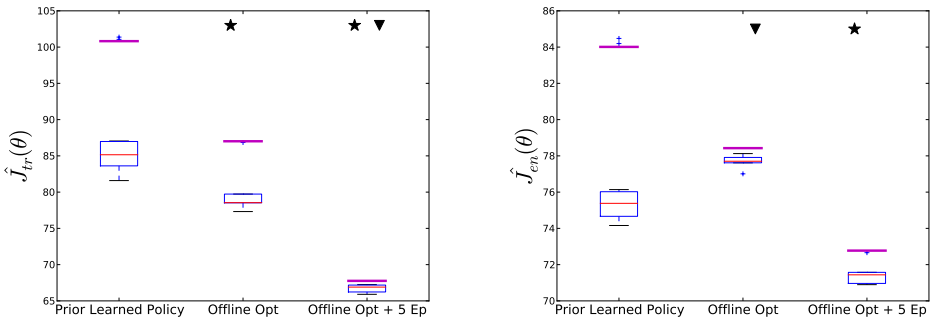
Figure 5: The learned risk-neutral policy (a) exploits the dynamics of the container to reliably perform the lifting task. With no additional learning trials, a risk-averse policy (b) is selected offline that reliably reduces translation. The total time duration of each of the above sequences is about 3 seconds.

Intuitively, these cases might correspond to when the robot's workspace is small, requiring that the policy that has a small footprint with high certainty, and when the battery charge is very low, requiring that the policy uses little energy with high certainty.

We represented a change in risk context by a reweighting of cost function terms. For example, to capture the low battery charge context, we simply increase the relative weight of the motor power term in eq. (12). We then recompute the cost of previous trajectories under this transformed cost function, $\hat{J}_{en}(\boldsymbol{\theta})$, and use SQP to minimize $\tilde{F}_{en}(\boldsymbol{\theta}, 2)$.

The result of applying offline policy selection for translation aversion is shown in Figure 6(a). With no additional trials, the system selects a policy that significantly reduces cumulative translation. An example run of the selected policy is shown in Figure 5(b). Using the translation-averse policy as a starting point, we performed an additional 5 episodes of risk-averse gradient descent. The result of

this short learning process was a very low average cost and low variance policy (see Figure 6(a)). We repeated this experiment for the energy-averse case and the result was very similar: the offline selected policy significantly increased performance with respect to the energy-averse criterion and 5 additional episodes of risk-averse online learning further increased performance leading to a very good policy (see Figure 6(b)).



(a) Translation-averse

(b) Energy-averse

Figure 6: Data from test runs of the prior learned policy, the offline selected risk-averse policy, and the policy after 5 episodes of risk-averse gradient descent. The solid magenta line corresponds to $\tilde{\mu} + 2\tilde{\sigma}$ computed using the test sample. 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 (Chi-squared, $p < 0.01$).

6 Conclusion

We presented an episodic policy search algorithm that efficiently descends the (natural) gradient of a risk-sensitive objective. Although the performance of the algorithm is not dependent on the state dimensionality, it is dependent on the dimensionality of the policy parameter space (as is generally the case with parameter perturbation algorithms [4, 5]). Thus, the expressiveness of a policy parameterization should be balanced with its parsimony to ensure the number of trials needed to find a suitable policy remains small.

We also showed that the local critic structure used in the update equation can be exploited to perform local offline policy optimization to rapidly change risk sensitivity in a completely model-free way. Most algorithms that could be used to model the local cost distribution require that assumptions be made regarding the smoothness of the expected cost and cost variance functions. Thus, care should be taken when selecting a critic structure so that, e.g., non-stationarity in the cost distribution is not overlooked. Our experiments suggest that the ERSAC algorithm is well suited for problems involving significant nonlinear dynamics and policy dependent noise, which may be large relative to the total magnitude of the cost.

References

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

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

[3] Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. 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, September 2004.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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