Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator

Maryam Fazel [*1] **Rong Ge** [*2] **Sham M. Kakade** [*1] **Mehran Mesbahi** [*1]

Abstract

Direct policy gradient methods for reinforcement learning and continuous control problems are a popular approach for a variety of reasons: 1) they are easy to implement without explicit knowledge of the underlying model, 2) they are an "end-to-end" approach, directly optimizing the performance metric of interest, 3) they inherently allow for richly parameterized policies. A notable drawback is that even in the most basic continuous control problem (that of linear quadratic regulators), these methods must solve a non-convex optimization problem, where little is understood about their efficiency from both computational and statistical perspectives. In contrast, system identification and model based planning in optimal control theory have a much more solid theoretical footing, where much is known with regards to their computational and statistical properties. This work bridges this gap showing that (model free) policy gradient methods globally converge to the optimal solution and are efficient (polynomially so in relevant problem dependent quantities) with regards to their sample and computational complexities.

1. Introduction

Recent years have seen major advances in the control of uncertain dynamical systems using reinforcement learning and data-driven approaches; examples range from allowing robots to perform more sophisticated controls tasks such as robotic hand manipulation (Tassa et al., 2012; Al Borno et al., 2013; Kumar et al., 2016; Levine et al., 2016; Tobin et al., 2017; Rajeswaran et al., 2017a), to sequential decision making in game domains, e.g., AlphaGo (Silver et al., 2016) and Atari game playing (Mnih et al., 2015). Deep reinforcement learning (DeepRL) is becoming increasingly popular for tackling such challenging sequential decision making problems.

Many of these successes have relied on sampling based reinforcement learning algorithms such as policy gradient methods, including the DeepRL approaches. For these approaches, there is little theoretical understanding of their efficiency, either from a statistical or a computational perspective. In contrast, control theory (optimal and adaptive control) has a rich body of tools, with provable guarantees, for related sequential decision making problems, particularly those that involve continuous control. These latter techniques are often model-based—they estimate an explicit dynamical model first (via system identification) and then design optimal controllers.

This work builds bridges between these two lines of work, namely, between optimal control theory and sample based reinforcement learning methods, using ideas from mathematical optimization.

1.1. The optimal control problem

In the standard optimal control problem, a dynamical system is described as

$$x_{t+1} = f_t(x_t, u_t, w_t)\,,$$

where f_t maps a state $x_t \in \mathbb{R}^d$, a control (the action) $u_t \in \mathbb{R}^k$, and a disturbance w_t, to the next state $x_{t+1} \in \mathbb{R}^d$, starting from an initial state x_0. The objective is to find the control input u_t which minimizes the long term cost,

$$\begin{aligned} \text{minimize} \quad & \sum_{t=0}^{T} c_t(x_t, u_t) \\ \text{such that} \quad & x_{t+1} = f_t(x_t, u_t, w_t) \;\; t = 0, \ldots, T. \end{aligned}$$

Here the u_t are allowed to depend on the history of observed states, and T is the time horizon (which can be finite or infinite). In practice, this is often solved by considering the linearized control (sub-)problem where the dynamics are approximated by

$$x_{t+1} = A_t x_t + B_t u_t + w_t,$$

[*]Equal contribution [1]University of Washington, Seattle, WA, USA [2]Duke University, Durham, NC, USA. Correspondence to: Rong Ge <rongge@cs.duke.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

and the costs are approximated by a quadratic function in x_t and u_t, e.g. (Todorov & Li, 2004). The present paper considers an important special case: the time homogenous, infinite horizon problem referred to as the linear quadratic regulator (LQR) problem. The results herein can also be extended to the finite horizon, time inhomogenous setting, discussed in Section 5.

We consider the following infinite horizon LQR problem,

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\left[\sum_{t=0}^{\infty}(x_t^\top Q x_t + u_t^\top R u_t)\right] \\ \text{such that} \quad & x_{t+1} = Ax_t + Bu_t\,, \quad x_0 \sim \mathcal{D}\,, \end{aligned}$$

where initial state $x_0 \sim \mathcal{D}$ is assumed to be randomly distributed according to distribution $\mathcal{D}$; the matrices $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times k}$ are referred to as system (or transition) matrices; $Q \in \mathbb{R}^{d\times d}$ and $R \in \mathbb{R}^{k\times k}$ are both positive definite matrices that parameterize the quadratic costs. For clarity, this work does not consider a noise disturbance but only a random initial state. The importance of (some) randomization for analyzing direct methods is discussed in Section 3.

Throughout, assume that A and B are such that the optimal cost is finite (for example, the controllability of the pair (A, B) would ensure this). Optimal control theory (Anderson & Moore, 1990; Evans, 2005; Bertsekas, 2011; 2017) shows that the optimal control input can be written as a linear function in the state,

$$u_t = -K^* x_t$$

where $K^* \in \mathbb{R}^{k\times d}$.

Planning with a known model. For the infinite horizon LQR problem, planning can be achieved by solving the Algebraic Riccati Equation (ARE),

$$P = A^T P A + Q - A^T P B(B^T P B + R)^{-1} B^T P A\,, \quad (1)$$

for a positive definite matrix P which parameterizes the "cost-to-go" (the optimal cost from a state going forward). The optimal control gain is then given as:

$$K^* = -(B^T P B + R)^{-1} B^T P A. \quad (2)$$

To find P, there are iterative methods, algebraic solution methods, and (convex) SDP formulations. Solving the ARE is extensively studied; one approach due to (Kleinman, 1968) (for continuous time) and (Hewer, 1971) (for discrete time) is to simply run the recursion $P_{k+1} = Q + A^T P_k A - A^T P_k B(R + B^T P_k B)^{-1} B^T P_k A$ where $P_1 = Q$, which converges to the unique positive semidefinite solution of the ARE (since the fixed-point iteration is contractive). Other approaches are direct and are based on linear algebra, which carry out an eigenvalue decomposition on a certain block matrix (called the Hamiltonian matrix) followed by a matrix inversion (Lancaster & Rodman, 1995). The LQR problem can also be expressed as a semidefinite program (SDP) with variable P as given in (Balakrishnan & Vandenberghe, 2003) (see Section A in the supplement).

However, these formulations: 1) do not directly parameterize the policy, 2) are not "end-to-end" approaches, in that they are not directly optimizing the cost function of interest, and 3) it is not immediately clear how to utilize these approaches in the model-free setting, where the agent only has simulation access. These issues are outlined in Section A of the supplement.

1.2. Contributions of this work

Even in the most basic case of the standard linear quadratic regulator model, little is understood as to how direct (model-free) policy gradient methods fare. This work provides rigorous guarantees, showing that, while in fact the approach deals with a non-convex problem, directly using (model free) local search methods leads to finding the globally optimal policy (i.e., a policy whose objective value is ϵ-close to the optimal). The main contributions are as follows:

- (Exact case) Even with access to exact gradient evaluation, little is understood about whether or not convergence to the optimal policy occurs, even in the limit, due to the non-convexity of the problem. This work shows that global convergence does indeed occur (and does so efficiently) for gradient descent methods.
- (Model free case) Without a model, this work shows how one can use simulated trajectories (as opposed to having knowledge of the model) in a stochastic policy gradient method, where provable convergence to a globally optimal policy is guaranteed, with (polynomially) efficient computational and sample complexities.
- (The natural policy gradient) Natural policy gradient methods (Kakade, 2001) — and related algorithms such as Trust Region Policy Optimization (Schulman et al., 2015) and the natural actor critic (Peters & Schaal, 2007) — are some of the most widely used and effective policy gradient methods (see (Duan et al., 2016)). While many results argue in favor of this method based on either information geometry (Kakade, 2001; Bagnell & Schneider, 2003) or based on connections to actor-critic methods (Deisenroth et al., 2013), these results do not provably show an improved convergence rate. This work is the first to provide a guarantee that the natural gradient method enjoys a considerably improved convergence rate over its naive gradient counterpart.

More broadly, the techniques in this work merge ideas from optimal control theory, mathematical optimization (first

order and zeroth order), and sample based reinforcement learning methods. These techniques may ultimately help in improving upon the existing set of algorithms, addressing issues such as variance reduction or improving upon the natural policy gradient method (with, say, a Gauss-Newton method as in Theorem 7). The Discussion section touches upon some of these issues.

1.3. Related work

In the reinforcement learning setting, the model is unknown, and the agent must learn to act through its interactions with the environment. Here, solution concepts are typically divided into: model-based approaches, where the agent attempts to learn a model of the world, and model-free approaches, where the agent directly learns to act and does not explicitly learn a model of the world. The related work on provably learning LQRs is reviewed from this perspective.

Model-based learning approaches. In the context of LQRs, the agent can attempt to learn the dynamics of "the plant" (i.e., the model) and then plan, using this model, for control synthesis. Here, the classical approach is to learn the model with subspace-based system identification (Ljung, 1999). Fiechter (1994) provides a provable learning (and non-asymptotic) result, where the quality of the policy obtained is shown to be near optimal (efficiency is in terms of the persistence of the training data and the controllability Gramian). Abbasi-Yadkori & Szepesvári (2011) also provides provable, non-asymptotic learning results in a regret context, using a bandit algorithm that achieves lower sample complexity (by balancing exploration-exploitation more effectively); the computational efficiency of this approach is less clear.

More recently, Dean et al. (2017) expands on an explicit system identification process, where a robust control synthesis procedure is adopted that relies on a coarse model of the plant matrices (A and B are estimated up to some accuracy level, naturally leading to a "robust control" setup to then design the controller based in the coarse model). Tighter analysis for sample complexity was given in Tu & Recht (2018); Simchowitz et al. (2018). Arguably, this is the most general (and non-asymptotic) result that is efficient from a statistical perspective. Computationally, the method works with a finite horizon to approximate the infinite horizon. This result only needs the plant to be controllable; the work herein needs the stronger assumption that the initial policy in the local search procedure is a stable controller (an assumption which may be inherent to local search procedures, discussed in Section 5). Another recent line of work (Hazan et al., 2017; 2018; Arora et al., 2018) treat the problem of learning a linear dynamical system as an online learning problem. (Hazan et al., 2017; Arora et al., 2018) are restricted to systems with symmetric dynamics (symmetric A matrix), while (Hazan et al., 2018) handles a more general setting. This line of work can handle the case when there are latent states (i.e., when the observed output is a linear function of the state, and the state is not observed directly) and does not need to do system identification first. On the other hand, they don't output a succinct linear policy as Dean et al. (2017) or this paper.

Model-free learning approaches. Model-free approaches that do not rely on an explicit system identification step typically either: 1) estimate value functions (or state-action values) through Monte Carlo simulation which are then used in some approximate dynamic programming variant (Bertsekas, 2011), or 2) directly optimize a (parameterized) policy, also through Monte Carlo simulation. Model-free approaches for learning optimal controllers are not well understood from a theoretical perspective. Here, Bradtke et al. (1994) provides an asymptotic learnability result using a value function approach, namely Q-learning.

2. Preliminaries and Background

2.1. Exact Gradient Descent

This work seeks to characterize the behavior of (direct) policy gradient methods, where the policy is linearly parameterized, as specified by a matrix $K \in \mathbb{R}^{k \times d}$ which generates the controls:

$$u_t = -Kx_t$$

for $t \geq 0$. The cost of this K is denoted as:

$$C(K) := \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} (x_t^\top Q x_t + u_t^\top R u_t) \right]$$

where $\{x_t, u_t\}$ is the trajectory induced by following K, starting with $x_0 \sim \mathcal{D}$. The importance of (some) randomization, either in x_0 or noise through having a disturbance, for analyzing gradient methods is discussed in Section 3. Here, K^* is a minimizer of $C(\cdot)$.

Gradient descent on $C(K)$, with a fixed stepsize η, follows the update rule:

$$K \leftarrow K - \eta \nabla C(K) \,.$$

It is helpful to explicitly write out the functional form of the gradient. Define P_K as the solution to:

$$P_K = Q + K^\top R K + (A - BK)^\top P_K (A - BK) \,.$$

and, under this definition, it follows that $C(K)$ can be written as:

$$C(K) = \mathbb{E}_{x_0 \sim \mathcal{D}} \, x_0^\top P_K x_0 \,.$$

Also, define Σ_K as the (un-normalized) state correlation matrix, i.e.

$$\Sigma_K = \mathbb{E}_{x_0 \sim \mathcal{D}} \sum_{t=0}^{\infty} x_t x_t^\top \,.$$

Lemma 1. *(Policy Gradient Expression) The policy gradient is:*

$$\nabla C(K) = 2\left((R + B^\top P_K B)K - B^\top P_K A\right)\Sigma_K$$

Later for simplicity, define E_K to be

$$E_K = \left((R + B^\top P_K B)K - B^\top P_K A\right),$$

as a result the gradient can be written as $\nabla C(K) = 2E_K\Sigma_K$.

Proof. Observe:

$$\begin{aligned} C_K(x_0) &= x_0^\top P_K x_0 \\ &= x_0^\top \left(Q + K^\top R K\right) x_0 \\ &\quad + x_0^\top (A - BK)^\top P_K (A - BK) x_0 \\ &= x_0^\top \left(Q + K^\top R K\right) x_0 \\ &\quad + C_K((A - BK)x_0)\,. \end{aligned}$$

Let ∇ denote the gradient with respect to K, note that $\nabla C_K((A - BK)x_0)$ has two terms (one with respect to K in the subscript and one with respect to the input $(A - BK)x_0$), this implies

$$\begin{aligned} \nabla C_K(x_0) &= 2RKx_0x_0^\top - 2B^\top P_K(A - BK)x_0x_0^\top \\ &\quad + \nabla C_K(x_1)|_{x_1=(A-BK)x_0} \\ &= 2\left((R + B^\top P_K B)K - B^\top P_K A\right)\sum_{t=0}^{\infty} x_t x_t^\top \end{aligned}$$

using recursion and that $x_1 = (A - BK)x_0$. Taking expectations completes the proof. □

2.2. Review: (Model free) sample based policy gradient methods

Sample based policy gradient methods introduce some randomization for estimating the gradient.

REINFORCE.(Williams, 1992; Sutton et al., 2000) Let $\pi_\theta(u|x)$ be a parametric stochastic policy, where $u \sim \pi_\theta(\cdot|x)$. The policy gradient of the cost, $C(\theta)$, is:

$$\nabla C(\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} Q_{\pi_{\theta_t}}(x_t, u_t)\nabla \log \pi_\theta(u_t|x_t)\right],$$

$$\text{where } Q_{\pi_\theta}(x, u) = \mathbb{E}\left[\sum_{t=0}^{\infty} c_t | x_0 = x, u_0 = u\right],$$

where the expectation is with respect to the trajectory $\{x_t, u_t\}$ induced under the policy π_θ and where $Q_{\pi_\theta}(x, u)$ is referred to as the state-action value. The REINFORCE algorithm uses Monte Carlo estimates of the gradient obtained by simulating π_θ.

The natural policy gradient. The natural policy gradient (Kakade, 2001) follows the update:

$$\theta \leftarrow \theta - \eta\, G_\theta^{-1}\nabla C(\theta), \text{where:}$$

$$G_\theta = \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla \log \pi_\theta(u_t|x_t)\nabla \log \pi_\theta(u_t|x_t)^\top\right],$$

where G_θ is the Fisher information matrix. There are numerous succesful related approaches (Peters & Schaal, 2007; Schulman et al., 2015; Duan et al., 2016). An important special case is using a linear policy with additive Gaussian noise (Rajeswaran et al., 2017b), i.e.

$$\pi_K(x, u) = \mathcal{N}(Kx, \sigma^2 \mathrm{I}) \tag{3}$$

where $K \in \mathbb{R}^{k\times d}$ and σ^2 is the noise variance. Here, the natural policy gradient of K (when σ is considered fixed) takes the form:

$$K \leftarrow K - \eta \nabla C(K)\Sigma_K^{-1} \tag{4}$$

To see this, one can verify that the Fisher matrix of size $kd \times kd$, which is indexed as $[G_K]_{(i,j),(i',j')}$ where $i, i' \in \{1, \dots k\}$ and $j, j' \in \{1, \dots d\}$, has a block diagonal form where the only non-zeros blocks are $[G_K]_{(i,\cdot),(i,\cdot)} = \Sigma_K$ (this is the block corresponding to the i-th coordinate of the action, as i ranges from 1 to k). This form holds more generally, for any diagonal noise.

Zeroth order optimization. Zeroth order optimization is a generic procedure (Conn et al., 2009; Nesterov & Spokoiny, 2015) for optimizing a function $f(x)$, using only query access to the function values of $f(\cdot)$ at input points x (and without explicit query access to the gradients of f). This is also the approach in using "evolutionary strategies" for reinforcement learning (Salimans et al., 2017). The generic approach can be described as follows: define the perturbed function as

$$f_{\sigma^2}(x) = \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,\sigma^2\mathrm{I})}[f(x + \varepsilon)]$$

For small σ, the smooth function is a good approximation to the original function. Due to the Gaussian smoothing, the gradient has the particularly simple functional form (see Conn et al. (2009); Nesterov & Spokoiny (2015)):

$$\nabla f_{\sigma^2}(x) = \frac{1}{\sigma^2}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,\sigma^2\mathrm{I})}[f(x + \varepsilon)\varepsilon]\,.$$

This expression implies a straightforward method to obtain an unbiased estimate of the $\nabla f_{\sigma^2}(x)$, through obtaining only the function values $f(x + \varepsilon)$ for random ε.

3. The (non-convex) Optimization Landscape

This section provides a brief characterization of the optimization landscape, in order to help provide intuition as to why global convergence is possible and as to where the analysis difficulties lie.

Lemma 2. *(Non-convexity) If $d \geq 3$, there exists an LQR optimization problem, $\min_K C(K)$, which is not convex, quasi-convex, and star-convex.*

The specific example is given in supplementary material (Section B). In particular, there can be two matrices K and K' where both $C(K)$ and $C(K')$ are finite, but $C((K + K')/2)$ is infinite.

For a general non-convex optimization problem, gradient descent may not even converge to the global optima in the limit. The optimization problem of LQR satisfies a special *gradient domination* condition, which makes it much easier to optimize:

Lemma 3. *(Gradient domination) Let K^* be an optimal policy. Suppose K has finite cost and $\sigma_{min}(\Sigma_K) > 0$. It holds that*

$$C(K) - C(K^*) \leq \frac{\|\Sigma_{K^*}\|}{\sigma_{min}(\Sigma_K)^2 \sigma_{min}(R)} \|\nabla C(K)\|_F^2.$$

This lemma can be proved by analyzing the "advantage" of the optimal policy Σ^* to Σ in every step. The detailed lemma and the full proof is deferred to supplementary material.

As a corollary, this lemma provides a characterization of the stationary points.

Corollary 4. *(Stationary point characterization) If $\nabla C(K) = 0$, then either K is an optimal policy or Σ_K is rank deficient.*

Note that the covariance $\Sigma_K \succeq \Sigma_0 := \mathbb{E}_{x_0 \sim \mathcal{D}} x_0 x_0^\top$. Therefore, this lemma is the motivation for using a distribution over x_0 (as opposed to a deterministic starting point): $\mathbb{E}_{x_0 \sim \mathcal{D}} x_0 x_0^\top$ being full rank guarantees that Σ_K is full rank, which implies all stationary points are a global optima. An additive disturbance in the dynamics model also suffices.

The concept of gradient domination is important in the non-convex optimization literature (Polyak, 1963; Nesterov & Polyak, 2006; Karimi et al., 2016). A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be gradient dominated if there exists some constant λ, such that for all x,

$$f(x) - \min_{x'} f(x') \leq \lambda \|\nabla f(x)\|^2 .$$

If a function is gradient dominated, this implies that if the magnitude of the gradient is small at some x, then the function value at x will be close to that of the optimal function value.

Using the fact that $\Sigma_K \succeq \Sigma_0$, the following corollary of Lemma 3 shows that $C(K)$ is gradient dominated.

Corollary 5. *(Gradient Domination) Suppose $\mathbb{E}_{x_0 \sim \mathcal{D}} x_0 x_0^\top$ is full rank. Then $C(K)$ is gradient dominated, i.e.*

$$C(K) - C(K^*) \leq \lambda \langle \nabla C(K), \nabla C(K) \rangle$$

where $\lambda = \frac{\|\Sigma_{K^*}\|}{\sigma_{min}(\Sigma_0)^2 \sigma_{min}(R)}$ is a problem dependent constant (and $\langle \cdot, \cdot \rangle$ denotes the trace inner product).

Naively, one may hope that gradient domination immediately implies that gradient descent converges quickly to the global optima. This would indeed be the case if the $C(K)$ were a smooth function[1]: if it were the case that $C(K)$ is both gradient dominated *and* smooth, then classical mathematical optimization results (Polyak, 1963) would not only immediately imply global convergence, these results would also imply convergence at a linear rate. These results are not immediately applicable due to it is not straightforward to characterize the (local) smoothness properties of $C(K)$; this is a difficulty well studied in the optimal control theory literature, related to robustness and stability.

Similarly, one may hope that recent results on escaping saddle points (Nesterov & Polyak, 2006; Ge et al., 2015; Jin et al., 2017) immediately imply that gradient descent converges quickly to the global optima, due to that there are no (spurious) local optima. Again, for reasons related to smoothness this is not the case.

The main reason that the LQR objective cannot satisfy the smoothness condition globally is that the objective becomes infinity when the matrix $A - BK$ becomes unstable (i.e. has an eigenvalue that is outside of the unit circle in the complex plane). At the boundary between stable and unstable policies, the objective function quickly becomes infinity, which violates the traditional smoothness conditions because smoothness conditions would imply quadratic upper-bounds for the objective function.

To solve this problem, it is observed that when the policy K is not too close to the boundary, the objective satisfies an almost-smoothness condition:

Lemma 6. *("Almost" smoothness) $C(K)$ satisfies:*

$$C(K') - C(K) = -2\mathrm{Tr}(\Sigma_{K'}(K - K')^\top E_K) + \mathrm{Tr}(\Sigma_{K'}(K - K')^\top (R + B^\top P_K B)(K - K'))$$

To see why this is related to smoothness (e.g. compare to Equation 13), suppose K' is sufficiently close to K so that:

$$\Sigma_{K'} \approx \Sigma_K + O(\|K - K'\|)$$

and the leading order term $2\mathrm{Tr}(\Sigma_{K'}(K' - K)^\top E_K)$ would then behave as $\mathrm{Tr}((K' - K)^\top \nabla C(K))$, and the remaining terms will be second order in $K - K'$.

Quantify the Taylor approximation $\Sigma_{K'} \approx \Sigma_K + O(\|K - K'\|)$ is one of the key steps in proving the convergence of policy gradient.

[1] A differentiable function $f(x)$ is said to be smooth if the gradients of f are continuous. Equivalently, see the definition in Equation 13.

4. Main Results

First, results on exact gradient methods are provided. From an analysis perspective, this is the natural starting point; once global convergence is established for exact methods, the question of using simulation-based, model-free methods can be approached with zeroth-order optimization methods (where gradients are not available, and can only be approximated using samples of the function value).

Notation. $\|Z\|$ denotes the spectral norm of a matrix Z; $\mathrm{Tr}(Z)$ denotes the trace of a square matrix; $\sigma_{\min}(Z)$ denotes the minimal singular value of a square matrix Z. Also, it is helpful to define

$$\mu := \sigma_{\min}(\mathbb{E}_{x_0 \sim \mathcal{D}} x_0 x_0^\top)$$

4.1. Model-based optimization: exact gradient methods

We consider three exact update rules. For gradient descent, the update is

$$K_{n+1} = K_n - \eta \nabla C(K_n). \tag{5}$$

For natural policy gradient descent, the direction is defined so that it is consistent with the stochastic case, as per Equation 4, in the exact case the update is:

$$K_{n+1} = K_n - \eta \nabla C(K_n) \Sigma_{K_n}^{-1} \tag{6}$$

For Gauss-Newton method, the update is:

$$K_{n+1} = K_n - \eta (R + B^\top P_{K_n} B)^{-1} \nabla C(K_n) \Sigma_{K_n}^{-1}. \tag{7}$$

The standard policy iteration algorithm(Howard, 1964) that tries to optimize a one-step deviation from the current policy is equivalent to a special case of the Gauss-Newton method when $\eta = 1$ (for the case of policy iteration, convergence in the limit is provided in (Todorov & Li, 2004; Ng et al., 2002; Liao & Shoemaker, 1991), along with local convergence rates.)

The Gauss-Newton method requires the most complex oracle to implement: it requires access to $\nabla C(K)$, Σ_K, and $R + B^\top P_K B$; it also enjoys the strongest convergence rate guarantee. At the other extreme, gradient descent requires oracle access to only $\nabla C(K)$ and has the slowest convergence rate. The natural policy gradient sits in between, requiring oracle access to $\nabla C(K)$ and Σ_K, and having a convergence rate between the other two methods.

Theorem 7. *(Global Convergence of Gradient Methods) Suppose $C(K_0)$ is finite and $\mu > 0$.*

- *Gauss-Newton case: For a stepsize $\eta = 1$ and for*

$$N \geq \frac{\|\Sigma_{K^*}\|}{\mu} \log \frac{C(K_0) - C(K^*)}{\varepsilon},$$

the Gauss-Newton algorithm (Equation 7) enjoys the following performance bound:

$$C(K_N) - C(K^*) \leq \varepsilon$$

- *Natural policy gradient case: For a stepsize*

$$\eta = \frac{1}{\|R\| + \frac{\|B\|^2 C(K_0)}{\mu}}$$

and for

$$N \geq \frac{\|\Sigma_{K^*}\|}{\mu} \left(\frac{\|R\|}{\sigma_{min}(R)} + \frac{\|B\|^2 C(K_0)}{\mu \sigma_{min}(R)} \right) \log \frac{C(K_0) - C(K^*)}{\varepsilon},$$

natural policy gradient descent (Equation 6) enjoys the following performance bound:

$$C(K_N) - C(K^*) \leq \varepsilon\,.$$

- *Gradient descent case: For an appropriate (constant) setting of the stepsize η,*

$$\eta = \mathrm{poly}\left(\frac{\mu \sigma_{min}(Q)}{C(K_0)}, \frac{1}{\|A\|}, \frac{1}{\|B\|}, \frac{1}{\|R\|}, \sigma_{min}(R) \right)$$

and for

$$N \geq \frac{\|\Sigma_{K^*}\|}{\mu} \log \frac{C(K_0) - C(K^*)}{\varepsilon} \mathrm{poly}\left(\frac{C(K_0)}{\mu \sigma_{min}(Q)}, \|A\|, \|B\|, \|R\|, \frac{1}{\sigma_{min}(R)} \right),$$

gradient descent (Equation 5) enjoys the following performance bound:

$$C(K_N) - C(K^*) \leq \varepsilon\,.$$

In comparison to model-based approaches, these results require the (possibly) stronger assumption that the initial policy is a stable controller, i.e. $C(K_0)$ is finite (an assumption which may be inherent to local search procedures). The Discussion mentions this as direction of future work.

The proof for Gauss-Newton algorithm is simple based on the characterizations in Lemma 3 and Lemma 6, and is given below. The proof for natural policy gradient and gradient descent are more involved, and are deferred to supplementary material.

Lemma 8. *Suppose that:*

$$K' = K - \eta (R + B^\top P_K B)^{-1} \nabla C(K) \Sigma_K^{-1},.$$

If $\eta \leq 1$, then

$$C(K') - C(K^*) \leq \left(1 - \frac{\eta \mu}{\|\Sigma_{K^*}\|}\right) (C(K) - C(K^*))$$

Algorithm 1 Model-Free Policy Gradient (and Natural Policy Gradient) Estimation

1: Input: K, number of trajectories m, roll out length ℓ, smoothing parameter r, dimension d
2: **for** $i = 1, \cdots m$ **do**
3: Sample a policy $\widehat{K}_i = K + U_i$, where U_i is drawn uniformly at random over matrices whose (Frobenius) norm is r.
4: Simulate $\widehat{K}_i$ for ℓ steps starting from $x_0 \sim \mathcal{D}$. Let $\widehat{C}_i$ and $\widehat{\Sigma}_i$ be the empirical estimates:

$$\widehat{C}_i = \sum_{t=1}^{\ell} c_t \,, \quad \widehat{\Sigma}_i = \sum_{t=1}^{\ell} x_t x_t^\top$$

where c_t and x_t are the costs and states on this trajectory.
5: **end for**
6: Return the (biased) estimates:

$$\widehat{\nabla C(K)} = \frac{1}{m}\sum_{i=1}^{m} \frac{d}{r^2}\widehat{C}_i U_i \,, \quad \widehat{\Sigma_K} = \frac{1}{m}\sum_{i=1}^{m}\widehat{\Sigma}_i$$

Proof. Observe $K' = K - \eta(R + B^\top P_K B)^{-1} E_K$. Using Lemma 6 and the condition on η,

$$\begin{aligned} & C(K') - C(K) \\ = \; & -2\eta \mathrm{Tr}(\Sigma_{K'} E_K^\top (R + B^\top P_K B)^{-1} E_K) + \\ & \eta^2 \mathrm{Tr}(\Sigma_{K'} E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ \leq \; & -\eta \mathrm{Tr}(\Sigma_{K'} E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ \leq \; & -\eta \sigma_{\min}(\Sigma_{K'}) \mathrm{Tr}(E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ \leq \; & -\eta \mu \mathrm{Tr}(E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ \leq \; & -\eta \frac{\mu}{\|\Sigma_{K^*}\|} (C(K) - C(K^*)) \,, \end{aligned}$$

where the last step uses Lemma 3. □

With this lemma, the proof of the convergence rate of the Gauss Newton algorithm is immediate.

Proof. (of Theorem 7, Gauss-Newton case) The theorem is due to that $\eta = 1$ leads to a contraction of $1 - \frac{\eta\mu}{\|\Sigma_{K^*}\|}$ at every step. □

4.2. Model free optimization: sample based policy gradient methods

In the model free setting, the controller has only simulation access to the model; the model parameters, A, B, Q and R, are unknown. The standard optimal control theory approach is to use system identification to learn the model, and then plan with this learned model This section proves that model-free, policy gradient methods also lead to globally optimal policies, with both polynomial computational and sample complexities (in the relevant quantities).

Using a zeroth-order optimization approach (see Section 2.2), Algorithm 1 provides a procedure to find (bounded bias) estimates, $\widehat{\nabla C(K)}$ and $\widehat{\Sigma}_K$, of both $\nabla C(K)$ and Σ_K. These can then be used in the policy gradient and natural policy gradient updates. For policy gradient we have

$$K_{n+1} = K_n - \eta \widehat{\nabla C(K_n)}. \tag{8}$$

For natural policy gradient we have:

$$K_{n+1} = K_n - \eta \widehat{\nabla C(K_n)} \widehat{\Sigma}_{K_n}^{-1} \,. \tag{9}$$

In both Equations (8) and (9), Algorithm 1 is called at every iteration to provide the estimates of $\nabla C(K_n)$ and Σ_{K_n}.

The choice of using zeroth order optimization vs using REINFORCE (with Gaussian additive noise, as in Equation 3) is primarily for technical reasons[2]. It is plausible that the REINFORCE estimation procedure has lower variance. One additional minor difference, again for technical reasons, is that Algorithm 1 uses a perturbation from the surface of a sphere (as opposed to a Gaussian perturbation).

Theorem 9. *(Global Convergence in the Model Free Setting) Suppose $C(K_0)$ is finite, $\mu > 0$, and that $x_0 \sim \mathcal{D}$ has norm bounded by L almost surely. Also, for both the policy gradient method and the natural policy gradient method, suppose Algorithm 1 is called with parameters:*

$$\begin{aligned} m, \ell, 1/r = & \mathit{poly}\left(C(K_0), \frac{1}{\mu}, \frac{1}{\sigma_{min}(Q)}, \|A\|, \|B\|, \|R\|, \right. \\ & \left. \frac{1}{\sigma_{min}(R)}, d, 1/\epsilon, L^2/\mu \right) . \end{aligned}$$

- *Natural policy gradient case: For a stepsize*

$$\eta = \frac{1}{\|R\| + \frac{\|B\|^2 C(K_0)}{\mu}}$$

and for

$$\begin{aligned} N \geq & \frac{\|\Sigma_{K^*}\|}{\mu}\left(\frac{\|R\|}{\sigma_{min}(R)} + \frac{\|B\|^2 C(K_0)}{\mu \sigma_{min}(R)}\right) \\ & \log \frac{2(C(K_0) - C(K^*))}{\varepsilon} \,, \end{aligned}$$

then, with high probability, i.e. with probability greater than $1 - \exp(-d)$, the natural policy gradient descent update (Equation 9) enjoys the following performance bound:

$$C(K_N) - C(K^*) \leq \varepsilon \,.$$

[2] The correlations in the state-action value estimates in REINFORCE are more challenging to analyze.

- *Gradient descent case: For an appropriate (constant) setting of the stepsize η,*

$$\eta = \text{poly}\left(\frac{\mu\sigma_{min}(Q)}{C(K_0)}, \frac{1}{\|A\|}, \frac{1}{\|B\|}, \frac{1}{\|R\|}, \sigma_{min}(R)\right)$$

and for

$$N \geq \frac{\|\Sigma_{K^*}\|}{\mu} \log \frac{C(K_0) - C(K^*)}{\varepsilon} \times \text{poly}\left(\frac{C(K_0)}{\mu\sigma_{min}(Q)}, \|A\|, \|B\|, \|R\|, \frac{1}{\sigma_{min}(R)}\right),$$

then, with high probability, gradient descent (Equation 8) enjoys the following performance bound:

$$C(K_N) - C(K^*) \leq \varepsilon\,.$$

This theorem gives the first polynomial time guarantee for policy gradient and natural policy gradient algorithms in the LQR problem.

Proof Sketch The model free results (Theorem 9) are proved in the following three steps:

1. Prove that when the roll out length ℓ is large enough, the cost function C and the covariance Σ are approximately equal to the corresponding quantities at infinite steps.
2. Show that with enough samples, Algorithm 1 can estimate both the gradient and covariance matrix within the desired accuracy.
3. Prove that both gradient descent and natural gradient descent can converge with a similar rate, even if the gradient/natural gradient estimates have some bounded perturbations.

The proofs are technical and are deferred to supplementary material. We have focused on proving polynomial relationships in our complexity bounds, and did not optimize for the best dependence on the relevant parameters.

5. Conclusions and Discussion

This work has provided provable guarantees that model-based gradient methods and model-free (sample based) policy gradient methods convergence to the globally optimal solution, with finite polynomial computational and sample complexities. Taken together, the results herein place these popular and practical policy gradient approaches on a firm theoretical footing, making them comparable to other principled approaches (e.g., subspace system identification methods and algebraic iterative approaches).

Finite $C(K_0)$ assumption, noisy case, and finite horizon case. These methods allow for extensions to the noisy case and the finite horizon case. This work also made the assumption that $C(K_0)$ is finite, which may not be easy to achieve in some infinite horizon problems. The simplest way to address this is to model the infinite horizon problem with a finite horizon one; the techniques developed in Section D.1 shows this is possible. This is an important direction for future work.

Open Problems.

- Variance reduction: This work only proved efficiency from a polynomial sample size perspective. An interesting future direction would be in how to rigorously combine variance reduction methods and model-based methods to further decrease the sample size.
- A sample based Gauss-Newton approach: This work showed how the Gauss-Newton algorithm improves over even the natural policy gradient method, in the exact case. A practically relevant question for the Gauss-Newton method would be how to both: a) construct a sample based estimator b) extend this scheme to deal with (non-linear) parametric policies.
- Robust control: In model based approaches, optimal control theory provides efficient procedures to deal with (bounded) model mis-specification. An important question is how to provably understand robustness in a model free setting.

Acknowledgments

Support from DARPA Lagrange Grant FA8650-18-2-7836 (to M. F., M. M., and S. K.) and from ONR award N00014-12-1-1002 (to M. F. and M. M.) is gratefully acknowledged. S. K. acknowledges funding from the Washington Research Foundation Fund for Innovation in Data-Intensive Discovery. S. K. thanks Emo Todorov, Aravind Rajeswaran, Kendall Lowrey, Sanjeev Arora, and Elad Hazan for helpful discussions. S. K. and M. F. also thank Ben Recht for helpful discussions. R. G. acknowledges funding from NSF CCF-1704656. We also thank Jingjing Bu from University of Washington for running the numerical simulations in Section E in supplementary material.

References

Abbasi-Yadkori, Y. and Szepesvári, C. Regret bounds for the adaptive control of linear quadratic systems. *Conference on Learning Theory*, 2011. ISSN 15337928.

Al Borno, M., de Lasa, M., and Hertzmann, A. Trajectory Optimization for Full-Body Movements with Complex Contacts. *IEEE Transactions on Visualization and Computer Graphics*, 2013.

Anderson, B. D. O. and Moore, J. B. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990. ISBN 0-13-638560-5.

Arora, S., Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. Towards provable control for unknown linear dynamical systems. 2018.

Bagnell, J. A. and Schneider, J. Covariant policy search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pp. 1019–1024, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1630659.1630805.

Balakrishnan, V. and Vandenberghe, L. Semidefinite programming duality and linear time-invariant systems. *IEEE Transactions on Automatic Control*, 48(1):30–41, 2003.

Bertsekas, D. P. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011. ISSN 16726340. doi: 10.1007/s11768-011-1005-3.

Bertsekas, D. P. *Dynamic Programming and Optimal Control*. Athena Scientific, 2017.

Bradtke, S., Ydstie, B., and a.G. Barto. Adaptive linear quadratic control using policy iteration. *Proceedings of American Control Conference*, 3(2):3475–3479, 1994. doi: 10.1109/ACC.1994.735224.

Camacho, E. and Bordons, C. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2004. ISBN 9781852336943.

Conn, A., Scheinberg, K., and Vicente, L. *Introduction to derivative-free optimization*, volume 8 of *MPS/SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2009.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. On the sample complexity of the linear quadratic regulator. *ArXiv e-prints*, 2017.

Deisenroth, M. P., Neumann, G., and Peters, J. A survey on policy search for robotics. *Found. Trends Robot*, 2(1–2):1–142, August 2013. ISSN 1935-8253. doi: 10.1561/2300000021. URL http://dx.doi.org/10.1561/2300000021.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *ICML*, 2016.

Evans, L. C. An introduction to mathematical optimal control theory. *University of California, Department of Mathematics*, pp. 126, 2005. ISSN 14712334. doi: 10.1186/1471-2334-10-32.

Fiechter, C.-N. PAC adaptive control of liner systems. In *Proceeding COLT '94 Proceedings of the seventh annual conference on Computational learning theory*, pp. 88–97, 1994.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 385–394. Society for Industrial and Applied Mathematics, 2005.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points - online stochastic gradient for tensor decomposition. *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, 2015.

Hazan, E., Singh, K., and Zhang, C. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 6705–6715, 2017.

Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. Spectral filtering for general linear dynamical systems. *arXiv preprint arXiv:1802.03981*, 2018.

Hewer, G. A. An iterative technique for the computation of steady state gains for the discrete optimal regulator. *IEEE Trans. Automat. Contr.*, pp. 382–384, 1971.

Howard, R. A. *Dynamic programming and Markov processes*. Wiley for The Massachusetts Institute of Technology, 1964.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1724–1732, 2017.

Kakade, S. A natural policy gradient. In *NIPS*, 2001.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *ICML*, 2002.

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College, London, 2003.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I*, pp. 795–811, 2016.

Kleinman, D. L. On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, 13 (1):114–115, 1968. ISSN 0018-9286. doi: 10.1109/TAC.1968.1098829.

Kumar, V., Todorov, E., and Levine, S. Optimal control with learned local models: Application to dexterous manipulation. In *ICRA*, 2016.

Lancaster, P. and Rodman, L. *Algebraic Riccati Equations*. Oxford science publications. Clarendon Press, 1995. ISBN 9780191591259.

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

Liao, L. Z. and Shoemaker, C. A. Convergence in unconstrained discrete-time differential dynamic programming. *IEEE Transactions on Automatic Control*, 36, 1991.

Ljung, L. (ed.). *System Identification (2Nd Ed.): Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999. ISBN 0-13-656695-2.

Mårtensson, K. Gradient methods for large-scale and distributed linear quadratic control. *Ph.D. Theses*, 2012.

Mårtensson, K. and Rantzer, A. Gradient methods for iterative distributed control synthesis. *Conference on Decision and Control*, pp. 1–6, 2009.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518, 2015.

Nesterov, Y. and Polyak, B. T. Cubic regularization of newton method and its global performance. *Math. Program.*, pp. 177–205, 2006.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, pp. 1–40, 2015. ISSN 1615-3383.

Ng, C.-K., Liao, L.-Z., and Li, D. A globally convergent and efficient method for unconstrained discrete-time optimal control. *J. Global Optimization*, 23:401–421, 2002.

Peters, J. and Schaal, S. Natural actor-critic. *Neurocomputing*, 71: 1180–1190, 2007.

Polak, E. An Historical Survey of Computational Methods in Optimal Control. *SIAM Review*, 15(2):pp. 553–584, 1973. ISSN 00361445. doi: 10.1137/1015071.

Polyak, B. T. Gradient methods for minimizing functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4): 864878, 1963.

Rajeswaran, A., Kumar, V., Gupta, A., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *CoRR*, abs/1709.10087, 2017a. URL http://arxiv.org/abs/1709.10087.

Rajeswaran, A., Lowrey, K., Todorov, E., and Kakade, S. Towards generalization and simplicity in continuous control. *CoRR*, abs/1703.02660, 2017b. URL http://arxiv.org/abs/1703.02660.

Rawlings, J. and Mayne, D. *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009. ISBN 9780975937709.

Salimans, T., Ho, J., Chen, X., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *ArXiv e-prints*, 2017.

Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. Trust region policy optimization. In *ICML*, 2015.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature*, 529, 2016.

Simchowitz, M., Mania, H., Tu, S., Jordan, M. I., and Recht, B. Learning without mixing: Towards a sharp analysis of linear system identification. In *COLT*, 2018.

Stewart, G. W. and Sun, J.-G. Matrix perturbation theory (computer science and scientific computing), 1990.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Tassa, Y., Erez, T., and Todorov, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. *International Conference on Intelligent Robots and Systems*, 2012.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. *ArXiv e-prints*, 2017.

Todorov, E. and Li, W. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *American Control Conference*, 2004.

Tropp, J. A. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

Tu, S. and Recht, B. Least-squares temporal difference learning for the linear quadratic regulator. In *ICML*, 2018.

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