State-Regularized Policy Search for Linearized Dynamical Systems

Hany Abdulsamad[1] **Oleg Arenz**[2] **Jan Peters**[1,3] **Gerhard Neumann**[4]

1: IAS, TU Darmstadt, Darmstadt, Germany
2: CLAS, TU Darmstadt, Darmstadt, Germany
3: Max Planck Institute, Tübingen, Germany
4: L-CAS, University of Lincoln, Lincoln, United Kingdom

Abstract

Trajectory-Centric Reinforcement Learning and Trajectory Optimization methods optimize a sequence of feedback-controllers by taking advantage of local approximations of model dynamics and cost functions. Stability of the policy update is a major issue for these methods, rendering them hard to apply for highly nonlinear systems. Recent approaches combine classical Stochastic Optimal Control methods with information-theoretic bounds to control the step-size of the policy update and could even be used to train nonlinear deep control policies. These methods bound the relative entropy between the new and the old policy to ensure a stable policy update. However, despite the bound in policy space, the state distributions of two consecutive policies can still differ significantly, rendering the used local approximate models invalid. To alleviate this issue we propose enforcing a relative entropy constraint not only on the policy update, but also on the update of the state distribution, around which the dynamics and cost are being approximated. We present a derivation of the closed-form policy update and show that our approach outperforms related methods on two nonlinear and highly dynamic simulated systems.

Introduction

Policy Search is a powerful class for learning optimal control policies of complex systems (Deisenroth, Neumann, and Peters 2013). By allowing a very broad description of a task, it is suitable for solving challenging applications, where expert knowledge is scarce or can not be easily analytically encoded (Kober, Bagnell, and Peters 2013; Rosenstein and Barto 2001). Trajectory-Centric Policy Search (TCPS) methods optimize a sequence of feedback-controllers by taking advantage of local approximations of model dynamics and cost function. They have been particularly successful in training deep neural network representations of policies for complex tasks (Levine et al. 2016).

Many TCPS methods rely on Stochastic Optimal Control (SOC) with learned linearized dynamics. SOC with linearized dynamics is an established Trajectory Optimization method for controlling nonlinear systems. By alternating between linearizing the system dynamics and optimizing local Linear-Quadratic Regulators (LQR) in closed-form, it offers a general scheme for finding locally optimal solutions and approaching control problems from a classical model-based perspective. Fundamental work on SOC includes the Differential Dynamic Programming (DDP) algorithm (Jacobson and Mayne 1970), the Approximate Inference Control (AICO) algorithm (Toussaint 2009; Rawlik, Toussaint, and Vijayakumar 2012), the iterative Linear Quadratic Gaussian (iLQG) algorithm (Todorov and Li 2005; Tassa, Erez, and Todorov 2012), iLQG with a regularized cost function (Rueckert et al. 2014), and a maximum entropy formulation of iLQG (Levine and Abbeel 2014). A key element in the stability of such procedure is a mechanism to control the step size of the policy update in a principled manner. Too aggressive policy updates often lead the trajectory distribution into regions where the used local approximations of the dynamics model are invalid, causing oscillations or even divergence of such algorithms. The aforementioned methods confront this problem by using a number of regularizations to guarantee stable learning.

The maximum entropy version of iLQG (maxEnt-iLQG), introduced in the Guided Policy Search (GPS) algorithm (Levine and Abbeel 2014), uses a relative entropy bound, inspired by (Peters, Mülling, and Altun 2010), to regularize the updates of the trajectory distribution. This regularization improves the robustness of learning and has been very successful for training neural network policies. It is trivial to show that a bound on the trajectory distribution is equivalent to an overall bound on the policy update for the whole trajectory. Although such a bound significantly improves the learning behavior, it still does not directly limit the step-size in the state distributions of the update. However, as the state distributions are used to obtain a new local approximation of the system model, the stability of the policy update can only be controlled indirectly. In order to guarantee consistent learning, these methods implicitly assume that small changes in the policy induce only small changes in the state-action distribution, under which the linearization of dynamics and quadratization of cost take place. Hence, we are either restricted to close-linear dynamical systems, or overly conservative policy updates must be enforced.

In this article, we introduce a new bound for regularizing Stochastic Optimal Control methods that are based on local linearization of the system dynamics and quadratization of the cost function. By applying a relative entropy constraint on the resulting state distributions, we limit the

state distribution from changing drastically, resulting in trajectories that stay within the narrow validity scope of the approximated local models. We present a unifying derivation for iLQG algorithms with information-theoretical constraints and show that our method allows for more effective policy updates and finds solutions of higher quality. In our experimental evaluation we use two simulated multi-link systems for comparison with state of the art under a range of different conditions.

Notation

In the scope of this article we concentrate on finite-horizon Markov Decision Processes (MDP). A finite-MDP is a mathematical abstraction of an environment defined over a state space $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ and an action space $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ with a fixed number of time steps T. The probability of a state transition from state s to state s' by applying action a is governed by the Markovian time-independent density function $\mathcal{P}(s'|s,a)$. The cost $c_t(s,a)$ is a time-variant function of the state s and action a. The policy $\pi_t(a|s)$ from which the actions are drawn is a time-variant function determining the probability of an action a given a state s at a time step $t < T$.

The general objective of a finite-horizon Optimal Control problem is to minimize the sum of expected cost $\sum_{t=1}^{T-1} \int_s \int_a c_t(s,a)\mu_t(s)\pi_t(a|s)\mathrm{d}a\mathrm{d}s$, where $\mu_t(s)$ is the state distribution induced under the current policy distribution $\pi_t(a|s)$. General solutions to this problem are rare and restricted to a small class of systems. Aside from discrete systems, the most important exception are systems with linear-Gaussian dynamics $\mathcal{P}(s'|s,a) = \mathcal{N}(s'|As+ba+c, \Sigma_s)$ and quadratic cost functions in the state and action $c_t(s,a) = s^T Rs + a^T Ha + s^T r + a^T h + s^T Wa + c_0$. Ziebart, Bagnell, and Dey show that the optimal controller for a system adhering to these so called *LQG assumptions* can be derived in closed-form by applying Dynamic Programming (DP) (Bellman 1956) to compute the optimal state and state-action value functions, $V_t(s)$ and $Q_t(s,a)$. The resulting optimal policy is a time-varying linear-Gaussian distribution $\pi_t(a|s) = \mathcal{N}(a|K_t s + k_t, \Sigma_{t,a})$.

Related Work

In the following section we list the contributions of the most fundamental and successful approaches in Stochastic Optimal Control under linearized dynamics and motivate our further inquiry into the subject.

Differential Dynamic Programming

Jacobson and Mayne (1970) introduce Differential Dynamic Programing. It follows the main scheme of SOC methods. By applying Dynamic Programming on the perturbations of the state-action value function $Q_t(\delta s, \delta a)$, the authors derive an optimal local linear controller δa_t^* that minimizes the expected cost-to-go

$$\begin{aligned}\delta a_t^* &= \operatorname*{argmin}_{\delta a} Q_t(\delta s, \delta a) \\ &= -Q_{aa,t}^{-1}(Q_{a,t} + Q_{as,t}\delta s) = K_t \delta s + k_t,\end{aligned}$$

where $Q_{a,t}, Q_{aa,t}$ and $Q_{as,t}$ are the Jacobians and Hessians of $Q_t(\delta s, \delta a)$. Applying the new policy to the nonlinear system to get a new trajectory completes one cycle of DDP. This formulation greedily exploits the local dynamics and produces policies that can be arbitrarily different between iterations, undermining the locality of the linearization. In most cases this leads to divergence of the learning. The authors address this issue by introducing a hand-tuned regularization μ to the action-Hessian $\tilde{Q}_{aa,t} = Q_{aa,t} + \mu I$, which is equivalent to applying an additional cost term for straying away from the policy inducing the current trajectory.

Iterative Linear Quadratic Gaussian

Iterative Linear Quadratic Gaussian aims to correct the shortcomings of DDP by proposing several improvements. Tassa, Erez, and Todorov (2012) artificially introduce a new regularization on the state cost, that punishes moving far from the last mean trajectory. Furthermore, they introduce a parameter $\alpha \in [0,1]$ to the updated policy $a_t^* = K_t \delta s + \alpha k_t + a_t$. The parameter α scales down the exploitation of local approximations in the action space and is optimized based on a heuristic that takes into consideration the discrepancy between the expected and actual cost.

Maximum Entropy iLQG

Levine and Abbeel (2014) introduce Maximum Entropy iLQG as an equivalent to iLQG in a completely stochastic Policy Search setup. However, the authors discard the state regularization introduced in earlier iLQG work. Maximum Entropy iLQG was originally formulated with a Kullback-Leibler-Divergence (KLD) constraint on the update of the trajectory distribution $p(\tau)$, where τ is a set of consecutive states and actions $\{s_1, a_1, ..., a_{T-1}, s_T\}$. For the purpose of this article we transform the optimization problem into a state-action notation. As a result the original KLD constraint can be rewritten as follows

$$\begin{aligned}&\int_\tau p(\tau) \log \frac{p(\tau)}{q(\tau)} \mathrm{d}\tau \\ &= \int_\tau p(\tau) \log \frac{\mu_1(s) \prod_{t=1}^{T-1} \mathcal{P}(s'|s,a)\pi_t(a|s)}{\mu_1(s) \prod_{t=1}^{T-1} \mathcal{P}(s'|s,a) q_t(a|s)} \mathrm{d}\tau.\end{aligned} \tag{1}$$

It becomes trivial to show that such a constraint is equivalent to an overall expected upper bound on the update of the policy distribution. Thus, the whole optimization problem can be reformulated as follows

$$\operatorname*{argmin}_{\pi_t(a|s)} \sum_{t=1}^{T-1} \int_s \int_a c_t(s,a)\mu_t(s)\pi_t(a|s)\mathrm{d}a\mathrm{d}s, \tag{2a}$$

$$\text{s.t.} \int_s \int_a \mu_t(s)\pi_t(a|s)\mathcal{P}(s'|s,a)\mathrm{d}a\mathrm{d}s = \mu_{t+1}(s'), \tag{2b}$$

$$\sum_{t=1}^{T-1} \int_s \mu_t(s) \int_a \pi_t(a|s) \log \frac{\pi_t(a|s)}{q_t(a|s)} \mathrm{d}a\mathrm{d}s \leq \epsilon, \tag{2c}$$

$$\int_a \pi_t(a|s)\mathrm{d}a = 1, \quad \mu_1(s) = p_1(s), \tag{2d}$$

where $q_t(a|s)$ is the policy distribution of the last iteration and ϵ is the bound hyperparameter. A *forward pass* constraint, Eq(2b), propagates the initial state distribution $\mu_1(s)$ under the approximated transition dynamics $\mathcal{P}(s'|s,a)$ and current policy $\pi_t(a|s)$ to obtain the state distribution $\mu_t(s)$ at each time step. This constraint can be calculated in closed-form under linear-Gaussian dynamics and policy. Finally, the relative entropy constraint, Eq(2c), is transformed into the sum of expected KLD between $\pi_t(a|s)$ and $q_t(a|s)$ under the distribution $\mu_t(s)$.

By solving the primal problem using the method of Lagrangian multipliers (Boyd and Vandenberghe 2004), the optimal closed-form policy update is a softmax distribution

$$\pi_t^*(a|s) \propto \exp\Big(\frac{Q_t(s,a)}{\alpha}\Big). \tag{3}$$

$Q_t(s,a)$ is the state-action value function, which includes a policy-dependent term that augments the effective cost optimized by maxEnt-iLQG as given in the equation

$$\begin{aligned} Q_t(s,a) = c_t(s,a) + \alpha \log q_t(a|s) \\ + \int_{s'} V_{t+1}(s')\mathcal{P}(s'|s,a)\mathrm{d}s'. \end{aligned} \tag{4}$$

The state value function $V_t(s)$ and α are Lagrangian multipliers associated with the transition dynamics and KLD constraints respectively. By plugging the optimal policy update, Eq(3), back into the primal problem we get the following dual objective

$$\begin{aligned} \mathcal{G} = \int_s V_1(s)p_1(s)\mathrm{d}s - \sum_{t=1}^{T}\int_{s'} V_t(s')\mu_t(s')\mathrm{d}s' + \alpha\epsilon \\ + \sum_{t=1}^{T-1}\int_s \alpha\mu_t(s)\log\int_a \exp\Big(\frac{Q_t(s,a)}{\alpha}\Big)\mathrm{d}a\mathrm{d}s \end{aligned} \tag{5}$$

By taking the gradient of $\mathcal{G}$ w.r.t. $\mu_t(s)$ and $V_t(s)$, we obtain two optimality conditions, a *backward* and a *forward pass*, for calculating the optimal state value function $V_t^*(s)$ and state distribution $\mu_t^*(s)$ in closed-form

$$V_t^*(s) = \alpha \log \int_a \exp\Big(\frac{Q_t(s,a)}{\alpha}\Big)\mathrm{d}a, \tag{6a}$$

$$\mu_{t+1}^*(s') = \int_s\int_a \mu_t(s)\pi_t(a|s)\mathcal{P}(s'|s,a)\mathrm{d}a\mathrm{d}s, \tag{6b}$$

whereas the gradient w.r.t. α reflects the original bound

$$\frac{\partial\mathcal{G}}{\partial\alpha} = \epsilon - \sum_{t=1}^{T-1}\int_s \mu_t(s)\int_a \pi_t(a|s)\log\frac{\pi_t(a|s)}{q_t(a|s)}\mathrm{d}a\mathrm{d}s, \tag{7}$$

allowing us to optimize α by applying gradient ascent on the dual objective, that finally simplifies to

$$\mathcal{G}(\mu, V, \alpha) = \int_s V_1^*(s)\mu_1(s)\mathrm{d}s + \alpha\epsilon, \tag{8}$$

when the optimality conditions Eq(6a) and Eq(6b) are met.

State Action Relative Entropy Regularization

While enforcing a policy KLD bound grants us some control over the exploration process, it still avoids the main issue regarding learning with approximate models. Particularly, it ignores that these approximations are valid only in a very small region around the current state distribution. Approaches, like maxEnt-iLQG, that regularize only the action are prone to greedy exploitation of the state, which introduces considerable errors into the dynamics and cost unless conservative updates are applied.

To address this issue we propose the introduction of a new relative entropy constraint on the update of the state distribution. Such a constraint gives an explicit guarantee regarding the integrity of the linear or quadratic approximation. Furthermore, we suggest spreading the previous action bound over the whole trajectory, by setting an explicit upper bound for every time step instead of the expected overall bound maxEnt-iLQG enforces. In the following we write the full optimization problem with all constraints

$$\operatorname*{argmin}_{\pi_t(a|s)} \sum_{t=1}^{T-1}\int_s\int_a c_t(s,a)\mu_t(s)\pi_t(a|s)\mathrm{d}a\mathrm{d}s, \tag{9a}$$

$$\text{s.t.} \int_s\int_a \mu_t(s)\pi_t(a|s)\mathcal{P}(s'|s,a)\mathrm{d}a\mathrm{d}s = \mu_{t+1}(s'), \tag{9b}$$

$$\forall t < T \int_s \mu_t(s)\int_a \pi_t(a|s)\log\frac{\pi_t(a|s)}{q_t(a|s)}\mathrm{d}a\mathrm{d}s \le \epsilon_t, \tag{9c}$$

$$\forall t < T \int_s \mu_t(s)\log\frac{\mu_t(s)}{q_t(s)}\mathrm{d}s \le \eta_t, \tag{9d}$$

$$\forall t < T \int_a \pi_t(a|s)\mathrm{d}a = 1, \quad \mu_1(s) = p_1(s), \tag{9e}$$

where $q_t(s)$ is the state distribution induced by the policy $q_t(a|s)$ and η_t are the state bound hyperparameters. The modification of the policy bound has two conflicting implications. On the one hand, it is important for imposing the state constraint, as it introduces more Lagrangian multipliers that provide the optimization with more degrees of freedom to find a local solution to satisfy both constraints. On the other hand, it makes the policy updates more conservative, since it limits the possibility of modulating the action updates on the trajectory-level and may thus inhibit exploration in regions of the trajectory associated with lower cost. However, this issue can be ultimately elevated by optimizing the bound hyperparameters accordingly. We also mention the possibility of setting a softer state bound by transforming Eq(9d) to constrain the overall sum of state distribution divergences over all time steps.

We solve this optimization analogously to our reformulation of maxEnt-iLQG by starting from the primal problem with the Lagrangian function. The resulting optimal policy update retains its form, Eq(3), while the state-action value function $Q_t(s,a)$ gains a new state-dependent cost term reflecting the newly introduced state constraints

$$\begin{aligned} Q_t(s,a) = c_t(s,a) + \alpha\log q_t(a|s) - \gamma_t\log\frac{\mu_t(s)}{q_t(s)} \\ - \gamma_t + \int_{s'} V_{t+1}(s')\mathcal{P}(s'|s,a)\mathrm{d}s', \end{aligned} \tag{10}$$

Algorithm 1: State Action Relative Entropy Regularization for Trajectory Optimization (STATO)

input : ϵ_t, η_t, $\mathcal{P}$, c_t, μ_1, q_t
output: π_t^*, μ_t^*, V_t^*
initialize: α_t, γ_t
while dual objective $\mathcal{G}$ not at maximum **do**
  $p_t \leftarrow q_t$
  do
    $V_t, \pi_t \leftarrow \text{backwardPass}(p_t, q_t, \mathcal{P}, c_t, \alpha_t, \gamma_t)$
    $\mu_t \leftarrow \text{forwardPass}(\mu_1, \mathcal{P}, \pi_t)$
    $p_t \leftarrow \mu_t$
  while $\text{KLD}(p_t||\mu_t) > \delta$
  $\mathcal{G} \leftarrow \text{updateDual}(V_t, \mu_t, \alpha_t, \gamma_t, \epsilon_t, \eta_t)$
  $\frac{\partial \mathcal{G}}{\partial \alpha_t}, \frac{\partial \mathcal{G}}{\partial \gamma_t} \leftarrow \text{dualGradient}(\mu_t, \pi_t, q_t, \epsilon_t, \eta_t)$
  $\alpha_t \leftarrow \alpha_t + \frac{\partial \mathcal{G}}{\partial \alpha_t}, \quad \gamma_t \leftarrow \gamma_t + \frac{\partial \mathcal{G}}{\partial \gamma_t}$
end

where γ_t are the Lagrangian multipliers associated with state constraint. By taking the gradients of the dual function w.r.t. $\mu_t(s)$, we get a *backward pass*, whose form corresponds to Eq(6a), and computes the optimal state value function

$$V_t^*(s) = \alpha_t \log \int_a \exp\Big(\frac{Q_t(s,a)}{\alpha_t}\Big) \mathrm{d}a. \tag{11}$$

In addition, we get to the necessary gradients for optimizing the multipliers α_t and γ_t by gradient ascent to maximize dual objective assuming that the aforementioned optimality conditions are met

$$\mathcal{G} = \int_s V_1^*(s)\mu_1(s)\mathrm{d}s + \sum_{t=1}^{T-1} \Big(\alpha_t \epsilon_t + \gamma_t \eta_t + \gamma_t\Big), \tag{12a}$$

$$\frac{\partial \mathcal{G}}{\partial \alpha_t} = \epsilon_t - \int_s \int_a \mu_t(s)\pi_t(a|s) \log \frac{\pi_t(a|s)}{q_t(a|s)} \mathrm{d}a\mathrm{d}s, \tag{12b}$$

$$\frac{\partial \mathcal{G}}{\partial \gamma_t} = \eta_t - \int_s \mu_t(s) \log \frac{\mu_t(s)}{q_t(s)} \mathrm{d}s. \tag{12c}$$

As a result of the new state bound a circular dependency arises between $\mu_t(s)$, the state distribution induced by the current policy $\pi_t(a|s)$, and the state value function $V_t(s)$, as implied by Eq(11) and (10). This dependency is a natural product of an optimization that aims to optimize a policy $\pi_t(a|s)$, a function of $V_t(s)$, to limit the relative divergence of the resulting state distribution $\mu_t(s)$ induced under $\pi_t(a|s)$. In order to resolve this dependency, we devise a scheme that alternates between updating $V_t(s)$ and $\mu_t(s)$ using the *backward* and *forward passes*. The complete procedure is shown in Algorithm 1.

However, this iterative process introduces significant computational complexity to the optimization, considering that $V_t(s)$ and $\mu_t(s)$ have to be constantly updated while optimizing α_t and γ_t. In our evaluations we found that applying block coordinate ascent on the dual variables $V_t(s)$, $\mu_t(s)$, α_t and γ_t leads to great reduction of the computational effort. This approach decouples the computation of

Algorithm 2: STATO with Batch Coordinate Ascent

input : ϵ_t, η_t, $\mathcal{P}$, c_t, μ_1, q_t
output: π_t^*, μ_t^*, V_t^*
initialize: α_t, γ_t
while dual objective $\mathcal{G}$ not at maximum **do**
  $p_t \leftarrow q_t$
  do
    $V_t, \pi_t \leftarrow \text{backwardPass}(p_t, q_t, \mathcal{P}, c_t, \alpha_t, \gamma_t)$
    $\mu_t \leftarrow \text{forwardPass}(\mu_1, \mathcal{P}, \pi_t)$
    $p_t \leftarrow \mu_t$
  while $\text{KLD}(p_t||\mu_t) > \delta$
  while sec. dual objective $\hat{\mathcal{G}}$ not at maximum **do**
    $\hat{V}_t, \hat{\pi}_t \leftarrow \text{modBackwardPass}(q_t, \mathcal{P}, c_t, \alpha_t, \gamma_t)$
    $\hat{\mu}_t \leftarrow \text{modForwardPass}(\hat{V}_t, V_t, \gamma_t)$
    $\hat{\mathcal{G}} \leftarrow \text{updateDual}(\hat{V}_t, \hat{\mu}_t, V_t, \alpha_t, \gamma_t, \epsilon_t, \eta_t)$
    $\frac{\partial \hat{\mathcal{G}}}{\partial \alpha_t}, \frac{\partial \hat{\mathcal{G}}}{\partial \gamma_t} \leftarrow \text{dualGradient}(\hat{\mu}_t, \hat{\pi}_t, q_t, \epsilon_t, \eta_t)$
    $\alpha_t \leftarrow \alpha_t + \frac{\partial \hat{\mathcal{G}}}{\partial \alpha_t}, \quad \gamma_t \leftarrow \gamma_t + \frac{\partial \hat{\mathcal{G}}}{\partial \gamma_t}$
  end
end

$V_t(s)$ and $\mu_t(s)$ from the optimization of α_t and γ_t, which now takes place under the secondary dual objective

$$\begin{aligned}\hat{\mathcal{G}} = &\int_s V_1(s)p_1(s)\mathrm{d}s - \sum_{t=1}^{T-1} \gamma_t \int_s \hat{\mu}_t(s) \log \hat{\mu}_t(s) \\ &- \sum_{t=1}^{T} \int_{s'} V_t(s')\hat{\mu}_t(s')\mathrm{d}s' + \sum_{t=1}^{T-1} \Big(\alpha_t \epsilon_t + \gamma_t \eta_t\Big) \\ &+ \sum_{t=1}^{T-1} \int_s \alpha_t \hat{\mu}_t(s) \log \int_a \exp\Big(\frac{\hat{Q}_t(s,a)}{\alpha_t}\Big) \mathrm{d}a\mathrm{d}s,\end{aligned} \tag{13}$$

where the term $\hat{Q}_t(s,a)$ stands for the modified state-action value function without a dependency on $\mu_t(s)$

$$\begin{aligned}\hat{Q}_t(s,a) = c_t(s,a) + \alpha_t \log q_t(a|s) + \gamma_t \log q_t(s) \\ - \gamma_t + \int_{s'} V_{t+1}(s')\mathcal{P}(s'|s,a)\mathrm{d}s',\end{aligned} \tag{14}$$

and its corresponding state value function $\hat{V}_t(s)$ is computed according to Eq(11), while $\hat{\mu}_t(s)$ is the modified state distribution derived from the following equation

$$\gamma_t \log \hat{\mu}_t(s) = \hat{V}_t(s) - V_t(s) - \gamma_t. \tag{15}$$

A description of our algorithm with batch coordinate ascent is shown in Algorithm 2.

Experimental Validation

We validate our approach, State Action Relative Entropy Trajectory Optimization (STATO), on two multi-link dynamical systems and compare it to maxEnt-iLQG. In the experiments we focus on two aspects to quantify the difference

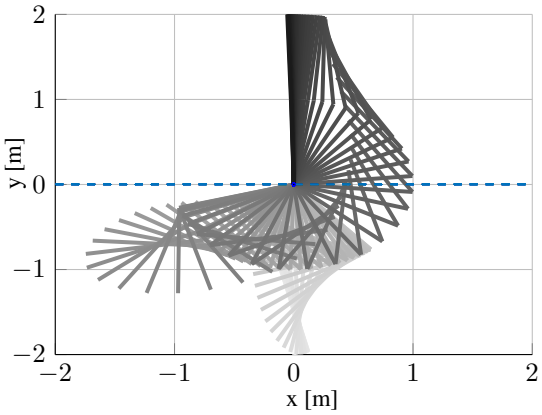

Figure 1: Illustration of the double-link swing-up task. The optimal policy (STATO) successfully achieves a full swing-up in two seconds, while subject to torque limits, friction and a non-quadratic cost function.

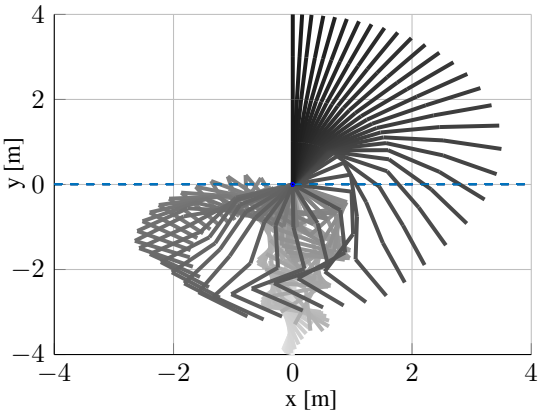

Figure 2: Illustration of the quad-link swing-up task. STATO's learned policy is able to achieve a full under-torqued swing-up in four seconds, while subject to highly nonlinear dynamics and a quadratic cost function.

in performance between the two algorithms. The first aspect is related to the degree of complexity and nonlinearity of the multi-link system dynamics, while the second is concerned with applying both quadratic non-quadratic cost functions.

The general objective in both tasks is to achieve an under-torqued swing-up and stabilization on a fully actuated double- and quad-link system. Figures 1 and 2 illustrate trajectories resulting from the learned controllers. Both systems possess highly nonlinear dynamics, although the quad-link is considerably more challenging. The state space consists of the joints and joint velocities, while the actions are the torques applied directly to the joints. Although we allow the state cost to become non-quadratic, the action cost is quadratic at all time steps and is summed up over the entire trajectory. All hyperparameters of both algorithms are optimized individually for each experiment to get the best results possible under the specified conditions. During all experiments we average over 10 trials to minimize statistical bias in the results. Furthermore, both algorithms use the same optimization toolbox under identical conditions.

At each iteration of the learning process, we draw a varying but sufficient number of samples under the current policy. We use these samples to fit linear-Gaussian dynamics at each time step of the current trajectory, which consists of 100 time steps in all experiments. In addition to approximating the linear dynamics, we also use the same sample batch to fit a quadratic cost function.

Double-Link To make the task challenging we apply torque limits, friction and a large initial distribution. The policy has two seconds, divided to 100 time steps, to finish. The first experiment implements a quadratic cost function in the joint space, while the second implements a cost defined in the task space by specifying only the target position of the end-effector. Figures 3 and 4 show STATO clearly outperforming maxEnt-iLQG regarding the quality of the final policy and overall efficiency, which is reflected in a lower number of iterations and total number of samples.

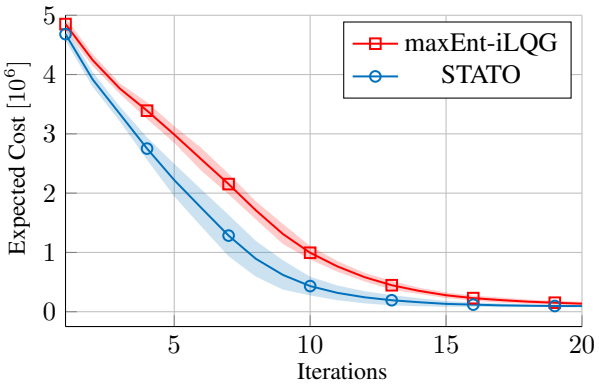

Figure 3: Double-Link swing-up results with a quadratic cost. STATO and maxEnt-iLQG learn comparable policies. However, STATO proves to have higher sample efficiency.

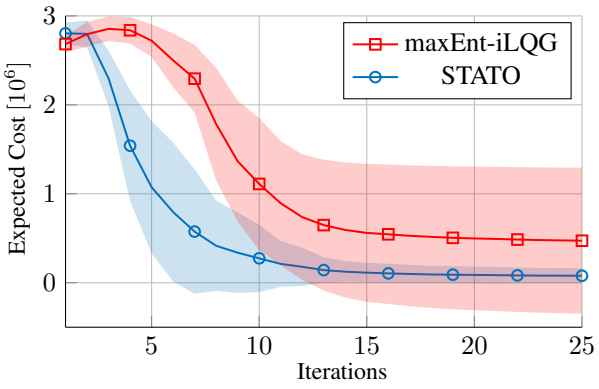

Figure 4: Double-Link swing-up results with a non-quadratic cost. STATO's final policy clearly outperforms maxEnt-iLQG, achieving a lower cost in less iterations.

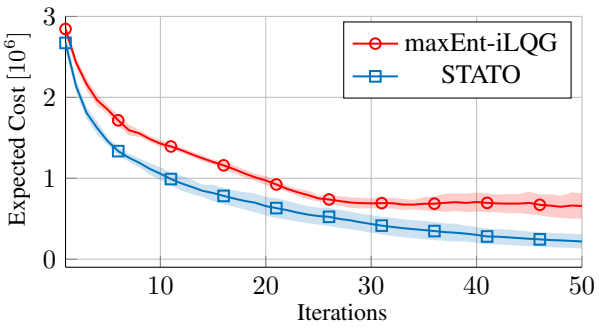

Figure 5: Quad-Link swing-up results with a quadratic cost. STATO's final policy is significantly better and more sample efficient compared to maxEnt-iLQG, even in a challenging task with soft joint limits and high nonlinearities.

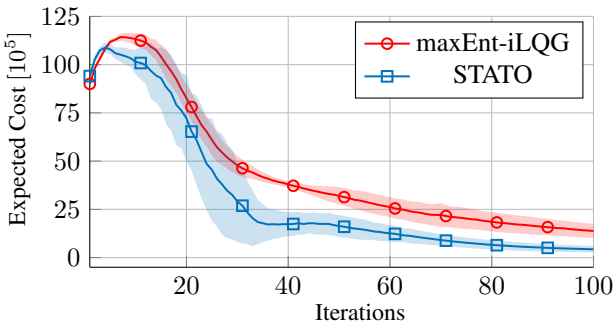

Figure 6: Quad-Link swing-up results with a non-quadratic cost. Once again STATO tops maxEnt-iLQG regarding the performance of the final policy and learning speed.

Quad-Link We compare STATO and maxEnt-iLQG in a swing-up task with both with quadratic and non-quadratic cost defined in the task space of the end-effector. Each experiment lasts for 4 seconds. Furthermore, instead of clipping the torques, we apply soft joint limits that become active if the absolute joint value exceeds $2\pi/3$. The results are depicted in Figures 5 and 6 and show that STATO is more capable of dealing with both the nonlinear dynamics of the quad-link and the approximation errors of the cost in comparison to maxEnt-iLQG. This is reflected in a lower expected final cost and considerably lower number of samples.

Conclusion

We proposed a new Trajectory Optimization algorithm with a novel relative entropy constraint on the update of the state distribution. By combining both the policy and state constraints, we offer an explicit measure of the divergence of the state-action distribution and a principled approach for updating the policy in a Stochastic Optimal Control setup. Our contribution proves its significance in guaranteeing the validity of approximate local models of cost and dynamics and the stability of learning. By validating our approach on two highly dynamic and nonlinear system, we showed that it outperforms state-of-the-art maxEnt-iLQG in both the quality of the final policy and sample efficiency, being able to learn better controllers in less number of iterations. Given its sample efficiency, our algorithm has a considerable advantage in robotic applications, where the number of trials drawn from the real system is crucial. Furthermore, we will investigate the behavior of our algorithm in the presence of high dimensional state input and explore its impact on the training of highly complex policies such as neural networks.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement # 645582 (RoMaNS) and grant agreement # 640554 (SKILLS4ROBOTS).

References

Bellman, R. 1956. Dynamic programming and lagrange multipliers. *National Academy of Sciences* 42(10):767–769.

Boyd, S., and Vandenberghe, L. 2004. Convex optimization.

Deisenroth, M. P.; Neumann, G.; and Peters, J. 2013. A survey on policy search for robotics. *Foundations and Trends in Robotics* 2(1–2):1–142.

Jacobson, D. H., and Mayne, D. Q. 1970. Differential dynamic programming.

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*.

Levine, S., and Abbeel, P. 2014. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, 1071–1079.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research* 17(39):1–40.

Peters, J.; Mülling, K.; and Altun, Y. 2010. Relative entropy policy search. In *Association for the Advancement of Artificial Intelligence*, 1607–1612.

Rawlik, K.; Toussaint, M.; and Vijayakumar, S. 2012. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems VIII*.

Rosenstein, M. T., and Barto, A. G. 2001. Robot weightlifting by direct policy search. In *International Joint Conference on Artificial Intelligence*, 839–846.

Rueckert, E.; Mindt, M.; Peters, J.; and Neumann, G. 2014. Robust policy updates for stochastic optimal control. In *IEEE-RAS International Conference on Humanoid Robots*, 388–393.

Tassa, Y.; Erez, T.; and Todorov, E. 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4906–4913.

Todorov, E., and Li, W. 2005. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *American Control Conference*, 300–306.

Toussaint, M. 2009. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning*, 1049–1056.

Ziebart, B. D.; Bagnell, J. A.; and Dey, A. K. 2010. Modeling interaction via the principle of maximum causal entropy. In *International Conference on Machine Learning*, 1247–1254.