Variable Risk Dynamic Mobile Manipulation

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

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

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

***Abstract*—The ability to operate effectively in a variety of contexts will be a critical attribute of deployed mobile manipulators. In general, a variety of properties, such as battery charge, workspace constraints, and the presence of dangerous obstacles, will determine the suitability of particular control policies. Some context changes will cause shifts in *risk sensitivity*, or tendency to seek or avoid policies with high performance variation. We describe a policy search algorithm designed to address the problem of variable risk control. We generalize the simple stochastic gradient descent update to the risk-sensitive case, 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. We evaluate the algorithm in experiments with a dynamically stable mobile manipulator lifting a heavy liquid-filled bottle while balancing.**

I. Introduction

Many interesting manipulation tasks involve a controlled interaction between an underactuated robot and a physical object in the environment that has significant dynamic properties of its own. Model-based techniques, such as motion planning [17, 29], can often be used to generate solutions to such problems, even if the model of the system and/or object is only approximate. However, to achieve high performance solutions that exploit subtle interactions between the dynamics of the robot and its environment, we typically must resort to online optimization procedures to improve on solutions produced by motion planners. For this reason, model-free policy search methods have become one of the standard tools for developing controllers in robot systems [23, 14, 26, 30, 12].

Mobile manipulators offer a somewhat more challenging setting for this type of policy optimization. A fundamental characteristic of these systems is that they must operate effectively in a variety of (possibly rapidly changing) contexts: in the laboratory vs. a crowded hallway, with high vs. low battery charge, with a nearly overheated elbow motor, or under environmentally-imposed time constraints. In many cases, different contexts will demand different sensitivity toward variation in performance, or *risk*. We aim to achieve the fine-tuned performance that policy search methods can produce while introducing the ability to adjust the system's risk sensitivity based on runtime context.

We present an efficient risk-sensitive policy search algorithm based on stochastic gradient descent. The algorithm shares several properties (such as scalability, local convergence, sample efficiency) with existing risk-neutral policy gradient algorithms that have been shown to perform well in robot learning tasks [25, 23]. We 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 results from a lifting experiment with a real mobile manipulator that demonstrate the ability to learn a policy that exploits dynamic interactions between the robot and manipulated object that would be very difficult to model. We also show how the learned policy can be adjusted at runtime to produce policies with different spatial and energetic risk sensitivity.

II. Related Work

Early work in risk-sensitive control was aimed at finding solutions to discrete Markov decision processes (MDPs) [8] and linear-quadratic-Gaussian problems [9, 33] 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 [3]. For continuous problems, Van den Broek et al. [32] generalized path integral methods to risk-sensitive stochastic optimal control. In our recent work [16], we 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 [5, 20, 21], which can be combined with policy selection criteria that take return variance into account. Heger [7] derived a worst-case Q-learning algorithm based on a minimax criterion. Mihatsch and Neuneier [19] 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. This algorithm was recently found to be consistent with behavioral and neurological measurements of humans learning a decision task that involving risky outcomes [22]. Recent motor control experiments suggest that humans select motor strategies in a risk-sensitive way [4].

Our contribution to this literature is an episodic risk-sensitive policy gradient algorithm that is sample-efficient and appropriate for domains that are continuous, noisy, and high-dimensional. Furthermore, our proposed method supports interweaving of offline optimization with online gradient descent to select local greedy optimal policies or adaptively change risk sensitivity.

III. 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, $\varepsilon_{\boldsymbol{\theta}} \sim \mathcal{N}(0, r_{\boldsymbol{\theta}}^2)$, is a function of the policy parameters. This policy-dependent noise is critical since, in general, the variance of the cost signal will not be constant across the policy space. For example, in 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

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

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

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.

IV. Episodic Risk-Sensitive Actor-Critic

Our goal is to perform the minimization (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 [28, 13, 25], so we focus on extending this approach to the risk-sensitive case.

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

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

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 $\tilde{J}_{\boldsymbol{\theta}}$ and $\tilde{r}_{\boldsymbol{\theta}}$ are estimates of the cost mean and standard deviation, respectively. Substituting (1) into (4) and taking the first order Taylor expansion, we have

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

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

$$\mathbb{E}[\Delta\boldsymbol{\theta}] = -\frac{\eta}{\tilde{r}_{\boldsymbol{\theta}}} \sigma^2 \nabla_{\boldsymbol{\theta}} \left(J_{\boldsymbol{\theta}} + \kappa\tilde{r}_{\boldsymbol{\theta}} \right). \tag{5}$$

Thus, (4) is an estimator of the gradient of expected cost that is biased by the estimated gradient of standard deviation, where the magnitude and direction of this bias is determined by the risk sensitivity parameter, κ.

If the estimator of the cost standard deviation is unbiased, we have

$$\mathbb{E}[\Delta\boldsymbol{\theta}] = -\frac{\eta}{r_{\boldsymbol{\theta}}} \sigma^2 \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \kappa), \tag{6}$$

a scaled unbiased estimate of the gradient of the risk-sensitive objective (3). Intuitively, (4) 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.

From (6) 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 [25], 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:

$$\mathbb{E}[\Delta\boldsymbol{\theta}] = -\frac{\eta}{r_{\boldsymbol{\theta}}} \boldsymbol{\Sigma} \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \kappa). \tag{7}$$

However, this is in the direction of the *natural gradient* [1]. 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 [1]. When the policy sampling distribution is mean-zero Gaussian with covariance $\boldsymbol{\Sigma}$, the inverse Fisher information matrix is $\mathbf{G}^{-1} = \boldsymbol{\Sigma}$.

A. Critic Representation

The update (4) 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 [2, 15]. 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 [6, 11, 31, 27, 34]. In our experiments, we used the Variational Heteroscedastic Gaussian Process (VHGP) model [18], 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 [18].

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 algorithm (ERSAC) is outlined in Algorithm 1.

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:*
      $\tilde{J}_{\boldsymbol{\theta}} = \mathbb{E}[\hat{J}_{\boldsymbol{\theta}} \mid \mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
      $\tilde{r}^2_{\boldsymbol{\theta}} = \mathbb{V}[\hat{J}_{\boldsymbol{\theta}} \mid \mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
      $\tilde{r}^2_{\boldsymbol{\theta}+\mathbf{z}} = \mathbb{V}[\hat{J}_{\boldsymbol{\theta}+\mathbf{z}} \mid \mathbf{X}_{\text{loc}}, \mathbf{y}_{\text{loc}}]$
      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** $\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}$

The local VHGP 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 V, 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. The local offline policy optimization procedure is illustrated in Algorithm 2.

Algorithm 2 Offline local policy optimization

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

B. Example

Figure 1 illustrates example runs of the above algorithms using the synthetic cost distribution in Figure 1(a). Figure 1(b) shows the result of applying the ERSAC algorithm with a risk-averse objective, $\kappa = 2$. The algorithm descends the gradient of the upper confidence bound to a local minimum while maintaining a reasonable local approximation of the cost distribution.

Figure 1(c) shows the result of applying offline local policy optimization using the local estimate of the cost distribution obtained during gradient descent. By performing an offline optimization using a risk-neutral objective, the algorithm directly selects a near-optimal average cost policy. Changing the value of the risk parameter in the offline optimization objective leads to selection of local risk-averse ($\kappa = 2$) and risk-seeking ($\kappa = -2$) objectives.

V. Experiments

We performed experiments with the uBot-5, a dynamically balancing mobile manipulator designed at the University of Massachusetts Amherst. The task we considered was lifting a 1 kg, partially-filled laundry detergent bottle from the ground to a height of about 120 cm (the robot's shoulder height above the ground is 60 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 robot balances using a simple linear-quadratic regulator (LQR) that models the upper body as a fixed mass. Thus, upper body motions act as disturbances to the stabilized system and violent lifting trajectories will cause the robot to fall. Finally, the bottle itself has significant dynamics since 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. These features make this problem well suited to a model-free policy search approach.

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

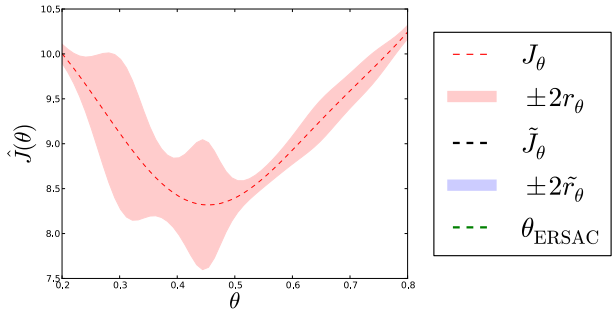
The cost function was defined as

$$J = \int_0^T \left(\mathbf{x}^\top \mathbf{Q} \mathbf{x} + cI(t)V(t) \right) dt, \tag{8}$$

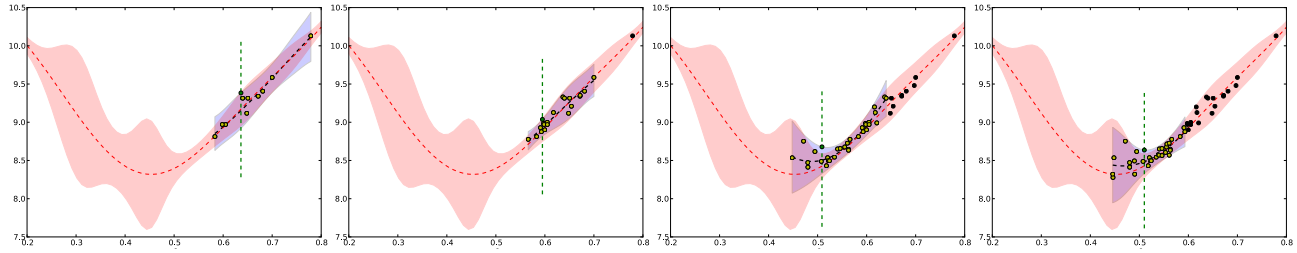
where $\mathbf{x} = [x_{wheel}, \dot{x}_{wheel}, \alpha_{body}, \dot{\alpha}_{body}, h_{error}]^\top$, $I(t)V(t)$ is the power being consumed by all motors at time t, $\mathbf{Q} = diag([0.001, 0.001, 0.5, 0.5, 0.05])$, and $c = 0.01$. The components of the state vector are the wheel position and velocity, body angle and angular velocity, and vertical error between the desired and actual bottle position, respectively. Intuitively, a cost function of this form encourages fast and energy efficient solutions that do not violently perturb the LQR. In each trial, the servo 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 wheel velocity. The parameter values in all experiments were $\eta = 0.5, \sigma = 0.075, \epsilon = 3.5\sigma$, and $\eta/\tilde{r}_{\boldsymbol{\theta}} \in [0.01, 0.5]$.

A. Risk-Neutral Learning

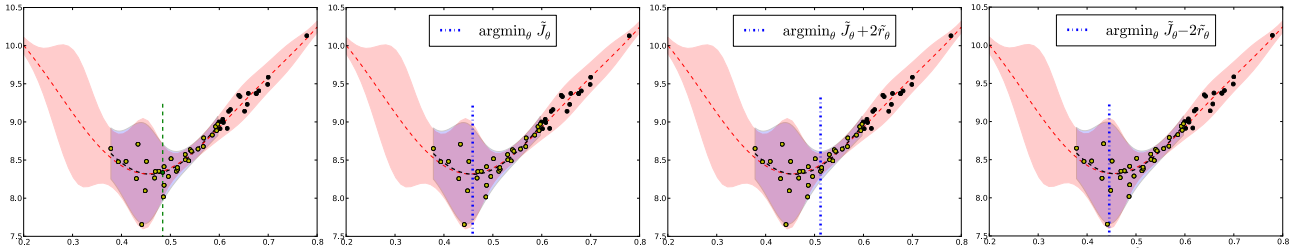
In the first experiment, we ran ERSAC with $\kappa = 0$. The VHGP model was used to locally construct the critic and model selection was performed using the NLOPT [10] implementation of SQP. A total of 30 trials (less than 2.5



(a) Example synthetic cost distribution



(b) Risk-averse gradient decent using ERSAC



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

Fig. 1. Figure (b) illustrates how risk-averse stochastic gradient descent descends the upper confidence bound of a synthetic cost distribution, (a). Subfigure (c) shows the result of performing offline local optimization using different risk-sensitive objectives given the local distribution learned during risk-neutral gradient descent.

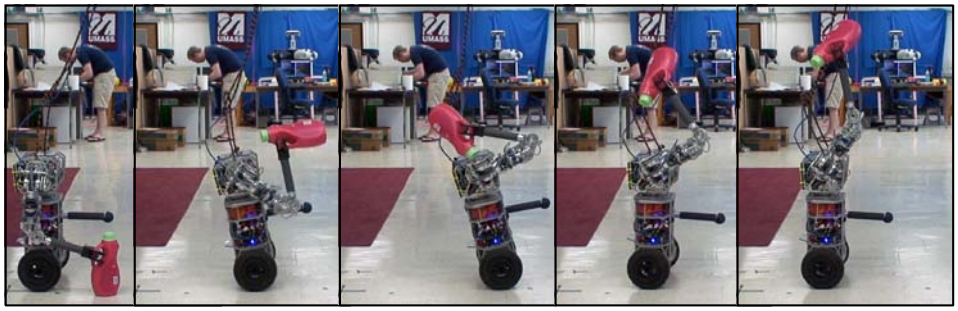
minutes of robot time) were performed and a reliable, low-cost policy was learned. Figure 2 illustrates the reduction in cost via empirical measurements taken at discrete times during learning. Interestingly, *the learned policy exploits the dynamics of the liquid in the bottle* by timing the motion such that the shifting bottle contents coordinate with the LQR controller to correct the angular displacement of the body. Figure 3(a) shows an example run of the learned policy.

B. Variable Risk Control

Given that we can learn a high performance policy in a small number of trials, we next examined the extent to which the policy could be adjusted on-the-fly to maintain high performance in different operating contexts. Our experiments were aimed at generating translation risk-averse and energy risk-averse policies. 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 very little energy with high certainty.

We represented a change in context by a reweighting of cost function terms. To capture the low battery charge context, we increased the relative weight of the motor power term in (8): $\mathbf{Q} = diag([0.0005, 0.0005, 0.25, 0.25, 0.05])$ and $c = 0.1$. We then recomputed the cost of previous trajectories under this transformed cost function, $\hat{J}_{en}(\boldsymbol{\theta})$, and used SQP to minimize $\tilde{F}_{en}(\boldsymbol{\theta}, 2)$. Likewise, to represent the translation risk-averse case we increased the relative weight

(a) Learned risk-neutral policy



(b) Translation risk-averse policy

Fig. 3. After 30 episodes of policy search, a risk-neutral policy (a) is learned that 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 the above sequences is about 3 seconds.

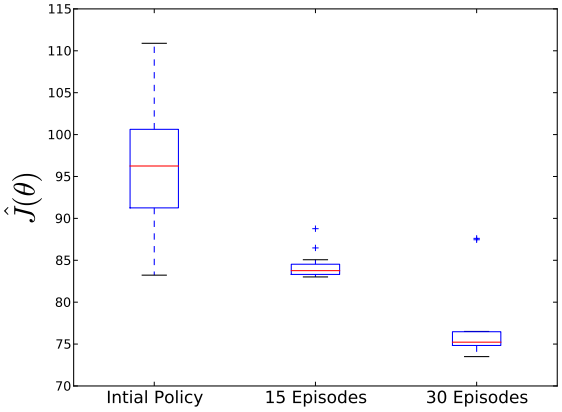


Fig. 2. Data collected from 10 test trials executing the initial lifting policy, the policy after 15 episodes of risk-neutral learning, and the final policy after 30 episodes of learning.

assigned to wheel translation in the cost function ($\mathbf{Q} = diag([0.002, 0.001, 0.5, 0.5, 0.05])$ and $c = 0.001$) and minimized $\tilde{F}_{tr}(\boldsymbol{\theta}, 2)$ offline.

The result of applying offline policy selection for translation risk-aversion is shown in Figure 4. With no additional trials, the system selected a policy that significantly reduced cumulative translation. An example run of the selected policy is shown in Figure 3(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, low variance policy (see Figure 4).

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

VI. Discussion

We presented a policy search algorithm that efficiently descends the (natural) gradient of a risk-sensitive objective. Although we focused on a particular manipulation task in our experiments, the ERSAC algorithm has much broader applicability to problems involving complex nonlinear dynamics, high-dimensionality, and policy dependent noise that may be large relative to the total magnitude of the cost. Since the algorithm performs local exploration, the quality of the final solution will depend on the initial policy. It is therefore good practice to combine such algorithms with methods for generating approximate initial solutions (e.g., sampling-based motion planning) when possible.

Although the performance of the algorithm is not dependent on the state dimensionality, it is dependent on the dimension-

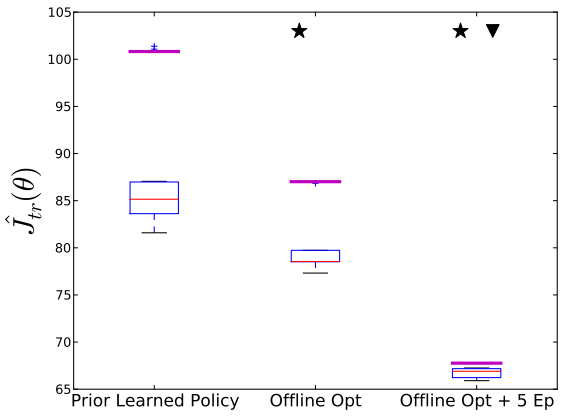


Fig. 4. Data collected from test runs of the previously learned policy, the offline selected translation risk-averse policy, and the policy after 5 episodes of risk-averse gradient descent. The solid magenta line corresponds to $\hat{\mu} + 2\hat{\sigma}$ computed using the test data. A star at the top of a column signifies a statistically significant reduction in the sample mean compared with the previous column (Behrens-Fisher, $p < 0.01$) and a triangle signifies a statistically significant reduction in the sample variance (Chi-squared, $p < 0.01$).

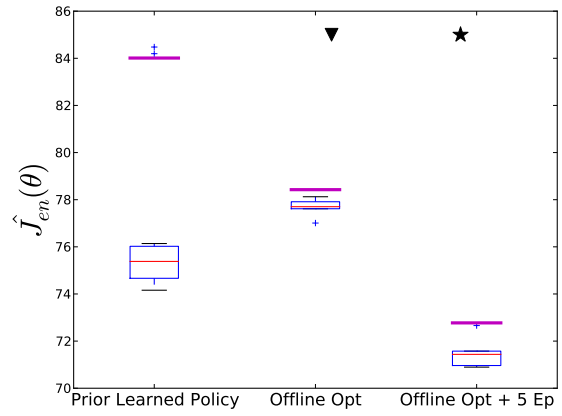


Fig. 5. Data collected from test runs of the previously learned policy, the offline selected energy risk-averse policy, and the policy after 5 episodes of risk-averse gradient descent.

ality of the policy parameter space (as is generally the case with parameter perturbation algorithms [25, 23]). Thus, the expressiveness of policy parameterizations should be balanced with their parsimony to ensure the number of trials needed to find a suitable policy remains small. In our experiments, we used a simple cubic spline parameterization, but more general closed-loop policy classes are possible [24]. The development of methods for autonomously identifying suitable local policy representations based on planned motions is an interesting possible direction for future work.

We 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 model-free way. Most algorithms that could be used to capture 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.

VII. Conclusion

Mobile systems designed meet manipulation and mobility objectives in many contexts must be adaptive—exploiting prior experience to make rapid adjustments to learned policies. In particular, some situations will require a non-neutral attitude toward risk. We examined this problem in the general context of model-free policy search. Our results demonstrate the potential for efficient online learning of a dynamically complex task and runtime adjustment of risk sensitivity in response to context changes.

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.

References

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

[2] 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.

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

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

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

[6] 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.

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

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

[9] 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.

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

[11] 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.

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

[13] 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.

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

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

[16] 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.

[17] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

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

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

[27] 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.

[28] 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.

[29] Russ Tedrake, Ian R. Manchester, Mark M. Tobenkin, and John W. Roberts. LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052, July 2010.

[30] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. 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, May 2010.

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

[32] 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.

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

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