Adaptive Smoothing Path Integral Control

Anonymous

Abstract

In Path Integral control problems a representation of an optimally controlled dynamical system can be formally computed and serve as a guidepost to learn a parametrized policy. The Path Integral Cross-Entropy (PICE) method tries to exploit this, but is hampered by poor sample efficiency. We propose a model-free algorithm called ASPIC (Adaptive Smoothing of Path Integral Control) that applies an inf-convolution to the cost function to speedup convergence of policy optimization. We identify PICE as the infinite smoothing limit of such technique and show that the sample efficiency problems that PICE suffers disappear for finite levels of smoothing. For zero smoothing this method becomes a greedy optimization of the cost, which is the standard approach in current reinforcement learning. We show analytically and empirically that intermediate levels of smoothing are optimal, which renders the new method superior to both PICE and direct cost-optimization.

1 Introduction

Optimal control of non-linear dynamical systems that are continuous in time and space is hard. Methods that have proven to work well introduce a parametrized policy like a neural network [16, 5] and directly optimize the expected cost using gradient descent [27, 19, 22, 11]. To achieve a robust decrease of the expected cost, it is important to ensure that at each step the policy stays in the proximity of the old policy [5]. This can be achieved by enforcing a trust region constraint [18, 22] or using adaptive regularization [11]. However the applicability of these methods is limited, as in each iteration of the algorithm, samples from the controlled system have to be computed. We want to increase the convergence rate of policy optimization to reduce the number of simulations needed.

To this end we consider Path Integral control problems [12], that offer an alternative approach to direct cost optimization and explore if this allows to speed up policy optimization. This class of control problems permits arbitrary non-linear dynamics and state cost, but requires a linear dependence of the control on the dynamics and a quadratic control cost [12, 2, 25]. These restrictions allow to obtain an explicit expression for the probability-density of optimally controlled system trajectories. Through this, an information-theoretical measure of the deviation of the current control policy from the optimal control can be calculated. The Path Integral Cross-Entropy (PICE) method [13] proposes to use this measure as a pseudo-objective for policy optimization.

In this work we analyze a new kind of smoothing technique for the cost function based on recently proposed smoothing techniques to speed up convergence in deep neural networks [4]. We adapt this technique to Path Integral control problems and show that *(i)*, in contrast to [4], smoothing in Path Integral control can be solved analytically, providing an expression of the gradient that can directly be computed from Monte Carlo samples and *(ii)*, we can interpolate between direct cost optimization and the PICE objective. Remarkably, the parameter governing the smoothing can be determined independently of the number of samples.

Based on these results, we introduce the ASPIC (Adaptive Smoothing of Path Integral Control) algorithm, a model-free algorithm that uses cost smoothing to speed up policy optimization. ASPIC adjusts the smoothing parameter in each step to keep the variance of the gradient estimator at a predefined level.

Preprint. Under review.

2 Path Integral Control Problems

Consider the (multivariate) dynamical system

$$\dot{x}_t = f(x_t, t) + g(x_t, t)\left(u(x_t, t) + \xi_t\right), \tag{1}$$

with initial condition x_0. The control policy is implemented in the control function $u(x, t)$, which is additive to the white noise ξ_t which has variance $\frac{\nu}{dt}$. Given a control function u and a time horizon T, this dynamical system induces a probability distribution $p_u(\tau)$ over state trajectories $\tau = \{x_t | \forall t : 0 < t \leq T\}$ with initial condition x_0.

We define the regularized expected cost

$$C(p_u) = \langle V(\tau) \rangle_{p_u} + \gamma KL(p_u || p_0), \tag{2}$$

with $V(\tau) = \int_0^T V(x_t, t) dt$, where the strength of the regularization $KL(p_u||p_0)$ is controlled by the parameter γ.

The Kullback-Leibler divergence $KL(p_u||p_0)$ puts high cost to controls u that bring the probability distribution p_u far away from the uncontrolled dynamics p_0 where $u(x_t, t) = 0$. We can also rewrite the regularizer $KL(p_u||p_0)$ directly in terms of the control function u by using the Girsanov theorem, c.f., [25]: $\log \frac{p_u(\tau)}{p_0(\tau)} = \frac{1}{\nu} \int_0^T \left(\frac{1}{2} u(x_t, t)^T u(x_t, t) + u(x_t, t)^T \xi_t\right) dt$. The regularization then takes the form of a quadratic control cost

$$KL(p_u||p_0) = \left\langle \frac{1}{\nu} \int_0^T \left(\frac{1}{2} u(x_t, t)^T u(x_t, t) + u(x_t, t)^T \xi_t\right) dt \right\rangle_{p_u} = \left\langle \frac{1}{\nu} \int_0^T \frac{1}{2} u(x_t, t)^T u(x_t, t) dt \right\rangle_{p_u},$$

where we used that $\left\langle u(x_t, t)^T \xi_t \right\rangle_{p_u} = 0$. This shows that the regularization $KL(p_u||p_0)$ puts higher cost for large values of the controller u.

The Path Integral control problem is to find the optimal control function u^* that minimizes

$$u^* = \arg\min_u C(p_u). \tag{3}$$

For a more complete introduction to Path Integral control problems, see [25, 13].

$-$**Direct cost optimization using gradient descent**: A standard approach to find an optimal control function is to introduce a parametrized controller $u_\theta(x_t, t)$ [11, 27, 22]. This parametrizes the path probabilities p_{u_θ} and allows to optimize the expected cost $C(p_{u_\theta})$ (2) using stochastic gradient descent on the cost function:

$$\nabla_\theta C(p_{u_\theta}) = \left\langle S^\gamma_{p_{u_\theta}}(\tau) \nabla_\theta \log p_{u_\theta}(\tau) \right\rangle_{p_{u_\theta}}, \tag{4}$$

with the stochastic cost $S^\gamma_{p_{u_\theta}}(\tau) \coloneqq V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)}$ (see App. A for details).

$-$**The Path Integral Cross-Entropy method**: An alternative approach to direct cost-optimization was introduced in [13], and takes advantage of the analytical expression for p_{u^*}, the probability density of state trajectories induced by a system with the optimal controller u^*, $p_{u^*} = \arg\min_{p_u} C(p_u)$ with $C(p_u)$ given by Eq. (2). Finding p_{u^*} is an optimization problem over probability distributions p_u that are induced by the controlled dynamical system (1). It has been shown [2, 25] that we can solve this by replacing the minimization over p_u with a minimization over all path probability distributions p:

$$p_{u^*} \equiv p^* := \arg\min_p C(p) = \arg\min_p \langle V(\tau) \rangle_p + \gamma KL(p||p_0) = \frac{1}{Z} p_0(\tau) \exp\left(-\frac{1}{\gamma} V(\tau)\right). \tag{5}$$

with the normalization constant $Z = \left\langle \exp\left(-\frac{1}{\gamma} V(\tau)\right) \right\rangle_{p_0}$. Note that the above is not a trivial statement, as we now take the minimum also over non-Markovian processes with non-Gaussian noise.

The PICE algorithm [13], instead of directly optimizing the expected cost, it minimizes the KL-divergence $KL\left(p^* || p_{u_\theta}\right)$ which measures the deviation of a parametrized distribution p_{u_θ} from the optimal one p^*. Although direct cost optimization and PICE are different methods, their global

minimum is the same if the parametrization of u_θ can express the optimal control $u^* = u_{\theta^*}$. The parameters θ^* of the optimal controller are found using gradient descent:

$$\nabla_\theta KL\left(p^*||p_{u_\theta}\right) = \frac{1}{Z_{p_{u_\theta}}} \left\langle \exp\left(-\frac{1}{\gamma} S^\gamma_{p_{u_\theta}}(\tau)\right) \nabla_\theta \log p_{u_\theta}(\tau) \right\rangle_{p_{u_\theta}}, \tag{6}$$

where $Z_{p_{u_\theta}} \coloneqq \left\langle \exp\left(-\frac{1}{\gamma} S^\gamma_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}$.

That PICE uses the optimal density as a guidepost for the policy optimization might give it an advantage compared to direct cost-optimization. In practice however, this method only works properly if the initial guess of the controller u_θ does not deviate too much from the optimal control, as a high value of $KL\left(p^*||p_{u_\theta}\right)$ leads to a high variance of the gradient estimator and results in bootstrapping problems of the algorithm [21, 24]. In the next section, we introduce a method that interpolates between direct cost-optimization and the PICE method, allowing us to take advantage of the analytical optimal density without being hampered by the same bootstrapping problems as PICE.

3 Interpolating Between Methods: Smoothing Stochastic Control Problems

Cost function smoothing was recently introduced as a way to speed up optimization of neural networks [4]: Optimization of a general cost function $f(\theta)$ can be speeded up by smoothing $f(\theta)$ using an inf-convolution with a distance kernel $d(\theta', \theta)$. The smoothed function

$$J^\alpha(\theta) = \inf_{\theta'} \alpha d(\theta', \theta) + f(\theta') \tag{7}$$

preserves the global minima of the function $f(\theta)$. To apply gradient descent based optimization on $J^\alpha(\theta)$ instead of $f(\theta)$ may significantly speed up convergence [4].

We want to use this accelerative effect to find the optimal parametrization of the controller u_θ. Therefore, we smooth the cost function $C(p_{u_\theta})$ as a function of the parameters θ. As $C(p_{u_\theta}) = \langle V(\tau)\rangle_{p_{u_\theta}} + \gamma KL(p_{u_\theta}||p_0)$ is a functional on the space of probability distributions p_{u_θ}, the natural "distance" is the KL-divergence $KL(p_{u_{\theta'}}||p_{u_\theta})$. So we replace

$$\begin{aligned} f(\theta) &\to C(p_{u_\theta}) \\ d(\theta', \theta) &\to KL(p_{u_{\theta'}}||p_{u_\theta}), \end{aligned}$$

and obtain the smoothed cost $J^\alpha(\theta)$ as

$$J^\alpha(\theta) = \inf_{\theta'} \alpha KL(p_{u_{\theta'}}||p_{u_\theta}) + C(p_{u_{\theta'}}) = \inf_{\theta'} \alpha KL(p_{u_{\theta'}}||p_{u_\theta}) + \gamma KL(p_{u_{\theta'}}||p_0) + \langle V(\tau)\rangle_{p_{u_{\theta'}}}. \tag{8}$$

Note the different roles of α and γ: the parameter α penalizes the deviation of $p_{u_{\theta'}}$ from p_{u_θ}, while the parameter γ penalizes the deviation of $p_{u_{\theta'}}$ from the uncontrolled dynamics p_0.

– **Computing the smoothed cost and its gradient**: The smoothed cost J^α is expressed as a minimization problem that has to be solved. Here we show that for Path Integral control problems this can be done analytically. To do this we first show that we can replace $\inf_{\theta'} \to \inf_{p'}$ and then solve the minimization over p' analytically. We replace the minimization over θ' by a minimization over p' in two steps: first we state an assumption that allows us to replace $\inf_{\theta'} \to \inf_{u'}$ and then proof that for Path Integral control problems we can replace $\inf_{u'} \to \inf_{p'}$.

We assume that for every u_θ and any $\alpha > 0$, the minimizer $\theta^*_{\alpha,\theta}$ over the parameter space

$$\theta^*_{\alpha,\theta} \coloneqq \arg\min_{\theta'} \alpha KL(p_{u_{\theta'}}||p_{u_\theta}) + C(p_{u_{\theta'}}) \tag{9}$$

is the parametrization of the minimizer $u^*_{\alpha,\theta}$ over the function space

$$u^*_{\alpha,\theta} \coloneqq \arg\min_{u'} \alpha KL(p_{u'}||p_{u_\theta}) + C(p_{u'}),$$

such that $u^*_{\alpha,\theta} \equiv u_{\theta^*_{\alpha,\theta}}$. We call this assumption *full parametrization*. Naturally it is sufficient for full parametrization if $u_\theta(x,t)$ is a universal function approximator with a fully observable state space x and the time t as input, although this may be difficult to achieve in practice. With this assumption

we can replace $\inf_{\theta'} \to \inf_{u'}$. Analogously, we replace $\inf_{u'} \to \inf_{p'}$: in App. B.1 we proof that for Path Integral control problems the minimizer $u^*_{\alpha,\theta}$ over the function space induces the minimizer $p^*_{\alpha,\theta}$ over the space of probability distributions

$$p^*_{\alpha,\theta} \coloneqq \arg\min_{p'} \alpha KL(p'||p_{u_\theta}) + C(p'), \tag{10}$$

such that $p^*_{\alpha,\theta} \equiv p_{u^*_{\alpha,\theta}}$. This step is similar to the derivation of of Eq. (5) in Section 2, but now we have added an additional term $\alpha KL(p_{u'}||p_{u_\theta})$.

Hence, given a Path Integral control problem and a controller u_θ that satisfies full parametrization, we can replace $\inf_{\theta'} \to \inf_{p'}$ and Eq. (8) becomes

$$J^\alpha(\theta) = \inf_{p'} \alpha KL(p'||p_{u_\theta}) + \gamma KL(p'||p_0) + \langle V(\tau) \rangle_{p'} \,. \tag{11}$$

This can be solved directly: first we compute the minimizer (see App. B.2)

$$p^*_{\alpha,\theta}(\tau) = \frac{1}{Z^\alpha_{p_{u_\theta}}} p_{u_\theta}(\tau) \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau)\right), \quad Z^\alpha_{p_{u_\theta}} = \left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}. \tag{12}$$

We plug this back in Eq. (11) and get the smoothed cost and its gradient (see App. B.3)

$$J^\alpha(\theta) = -(\gamma+\alpha) \log \left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}} \tag{13}$$

$$\nabla_\theta J^\alpha(\theta) = -\frac{\alpha}{Z^\alpha_{p_{u_\theta}}} \left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau)\right) \nabla_\theta \log p_{u_\theta}(\tau)\right\rangle_{p_{u_\theta}}. \tag{14}$$

Both can be estimated by samples from the distribution p_{u_θ}.

4 The ASPIC Algorithm

In this section, we derive an iterative algorithm that takes a parametrized control function u_θ and performs smooth parameter updates starting from initial parameters θ_0. We focus on the effect that a finite $\alpha > 0$ has on the iterative optimization of the control u_θ for a fixed value of γ. For our theoretical results, we refer the reader to App. C, where we identify several existing settings as limiting cases of the parameters α and γ, and to App. D, where we proof that smooth updates are optimal in two-step sequential decision problems.

To simplify notation, we overload $p_{u_\theta} \to \theta$ so that we get $C(p_{u_\theta}) \to C(\theta)$ and $KL(p_{u_{\theta'}}||p_{u_\theta}) \to KL(\theta'||\theta)$. We use a trust region constraint to robustly optimize the policy, c.f., [18, 22, 10]. We define the smoothed update with stepsize $\mathcal{E}$ as an update $\theta \to \theta'$ with $\theta' = \Theta^{J^\alpha}_\mathcal{E}(\theta)$ and

$$\Theta^{J^\alpha}_\mathcal{E}(\theta) \coloneqq \underset{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta) \leq \mathcal{E}}}{\arg\min} J^\alpha(\theta'). \tag{15}$$

−Smoothed and direct updates using natural gradients: We first express the constraint optimization (15) as an unconstrained optimization problem introducing a Lagrange multiplier β

$$\theta_{n+1} = \arg\min_{\theta'} J^\alpha(\theta') + \beta KL(\theta'||\theta_n). \tag{16}$$

Following [22] we assume that the trust region size $\mathcal{E}$ is small. For $\mathcal{E} \ll 1$ we get $\beta \gg 1$ and can expand $J^\alpha(\theta')$ to first and $KL(\theta'||\theta_n)$ to second order (see App. E.1 for the details). This gives

$$\theta_{n+1} = \theta_n - \beta^{-1} F^{-1} \left.\nabla_{\theta'} J^\alpha(\theta')\right|_{\theta'=\theta_n}, \tag{17}$$

a natural gradient update with the Fisher-matrix $F = \nabla_\theta \nabla^T_\theta KL(\theta'||\theta_n)\big|_{\theta'=\theta_n}$ (we use the conjugate gradient method to approximately compute the natural gradient for high dimensional parameter spaces. See App. E.2 or [22] for details). Parameter β is determined using a line search such that

$$KL(\theta_n||\theta_{n+1}) = \mathcal{E}. \tag{18}$$

Note that for direct updates this derivation is the same, just replace J^α by C.

$-$**Reliable gradient estimation using adaptive smoothing**: To compute smoothed updates using Eq. (17) we need the gradient of the smoothed cost. We assume full parametrization and use Eq. (14), which can be estimated using N weighted samples drawn from the distribution p_{u_θ}:

$$\nabla_\theta J^\alpha(\theta) \approx \alpha \sum_{i=1}^{N} w^i \log p_{u_\theta}(\tau^i). \tag{19}$$

The weights are given by

$$w^i = \frac{1}{\tilde{Z}} \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau^i)\right), \qquad \tilde{Z} = \sum_{i=1}^{N} \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau^i)\right).$$

The variance of this estimator depends sensitively on the entropy of the weights $H_N(w) = -\sum_{i=1}^N w^i \log(w^i)$. If the entropy is low, the total weight is concentrated on a few particles. This results in a poor gradient estimator where only a few of the particles actually contribute. This concentration is dependent on the smoothing parameter α: for small α, the weights are very concentrated in a few samples, resulting in a large weight-entropy and thus a high variance of the gradient estimator. As small α corresponds to strong smoothing, we want α to be as small as possible, but large enough to allow a reliable gradient estimation. Therefore, we set a bound to the weight entropy $H_N(w)$. To get a bound that is independent of the number of samples N, we use that in the limit of $N \to \infty$ the weight entropy is monotonically related to the KL-Divergence $KL(p^*_{\alpha,u_\theta}||p_{u_\theta})$

$$KL(p^*_{\alpha,u_\theta}||p_{u_\theta}) = \lim_{N\to\infty} \log N - H_N(w)$$

(see App. E.3). This provides a method for choosing α *independently of the number of samples*: we set the constraint $KL(p^*_{\alpha,u_\theta}||p_{u_\theta}) \leq \Delta$ and determine the smallest α that satisfies this condition by using a line search. Large values of Δ correspond to small values of α (see App. E.4) and therefore strong smoothing, we thus call parameter Δ the *smoothing strength*.

$-$**A model-free algorithm**: We can compute the gradient (19) and the KL-divergence while treating the dynamical system as a black-box. For this we write the probability distribution p_{u_θ} over trajectories τ as a Markov process $p_{u_\theta}(\tau) = \prod_{0<t<T} p_{u_\theta}(x_{t+dt}|x_t, t)$, where the product runs over the time t, which is discretized with time step dt. We define the noisy action $a_t = u(x_t, t) + \xi_t$ and formulate the transitions $p_{u_\theta}(x_{t+dt}|x_t)$ for the dynamical system (1) as

$$p_{u_\theta}(x_{t+dt}|x_t) = \delta\left(x_{t+dt} - \mathcal{F}\left(x_t, a_t, t\right)\right) \cdot \pi_\theta(a_t|t, x_t),$$

with $\delta(\cdot)$ the Dirac delta function. This splits the transitions up into the deterministic dynamical system $\mathcal{F}\left(x_t, a_t, t\right)$ and a Gaussian policy $\pi_\theta(a_t|t, x_t) = \mathcal{N}\left(a_t|u_\theta(x_t, t), \frac{\nu}{dt}\right)$ with mean $u_\theta(x_t, t)$ and variance $\frac{\nu}{dt}$. Using this we get a simplified expression for the gradient of the smoothed cost (19) that is independent of the system dynamics, given the samples drawn from the controlled system p_{u_θ}:

$$\nabla_\theta J^\alpha(\theta) \approx \alpha \sum_{i=1}^{N} \sum_{0<t<T} w^i \nabla_\theta \log \pi_\theta(a^i_t|t, x^i_t).$$

Similarly we obtain an expression for the estimator of the KL divergence $KL(\theta_n||\theta_{n+1}) \approx \frac{1}{N}\sum_{i=1}^N \sum_{0<t<T} \log \frac{\pi_{\theta_n}(a^i_t|t,x^i_t)}{\pi_{\theta_{n+1}}(a^i_t|t,x^i_t)}$. With this we formulate ASPIC (Algorithm 1) which optimizes the parametrized policy π_θ by iteratively drawing samples from the controlled system.

5 Numerical Experiments

We compare experimentally the convergence speed of policy optimization with and without smoothing. For the optimization with smoothing, we use ASPIC and for the optimization without smoothing, we use a version of ASPIC where we replaced the gradient of the smoothed cost with the gradient of the cost itself. We consider three non-linear control problems, which violate the full parametrization assumption (pendulum swing-up task, Acrobot, and 2D walker). The latter was simulated using OpenAI gym [3]. For pendulum swing-up and the Acrobot tasks we used time-varying linear feedback

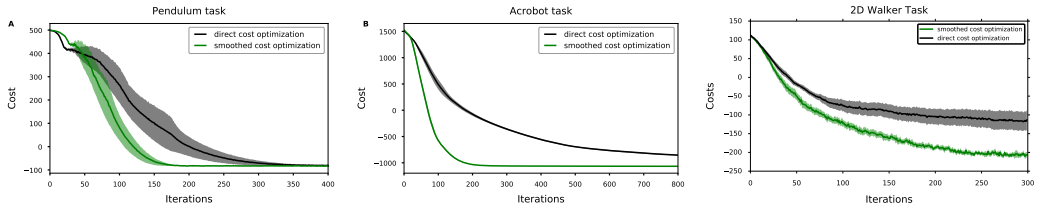

Figure 1: Smoothed cost-optimization exhibits faster convergence than direct cost-optimization in a variety of tasks. Plots show mean and standard deviation of the cost per iteration for 10 runs of the algorithm. For pendulum and acrobot tasks, we used $\Delta = 0.5$ and $\mathcal{E} = 0.1$ whereas for the walker, we used $\Delta = 0.05 \log N$ and $\mathcal{E} = 0.01$. See App. F for more details.

controllers, whereas for the 2D walker task we parametrized the control u_θ using a neural network. We provide more details about the experimental settings and additional results in App. F.

$-$**Convergence rate of policy optimization**: Fig. 4 shows the comparison of ASPIC algorithm with smoothing against direct-cost optimization. In all three tasks, smoothing improves the convergence rate of policy optimization. Smoothed cost optimization requires less iterations to achieve the same cost reduction as direct cost-optimization, with only a negligible amount of additional computational steps that do not depend on the complexity of the simulation runs.

We can thus conclude that even in cases when the parametrized controller does not strictly meet the requirement of full parametrization that we need to derive the gradient of the smoothed cost, a strong performance boost can also be achieved.

6 Discussion

Many policy optimization algorithms update the control policy based on a direct optimization of the cost; examples are Trust Region Policy Optimization (TRPO) [22] or the Path-Integral Relative Entropy Policy Search (PIREPS) [10], where the later is particularly developed for Path Integral control problems. The main novelty of this work is the application of the idea of smoothing as introduced in [4] to Path Integral control problems. This allows to outperform direct cost-optimization and achieve faster convergence rates with only a negligible amount of computational overhead.

This procedure bears similarities to an adaptive annealing scheme, with the smoothing parameter playing the role of an artificial temperature. However in contrast to classical annealing schemes, such as simulated annealing, changing the smoothing parameter does not change the optimization target: the minimum of the smoothed cost remains the optimal control solution for all levels of smoothing.

In the weak smoothing limit, ASPIC directly optimizes the cost using trust region constrained updates, similar to the TRPO algorithm [22]. TRPO differs from ASPIC's weak smoothing limit by additionally using certain variance reduction techniques for the gradient estimator: They replace the stochastic cost in the gradient estimator by the easier-to-estimate advantage function, which has a state dependent baseline and only takes into account future expected cost. Since this depends on the linearity of the gradient in the stochastic cost and this dependence is non-linear for the gradient of the smoothed cost, we cannot directly incorporate these variance reduction techniques in ASPIC.

In the strong smoothing limit ASPIC becomes a version of PICE [13] that—unlike the plain PICE algorithm—uses a trust region constraint to achieve robust updates. The gradient estimation problem that appears in the PICE algorithm was previously addressed in [21]: they proposed a heuristic that allows to reduce the variance of the gradient estimator by adjusting the particle weights used to compute the policy gradient. In [21] this heuristic is introduced as an ad hoc fix of the sampling problem and the adjustment of the weights introduces a bias with possible unknown side effects. Our study sheds a new light on this, as adjusting the particle weights corresponds to a change of the smoothing parameter in our case.

Acknowledgments

References

[1] O. Arenz, M. Zhong, and G. Neumann. Trust-region variational inference with Gaussian mixture models. *arXiv preprint arXiv:1907.04710*, 2019.

[2] J. Bierkens and H. J. Kappen. Explicit solution of relative entropy weighted control. *Systems & Control Letters*, 72:36–43, 2014.

[3] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[4] P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(3):30, 2018.

[5] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[6] W. Fleming and S. Sheu. Risk-sensitive control and an optimal investment model ii. *Annals of Applied Probability*, pages 730–767, 2002.

[7] W. H. Fleming and W. M. McEneaney. Risk-sensitive control on an infinite time horizon. *SIAM Journal on Control and Optimization*, 33(6):1881–1915, 1995.

[8] P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[10] V. Gómez, H. J. Kappen, J. Peters, and G. Neumann. Policy search for path integral control. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 482–497. Springer, 2014.

[11] N. Heess, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, A. Eslami, M. Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[12] H. J. Kappen. Linear theory for control of nonlinear stochastic systems. *Physical Review Letters*, 95(20):200201, 2005.

[13] H. J. Kappen and H. C. Ruiz. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 162(5):1244–1266, 2016.

[14] H. Kwakernaak and R. Sivan. *Linear optimal control systems*, volume 1. Wiley-Interscience New York, 1972.

[15] J. Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[17] M. Papini, D. Binaghi, G. Canonaco, M. Pirotta, and M. Restelli. Stochastic variance-reduced policy gradient. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4026–4035. PMLR, 2018.

[18] J. Peters, K. Mülling, and Y. Altün. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1607–1612. AAAI Press, 2010.

[19] J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

[20] R. Ranganath, S. Gerrish, and D. Blei. Black Box Variational Inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33, pages 814–822. PMLR, 2014.

[21] H.-C. Ruiz and H. J. Kappen. Particle smoothing for hidden diffusion processes: Adaptive path integral smoother. *IEEE Transactions on Signal Processing*, 65(12):3191–3203, 2017.

[22] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1889–1897. PMLR, 2015.

[23] M. W. Spong. The swing up control problem for the acrobot. *IEEE control systems*, 15(1):49–55, 1995.

[24] D. Thalmeier, V. Gómez, and H. J. Kappen. Action selection in growing state spaces: Control of network structure growth. *Journal of Physics A: Mathematical and Theoretical*, 50(3):034006, 2016.

[25] S. Thijssen and H. J. Kappen. Path integral control and state-dependent feedback. *Physical Review E*, 91(3):032104, 2015.

[26] B. van den Broek, W. Wiegerinck, and H. Kappen. Risk sensitive path integral control. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 615–622. AUAI Press, 2010.

[27] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

A Derivation of the policy gradient

Here we derive Eq. (4). We write $C(p_{u_\theta}) = \left\langle S^\gamma_{u_\theta}(\tau)\right\rangle_{p_{u_\theta}}$, with $S^\gamma_{u_\theta}(\tau) \coloneqq V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)}$ and take the derivative of Eq. (2):

$$\nabla_\theta \left\langle S^\gamma_{u_\theta}(\tau)\right\rangle_{p_{u_\theta}} = \nabla_\theta \left\langle V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)} \right\rangle_{p_{u_\theta}} \tag{20}$$

Now we introduce the importance sampler $p_{u_{\theta'}}$ and correct for it.

$$\nabla_\theta \left\langle S^\gamma_{u_\theta}(\tau)\right\rangle_{p_{u_\theta}} = \nabla_\theta \left\langle \frac{p_{u_\theta}(\tau)}{p_{u_{\theta'}}(\tau)} \left(V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)} \right) \right\rangle_{p_{u_{\theta'}}} \tag{21}$$

This is true for all θ' as long as $p_{u_\theta}(\tau)$ and $p_{u_{\theta'}}(\tau)$ are absolutely continuous to each other. Taking the derivative we get:

$$= \left\langle \frac{\nabla_\theta p_{u_\theta}(\tau)}{p_{u_{\theta'}}(\tau)} \left(V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)} \right) \right\rangle_{p_{u_{\theta'}}} + \left\langle \frac{p_{u_\theta}(\tau)}{p_{u_{\theta'}}(\tau)} \left(\gamma \frac{1}{p_{u_\theta}(\tau)} \nabla_\theta p_{u_\theta}(\tau) \right) \right\rangle_{p_{u_{\theta'}}} \tag{22}$$

$$= \left\langle (\nabla_\theta \log p_{u_\theta}(\tau)) \left(V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)} \right) \right\rangle_{p_{u_\theta}} + \gamma \nabla_\theta \left\langle \frac{1}{p_{u_{\theta'}}(\tau)} p_{u_\theta}(\tau) \right\rangle_{p_{u_{\theta'}}} \tag{23}$$

$$= \left\langle S^\gamma_{u_\theta}(\tau) \nabla_\theta \log p_{u_\theta}(\tau) \right\rangle_{p_{u_\theta}} + \gamma \nabla_\theta \left\langle 1 \right\rangle_{p_{u_\theta}} \tag{24}$$

$$= \left\langle S^\gamma_{u_\theta}(\tau) \nabla_\theta \log p_{u_\theta}(\tau) \right\rangle_{p_{u_\theta}} . \tag{25}$$

B Smoothing Stochastic Control Problems

B.1 Replacing Minimization over u by Minimization over p'

Here we show that for

$$J^{\alpha}(\theta) = \inf_{u'} \alpha KL(p_{u'}||p_{u_\theta}) + \gamma KL(p_{u'}||p_0) + \langle V(\tau)\rangle_{p'} \tag{26}$$

we can replace the minimization over u by a minimization over p' to obtain Eq. (11). For this, we need to show that the minimizer $p^*_{\alpha,\theta}$ of Eq. (11) is induced by $u^*_{\alpha,\theta}$, the minimizer of Eq. (26):

$$p^*_{\alpha,\theta} \equiv p_{u^*_{\alpha,\theta}}.$$

The solution to (11) is given by (see App. B.2)

$$p^*_{\alpha,\theta} = \frac{1}{Z} p_{u_\theta}(\tau) \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right) = \frac{1}{Z} p_{u_\theta}(\tau) \left(\frac{p_0(\tau)}{p_{u_\theta}(\tau)}\right)^{\frac{\gamma}{\gamma+\alpha}} \exp\left(-\frac{1}{\gamma+\alpha} V(\tau)\right). \tag{27}$$

We rewrite

$$p_0(\tau)\left(\frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right)^{1-\frac{\gamma}{\gamma+\alpha}} = p_0(\tau)\exp\left(\left(1-\frac{\gamma}{\gamma+\alpha}\right)\int_0^T \left(\frac{1}{2}u_\theta(x_t,t)^T u_\theta(x_t,t) + u_\theta(x_t,t)^T\xi_t\right)dt\right),$$

where we used the Girsanov theorem [2, 25] (and set $\nu = 1$ for simpler notation). With $\tilde{u}_\theta(x_t,t) \coloneqq \left(1-\frac{\gamma}{\gamma+\alpha}\right)u_\theta(x_t,t)$ this gives

$$\begin{aligned} p_0(\tau)\left(\frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right)^{1-\frac{\gamma}{\gamma+\alpha}} &= p_0(\tau)\exp\left(\int_0^T \left(\frac{1}{2}\tilde{u}_\theta(x_t,t)^T\tilde{u}_\theta(x_t,t) + \tilde{u}_\theta(x_t,t)^T\xi_t\right)dt\right)\cdot \\ &\quad \cdot \exp\left(\int_0^T \left(\frac{1}{2}\frac{\gamma}{\alpha}\tilde{u}_\theta(x_t,t)^T\tilde{u}_\theta(x_t,t)\right)dt\right) \\ &= p_{\tilde{u}_\theta}(\tau)\exp\left(\int_0^T \left(\frac{1}{2}\frac{\gamma}{\alpha}\tilde{u}_\theta(x_t,t)^T\tilde{u}_\theta(x_t,t)\right)dt\right). \end{aligned}$$

So we get

$$p^*_{\alpha,\theta} = \frac{1}{Z} p_{\tilde{u}_\theta}(\tau)\exp\left(\int_0^T \left(\frac{1}{2}\frac{\gamma}{\alpha}\tilde{u}_\theta(x_t,t)^T\tilde{u}_\theta(x_t,t)\right)dt\right)\exp\left(-\frac{1}{\gamma+\alpha}V(\tau)\right). \tag{28}$$

This has the form of an optimally controlled distribution with dynamics

$$\dot{x}_t = f(x_t,t) + g(x_t,t)\left(\tilde{u}_\theta(x_t,t) + \hat{u}(x_t,t) + \xi_t\right) \tag{29}$$

and cost

$$\left\langle \int_0^T \frac{1}{\gamma+\alpha}V(x_t,t) - \frac{1}{2}\frac{\gamma}{\alpha}\tilde{u}_\theta(x_t,t)^T\tilde{u}_\theta(x_t,t)dt + \int_0^T \left(\frac{1}{2}\hat{u}(x_t,t)^T\hat{u}(x_t,t) + \hat{u}(x_t,t)^T\xi_t\right)dt \right\rangle_{p_{\hat{u}}}. \tag{30}$$

This is a Path Integral control problem with state cost $\int_0^T \frac{1}{\gamma+\alpha}V(x_t,t) - \frac{1}{2}\frac{\gamma}{\alpha}\tilde{u}_\theta(x_t,t)^T\tilde{u}_\theta(x_t,t)dt$ which is well defined with $\tilde{u}_\theta(x_t,t) = \left(1-\frac{\gamma}{\gamma+\alpha}\right)u_\theta(x_t,t)$.

Let $\hat{u}^*$ be the optimal control of this Path Integral control problem. Then $p^*_{\alpha,\theta}$ is induced by Eq. (29) with $\hat{u} = \hat{u}^*$. This is equivalent to say that $p^*_{\alpha,\theta}$ is induced by Eq. (1) As $p^*_{\alpha,\theta}$ is the density that minimizes Eq. (11), $\tilde{u}_\theta + \hat{u}^*$ is minimizing Eq. (26).

B.2 Minimizer of smoothed cost

Here we want to proof Eq. (12):

$$p^*_{\alpha,\theta}(\tau) \coloneqq \arg\min_{p'} \alpha KL(p'||p_{u_\theta}) + \left\langle S^\gamma_{p_{u_\theta}}(\tau) \right\rangle_{p'} \tag{31}$$

$$= \arg\min_{p'} \left\langle \alpha \log \frac{p'(\tau)}{p_{u_\theta}(\tau)} + V(\tau) + \gamma \log \frac{p'(\tau)}{p_0(\tau)} \right\rangle_{p'} . \tag{32}$$

For this we take the variational derivative and set it to zero:

$$0 = \frac{\delta}{\delta p'(\tau)} \left\langle \alpha \log \frac{p'(\tau)}{p_{u_\theta}(\tau)} + V(\tau) + \gamma \log \frac{p'(\tau)}{p_0(\tau)} + \kappa \right\rangle_{p'} \Bigg|_{p'=p^*_{\alpha,\theta}} , \tag{33}$$

where we added a Lagrange multiplier κ to ensure normalization. We get

$$0 = \alpha \log \frac{p'(\tau)}{p_{u_\theta}(\tau)} + V(\tau) + \gamma \log \frac{p'(\tau)}{p_0(\tau)} + \kappa \Bigg|_{p'=p^*_{\alpha,\theta}} , \tag{34}$$

from which follows

$$p^*_{\alpha,\theta}(\tau) = \exp\left(\frac{\kappa}{\alpha+\gamma}\right) p_{u_\theta}(\tau)^{\frac{\alpha}{\alpha+\gamma}} p_0(\tau)^{\frac{\gamma}{\alpha+\gamma}} \exp\left(-\frac{1}{\gamma+\alpha} V(\tau)\right) \tag{35}$$

$$= \exp\left(\frac{\kappa}{\alpha+\gamma}\right) p_{u_\theta}(\tau) \exp\left(-\frac{1}{\gamma+\alpha} V(\tau) - \frac{\gamma}{\alpha+\gamma} \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right) \tag{36}$$

$$= \exp\left(\frac{\kappa}{\alpha+\gamma}\right) p_{u_\theta}(\tau) \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau)\right), \tag{37}$$

where κ is chosen such that the distribution is normalized.

B.3 Derivation of the gradient of the smoothed cost function

Here we derive Eq. (14) by taking the derivative of Eq. (13):

$$\nabla_\theta J^\alpha(\theta) = -(\gamma+\alpha) \nabla_\theta \log \left\langle \exp\left(-\frac{1}{\gamma+\alpha}\left(V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right)\right)\right\rangle_{p_{u_\theta}} \tag{38}$$

$$= -\frac{\gamma+\alpha}{Z^\alpha_{p_{u_\theta}}} \nabla_\theta \left\langle \exp\left(-\frac{1}{\gamma+\alpha}\left(V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right)\right)\right\rangle_{p_{u_\theta}} . \tag{39}$$

Now we introduce the importance sampler $p_{u_{\theta'}}$ and correct for it.

$$\nabla_\theta J^\alpha(\theta) = -\frac{\gamma+\alpha}{Z^\alpha_{p_{u_\theta}}} \nabla_\theta \left\langle \frac{p_{u_\theta}(\tau)}{p_{u_{\theta'}}(\tau)} \exp\left(-\frac{1}{\gamma+\alpha}\left(V(\tau) + \gamma \log \frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right)\right)\right\rangle_{p_{u_{\theta'}}} \tag{40}$$

$$= -\frac{\gamma+\alpha}{Z^\alpha_{p_{u_\theta}}} \nabla_\theta \left\langle \frac{p_0(\tau)^{\frac{\gamma}{\gamma+\alpha}}}{p_{u_{\theta'}}(\tau)} \left(p_{u_\theta}(\tau)\right)^{\frac{\alpha}{\gamma+\alpha}} \exp\left(-\frac{1}{\gamma+\alpha} V(\tau)\right)\right\rangle_{p_{u_{\theta'}}} \tag{41}$$

$$= -\frac{\alpha}{Z^\alpha_{p_{u_\theta}}} \left\langle \frac{1}{p_{u_{\theta'}}(\tau)} \left(\frac{p_{u_\theta}(\tau)}{p_0(\tau)}\right)^{-\frac{\gamma}{\gamma+\alpha}} \exp\left(-\frac{1}{\gamma+\alpha} V(\tau)\right) \nabla_\theta p_{u_\theta}\right\rangle_{p_{u_{\theta'}}} \tag{42}$$

$$= -\frac{\alpha}{Z^\alpha_{p_{u_\theta}}} \left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_\theta}}(\tau)\right) \nabla_\theta \log p_{u_\theta}(\tau)\right\rangle_{p_{u_\theta}} . \tag{43}$$

C PICE, Direct Cost-Optimization and Risk Sensitivity as Limiting Cases of Smoothed Cost Optimization

The smoothed cost and its gradient depend on the two parameters α and γ, which come from the smoothing Eq. (7) and the definition of the control problem (2) respectively. Although at first glance

the two parameters seem to play a similar role, they change different properties of the smoothed cost $J^\alpha(\theta)$ when they are varied.

In the expression for the smoothed cost (13), the parameter α only appears in the sum $\gamma + \alpha$. Varying it changes the effect of the smoothing but leaves the optimum $\theta^* = \arg\min_\theta J^\alpha(\theta)$ of the smoothed cost invariant. Here we show that smoothing leaves the global optimum of the cost $C(p_{u_\theta})$ invariant. As $KL(p_{u_{\theta'}}||p_{u_\theta}) \geq 0$ we have that

$$J^\alpha(\theta) = \inf_{\theta'} C(p_{u_{\theta'}}) + \alpha KL(p_{u_{\theta'}}||p_{u_\theta}) \geq \inf_{\theta'} C(p_{u_{\theta'}}) = C(p_{u_{\theta^*}}).$$

To show that the global minimum θ^* of C is also the global minimum of J^α, it is thus sufficient to show that

$$J^\alpha(\theta^*) \leq C(p_{u_{\theta^*}}).$$

We have

$$J^\alpha(\theta^*) = \inf_{\theta'} C(p_{u_{\theta'}}) + \alpha KL(p_{u_{\theta'}}||p_{u_{\theta^*}}).$$

Using that the minimum of a sum of terms is never larger than the sum of the minimum of terms, we get

$$\begin{aligned} J^\alpha(\theta^*) &\leq \left(\inf_{\theta'} C(p_{u_{\theta'}})\right) + \left(\inf_{\theta'} \alpha KL(p_{u_{\theta'}}||p_{u_{\theta^*}})\right) \\ &= C(p_{u_{\theta^*}}) + \alpha KL(p_{u_{\theta^*}}||p_{u_{\theta^*}}) \\ &= C(p_{u_{\theta^*}}). \end{aligned}$$

We also expect local maxima to be also preserved for large-enough smoothing parameter α. This would correspond to small time smoothing by the associated Hamilton-Jacobi partial differential equation [4].

We therefore call α the *smoothing parameter*. The larger α, the weaker the smoothing; in the limiting case $\alpha \to \infty$, smoothing is turned off as we can see from Eq. (13): for very large α, the exponential and the logarithmic function linearise, $J^\alpha(\theta) \to C(p_{u_\theta})$ and we recover direct cost-optimization. For the limiting case $\alpha \to 0$, we recover the PICE method: the optimizer $p^*_{\alpha,\theta}$ becomes equal to the optimal density p^* and the gradient on the smoothed cost (14) becomes proportional to the PICE gradient (6):

$$\lim_{\alpha \to 0} \frac{1}{\alpha} \nabla_\theta J^\alpha(\theta) = \nabla_\theta KL(p^*||p_{u_\theta}).$$

Varying γ changes the control problem and thus its optimal solution. For $\gamma \to 0$, the control cost becomes zero. In this case the cost only consists of the state cost and arbitrary large controls are allowed. We get

$$J^\alpha(\theta) = -\alpha \log \left\langle \exp\left(-\frac{1}{\alpha} V(\tau)\right) \right\rangle_{p_{u_\theta}}.$$

This expression is identical to the risk sensitive control cost proposed in [6, 7, 26]. Thus, for $\gamma = 0$, the smoothing parameter α controls the risk-sensitivity, resulting in risk seeking objectives for $\alpha > 0$ and risk avoiding objectives for $\alpha < 0$. In the limiting case $\gamma \to \infty$, the problem becomes trivial; the optimal controlled dynamics becomes equal to the uncontrolled dynamics: $p^* \to p_0$, c.f., Eq. (5), and $u^* \to 0$.

If both parameters α and γ are small, the problem is hard (see [21, 24]) as many samples are needed to estimate the smoothed cost. The problem becomes feasible if either α or γ is increased. Increasing γ however, changes the control problem, while increasing α weakens the effect of smoothing.

D The effect of cost function smoothing on policy optimization

We introduced smoothing as a way to speed up policy optimization compared to a direct optimization of the cost. In this section we analyse policy optimization with and without smoothing and show

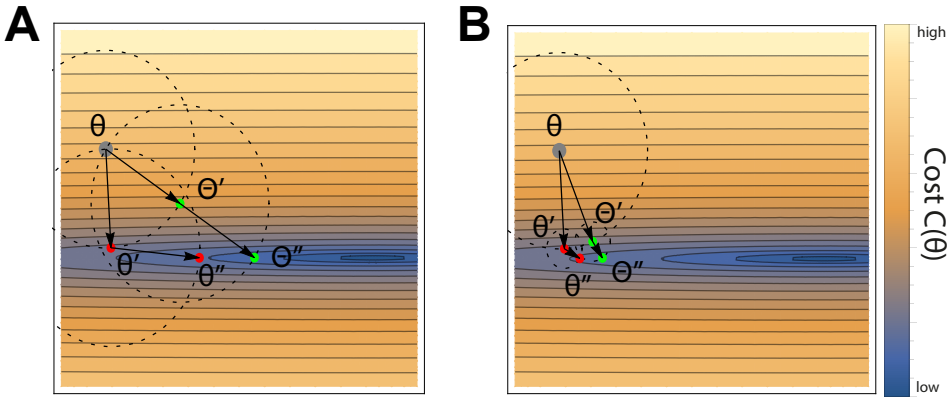

Figure 2: Illustration of optimal two-step updates compared with two consecutive direct updates. Illustrated is a two-dimensional cost landscape $C(\theta)$ parametrized by θ. Dark colors represent low cost, while light colors represent high cost. Green dots indicate the optimal two-step update $\theta \to \Theta' \to \Theta''$ while red dots indicates two consecutive direct updates $\theta \to \theta' \to \theta''$ with $\theta' = \Theta^C_{\mathcal{E}}(\theta)$ and $\theta'' = \Theta^C_{\mathcal{E}'}(\theta')$. The dashed circles indicate trust regions. θ', θ'' and Θ'' are the minimizers of the cost in the trust regions around θ, θ' and Θ' respectively. Θ' is chosen such that the cost $C(\Theta'')$ after the subsequent direct update is minimized. In both panels, the final cost after an optimal two-step update $C(\Theta'')$ is smaller than the final cost after two direct updates $C(\theta'')$. (A) Equal sizes of the update steps, $\mathcal{E} = \mathcal{E}'$. (B) When the size of the second step becomes small $\mathcal{E}' \ll \mathcal{E}$, the smoothed update $\theta \to \Theta'$ becomes more similar to the direct update $\theta \to \theta'$.

analytically how smoothing can speed up policy optimization. To simplify notation, we overload $p_{u_\theta} \to \theta$ so that we get $C(p_{u_\theta}) \to C(\theta)$ and $KL(p_{u_{\theta'}}||p_{u_\theta}) \to KL(\theta'||\theta)$.

We use a trust region constraint to robustly optimize the policy, c.f., [18, 22, 10]. There are two options. On the one hand, we can directly optimize the cost C:

Definition 1. *We define the direct update with stepsize $\mathcal{E}$ as an update $\theta \to \theta'$ with $\theta' = \Theta^C_{\mathcal{E}}(\theta)$ and*

$$\Theta^C_{\mathcal{E}}(\theta) := \underset{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta) \leq \mathcal{E}}}{\arg\min} \quad C(\theta'). \tag{44}$$

The direct update results in the minimal cost that can be achieved after one single update. We define the optimal one-step cost

$$C^*_{\mathcal{E}}(\theta) := \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta) \leq \mathcal{E}}} C(\theta').$$

On the other hand we can optimize the smoothed cost J^α:

Definition 2. *We define the smoothed update with stepsize $\mathcal{E}$ as an update $\theta \to \theta'$ with $\theta' = \Theta^{J^\alpha}_{\mathcal{E}}(\theta)$ and*

$$\Theta^{J^\alpha}_{\mathcal{E}}(\theta) := \underset{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta) \leq \mathcal{E}}}{\arg\min} \quad J^\alpha(\theta'). \tag{45}$$

While a direct update achieves the minimal cost that can be achieved after a single update, we show below that a smoothed update can result in a faster cost reduction if more than one update step is performed.

Definition 3. *We define the optimal two-step update $\theta \to \Theta' \to \Theta''$ as an update that results in the lowest cost that can be achieved with a two-step update $\theta \to \theta' \to \theta''$ with fixed stepsizes $\mathcal{E}$ and $\mathcal{E}'$ respectively:*

$$\Theta', \Theta'' := \underset{\substack{\theta', \theta'' \\ \text{s.t. } KL(\theta''||\theta') \leq \mathcal{E}' \\ KL(\theta'||\theta) \leq \mathcal{E}}}{\arg\min} \quad C(\theta'')$$

and the corresponding optimal two-step cost

$$C^*_{\mathcal{E},\mathcal{E}'}(\theta) \coloneqq \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} \min_{\substack{\theta'' \\ \text{s.t. } KL(\theta''||\theta')\leq\mathcal{E}'}} C(\theta'') = \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} C\left(\Theta^C_{\mathcal{E}'}(\theta')\right). \tag{46}$$

In Fig. 2 we illustrate how such an optimal two-step update leads to a faster decrease of the cost than two consecutive direct updates.

Theorem 1. Statement 1: *For all $\mathcal{E}$, α there exists an $\mathcal{E}'$, such that a smoothed update with stepsize $\mathcal{E}$ followed by a direct update with stepsize $\mathcal{E}'$ is an optimal two-step update:*

$$\Theta' = \Theta^{J^\alpha}_{\mathcal{E}}(\theta)$$
$$\Theta'' = \Theta^C_{\mathcal{E}'}(\Theta')$$

$$\Rightarrow C\left(\Theta''\right) = C^*_{\mathcal{E},\mathcal{E}'}(\theta)$$

The size of the second step $\mathcal{E}'$ is a function of θ and α.

Statement 2: *$\mathcal{E}'$ is monotonically decreasing in α.*

While it is evident from Eq. (46) that the second step of the optimal two-step update must be a direct update, the statement that the first step is a smoothed update is non-trivial.

We split the proof into three subsections: in the first subsection, we state and proof a lemma that we need to proof statement 1. In the second subsection, we proof statement 1 and in the third subsection, we proof statement 2.

D.1 Lemma

Lemma 1. *With $\theta^*_{\alpha,\theta}$ defined as in Eq.* (9) *and $\mathcal{E}_\alpha(\theta) = KL\left(\theta^*_{\alpha,\theta}||\theta\right)$ we can rewrite $J^\alpha(\theta)$:*

$$J^\alpha(\theta) = C\left(\Theta^C_{\mathcal{E}'}(\theta)\right)\big|_{\mathcal{E}'=\mathcal{E}_\alpha(\theta)} + \alpha\mathcal{E}_\alpha(\theta). \tag{47}$$

Proof. With the definition of $\theta^*_{\alpha,\theta}$ as the minimizer of $C(\theta') + \alpha KL(\theta'||\theta)$ (see (9)) we have

$$\begin{aligned} J^\alpha(\theta) &= C\left(\theta^*_{\alpha,\theta}\right) + \alpha KL(\theta^*_{\alpha,\theta}||\theta) \\ &= C\left(\theta^*_{\alpha,\theta}\right) + \alpha\mathcal{E}_\alpha(\theta). \end{aligned}$$

What is left to show is that

$$\theta^*_{\alpha,\theta} \equiv \Theta^C_{\mathcal{E}_\alpha(\theta)}(\theta).$$

As $\Theta^C_{\mathcal{E}_\alpha(\theta)}(\theta)$ is the minimizer of the cost C within the trust region defined by $\{\theta' : KL(\theta'||\theta) \leq \mathcal{E}_\alpha(\theta)\}$ we have to show that

1. $\theta^*_{\alpha,\theta}$ lies within this trust region,
2. $C\left(\theta^*_{\alpha,\theta}\right)$ is a minimizer of the cost C within this trust region.

The first point is trivially true as $KL(\theta^*_{\alpha,\theta}||\theta) = \mathcal{E}_\alpha(\theta)$ by definition. Hence $\theta^*_{\alpha,\theta}$ lies at the boundary of this trust region and therefore in it, as the boundary belongs to the trust region. The second point we proof by contradiction: Given $\theta^*_{\alpha,\theta}$ is not minimizing the cost within the trust region, then there exists a $\hat{\theta}$ with $C(\hat{\theta}) < C(\theta^*_{\alpha,\theta})$ and $KL(\hat{\theta}||\theta) \leq \mathcal{E}_\alpha(\theta) = KL(\theta^*_{\alpha,\theta}||\theta)$. Therefore it must hold that

$$C(\hat{\theta}) + \alpha KL(\hat{\theta}||\theta) < C(\theta^*_{\alpha,\theta}) + \alpha KL(\theta^*_{\alpha,\theta},\theta)$$

which is a contradiction, as $\theta^*_{\alpha,\theta}$ is the minimizer of $C(\theta') + \alpha KL(\theta'||\theta)$. □

D.2 Proof of Statement 1

Here we show that for every α and θ there exists an $\mathcal{E}' = \mathcal{E}^*_\alpha(\theta)$ such that

$$C\left(\Theta^C_{\mathcal{E}'}\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right)\right)\Big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} = C^*_{\mathcal{E},\mathcal{E}'}\big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)}. \tag{48}$$

Proof. As $J^\alpha(\theta)$ is the infimum of $C(\theta') + \alpha KL(\theta'||\theta)$, we have for any $\mathcal{E}' > 0$

$$J^\alpha(\theta) \leq C\left(\Theta^C_{\mathcal{E}'}(\theta)\right) + \alpha KL\left(\Theta^C_{\mathcal{E}'}(\theta)||\theta\right).$$

Further, as $\Theta^C_{\mathcal{E}'}(\theta)$ lies in the trust region $\{\theta' : KL(\theta'||\theta) \leq \mathcal{E}'\}$ we have that $KL\left(\Theta^C_{\mathcal{E}'}(\theta)||\theta\right) \leq \mathcal{E}'$, so we can write

$$C\left(\Theta^C_{\mathcal{E}'}(\theta)\right) + \alpha KL\left(\Theta^C_{\mathcal{E}'}(\theta)||\theta\right) \leq C\left(\Theta^C_{\mathcal{E}'}(\theta)\right) + \alpha\mathcal{E}'$$

and thus

$$J^\alpha(\theta) \leq C\left(\Theta^C_{\mathcal{E}'}(\theta)\right) + \alpha\mathcal{E}'.$$

Next we minimize both sides of this inequality within the trust region $\{\theta' : KL(\theta'||\theta) \leq \mathcal{E}\}$. We use that

$$J^\alpha\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right) = \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} J^\alpha(\theta')$$

and get

$$J^\alpha\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right) \leq \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} \left(C\left(\Theta^C_{\mathcal{E}'}(\theta')\right) + \alpha\mathcal{E}'\right). \tag{49}$$

Now we use Lemma 1 and rewrite the left hand side of this inequality.

$$J^\alpha\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right) = C\left(\Theta^C_{\mathcal{E}'}\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right)\right)\Big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} + \alpha\mathcal{E}^*_\alpha(\theta)$$

with $\mathcal{E}^*_\alpha(\theta) \coloneqq \mathcal{E}_\alpha(\Theta^{J^\alpha}_{\mathcal{E}}(\theta))$. Plugging this back to (49) we get

$$C\left(\Theta^C_{\mathcal{E}'}\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right)\right)\Big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} + \alpha\mathcal{E}^*_\alpha(\theta) \leq \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} \left(C\left(\Theta^C_{\mathcal{E}'}(\theta')\right) + \alpha\mathcal{E}'\right).$$

As this inequality holds for any $\mathcal{E}' > 0$ we can plug in $\mathcal{E}^*_\alpha(\theta)$ on the right hand side of this inequality and obtain

$$C\left(\Theta^C_{\mathcal{E}'}\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right)\right)\Big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} + \alpha\mathcal{E}^*_\alpha(\theta) \leq \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} C\left(\Theta^C_{\mathcal{E}'}(\theta')\right)\big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} + \alpha\mathcal{E}^*_\alpha(\theta).$$

We subtract $\alpha\mathcal{E}^*_\alpha(\theta)$ on both sides

$$C\left(\Theta^C_{\mathcal{E}'}\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right)\right)\Big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} \leq \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} C\left(\Theta^C_{\mathcal{E}'}(\theta')\right)\big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)}.$$

Using Eq. (46) gives

$$C\left(\Theta^C_{\mathcal{E}'}\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right)\right)\Big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)} \leq C^*_{\mathcal{E},\mathcal{E}'}(\theta)\big|_{\mathcal{E}'=\mathcal{E}^*_\alpha(\theta)},$$

which concludes the proof. □

D.3 Proof of Statement 2

Here we show that $\mathcal{E}' = \mathcal{E}^*_\alpha(\theta)$ is a monotonically decreasing function of α. $\mathcal{E}^*_\alpha(\theta)$ is given by

$$\mathcal{E}^*_\alpha(\theta) = \mathcal{E}_\alpha\left(\Theta^{J^\alpha}_{\mathcal{E}}(\theta)\right) = KL(\theta^*_{\alpha,\theta'}||\theta')\big|_{\theta'=R^{J^\alpha}_{\mathcal{E}}(\theta)}.$$

We have

$$\begin{aligned}\left(\alpha KL(\theta^*_{\alpha,\theta'}||\theta') + C\left(\theta^*_{\alpha,\theta'}\right)\right)\big|_{\theta'=R^{J^\alpha}_{\mathcal{E}}(\theta)} &= \left(\inf_{\theta''} \alpha KL(\theta''||\theta') + C(\theta'')\right)\Big|_{\theta'=R^{J^\alpha}_{\mathcal{E}}(\theta)} \\ &= \min_{\substack{\theta' \\ \text{s.t. } KL(\theta'||\theta)\leq\mathcal{E}}} \inf_{\theta''} \alpha KL(\theta''||\theta') + C(\theta'').\end{aligned}$$

For convenience we introduce a shorthand notation for the minimizers

$$\begin{aligned}\theta_\alpha &\coloneqq \Theta^{J^\alpha}_{\mathcal{E}}(\theta) \\ \theta'_\alpha &\coloneqq \theta^*_{\alpha,\theta'}|_{\theta'=\Theta^{J^\alpha}_{\mathcal{E}}(\theta)}.\end{aligned}$$

We compare $\alpha_1 \geq 0$ with $\mathcal{E}^*_{\alpha_1}(\theta) \coloneqq KL(\theta'_{\alpha_1}||\theta_{\alpha_1})$ and $\alpha_2 \geq 0$ with $\mathcal{E}^*_{\alpha_2}(\theta) \coloneqq KL(\theta'_{\alpha_2}||\theta_{\alpha_2})$ and assume that $\mathcal{E}^*_{\alpha_1}(\theta) < \mathcal{E}^*_{\alpha_2}(\theta)$. We show that from this it follows that $\alpha_1 > \alpha_2$.

Proof. As θ'_{α_1},θ_{α_1} minimize $\alpha_1 KL(\theta'||\theta) + C(\theta')$ we have

$$\begin{aligned}&\alpha_1 KL(\theta'_{\alpha_1}||\theta_{\alpha_1}) + C(\theta'_{\alpha_1}) \leq \alpha_1 KL(\theta'_{\alpha_2}||\theta_{\alpha_2}) + C(\theta'_{\alpha_2}) \\ &\Rightarrow \alpha_1 \mathcal{E}_{\alpha_1}(\theta) + C(\theta'_{\alpha_1}) \leq \alpha_1 \mathcal{E}_{\alpha_2}(\theta) + C(\theta'_{\alpha_2})\end{aligned}$$

and analogous for α_2

$$\begin{aligned}&\alpha_2 KL(\theta'_{\alpha_1}||\theta_{\alpha_1}) + C(\theta'_{\alpha_1}) \geq \alpha_2 KL(\theta'_{\alpha_2}||\theta_{\alpha_2}) + C(\theta'_{\alpha_2}) \\ &\Rightarrow \alpha_2 \mathcal{E}_{\alpha_1}(\theta) + C(\theta'_{\alpha_1}) \geq \alpha_2 \mathcal{E}_{\alpha_2}(\theta) + C(\theta'_{\alpha_2})\end{aligned}$$

With $\mathcal{E}_{\alpha_1}(\theta) < \mathcal{E}_{\alpha_2}(\theta)$ we get

$$\alpha_1 \geq \frac{C(\theta'_{\alpha_1}) - C(\theta'_{\alpha_2})}{\mathcal{E}_{\alpha_2}(\theta) - \mathcal{E}_{\alpha_1}(\theta)} \geq \alpha_2.$$

□

We showed that from $\mathcal{E}_{\alpha_1}(\theta) < \mathcal{E}_{\alpha_2}(\theta)$ it follows that $\alpha_1 \geq \alpha_2$ which proofs that $\mathcal{E}_\alpha(\theta)$ is monotonously decreasing in α.

Direct updates are myopic and do not take into account successive steps and are thus suboptimal when more than one update is needed. Smoothed updates on the other hand, as we see on theorem 1, anticipate a subsequent step and minimize the cost that results from this this two-step update. Hence smoothed updates favour a greater cost reduction in the future over maximal cost reduction in the current step. The strength of this anticipatory effect depends on the smoothing strength, which is controlled by the smoothing parameter α: For large α, smoothing is weak and the size $\mathcal{E}'$ of this anticipated second step becomes small. Fig. 2 B illustrates that for this case, when $\mathcal{E}'$ becomes small, smoothed updates become more similar to direct updates. In the limiting case $\alpha \to \infty$ the difference between smoothed and direct updates vanishes completely, as $J^\alpha(\theta) \to C(\theta)$ (see section C).

We expect that also with multiple update steps due to this anticipatory effect, iterating smoothed updates leads to a faster decrease of the cost than iterating direct updates. We will confirm this by numerical studies. Furthermore, we expect that this accelerating effect of smoothing is stronger for smaller values of α. On the other hand, as we will discuss in the next section, for smaller values of α it is harder to accurately perform the smoothed updates. Therefore we expect an optimal performance for an intermediate value of α. Based on this we build an algorithm in the next section that aims to accelerate policy optimization by cost function smoothing.

E Additional Theoretical Results for Section 4

E.1 Smoothed Updates for Small Update Steps $\mathcal{E}$

We want to compute Eq. (16) for small $\mathcal{E}$ which corresponds to large β. Assuming a smooth dependence of p_{u_θ} on θ, bounding $KL(\theta||\theta_n)$ to a very small value allows us to do a Taylor expansion which we truncate at second order:

$$\arg\min_{\theta'} J^\alpha(\theta') + \beta KL(\theta'||\theta_n) \approx \tag{50}$$

$$\approx \arg\min_{\theta'} \ (\theta' - \theta_n)^T \nabla_{\theta'} J^\alpha(\theta') + \frac{1}{2}(\theta' - \theta_n)^T \left(H + \beta F\right)(\theta' - \theta_n) \tag{51}$$

$$= \theta_n - \beta^{-1} F^{-1} \ \nabla_{\theta'} J^\alpha(\theta')|_{\theta'=\theta_n} + \mathcal{O}(\beta^{-2}) \tag{52}$$

with

$$H = \nabla_{\theta'} \nabla_{\theta'}^T J^\alpha(\theta')\big|_{\theta'=\theta_n}$$
$$F = \nabla_{\theta'} \nabla_{\theta'}^T KL(\theta'||\theta_n)\big|_{\theta'=\theta_n}.$$

See also [15]. We used that $\mathcal{E} \ll 1 \Leftrightarrow \beta \gg 1$. With this the Fisher information F dominates over the Hessian H and thus the Hessian does not appear anymore in the update equation. This defines a natural gradient update with stepsize β^{-1}.

E.2 Inversion of the Fisher matrix

We compute an approximation to the natural gradient $g_f = F^{-1}g$ by approximately solving the linear equation $Fg_f = g$ using truncated conjugate gradient. With the normal gradient g and the Fisher matrix $F = \nabla_\theta \nabla_\theta^T KL(p_{u_\theta}||p_{u_{\theta_n}})$ (see App. E.1).

We use an efficient way to compute the Fisher vector product Fy [22] using an automated differentiation package: First for each rollout i and timepoint t the symbolic expression for the gradient on the KL multiplied by a vector y is computed:

$$a_{i,t}(\theta_{n+1}) = \left(\nabla_{\theta_{n+1}}^T \log \frac{\pi_{\theta_n}(a_t^i|t, x_t^i)}{\pi_{\theta_{n+1}}(a_t^i|t, x_t^i)}\right) y.$$

Then we take the second derivative on this scalar quantity, sum over all times and average over the samples. This gives then the Fisher vector

$$Fy = \frac{1}{N}\sum_{i=1}^{N} \sum_{0<t<T} \nabla_{\theta_{n+1}} a_{i,t}(\theta_{n+1}).$$

For practical reasons, we reverse the arguments of the KL, since it is easier to estimate it from samples drawn from the first argument. For very small values, the KL is approximately symmetric in its arguments. Also, the equality in (18) differs from [22], which optimizes a value function within the trust region, e.g., $KL(\theta_n||\theta_{n+1}) \leq \mathcal{E}$.

E.3 Proof for equivalence of weight entropy and KL-divergence

We want to show that

$$\lim_{N\to\infty} \log N - H_N(w) = \lim_{N\to\infty} \log N + \sum_{i=1}^{N} w^i \log(w^i)$$
$$= KL(p^*_{\alpha,\theta}||p_{u_\theta}).$$

Where the samples i are drawn from p_{u_θ} and the w^i are given by

$$w^i = \frac{1}{\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S_{p_{u_\theta}}(\tau^i)\right),$$

We get

$$
\begin{aligned}
&\lim_{N\to\infty} \log N + \sum_{i=1}^{N} w^i \log(w^i) = \\
&= \lim_{N\to\infty} \log N + \sum_{i=1}^{N} \frac{1}{\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right) \cdot \\
&\qquad\qquad \cdot \log\left(\frac{1}{\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)\right) \\
&= \lim_{N\to\infty} \log N + \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\frac{1}{N}\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right) \cdot \\
&\qquad\qquad \cdot \log\left(\frac{\frac{1}{N}}{\frac{1}{N}\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)\right) \\
&= \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\frac{1}{N}\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right) \cdot \\
&\qquad\qquad \cdot \log\left(\frac{1}{\frac{1}{N}\sum_i^N \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau^i)\right)\right)
\end{aligned}
$$

Now we replace in the limit $N \to \infty$, $\frac{1}{N}\sum_i^N \to \langle\rangle_{p_{u_\theta}}$:

$$
\begin{aligned}
&= \left\langle \frac{1}{\left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right) \cdot \right. \\
&\qquad \left. \cdot \log\left(\frac{1}{\left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\right)\right\rangle_{p_{u_\theta}}
\end{aligned}
$$

Using Eq. (12) this gives

$$
\begin{aligned}
&= \left\langle \log\left(\frac{1}{\left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\right)\right\rangle_{p^*_{\alpha,\theta}} \\
&= \left\langle \log\left(\frac{1}{\left\langle \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}} \exp\left(-\frac{1}{\gamma+\alpha} S^{\gamma}_{p_{u_\theta}}(\tau)\right)\frac{p_{u_\theta}(\tau)}{p_{u_\theta}(\tau)}\right)\right\rangle_{p^*_{\alpha,\theta}} \\
&= \left\langle \log \frac{p^*_{\alpha,\theta}(\tau)}{p_{u_\theta}(\tau)}\right\rangle_{p^*_{\alpha,\theta}} \\
&= KL(p^*_{\alpha,\theta}||p_{u_\theta}).
\end{aligned}
$$

E.4 The Smoothness Parameter Δ is monotonic in α

Now we show that

$$\Delta = KL(p^*_{\alpha,\theta}||p_{u_\theta})$$

is a monotonic function of α.

$$\begin{aligned}
\frac{\partial}{\partial\alpha}KL(p^*_{\alpha,\theta}||p_{u_\theta}) &= \frac{\partial}{\partial\alpha}\left\langle \ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p^*_{\alpha,\theta}} \\
&= \frac{\partial}{\partial\alpha}\left\langle \frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}} \\
&= \left\langle \left(\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right)\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}} + \left\langle \frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\frac{\partial}{\partial\alpha}\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}} \\
&= \left\langle \left(\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right)\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}} + \left\langle \frac{1}{p_{u_\theta}}\frac{\partial}{\partial\alpha}p^*_{\alpha,\theta}\right\rangle_{p_{u_\theta}} \\
&= \left\langle \left(\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right)\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}} + \frac{\partial}{\partial\alpha}\left\langle 1\right\rangle_{p^*_{\alpha,\theta}} \\
&= \left\langle \left(\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right)\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}}.
\end{aligned}$$

Now let us look at

$$\begin{aligned}
\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}} &= \frac{\partial}{\partial\alpha}\left(\frac{1}{Z^\alpha_{p_{u_\theta}}}\exp\left(-\frac{1}{\gamma+\alpha}S^\gamma_{p_{u_\theta}}(\tau)\right)\right) \\
Z^\alpha_{p_{u_\theta}} &= \left\langle \exp\left(-\frac{1}{\gamma+\alpha}S^\gamma_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}.
\end{aligned}$$

we get

$$\begin{aligned}
\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}} &= \frac{1}{(\gamma+\alpha)^2}S^\gamma_{p_{u_\theta}}(\tau)\frac{p^*_{\alpha,\theta}}{p_{u_\theta}} - \frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\frac{1}{Z^\alpha_{p_{u_\theta}}}\frac{\partial}{\partial\alpha}Z^\alpha_{p_{u_\theta}} \\
\frac{\partial}{\partial\alpha}Z^\alpha_{p_{u_\theta}} &= \left\langle \frac{1}{(\gamma+\alpha)^2}S^\gamma_{p_{u_\theta}}\exp\left(-\frac{1}{\gamma+\alpha}S^\gamma_{p_{u_\theta}}(\tau)\right)\right\rangle_{p_{u_\theta}}.
\end{aligned}$$

and thus

$$\begin{aligned}
\frac{\partial}{\partial\alpha}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}} &= \frac{1}{(\gamma+\alpha)^2}S^\gamma_{p_{u_\theta}}(\tau)\frac{p^*_{\alpha,\theta}}{p_{u_\theta}} - \frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\frac{1}{(\gamma+\alpha)^2}\left\langle S^\gamma_{p_{u_\theta}}\right\rangle_{p^*_{\alpha,\theta}} \\
&= \frac{1}{(\gamma+\alpha)^2}\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\left(S^\gamma_{p_{u_\theta}}(\tau) - \left\langle S^\gamma_{p_{u_\theta}}\right\rangle_{p^*_{\alpha,\theta}}\right).
\end{aligned}$$

So finally we get

$$\begin{aligned}
\frac{\partial}{\partial\alpha}KL(p^*_{\alpha,\theta}||p_{u_\theta}) &= \frac{1}{(\gamma+\alpha)^2}\left\langle \frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\left(S^\gamma_{p_{u_\theta}}(\tau) - \left\langle S^\gamma_{p_{u_\theta}}\right\rangle_{p^*_{\alpha,\theta}}\right)\ln\frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\right\rangle_{p_{u_\theta}} \\
&= \frac{1}{(\gamma+\alpha)^2}\left\langle \frac{p^*_{\alpha,\theta}}{p_{u_\theta}}\left(S^\gamma_{p_{u_\theta}}(\tau) - \left\langle S^\gamma_{p_{u_\theta}}\right\rangle_{p^*_{\alpha,\theta}}\right)\left(-\frac{1}{\gamma+\alpha}S^\gamma_{p_{u_\theta}}(\tau) - \log Z^\alpha_{p_{u_\theta}}\right)\right\rangle_{p_{u_\theta}} \\
&= \frac{1}{(\gamma+\alpha)^2}\left\langle \left(S^\gamma_{p_{u_\theta}}(\tau) - \left\langle S^\gamma_{p_{u_\theta}}\right\rangle_{p^*_{\alpha,\theta}}\right)\left(-\frac{1}{\gamma+\alpha}S^\gamma_{p_{u_\theta}}(\tau) - \log Z^\alpha_{p_{u_\theta}}\right)\right\rangle_{p^*_{\alpha,\theta}} \\
&= -\frac{1}{(\gamma+\alpha)^3}\left(\left\langle \left(S^\gamma_{p_{u_\theta}}\right)^2\right\rangle_{p^*_{\alpha,\theta}} - \left\langle S^\gamma_{p_{u_\theta}}\right\rangle^2_{p^*_{\alpha,\theta}}\right) \\
&= -\frac{1}{(\gamma+\alpha)^3}\mathrm{Var}\left(S^\gamma_{p_{u_\theta}}\right) \leq 0.
\end{aligned}$$

Algorithm 1 ASPIC - Adaptive Smoothing of Path Integral Control

Require: State cost function $V(x,t)$
control cost parameter γ
base policy that defines uncontrolled dynamics π_0
simulator of system dynamics with a parametrized policy π_θ
trust region sizes $\mathcal{E}$
smoothing strength Δ
number of samples N

initialize θ_0
$n = 0$
repeat
  draw samples τ^i, with $i = 1, \ldots, N$, from simulator controlled by parametrized policy π_{θ_n}
  for each sample i compute $S^\gamma_{p_{u_{\theta_n}}}(\tau^i) = \sum_{0<t<T} V(x^i_t, t) + \gamma \log \frac{\pi_{\theta_n}(a^i_t|t,x^i_t)}{\pi_0(a^i_t|t,x^i_t)}$
  {Find minimal α such that $KL \leq \Delta$}
  $\alpha \leftarrow 0$
  repeat
    increase α
    $S^i_\alpha \leftarrow S^\gamma_{p_{u_{\theta_n}}}(\tau^i) \cdot \frac{1}{\gamma+\alpha}$
    compute weights $w_i \leftarrow \exp(-S^i_\alpha)$
    normalize weights $w_i \leftarrow \frac{w_i}{\sum_i(w_i)}$
    compute sample size independent weight entropy $KL \leftarrow \log N + \sum_i w_i \log(w_i)$
  until $KL \leq \Delta$
  {whiten the weigths}
  $\hat{w}_i \leftarrow \frac{w_i - \text{mean}(w_i)}{\text{std}(w_i)}$
  {compute the gradient on the smoothed cost}
  $g \leftarrow \sum_i \sum_t \hat{w}_i \; \frac{\partial}{\partial \theta} \log \pi_\theta(a^i_t|t, x^i_t)\big|_{\theta=\theta_n}$
  {compute Fisher matrix}
  use conjugate gradient descent to compute an approximate solution to the natural gradient $g_F = F^{-1}g$ (see App. E.2)
  do line search to compute step size η such $KL(\theta_n||\theta_{n+1}) = \mathcal{E}$.
  update parameters $\theta_{n+1} \leftarrow \theta_n + \eta \cdot g_F$
  $n = n + 1$
until convergence

Therefore

$$\Delta = KL(p^*_{\alpha,\theta}||p_{u_\theta})$$

is a monotonically decreasing function of α.

F Experimental Details and Additional Results

Algorithm 1 summarizes ASPIC. We first analyze the behavior of ASPIC in a simple linear-quadratic control problem, F.1,F.2. We then look at the dependence on the number of rollouts per iteration N in F.3 and the interplay between smoothing strength Δ and trust region size $\mathcal{E}$ in F.4. Finally, we describe the parameter settings for all tasks in F.5.

F.1 A Simple Linear-Quadratic Control Problem: Brownian Viapoints

We analyse the convergence speed for different values of the smoothing strength Δ in the task of controlling a one-dimensional Brownian particle

$$\dot{x} = u(x,t) + \xi. \tag{53}$$

We define the state cost as a quadratic penalty for deviating from the viapoints x_i at the different times t_i: $V(x,t) = \sum_i \delta(t - t_i) \frac{(x-x_i)^2}{2\sigma^2}$ with $\sigma = 0.1$. As a parametrized controller we use a time

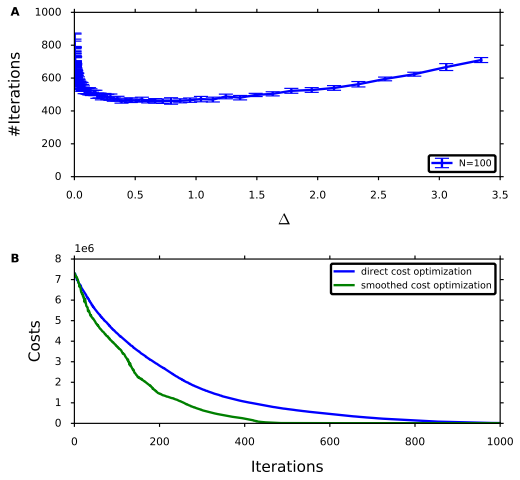

Figure 3: LQ control problem: Brownian viapoints. For each iteration we used $N = 100$ rollouts to compute the gradient. A) Number of iterations needed for the cost to cross a threshold $C \leq 2 \cdot 10^4$ versus the smoothing strength Δ. For $\Delta = 0$ there is no smoothing. Increasing the smoothing strength results in a faster decrease of the cost; when Δ is increased further the performance decreases again. Errorbars denote mean and standard deviation over 10 runs of the algorithm. B) Cost versus the iterations of the algorithm. Direct optimization of the cost exhibits a slower convergence rate than optimization of the smoothed cost with $\Delta = 0.2 \log 100$.

varying linear feedback controller, i.e., $u_\theta(x,t) = \theta_{1,t}x + \theta_{0,t}$. This controller fulfils the requirement of full parametrization for this task (see App. F.2). For further details of the numerical experiment see appendix F.5.

We apply ASPIC to this control problem and compare its performance for different sizes of the smoothing strength Δ (see Fig. 3). The results confirm our expectations from our theoretical analysis. As predicted by theory we observe an acceleration of the policy optimization when smoothing is switched on. This acceleration becomes more pronounced when Δ is increased, which we attribute to an increase of the anticipatory effect of the smoothed updates as smoothing becomes stronger (see section D). When Δ is too large the performance of the algorithm deteriorates again, which is in line with our discussion of gradient estimation problems that arise for strong smoothing.

F.2 Full parametrization in LQ problem

Here we discuss why for a linear quadratic problem a time varying linear controller is a full parametrization. We want to show that for every

$$p^*_{\alpha,\theta_0} = \frac{1}{Z} p_{u_0}(\tau) \exp\left(-\frac{1}{\gamma+\alpha} S^\gamma_{p_{u_{\theta_0}}}(\tau)\right) \tag{54}$$

there is a time varying linear controller $u_{\theta^*_{\alpha,\theta_0}}$ such that $p_{u_{\theta^*_{\alpha,\theta_0}}} = p^*_{\alpha,\theta_0}$. We assume that u_{θ_0} is a time varying linear controller. In App. B.1 we have shown that u^*_{α,θ_0} is the solution to the Path Integral control problem with dynamics

$$\dot{x}_t = f(x_t,t) + g(x_t,t)\left(\tilde{u}(x_t,t) + \hat{u}(x_t,t) + \xi_t\right)$$

and cost

$$\left\langle \int_0^T \frac{1}{\gamma} V(x_t,t) - \frac{1}{2}\frac{\gamma}{\alpha}\tilde{u}(x_t,t)^T\tilde{u}(x_t,t)dt + \int_0^T \left(\frac{1}{2}\hat{u}(x_t,t)^T\hat{u}(x_t,t) + \hat{u}(x_t,t)^T\xi_t\right)dt \right\rangle_{p_{\hat{u}}}, \tag{55}$$

with $\tilde{u} = \left(1 - \frac{\gamma}{\gamma+\alpha}\right) u_{\theta_0}(x_t,t)$.

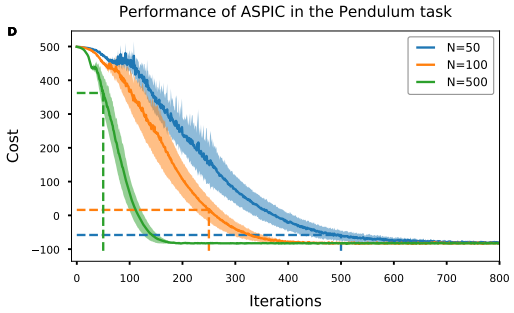

Figure 4: Performance as a function of the number of iterations for different values of $N \in \{50, 100, 500\}$ in the Pendulum swing-up task. Dashed lines denote the solution for a total fixed budget of 25K rollouts, i.e., 500, 250, and 50 iterations, respectively. In this case, $N = 50$ achieves near optimal performance whereas using larger values of N leads to worse solutions.

It is now easy to see that if u_{θ_0} is a time varying linear controller, thus a linear function of the state, the cost is a quadratic function of the state x (note that $V(x_t, t)$ is quadratic in the LQ case). Thus for all values of α, u^*_{α,θ_0} is the solution to a linear quadratic control problem and thus a time varying linear controller (see e.g. [14]). Therefore a time varying linear controller is a full parametrization.

F.3 Dependence on the Number of Rollouts per Iteration N

We now analyze the dependence of the performance of ASPIC on the number of rollouts per iteration N. In general, using larger values of N allows for more reliable gradient estimates and achieves convergence in fewer iterations. However, too large N may be inefficient and lead to suboptimal solutions in the presence of a fixed budget of rollouts.

Figure 4 illustrates this trade-off in the Pendulum swing-up task for three values of N, including the previous one $N = 500$. For a total budget of 25K rollouts (dashed lines) the lowest value of $N = 50$ achieves near optimal performance and is preferable to the other choices, despite resulting in higher variance estimates and requiring more iterations until convergence.

F.4 Interplay Between Smoothing Strength Δ and Trust Region Size $\mathcal{E}$

To understand better the relation between the smoothing strength and the trust region sizes, we analyze empirically the performance of ASPIC as a function of both Δ and $\mathcal{E}$ parameters. We focus on the Acrobot task and in the setting of $N = 500$ and intermediate smoothing strength, when smoothing is most beneficial.

Figure 5 shows the cost as a function of Δ and $\mathcal{E}$ averaged over the first 500 iterations of the algorithm, and for 10 different runs. Larger (averaged) costs correspond runs where the algorithm fails to converge. Conversely, the lower cost, the fastest the convergence. In general, larger values of $\mathcal{E}$ lead to faster convergence. However, the convergence is less stable for smaller values of Δ. For stronger smoothing, the algorithm is more sensitive to $\mathcal{E}$.

F.5 Details of Numerical Experiments

Linear-Quadratic control Task

Dynamics: The dynamics are ODEs integrated by an Euler scheme (see section F.1). The differential equation is initialized at $x = 0$. $dt = 0.1$

Control problem: $\gamma = 1$. Time-Horizon $T = 10s$. State-Cost function: see section F.1. $(x_0, t_0) = (-10, 1)$, $(x_1, t_1) = (10, 2)$,$(x_2, t_2) = (-10, 3)$, $(x_3, t_3) = (-20, 4)$, $(x_4, t_4) = (-100, 5)$, $(x_5, t_5) = (-50, 6)$, $(x_6, t_6) = (10, 7)$, $(x_7, t_7) = (20, 8)$, $(x_8, t_8) = (30, 9)$. Variance of uncontrolled dynamics $\nu = 1$.

Algorithm: Batchsize: $N = 100$. $\mathcal{E} = 0.1$. $\Delta = 0.2 \log 100$. Conjugate gradient iterations: 2 (for each time step separately). The parametrized controller was initialized at $\theta = 0$.

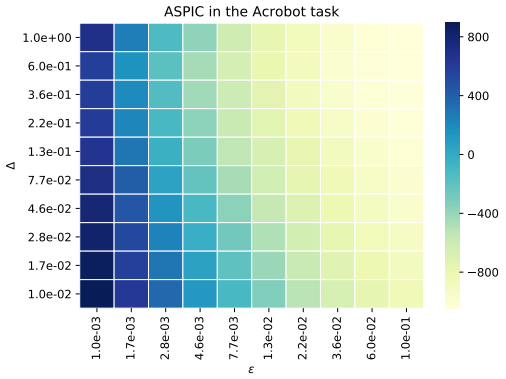

Figure 5: Solution cost as a function of the smoothing strength Δ and the trust region size ε in the Acrobot task. Shown is the cost averaged over the first 500 iterations of the algorithm, and for 10 different runs. Blue indicates failure to convergence. White indicates the solutions which converged fastest.

Pendulum Task

Dynamics: The differential equation for the pendulum is:

$$\ddot{x} + c\omega_0 \dot{x} + \omega_0^2 \sin(x) = \lambda\,(u + \xi)$$

with

- $c\omega_0 = 0.1\ [s^{-1}]$
- $\omega_0^2 = 10.\ [s^{-2}]$
- $\lambda = 0.2$

We implemented this differential equation as a first order differential equation and integrated it with an Euler scheme with $dt = 0.01$. The pendulum is initialized resting at the bottom:

$$\dot{x} = 0, x = 0.$$

As a parametrized controller we use a time varying linear feedback controller:

$$u_\theta(x, \dot{x}, t) = \theta_{3,t} \cos(x) + \theta_{2,t} \sin(x) + \theta_{1,t}\dot{x} + \theta_{0,t}.$$

The parametrized controller was initialized at $\theta = 0$.

Control-problem: $\gamma = 1.$. $T = 3.0s$. The State-Cost function has End-Cost only:

$$V(x, \dot{x}, t) = \delta(t - T)\left(-500Y + 10\dot{x}^2\right)$$

with $Y = -\cos(x)$ (height of tip). Variance of uncontrolled dynamics $\nu = 1$

Algorithm: Batchsize: $N = 500$. $\mathcal{E} = 0.1$. $\Delta = 0.5$. The Fisher-matrix was inverted for each time step separately using the scipy pseudo-inverse with rcond=1e-4.

Acrobot Task

Dynamics: We use the definition of the acrobot as in [23]. The differential equations for the acrobot are:

$$\begin{aligned} d_{11}(x)\ddot{x}_1 + d_{12}(x)\ddot{x}_2 + h_1(x, \dot{x}) + \phi_1(x) &= 0 \\ d_{21}(x)\ddot{x}_1 + d_{22}\ddot{x}_2 + h_2(x, \dot{x}) + \phi_2(x) &= \lambda \cdot (u + \xi) \end{aligned}$$

with

$$
\begin{aligned}
d_{11} &= m_1 l_{c1}^2 + m_2 \left(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} \cos(x_2)\right) + I_1 + I_2 \\
d_{12} &= m_2 \left(l_{c2}^2 + l_1 l_{c2} \cos(x_2)\right) + I_2 \\
d_{21} &= d_{12} \\
d_{22} &= m_2 l_{c2}^2 + I_2 \\
h_1 &= -m_2 l_1 l_{c2} \sin(x_2) \left(\dot{x}_2^2 + 2 \dot{x}_1 \dot{x}_2\right) \\
h_2 &= m_2 l_1 l_{c2} \sin(x_2) \dot{x}_1^2 \\
\phi_2 &= m_2 l_{c2} G \cos\left(x_1 + x_2\right) \\
\phi_1 &= (m_1 l_{c1} + m_2 l_1) g \cos\left(x_1\right) + \phi_2
\end{aligned}
$$

with the parameter values

- $G = 9.8$
- $l_1 = 1.$ [m]
- $l_2 = 2.$ [m]
- $m_1 = 1.$ [kg] mass of link 1
- $m_2 = 1.$ [kg] mass of link 2
- $l_{c1} = 0.5$ [m] position of the center of mass of link 1
- $l_{c2} = 1.0$ [m] position of the center of mass of link 2
- $I_1 = 0.083$ moments of inertia for both links
- $I_2 = 0.33$ moments of inertia for both links
- $\lambda = 0.2$

We implemented this differential equation as a first order differential equation and integrated it with an Euler scheme with $dt = 0.01$. The acrobot is initialized resting at the bottom:

$$\dot{x}_1 = 0, \dot{x}_2 = 0, x_1 = -\frac{1}{2}\pi, x_2 = 0.$$

As a parametrized controller we use a time varying linear feedback controller:

$$
\begin{aligned}
u_\theta(x, \dot{x}, t) =& \theta_{8,t} \cos(x_1) + \theta_{7,t} \sin(x_2) + \theta_{6,t} \cos(x_2) + \theta_{5,t} \sin(x_2) + \\
&+ \theta_{4,t} \sin(x_1 + x_2) + \theta_{3,t} \cos(x_1 + x_2) + \theta_{2,t} \dot{x_1} + \theta_{1,t} \dot{x_2} + \theta_{0,t}.
\end{aligned}
$$

The parametrized controller was initialized at $\theta = 0$.

Control-problem: $\gamma = 1.$. Time-Horizon: $T = 3.0s$. The State-Cost function has End-Cost only:

$$V(x, \dot{x}, t) = \delta(t - T) \left(-500 Y + 10(\dot{x_1}^2 + \dot{x_2}^2)\right)$$

with $Y = -l_1 \cos(x_1) - l_2 \cos(x_1 + x_2)$ (height of tip). Variance of uncontrolled dynamics $\nu = 1$.

Algorithm: Batchsize: $N = 500$. $\mathcal{E} = 0.1$. $\Delta = 0.5$. The Fisher-matrix was inverted for each time step separately using the scipy pseudo-inverse with rcond=1e-4.

Walker

Dynamics: For dynamics and the state cost function we used "BipedalWalker-v2" from the OpenAI gym [3]. The policy was a Gaussian policy, with static variance $\sigma = 1$. The state dependent mean of the Gaussian policy was a neural network controller with two hidden layers with 32 neurons, each. The activation function is a tanh. For the initialization we used Glorot Uniform (see [9]). The inputs to the neural network was the observation space provided by OpenAI gym task "BipedalWalker-v2": State consists of hull angle speed, angular velocity, horizontal speed, vertical speed, position of joints and joints angular speed, legs contact with ground, and 10 lidar rangefinder measurements.

Control-problem: $\gamma = 0$. Time-Horizon: defined by OpenAI gym task “BipedalWalker-v2”. State-Cost function defined by OpenAI gym task "BipedalWalker-v2": Reward is given for moving forward, total 300+ points up to the far end. If the robot falls, it gets -100. Applying motor torque costs a small amount of points, more optimal agent will get better score.

Algorithm: Batchsize: $N = 100$. $\mathcal{E} = 0.01$. $\Delta = 0.05 \log 100$. Conjugate gradient iterations: 10.