Revisiting LQR Control from the Perspective of Receding-Horizon Policy Gradient

Xiangyuan Zhang   Tamer Başar

Abstract— We revisit in this paper the discrete-time linear quadratic regulator (LQR) problem from the perspective of receding-horizon policy gradient (RHPG), a newly developed model-free learning framework for control applications. We provide a fine-grained sample complexity analysis for RHPG to learn a control policy that is both stabilizing and ϵ-close to the optimal LQR solution, and our algorithm does not require knowing a stabilizing control policy for initialization. Combined with the recent application of RHPG in learning the Kalman filter, we demonstrate the general applicability of RHPG in linear control and estimation with streamlined analyses.

1. Introduction

Model-free policy gradient (PG) methods promise a universal end-to-end framework for controller designs. By utilizing input-output data of a black-box simulator, PG methods directly search the prescribed policy space until convergence, agnostic to system models, objective function, and design criteria/constraints. The general applicability of PG methods leads to countless empirical successes in continuous control, but the theoretical understanding of these PG methods is still in its early stage. Stemmed from the convergence theory of PG methods for general reinforcement learning tasks [1], [2], a recent thrust of research has specialized the analysis for the convergence and sample complexity of PG methods into several linear state-feedback control benchmarks [3]–[9]. However, incorporating imperfect-state measurements leads to a deficit of most, if not all, favorable landscape properties crucial for PG methods to converge (globally) in the state-feedback settings [9], [10]. Even worse, the control designer now faces several challenges unique to control applications: a) convergence might be toward a suboptimal stationary point without system-theoretic interpretations; b) provable stability and robustness guarantee could be lacking; c) convergence depends heavily on the initialization (e.g., the initial policy should be stabilizing), which would be challenging to hand-craft; and d) algorithm could be computationally inefficient. These bottlenecks blur the applicability of model-free PG methods in real-world control scenarios since the price for each of the above disadvantages could be unaffordable.

On the other hand, classic theories provide both elegant analytic solutions and efficient computational means (e.g., Riccati recursions) to a wide range of control problems [11]–[13]. They further reveal the intricate structure in various control settings and offer system-theoretical interpretations and guarantees to their characterized solutions. They suggest that, compared to viewing the dynamical system as a black box and studying the properties of PG methods from a (nonconvex) optimization perspective, it is better to incorporate those properties unique to decision and control into the design of learning algorithms.

In this work, we revisit the classical linear quadratic regulator (LQR) problem from the perspective of the newly-developed receding-horizon PG (RHPG) framework [14], which integrates Bellman's principle of optimality into the development of a model-free PG framework. First, RHPG approximates infinite-horizon LQR using a finite-horizon problem formulation and further decomposes the finite-horizon problem into a sequence of one-step sub-problems. Second, RHPG solves each sub-problem recursively using model-free PG methods. To accommodate the inevitable computational errors in solving these sub-problems, we establish the generalized principle of optimality that bounds the accumulated bias by controlling the inaccuracies in solving each sub-problem. We characterize the convergence and sample complexity of RHPG in §3-D and emphasize that the RHPG algorithm does not require knowing a stabilizing initial control policy *a priori*.

A. Literature Review

We mainly compare with [3], [4] and [8], [15], [16], where [3], [4] are the foundational work in applying policy optimization to LQR and [8], [15], [16] remove the assumption on an initial stabilizing point in LQR by adding a discount factor to the objective function as an extra parameter.

In contrast to [3], [4] that parametrizes LQR as a single nonconvex (constrained) optimization problem over the policy space, we provide a new parametrization and perspective to learning LQR control. The critical difference between the RHPG algorithm and [3], [4] is that RHPG incorporates existing theories into the design of model-free learning algorithms, which is more than just exploiting them for the convergence analysis. We view this work as an initial step toward the goal of "*designing control-specific learning algorithms with performance guarantees*" rather than "*analyzing existing learning algorithms for control*". This line of research is motivated by the observation that viewing the dynamical system as a black box and directly searching in the policy space leads to a deficit of most, if not all, favorable landscape properties beyond LQR [9], [10]. This implies that the excellent properties in LQR following the parametrization in [3], [4], such as coercivity

and gradient domination, are rare, problem-dependent, and hard to generalize. However, when the model information is known, existing theories have provided extremely efficient solutions (e.g., Riccati recursions) to these problems that seemed unsolvable in the PO paradigm. Hence, our rationale is that control settings inherently have more structures (that are problem-agnostic) than a black-box system. One should always exploit these structures in *developing learning-based algorithms with performance guarantees*. In our work and [14], [17], we have identified causality and the dynamic programming principle as the fundamental properties in all control settings and exploited them in the model-free learning paradigm. As demonstrated in our work and [14], [17], the RHPG framework efficiently solves LQR and the seemed-to-be-unsolvable Kalman filtering problem in the PO fashion, which serves as a fundamental benchmark in output feedback control. As a by-product, the RHPG framework also removes all assumptions inappropriate in model-free learning settings and is more consistent with existing theories in control and estimation. Our work and [14], [17] together lead to a promising path toward the theoretical foundation of model-free learning in partially observable settings and nonlinear control through the lens of RHPG.

In comparison with [8], [15], [16], we note that the γ-discounted LQR problems therein are equivalent to the standard non-discounted LQR with system matrices being $\sqrt{\gamma}A$ and $\sqrt{\gamma}B$. Then, for any $\gamma \in (0,1)$, the set of stabilizing policies with system matrices being $\sqrt{\gamma}A$ and $\sqrt{\gamma}B$ is strictly larger than that of the un-discounted case. Hence, when γ is sufficiently small, one can initialize the PG algorithm with an arbitrary control policy. This removes the requirement of knowing a stabilizing policy in advance, but it comes with the price of solving multiple LQRs instead of only one. Moreover, the criterion for increasing γ is more complex than our selection rule for N in RHPG. Besides removing the assumption on the initial stabilizing point, it is more important to determine if the results/insights can be further generalized/extended to more complicated control problems, where the critical difference between our work and [8], [15], [16] appears. In [8], [15], [16], the landscape of discounted LQR is identical to those in [3], [4] for un-discounted LQR, with essentially scaled versions of system parameters. Thus, the same difficulties reported in [9], [10] will appear when considering the output-feedback setting with an additional discount factor. In contrast, RHPG can be directly extended to solve output-feedback problems [17].

B. Notations

We use $\|X\|$, κ_X, and $\rho(X)$ to denote, respectively, the spectral norm, condition number, and the spectral radius of a square matrix X. If X is symmetric, we use $X > 0$ and $X \geq 0$ to denote that X is positive definite (pd) and positive semi-definite (psd), respectively. For a pd matrix W of appropriate dimensions, we define the W-induced norm of X as

$$\|X\|_W^2 := \sup_{z \neq 0} \frac{z^\top X^\top W X z}{z^\top W z}.$$

2. PRELIMINARIES

A. Infinite-Horizon LQR

Consider the discrete-time linear dynamical system[1]

$$x_{t+1} = Ax_t + Bu_t, \qquad (2.1)$$

where $x_t \in \mathbb{R}^n$ is the state; $u_t \in \mathbb{R}^m$ is the control input; $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are system matrices unknown to the control designer; and the initial state $x_0 \in \mathbb{R}^n$ is sampled from a zero-mean distribution $\mathcal{D}$ that satisfies $\mathrm{Cov}(x_0) = \Sigma_0 > 0$. The goal in the LQR problem is to obtain the optimal controller $u_t = \phi_t(x_t)$ that minimizes the cost

$$J_\infty := \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} \left(x_t^\top Q x_t + u_t^\top R u_t \right) \right], \qquad (2.2)$$

where $Q > 0$ and $R > 0$ are symmetric pd weightings chosen by the control designer. For the LQR problem as posed to admit a solution, we require (A, B) to be stabilizable. Note that here $Q > 0$ implies the observability of $(A, Q^{1/2})$. Then, the unique optimal LQR controller is linear state-feedback, i.e., $u_t^* = -K^* x_t$, and $K^* \in \mathbb{R}^{m \times n}$, which with a slight abuse of terminology we will call optimal control policy, can be computed by

$$K^* = (R + B^\top P^* B)^{-1} B^\top P^* A, \qquad (2.3)$$

where P^* is the unique positive definite (pd) solution to the algebraic Riccati equation (ARE)

$$P = Q + A^\top P A - A^\top P B (R + B^\top P B)^{-1} B^\top P A. \quad (2.4)$$

Moreover, the optimal control policy K^* is guaranteed to be stabilizing, i.e., $\rho(A - BK^*) < 1$. Therefore, we can parametrize LQR as an optimization problem over the policy space $\mathbb{R}^{m \times n}$, subject to the stability condition [3]:

$$\min_K \ J_\infty(K) = \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} \left(x_t^\top (Q + K^\top RK) x_t \right) \right] \quad (2.5)$$

$$\text{s.t.} \quad K \in \mathcal{K} := \{K \mid \rho(A - BK) < 1\}. \qquad (2.6)$$

Theoretical properties of model-free (zeroth-order) PG methods in solving (2.5) have been well understood [3], [4], [18]. In particular, the objective function (2.5), even though nonconvex, is coercive and (globally) gradient dominated [18]. Hence, if an initial control policy $K_0 \in \mathcal{K}$ is known *a priori*, then any descent direction of the objective value (e.g., vanilla PG) suffices to ensure that all the iterates will remain in the interior of $\mathcal{K}$ while quickly converging toward the unique stationary point. Removing the assumption on K_0 (that an initial stabilizing policy can readily be found) has remained an active research topic [8], [15], [16].

---

[1]For extensions to stochastic LQR with i.i.d. additive noises, as well as the setting with an arbitrary (deterministic) initial state, see §5.

B. Finite-Horizon LQR

The finite-N-horizon version of the LQR problem is also described by the system dynamics (2.1), but with the objective function summing up only up to time $t = N$. Similar to (2.5), we can parametrize the finite-horizon LQR problem as $\min_{\{K_t\}} J(\{K_t\})$, where

$$J(\{K_t\}) := \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{N-1} x_t^\top (Q + K_t^\top R K_t) x_t + x_N^\top Q_N x_N \right], \quad (2.7)$$

and Q_N is a symmetric psd terminal-state weighting to be chosen. The unique optimal control policy in the finite-horizon LQR is time-varying and can be computed by

$$K_t^* = (R + B^\top P_{t+1}^* B)^{-1} B^\top P_{t+1}^* A, \quad (2.8)$$

where P_t^*, for all $t \in \{0, \cdots, N-1\}$, are generated by the Riccati difference equation (RDE) starting with $P_N^* = Q_N$:

$$\begin{aligned} P_t^* = Q &+ A^\top P_{t+1}^* A \\ &- A^\top P_{t+1}^* B (R + B^\top P_{t+1}^* B)^{-1} B^\top P_{t+1}^* A. \end{aligned} \quad (2.9)$$

Theoretical properties of zeroth-order PG methods in addressing (2.7) have been studied in [6], [7]. Compared to the infinite-horizon setting (2.5), the finite-horizon LQR problem (2.7) is also a nonconvex and gradient-dominated problem, but it does not naturally require the stability condition (2.6).

3. RECEDING-HORIZON POLICY GRADIENT

A. LQR with Dynamic Programming

It is well known that the solution of the RDE (2.9) converges monotonically to the stabilizing solution of the ARE (2.4) exponentially [19]. It then readily follows that the sequence of time-varying LQR policies (2.8), denoted as $\{K_t\}_{t \in \{N-1, \cdots, 0\}}$, converges monotonically to the time-invariant LQR policy K^* as $N \to \infty$. Now, we formally present this non-asymptotic convergence result.

Theorem 3.1: Let $A_K^* := A - BK^*$, use $\|\cdot\|_*$ to denote the P^*-induced norm, and define

$$N_0 = \frac{1}{2} \cdot \frac{\log\left(\frac{\|Q_N - P^*\|_* \cdot \kappa_{P^*} \cdot \|A_K^*\| \cdot \|B\|}{\epsilon \cdot \lambda_{\min}(R)}\right)}{\log\left(\frac{1}{\|A_K^*\|_*}\right)} + 1. \quad (3.1)$$

where it holds that $\|A_K^*\|_* < 1$. Then, for all $N \geq N_0$, the control policy K_0^* computed by (2.8) satisfies $\|K_0^* - K^*\| \leq \epsilon$ for any $\epsilon > 0$.

We provide the proof of Theorem 3.1 in §A. Theorem 3.1 demonstrates that if selecting $N \sim \mathcal{O}(\log(\epsilon^{-1}))$, then solving the finite-horizon LQR will result in a policy K_0^* that is ϵ-close to K^*, for any $\epsilon > 0$. Furthermore, if one chooses a small enough ϵ such that an ϵ-ball centered at K^* lies entirely in $\mathcal{K}$, then this condition on ϵ constitutes a sufficient condition for K_0^* to be stabilizing, i.e., $K_0^* \in \mathcal{K}$.

Algorithm 1: Receding-Horizon Policy Gradient

Input: horizon N, max iterations $\{T_h\}$, smoothing radius $\{r_h\}$, stepsizes $\{\eta_h\}$

1 **for** $h = N-1, \cdots, 0$ **do**
2   Initialize $K_{h,0}$ arbitrarily (e.g., the convergent policy from the prev. iter. $K_{h+1,T_{h+1}}$ or 0);
3   **for** $i = 0, \cdots, T_h - 1$ **do**
4   // sample PG update via a zeroth-order oracle
5   Sample $K_{h,i}^+ = K_{h,i} + r_h U$ and $K_{h,i}^- = K_{h,i} - r_h U$, where U is uniformly drawn from the surface of a unit sphere, i.e., $\|U\|_F = 1$;
6   Sample $x_h \sim \mathcal{D}$ and simulate two trajectories with policies $K_{h,i}^+$ and $K_{h,i}^-$, respectively. Compute values $J_h(K_{h,i}^+)$ and $J_h(K_{h,i}^-)$;
7   Compute the estimated PG $\widetilde{\nabla} J_h(K_{h,i}) = \frac{mn}{2r_h}\big[J_h(K_{h,i}^+) - J_h(K_{h,i}^-)\big]U$
8   and update $K_{h,i+1} = K_{h,i} - \eta_h \cdot \widetilde{\nabla} J_h(K_{h,i})$;
9   **end**
10 **end**
11 Return K_{0,T_0};

B. Algorithm Design

We propose the RHPG algorithm (cf., Algorithm 1), which first selects N by Theorem 3.1, and then sequentially decomposes the finite-N-horizon LQR problem backward in time. In particular, for every iteration indexed by $h \in \{N-1, \cdots, 0\}$, the RHPG algorithm solves an LQR problem from $t = h$ to $t = N$, where we only optimize for the current policy K_h and fix all the policies $\{K_t\}$ for $t \in \{h+1, \cdots, N-1\}$ to be the convergent solutions generated from earlier iterations. Concretely, for every h, the RHPG algorithm solves the following *quadratic* program in K_h:

$$\min_{K_h} J_h(K_h) := \mathbb{E}_{x_h \sim \mathcal{D}} \bigg[\sum_{t=h+1}^{N-1} x_t^\top \big(Q + (K_t^*)^\top R K_t^*\big) x_t$$
$$+ x_h^\top \big(Q + K_h^\top R K_h\big) x_h + x_N^\top Q_N x_N \bigg]. \quad (3.2)$$

Due to the quadratic optimization landscape of (3.2) in K_h for every h, applying any PG method with an arbitrary finite initial point (e.g., zero) would lead to convergence to the globally optimal solution of (3.2).

C. Bias of Model-Free Receding-Horizon Control

The RHPG algorithm builds on Bellman's principle of optimality, which requires solving each iteration to the exact optimal solution. However, PG methods can only return an ϵ-accurate solution after a finite number of steps. To generalize Bellman's principle of optimality, we analyze how computational errors accumulate and propagate in the (backward) dynamic programming process. In the theorem below, we show that if one solves every iteration of the RHPG algorithm to the $\mathcal{O}(\epsilon)$-neighborhood of the unique

Fig. 1. We first show that the output policy $\widetilde{K}_0$ can be made ϵ-close to K^* in two steps. First, Theorem 3.1 proves that K_0^* is ϵ-close to K^* by selecting N accordingly. Then, Theorem 3.2 analyzes the backward propagation of the computational errors from solving each subproblem, denoted as $\delta_t := \widetilde{K}_t - \widetilde{K}_t^*$ for all t, where $\widetilde{K}_t^*$ represents the current optimal LQR policy after absorbing errors from all previous iterations. Then, we show that if one requires a small enough optimality gap ϵ between $\widetilde{K}_0$ and K^*, then the RHPG output $\widetilde{K}_0$ can automatically acquire a closed-loop stability certificate.

optimum, then the RHPG algorithm will output a policy that is ϵ-close to the infinite-horizon LQR policy K^*.

Theorem 3.2: Choose N according to Theorem 3.1 and assume that one can compute, for all $h \in \{N-1, \cdots, 0\}$ and some $\epsilon > 0$, a policy $\widetilde{K}_h$ that satisfies

$$\left\| \widetilde{K}_h - \widetilde{K}_h^* \right\| \sim \mathcal{O}(\epsilon)\mathcal{O}(\texttt{poly}(\text{system parameters})),$$

where $\widetilde{K}_h^*$ is the optimum of the LQR from h to N, after absorbing errors in all previous iterations of Algorithm 1. Then, the RHPG algorithm outputs a control policy $\widetilde{K}_0$ that satisfies $\left\| \widetilde{K}_0 - K^* \right\| \leq \epsilon$. Furthermore, if ϵ is sufficiently small such that $\epsilon < \frac{1 - \|A - BK^*\|_*}{\|B\|}$, then $\widetilde{K}_0$ is stabilizing.

We illustrate Theorem 3.2 in Figure 1 and defer its proof to §B. Additionally, we discuss the implications of Theorem 3.2 in the following remark.

Remark 3.3: Theorem 3.2 ensures that if each RHPG iteration satisfies a certain error tolerance level, then the RHPG output $\widetilde{K}_0$ will reach an ϵ-neighborhood of K^*. Notably, upon selecting a sufficiently small ϵ, such that the ϵ-ball around K^* lies entirely within $\mathcal{K}$, we can guarantee closed-loop stability of $\widetilde{K}_0$ solely based on its near-optimality. This approach differs significantly from existing PG for LQR literature (e.g., [3], [4]), which requires a stabilizing initial policy and focus on preserving the closed-loop stability during learning. Conversely, RHPG starts with an arbitrary initial point, potentially very far from the stabilizing region (see our numerical experiments in §4), yet still converges globally to K^*. The closed-loop stability certificate then comes for free, given near-optimality.

Now, it remains to establish the sample complexity for the convergence of (zeroth-order) PG methods in every iteration of the algorithm, which is done next.

D. PG Update and Sample Complexity

We analyze here the sample complexity of the zeroth-order PG update in solving each iteration of the RHPG algorithm. Specifically, the zeroth-order PG update is defined as

$$K_{h,i+1} = K_{h,i} - \eta_h \cdot \widetilde{\nabla} J_h(K_{h,i}) \qquad (3.3)$$

where $\eta_h > 0$ is the stepsize to be determined and $\widetilde{\nabla} J_h(K_{h,i})$ is the estimated PG sampled from a (two-point) zeroth-order oracle. We formally present the sample complexity result in the following proposition.

Proposition 3.4: For all $h \in \{0, \cdots, N-1\}$, choose a constant smoothing radius $r_h \sim \mathcal{O}(\epsilon)$ and a constant stepsize $\eta_h \sim \mathcal{O}(\epsilon^2)$. Then, the zeroth-order PG update (3.3) converges after $T_h \sim \mathcal{O}(\frac{1}{\epsilon^2} \log(\frac{1}{\delta\epsilon^2}))$ iterations in the sense that $\left\| K_{h,T_h} - \widetilde{K}_h^* \right\| \leq \epsilon$ with a probability of at least $1 - \delta$.

For completeness, we provide a supplementary proof of Proposition 3.4 in §C, which mostly follows existing results in the literature [4]. Combining Theorem 3.2 with Proposition 3.4, we conclude that if we spend $\widetilde{\mathcal{O}}(\epsilon^{-2} \log(\delta^{-1}))$ iterations in solving every subproblem to $\mathcal{O}(\epsilon)$-accuracy with a probability of $1 - \delta$, for all $h \in \{0, \cdots, N-1\}$, then the RHPG algorithm will output a $\widetilde{K}_0$ that satisfies $\|\widetilde{K}_0 - K^*\| \leq \epsilon$ with a probability of at least $1 - N\delta$. By (3.1), this implies that the total iteration complexity of RHPG is also $\widetilde{\mathcal{O}}(\epsilon^{-2} \log(\delta^{-1}))$ with the dependence on various system parameters being polynomial.

We discuss the tradeoffs in selecting N to balance minimizing finite-to-infinite error and minimizing errors through the backward propagation in §5-A. To compare our sample complexity bound with the sharpest result in the literature [4], our dependence on ϵ matches that of [4][2]. Our sample complexity and that of [4] have polynomial dependence on system parameters. However, it is not clear how to compare the polynomial dependencies between our bounds and those of [4], and these polynomial factors might affect the overall computational efficiency of both algorithms in a substantial way. We leave this comparison as an important future research topic.

4. NUMERICAL EXPERIMENTS

We verify our theories on a scalar linear system studied in [4], where $A = 5$, $B = 0.33$, $Q = R = 1$, and the

---

[2]Note that the $\widetilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity presented in [4] is for the convergence in objective value (e.g., $f(K) - f(K^*) \leq \epsilon$), and is equivalent to an $\widetilde{\mathcal{O}}(\epsilon^{-2}) \cdot \mathcal{O}(\texttt{poly}(\text{system parameters}))$ sample complexity for the convergence in policy (i.e., $\|K - K^*\| \leq \epsilon$).

Fig. 2. For six different values of ϵ: *Left*: policy error between the output and K^*. *Right*: the total number of calls to the (two-point) zeroth-order oracle.

unique optimal LQR policy is $K^* = 14.5482$. For the PG method in [3], [4] to converge in this simple setting, one has to initialize with a policy K_0 that satisfies $K_0 \in \underline{\mathcal{K}} := \{K \mid 12.12 < K \pm r < 18.18\}$, which is necessary to prevent the zeroth-order oracle with a smoothing radius of r from perturbing K_0 outside of the stabilizing region $\mathcal{K}$ during the first iteration of the PG update. In contrast, we initialize the PG updates in Algorithm 1 with a zero policy $K_h = 0$, set $Q_N = 3$, and choose $N = \texttt{ceil}(\log(\epsilon^{-1}))$ according to (3.1). Furthermore, we choose $r_h = \sqrt{\epsilon}$, select a constant stepsize in each iteration of the RHPG algorithm, and run the algorithm to solve the LQR problem under six different ϵ, namely $\epsilon \in \{10^{-3}, 3.16 \times 10^{-3}, 10^{-2}, 3.16 \times 10^{-2}, 10^{-1}, 3.16 \times 10^{-1}\}$. We apply the zeroth-order PG update in solving every subproblem to $\|\widetilde{K}_h - \widetilde{K}_h^*\| \le \epsilon$. As shown in Figure 2, the empirical observation of the iteration complexity of RHPG (right) for the convergence in policy (left) is around $\mathcal{O}(\epsilon^{-2})$ under varying ϵ, which corroborates our theoretical findings.

5. DISCUSSIONS

In this section, we provide two discussion paragraphs. The first paragraph discusses the tradeoffs in selecting the problem horizon N. The second paragraph covers extensions of our results to the stochastic LQR setting that incorporates a zero-mean, independent stochastic disturbance in the dynamics and, additionally, another setting where the initial state x_0 could be arbitrary.

A. Tradeoffs in Selecting the Problem Horizon N

There exist two potential usages of the RHPG algorithm in practice. In most real-world scenarios, the user wants to address the finite-horizon LQR problem with *time-varying* system parameters and does not necessarily care whether the resulting control policy is close to the infinite-horizon solution or not. Model-predictive or receding-horizon control is the most prevalent choice when detailed model information is available. The RHPG solution, on the other hand, extends receding-horizon control to the setting with an unknown model, and the user could specify a desired horizon N.

In more theory-oriented cases, the user wants to address the infinite-horizon LQR problem with time-invariant system parameters. Then the user should choose the problem horizon N carefully, balancing between i) reducing the finite-to-infinite horizon error by increasing N, and ii) potentially creating more computational error in the approximate dynamic programming when N is large. The RHPG framework performs these two tasks sequentially: first dealing with the finite-to-infinite horizon error and fixing an N, and then addressing the finite-N-horizon problem step-by-step in time. The main rationale comes from the different rates of N contributing to the two errors. In the first step (cf., Theorem 3.1), the convergence of the finite-horizon solution toward the infinite-horizon solution is exponential, meaning that it suffices to choose a problem horizon $N = \mathcal{O}(\log(\epsilon^{-1}))$ for an accurate approximation (the dependence to other system parameters are also logarithmic). After deciding on an N based on the desired ϵ, the user solves the finite-N-horizon problem iteratively, where the subproblem in each iteration is strongly convex and smooth. In the sample complexity analysis, we have considered the choice of N since one would require a higher solution accuracy in each iteration if N is large (cf., §B). Moreover, N also appears in the probability term of $1 - N\delta$ due to the utilization of Boole's inequality. In solving each sub-problem, the complexity in the failure probability δ is $\mathcal{O}(\log(\delta^{-1}))$. Replacing δ with δ/N yields the rate of $\mathcal{O}(\log(N/\delta)) = \mathcal{O}(\log(\delta^{-1})) + \mathcal{O}(\log\log(\epsilon^{-1}))$. Hence, with a fixed N chosen according to Theorem 3.1, the impact of N on the total complexity is an additional $\mathcal{O}(\log\log(\epsilon^{-1}))$ factor and is therefore minor. In summary, the user in the infinite-horizon setting should prioritize the choice of N toward reducing the finite-to-infinite horizon error according to Theorem 3.1 and then solve the finite-N-horizon problem.

B. Extensions to Arbitrary x_0 and Stochastic LQR

We discuss here extensions of the RHPG framework to i) the setting where $x_0 \in \mathbb{R}^n$ is an arbitrary (deterministic) vector that is unknown to the designer, and ii) stochastic LQR with $x_0 \sim \mathcal{D}$ and the system dynamics being

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad w_t \sim \mathcal{N}(0, W), \quad W > 0.$$

Note that in the stochastic LQR setting, the objective function (2.2) should be replaced with the time-average cost

$$J_\infty := \limsup_{N \to \infty} \frac{1}{N} \, \mathbb{E}_{x_0, w_t} \left[\sum_{t=0}^{N-1} \left(x_t^\top Q x_t + u_t^\top R u_t \right) \right].$$

In both settings and under the stabilizability and detectability assumptions, a unique stationary state-feedback control policy exists and is identical to (2.3), which also stabilizes the closed-loop system. The difference is that in setting i), the LQR problem is deterministic, which allows implementing RHPG with a two-point zeroth-order oracle (cf., Algorithm 1). In contrast, setting ii) involves additive noises in the system dynamics, which necessitates using a one-point zeroth-order oracle, and thus, the gradient sampling will be noisier.

We note that the procedure of RHPG is identical whether solving deterministic or stochastic problems and with either finite or infinite problem horizons. The only difference is in the choice of the inner-loop oracles (two-point v.s. one-point). In both settings, the receding-horizon parametrization, the convergence of the Riccati equations in Theorem 3.1, the analysis of dynamic programming in Theorem 3.2, and the quadratic optimization landscape in each subproblem are the same as in the presentation in the main body of this paper. Simply combining Theorems 3.1-3.2 with the corresponding inner-loop convergence result (for quadratic minimization and as a replacement of Proposition 3.3) yields the overall complexity in these two extended settings.

6. CONCLUSION

We have revisited discrete-time LQR from the perspective of RHPG and provided a fine-grained sample complexity analysis for RHPG to learn a control policy that is stabilizing and ϵ-close to the optimal LQR policy. Our result demonstrates the potential of RHPG in addressing various tasks in linear control and estimation with streamlined analyses.

REFERENCES

[1] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in Markov decision processes. In *Conference on Learning Theory*, pages 64–66, 2020.

[2] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020.

[3] Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476, 2018.

[4] Dhruv Malik, Ashwin Pananjady, Kush Bhatia, Koulik Khamaru, Peter L Bartlett, and Martin J Wainwright. Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. *Journal of Machine Learning Research*, 21(21):1–51, 2020.

[5] Yingying Li, Yujie Tang, Runyu Zhang, and Na Li. Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach. *arXiv preprint arXiv:1912.09135*, 2019.

[6] Ben M Hambly, Renyuan Xu, and Huining Yang. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *arXiv preprint arXiv:2011.10300*, 2020.

[7] Kaiqing Zhang, Xiangyuan Zhang, Bin Hu, and Tamer Başar. Derivative-free policy optimization for linear risk-sensitive and robust control design: Implicit regularization and sample complexity. *Advances in Neural Information Processing Systems*, pages 2949–2964, 2021.

[8] Juan C Perdomo, Jack Umenberger, and Max Simchowitz. Stabilizing dynamical systems via policy gradient methods. *Advances in Neural Information Processing Systems*, 34, 2021.

[9] Bin Hu, Kaiqing Zhang, Na Li, Mehran Mesbahi, Maryam Fazel, and Tamer Başar. Toward a theoretical foundation of policy optimization for learning control policies. *Annual Review of Control, Robotics, and Autonomous Systems*, 6:123–158, 2023.

[10] Yang Zheng, Yujie Tang, and Na Li. Analysis of the optimization landscape of linear quadratic Gaussian (LQG) control. *arXiv preprint arXiv:2102.04393*, 2021.

[11] Brian DO Anderson and John B Moore. *Optimal Filtering*. Prentice-Hall, 1979.

[12] Brian DO Anderson and John B Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc., 1990.

[13] Tamer Başar and Pierre Bernhard. *H-infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhäuser, Boston, 1995.

[14] Xiangyuan Zhang, Bin Hu, and Tamer Başar. Learning the Kalman filter with fine-grained sample complexity. In *American Control Conference*, pages 4549–4554, 2023.

[15] Andrew Lamperski. Computing stabilizing linear controllers via policy iteration. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1902–1907. IEEE, 2020.

[16] Feiran Zhao, Xingyun Fu, and Keyou You. Learning stabilizing controllers of linear systems via discount policy gradient. *arXiv preprint arXiv:2112.09294*, 2021.

[17] Xiangyuan Zhang, Saviz Mowlavi, Mouhacine Benosman, and Tamer Başar. Global convergence of receding-horizon policy search in learning estimator designs. *arXiv preprint arXiv:2309.04831*, 2023.

[18] Jingjing Bu, Afshin Mesbahi, Maryam Fazel, and Mehran Mesbahi. LQR through the lens of first order methods: Discrete-time case. *arXiv preprint arXiv:1907.08921*, 2019.

[19] Babak Hassibi, Ali H Sayed, and Thomas Kailath. *Indefinite-Quadratic Estimation and Control: A Unified Approach to H_2 and H_∞ Theories*. SIAM, 1999.

[20] Luca Furieri, Yang Zheng, and Maryam Kamgarpour. Learning the globally optimal distributed LQ regulator. In *Learning for Dynamics and Control*, pages 287–297, 2020.

[21] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

APPENDIX

A. Proof of Theorem 3.1

This proof is dual to the proof of Theorem 3.1 in [14]. We first identify one-to-one correspondences between system parameters in LQR and those in Kalman filtering [14]:

LQR:	A	B	Q	R	Q_N
	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
KF [14]:	$A^\top$	$C^\top$	W	V	X_0

We also identify direct correspondences between our P_t, P^*, K_t, and K^* and [14]'s Σ_{N-t}, Σ^*, L_{N-1-t}, and L^*, respectively. Then, letting

$$\widetilde{P}_t := P_t^* - P^*, \quad \widetilde{R} := R + B^\top P^* B,$$
$$\overline{A} := A - B\widetilde{R}^{-1}B^\top P^* A,$$

and following equations (A.2)-(A.3) of [14], we have

$$
\begin{aligned}
\widetilde{P}_t &= \overline{A}^\top \widetilde{P}_{t+1}\overline{A} - \overline{A}^\top \widetilde{P}_{t+1}B(\widetilde{R} + B^\top \widetilde{P}_{t+1}B)^{-1}B^\top \widetilde{P}_{t+1}\overline{A}\\
&= \overline{A}^\top \widetilde{P}_{t+1}^{1/2}\big[I + \widetilde{P}_{t+1}^{1/2}B\widetilde{R}^{-1}B^\top \widetilde{P}_{t+1}^{1/2}\big]^{-1}\widetilde{P}_{t+1}^{1/2}\overline{A}\\
&\leq \big[1 + \lambda_{\min}(\widetilde{P}_{t+1}^{1/2}B\widetilde{R}^{-1}B^\top \widetilde{P}_{t+1}^{1/2})\big]^{-1}\overline{A}^\top \widetilde{P}_{t+1}\overline{A}\\
&=: \mu_t \overline{A}^\top \widetilde{P}_{t+1}\overline{A}, \quad\quad\quad\quad\quad\quad\quad\quad\quad (A.1)
\end{aligned}
$$

where $\widetilde{P}_{t+1}^{1/2}$ denotes the unique positive semi-definite (psd) square root of the psd matrix $\widetilde{P}_{t+1}$, $0 < \mu_t \leq 1$ for all t, and $\overline{A}$ satisfies $\rho(\overline{A}) < 1$. We now use $\|\cdot\|_*$ to represent the P^*-induced matrix norm and invoke Theorem 14.4.1 of [19], where our $\widetilde{P}_t$, $\overline{A}^\top$ and P^* correspond to $P_i - P^*$, F_p and W in [19], respectively. By Theorem 14.4.1 of [19] and (A.1), we obtain $\|\overline{A}\|_* < 1$ and given that $\mu_t \leq 1$,

$$\|\widetilde{P}_t\|_* \leq \|\overline{A}\|_*^2 \cdot \|\widetilde{P}_{t+1}\|_*.$$

Therefore, the convergence is exponential such that $\|\widetilde{P}_t\|_* \leq \|\overline{A}\|_*^{2(N-t)} \cdot \|\widetilde{P}_N\|_*$. As a result, the convergence of $\widetilde{P}_t$ to 0 in spectral norm can be characterized as

$$\|\widetilde{P}_t\| \leq \kappa_{P^*} \cdot \|\widetilde{P}_t\|_* \leq \kappa_{P^*} \cdot \|\overline{A}\|_*^{2(N-t)} \cdot \|\widetilde{P}_N\|_*,$$

where we have used κ_X to denote the condition number of X. That is, to ensure $\|\widetilde{P}_1\| \leq \epsilon$, it suffices to require

$$N \geq \frac{1}{2} \cdot \frac{\log\left(\frac{\|\widetilde{P}_N\|_* \cdot \kappa_{P^*}}{\epsilon}\right)}{\log\left(\frac{1}{\|\overline{A}\|_*}\right)} + 1. \tag{A.2}$$

Lastly, we show that the (monotonic) convergence of K_t^* to K^* follows from the convergence of P_t^* to P^*. Similar to (A.5) of [14], this can be verified through:

$$\begin{aligned}
K_t^* - K^* &= (R + B^\top P_{t+1}^* B)^{-1} B^\top P_{t+1}^* A \\
&\quad - (R + B^\top P^* B)^{-1} B^\top P^* A \\
&= \left[(R + B^\top P_{t+1}^* B)^{-1} - (R + B^\top P^* B)^{-1}\right] B^\top P^* A \\
&\quad + (R + B^\top P_{t+1}^* B)^{-1} B^\top (P_{t+1}^* - P^*) A \\
&= (R + B^\top P_{t+1}^* B)^{-1} B^\top (P^* - P_{t+1}^*) B K^* \\
&\quad - (R + B^\top P_{t+1}^* B)^{-1} B^\top (P^* - P_{t+1}^*) A \\
&= (R + B^\top P_{t+1}^* B)^{-1} B^\top (P^* - P_{t+1}^*)(B K^* - A) \tag{A.3}
\end{aligned}$$

Hence, we have $\|K_t^* - K^*\| \leq \frac{\|\overline{A}\| \cdot \|B\|}{\lambda_{\min}(R)} \cdot \|P_{t+1}^* - P^*\|$ and

$$\|K_0^* - K^*\| \leq \frac{\|\overline{A}\| \cdot \|B\|}{\lambda_{\min}(R)} \cdot \|\widetilde{P}_1\|.$$

Substituting ϵ in (A.2) with $\frac{\epsilon \cdot \lambda_{\min}(R)}{\|\overline{A}\| \cdot \|B\|}$ completes the proof.

B. Proof of Theorem 3.2

This proof is dual to the proof of Theorem 3.3 in [14]. First, according to Theorem 3.1, we select

$$N = \frac{1}{2} \cdot \frac{\log\left(\frac{2\|Q_N - P^*\|_* \cdot \kappa_{P^*} \cdot \|A_K^*\| \cdot \|B\|}{\epsilon \cdot \lambda_{\min}(R)}\right)}{\log\left(\frac{1}{\|A_K^*\|_*}\right)} + 1, \tag{B.4}$$

where $A_K^* := A - BK^*$. This ensures that K_0^* is stabilizing and $\|K_0^* - K^*\| \leq \epsilon/2$. Then, it remains to show that the output $\widetilde{K}_0$ satisfies $\|\widetilde{K}_0 - K_0^*\| \leq \epsilon/2$.

Recall that the RDE (2.9) is a backward iteration starting with $P_N^* = Q_N \geq 0$, and can also be represented as:

$$P_t^* = A^\top P_{t+1}^* (A - BK_t^*) + Q \tag{B.5}$$
$$= (A - BK_t^*)^\top P_{t+1}^* (A - BK_t^*) + (K_t^*)^\top R K_t^* + Q. \tag{B.6}$$

Moreover, for any K_t, we introduce the Lyapunov equation:

$$P_t = (A - BK_t)^\top P_{t+1}(A - BK_t) + K_t^\top R K_t + Q. \tag{B.7}$$

Furthermore, for clarity of the proof, we define/recall:

K_t^*: Exact LQR policy at time t defined in (2.8)

$\widetilde{K}_t^*$: Optimal policy of the current cost-to-go function, absorbing errors in all prior steps

$\widetilde{K}_t$: An approximation of $\widetilde{K}_t^*$ obtained by the PG update (3.3)

$\delta_t := \widetilde{K}_t - \widetilde{K}_t^*$: Policy optimization error at time t

$\widetilde{P}_t^*$: Generated by (B.6) with $K_t^* = \widetilde{K}_t^*$ and $P_{t+1}^* = \widetilde{P}_{t+1}$.

We argue that $\|\widetilde{K}_0 - K_0^*\| \leq \epsilon/2$ can be achieved by carefully controlling δ_t for all t. At $t = 0$, it holds that

$$\|\widetilde{K}_0 - K_0^*\| \leq \|\widetilde{K}_0^* - K_0^*\| + \|\delta_0\|,$$

where substituting K_t^* and K^* in (A.3), respectively, with $\widetilde{K}_0^*$ and K_0^* leads to

$$\widetilde{K}_0^* - K_0^* = (R + B^\top \widetilde{P}_1 B)^{-1} B^\top (P_1^* - \widetilde{P}_1)(BK_0^* - A).$$

Hence, the error size $\|\widetilde{K}_0^* - K_0^*\|$ could be bounded by

$$\|\widetilde{K}_0^* - K_0^*\| \leq \frac{\|A - BK_0^*\| \cdot \|B\|}{\lambda_{\min}(R)} \cdot \|P_1^* - \widetilde{P}_1\|. \tag{B.8}$$

Define the helper constants

$$C_1 := \frac{\varphi \cdot \|B\|}{\lambda_{\min}(R)} > 0, \quad \varphi := \max_{t \in \{0, \cdots, N-1\}} \|A - BK_t^*\|.$$

Next, we require $\|\delta_0\| \leq \epsilon/4$ and $\|\widetilde{K}_0^* - K_0^*\| \leq \epsilon/4$ to fulfill $\|\widetilde{K}_0 - K_0^*\| \leq \epsilon/2$. We select a fixed scalar $a > 0$ that is independent of system parameters and ϵ, and additionally require $\|P_1^* - \widetilde{P}_1\| \leq a$ to upper-bound the pd solutions of (B.7). Then, by (B.8), in order to fulfill $\|\widetilde{K}_0^* - K_0^*\| \leq \epsilon/4$, it suffices to require

$$\|P_1^* - \widetilde{P}_1\| \leq \min\left\{a, \frac{\epsilon}{4C_1}\right\}. \tag{B.9}$$

Subsequently, by (B.7), we have

$$P_1^* - \widetilde{P}_1 = (P_1^* - \widetilde{P}_1^*) + (\widetilde{P}_1^* - \widetilde{P}_1). \tag{B.10}$$

The first difference term on the RHS of (B.10) is

$$\begin{aligned}
P_1^* - \widetilde{P}_1^* &= A^\top P_2^* (A - BK_1^*) - A^\top \widetilde{P}_2 (A - B\widetilde{K}_1^*) \\
&= A^\top (P_2^* - \widetilde{P}_2)(A - BK_1^*) + A^\top \widetilde{P}_2 B(\widetilde{K}_1^* - K_1^*). \tag{B.11} \\
&= A^\top (P_2^* - \widetilde{P}_2)(A - BK_1^*) \\
&\quad - A^\top \widetilde{P}_2 B(R + B^\top \widetilde{P}_2 B)^{-1} B^\top (P_2^* - \widetilde{P}_2)(A - BK_1^*) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \tag{B.12} \\
&= A^\top [I - \widetilde{P}_2 B(R + B^\top \widetilde{P}_2 B)^{-1} B^\top](P_2^* - \widetilde{P}_2)(A - BK_1^*) \\
&= A^\top (I + \widetilde{P}_2 B R^{-1} B^\top)^{-1}(P_2^* - \widetilde{P}_2)(A - BK_1^*), \tag{B.13}
\end{aligned}$$

where applying (B.8) in deriving (B.12) and (B.13) is due to the matrix inversion lemma. Moreover, the second term on the RHS of (B.10) is

$$\begin{aligned}
\widetilde{P}_1^* - \widetilde{P}_1 &= (A - B\widetilde{K}_1^*)^\top \widetilde{P}_2 (A - B\widetilde{K}_1^*) + (\widetilde{K}_1^*)^\top R \widetilde{K}_1^* \\
&\quad - (A - B\widetilde{K}_1)^\top \widetilde{P}_2 (A - B\widetilde{K}_1) - (\widetilde{K}_1)^\top R \widetilde{K}_1 \\
&= -(\widetilde{K}_1^*)^\top B^\top \widetilde{P}_2 A - A^\top \widetilde{P}_2 B \widetilde{K}_1^* + (\widetilde{K}_1^*)^\top (R + B^\top \widetilde{P}_2 B)\widetilde{K}_1^* \\
&\quad + \widetilde{K}_1^\top B^\top \widetilde{P}_2 A + A^\top \widetilde{P}_2 B \widetilde{K}_1 - \widetilde{K}_1^\top (R + B^\top \widetilde{P}_2 B)\widetilde{K}_1 \\
&= \left[(R + B^\top \widetilde{P}_2 B)^{-1} B^\top \widetilde{P}_2 A - \widetilde{K}_1^*\right]^\top (R + B^\top \widetilde{P}_2 B) \cdot \\
&\quad \left[(R + B^\top \widetilde{P}_2 B)^{-1} B^\top \widetilde{P}_2 A - \widetilde{K}_1^*\right] \\
&\quad - \left[(R + B^\top \widetilde{P}_2 B)^{-1} B^\top \widetilde{P}_2 A - \widetilde{K}_1\right]^\top (R + B^\top \widetilde{P}_2 B) \cdot \\
&\quad \left[(R + B^\top \widetilde{P}_2 B)^{-1} B^\top \widetilde{P}_2 A - \widetilde{K}_1\right] \tag{B.14} \\
&= -\delta_1^\top (R + B^\top \widetilde{P}_2 B)\delta_1, \tag{B.15}
\end{aligned}$$

where (B.14) follows from completion of squares. Thus, combining (B.10), (B.11), and (B.15) yields

$$\begin{aligned}
&\|P_1^* - \widetilde{P}_1\| \\
&\leq \|P_2^* - \widetilde{P}_2\| \cdot \varphi\|A\| \|(I + \widetilde{P}_2 B R^{-1} B^\top)^{-1}\| \\
&\quad + \|\delta_1\|^2 \|R + B^\top \widetilde{P}_2 B\| \\
&\leq \varphi\|A\| \cdot \|P_2^* - \widetilde{P}_2\| + \|\delta_1\|^2 \|R + B^\top \widetilde{P}_2 B\|, \tag{B.16}
\end{aligned}$$

where (B.16) is due to that $\widetilde{P}_2 BR^{-1}B^\top \geq 0$ and thus $\|(I + \widetilde{P}_2 BR^{-1}B^\top)^{-1}\| \leq 1$. Now, we require

$$\|P_2^* - \widetilde{P}_2\| \leq \min\left\{a, \frac{a}{C_2}, \frac{\epsilon}{4C_1C_2} \cdot\right\} \qquad (B.17)$$

$$\|\delta_1\| \leq \min\left\{\sqrt{\frac{a}{C_3}}, \frac{1}{2}\sqrt{\frac{\epsilon}{C_1C_3}}\right\}, \qquad (B.18)$$

where C_2 and C_3 are positive constants defined as[3]

$$C_2 := 2\varphi\|A\| > 0, \quad C_3 := 2\|R + B^\top(P_{\max} + aI)B\| > 0$$
$$P_{\max} := \max_{t\in\{0,\cdots,N-1\}}\{P_t^*\}.$$

Then, conditions (B.17) and (B.18) are sufficient for (B.9) (and thus for $\|\widetilde{K}_0 - K_0^*\| \leq \epsilon/2$) to hold. Subsequently, we can propagate the required accuracies in (B.17) and (B.18) forward in time. Specifically, we iteratively apply the arguments in (B.16) (i.e., by plugging quantities with subscript t into the LHS of (B.16) and plugging quantities with subscript $t+1$ into the RHS of (B.16)) to obtain the result that if at all $t \in \{2, \cdots, N-1\}$, we require

$$\|P_t^* - \widetilde{P}_t\| \leq \min\left\{a, \frac{a}{C_2^{t-1}}, \frac{\epsilon}{4C_1C_2^{t-1}}\right\} \qquad (B.19)$$

$$\|\delta_t\| \leq \min\left\{\sqrt{\frac{a}{C_3}}, \sqrt{\frac{a}{C_2^{t-2}C_3}}, \frac{1}{2}\sqrt{\frac{\epsilon}{C_1C_2^{t-2}C_3}}\right\},$$

then (B.17) holds true and therefore (B.9) is satisfied.

We now compute the required accuracy for δ_{N-1}. Note that $P_{N-1}^* = \widetilde{P}_{N-1}^*$ since no prior computational errors happened at $t = N$. By (B.16), the distance between P_{N-1}^* and $\widetilde{P}_{N-1}$ can be bounded as

$$\|P_{N-1}^* - \widetilde{P}_{N-1}\| = \|\widetilde{P}_{N-1}^* - \widetilde{P}_{N-1}\| \leq \|\delta_{N-1}\|^2 \cdot C_3.$$

To fulfill the requirement (B.19) for $t = N-1$, which is

$$\|P_{N-1}^* - \widetilde{P}_{N-1}\| \leq \min\left\{a, \frac{a}{C_2^{N-2}}, \frac{\epsilon}{4C_1C_2^{N-2}}\right\},$$

it suffices to let

$$\|\delta_{N-1}\| \leq \min\left\{\sqrt{\frac{a}{C_3}}, \sqrt{\frac{a}{C_2^{N-2}C_3}}, \frac{1}{2}\sqrt{\frac{\epsilon}{C_1C_2^{N-2}C_3}}\right\}. \quad (B.20)$$

Finally, we analyze the worst-case complexity of RHPG by computing, at the most stringent case, the required size of $\|\delta_t\|$. When $C_2 \leq 1$, the most stringent dependence of $\|\delta_t\|$ on ϵ happens at $t = 0$, which is of the order $\mathcal{O}(\epsilon)$, and the dependences on system parameters are $\mathcal{O}(1)$. We then analyze the case where $C_2 > 1$, where the requirement on $\|\delta_0\|$ is still $\mathcal{O}(\epsilon)$. Note that in this case, $\|\delta_{N-1}\| \leq \|\delta_t\|$ for all $t \in \{1, \cdots, N-1\}$ and by (B.20):

$$\|\delta_{N-1}\| \sim \mathcal{O}\left(\sqrt{\frac{\epsilon}{C_1C_2^{N-2}C_3}}\right). \qquad (B.21)$$

Since we require N to satisfy (B.4), the dependence of $\|\delta_{N-1}\|$ on ϵ in (B.21) becomes $\|\delta_{N-1}\| \sim \mathcal{O}(\epsilon^{\frac{3}{4}})$ with additional polynomial dependences on system parameters, but one can observe that the dependence on

[3]As the scalar $a > 0$ increases, the constant C_3 grows correspondingly.

ϵ is still milder than the requirement for $\|\delta_0\|$. Therefore, it suffices to require error bound for all t to be $\|\delta_t\| \sim \mathcal{O}(\epsilon)\mathcal{O}(\texttt{poly}(\text{system parameters}))$ to reach the ϵ-neighborhood of the infinite-horizon LQR policy. Lastly, for $\widetilde{K}_0$ to be stabilizing, it suffices to select a sufficiently small ϵ such that the ϵ-ball centered at the infinite-horizon LQR policy K^* lies entirely in the set of stabilizing policies. A crude bound that satisfies this requirement is

$$\epsilon < \frac{1 - \|A - BK^*\|_*}{\|B\|} \implies \|A - B\widetilde{K}_0\|_* < 1.$$

This completes the proof.

C. Proof of Proposition 3.4

Recall that for all h, the objective function J_h is L_h-smooth and α_h-strongly-convex. Define $\varsigma_h := \frac{\epsilon^2\alpha_h}{2}$ and $\varsigma := \min_h \varsigma_h > 0$. We argue that if with a probability of at least $1 - \delta$, it holds that

$$J_h(K_{h,T_h}) - J_h(\widetilde{K}_h^*) \leq \varsigma, \qquad (C.22)$$

then $\|K_{h,T_h} - \widetilde{K}_h^*\| \leq \epsilon$ also holds with a probability of at least $1-\delta$ and the proof of Proposition 3.4 is complete. This is due to the α_h-strong convexity and $\nabla J_h(\widetilde{K}_h^*) = 0$. Thus,

$$J_h(K_{h,T_h}) - J_h(\widetilde{K}_h^*)$$
$$\geq \nabla J_h(\widetilde{K}_h^*)^\top(K_{h,T_h} - \widetilde{K}_h^*) + \frac{\alpha_h}{2}\|K_{h,T_h} - \widetilde{K}_h^*\|_F^2$$
$$\implies \|K_{h,T_h} - \widetilde{K}_h^*\|_F^2 \leq \frac{2}{\alpha_h}\left[J_h(K_{h,T_h}) - J_h(\widetilde{K}_h^*)\right] \leq \epsilon.$$

As a result, we will focus on proving (C.22) with a high probability of at least $1-\delta$. First, define the cost difference $\Delta_t := J_h(K_{h,t}) - J_h(\widetilde{K}_h^*)$ and the stopping time $\tau := \min\{t \mid \Delta_t > 10\delta^{-1}\Delta_0\}$. Let $\mathbb{E}^t[\cdot]$ denote the expectation conditioned on all the randomness up to t. Then, we state the following helper lemma and defer its proof to §D.

Lemma 1.1: For all h, choose the parameters of Algorithm 1 according to

$$\eta_h \leq \frac{1}{2L_h}, \quad r_h \leq \min\left\{\frac{\alpha_h}{4L_h}\sqrt{\frac{\varsigma\delta}{10}}, \frac{1}{2L_h}\sqrt{\frac{\alpha_h\varsigma\delta}{5}}\right\}.$$

Then, for all t, it holds that

$$\mathbb{E}^t[\Delta_{t+1}] \leq \left(1 - \frac{\eta_h\alpha_h}{4}\right)\Delta_t + \frac{L_h\eta_h^2}{2}G_2 + \eta_h\alpha_h\frac{\varsigma\delta}{20},$$

where α_h and L_h are the strong convexity and smoothness constants of J_h, respectively, and G_2 is a uniform constant to be introduced shortly.

Following the proof of Theorem 8 in [4], [20], we first consider the case of $\tau > T_h$. In this case, we can bound $\mathbb{E}^t[\Delta_{t+1}]$ using Lemma 1.1 directly. When $\tau \leq T_h$, it implies that $\mathbb{E}^t[\Delta_{t+1}]1_{\tau > t} = 0$. We require $\eta_h \leq \frac{\varsigma\delta\alpha_h}{40L_hG_2}$ and show

$$\mathbb{E}^t[\Delta_{t+1}]1_{\tau > t+1} \leq \left(1 - \frac{\eta_h\alpha_h}{4}\right)^{t+1}\Delta_0$$
$$+ \left(\frac{L_h\eta_h^2}{2}G_2 + \eta_h\alpha_h\frac{\varsigma\delta}{20}\right)\sum_{i=0}^{t}\left(1 - \frac{\eta_h\alpha_h}{4}\right)^i$$
$$\leq \left(1 - \frac{\eta_h\alpha_h}{4}\right)^{t+1}\Delta_0 + \frac{\varsigma\delta}{4}$$

Setting $t+1 = T_h$, it suffices to let $T_h = \frac{4}{\eta_h \alpha_h} \log(\frac{4\Delta_0}{\delta \varsigma})$ to ensure that

$$\mathbb{E}[\Delta_{T_h} 1_{\tau > T_h}] \leq \left(1 - \frac{\eta_h \alpha_h}{4}\right)^{T_h} \Delta_0 + \frac{\varsigma \delta}{4} \leq \frac{\varsigma \delta}{2}.$$

Next, we prove that the event $\tau \leq T_h$ has a probability smaller than $\frac{\delta}{2}$. For all t, we define the stopping process as

$$Y_t := \Delta_{\min\{\tau, t\}} + (T_h - t)\left(\frac{L_h \eta_h^2}{2} G_2 + \eta_h \alpha_h \frac{\varsigma \delta}{20}\right),$$

By Eq. (20)-(21) of [4], Y_t is a super-martingale. Applying Doob's maximal inequality yields

$$P\left(\max_{t=1,\cdots,T_h} Y_t \geq \frac{10\Delta_0}{\delta}\right) \leq \frac{\delta \mathbb{E}[Y_0]}{10\Delta_0}$$
$$= \frac{\delta}{10\Delta_0}\left(\Delta_0 + T_h\left(\frac{L_h \eta_h^2}{2} G_2 + \eta_h \alpha_h \frac{\varsigma \delta}{20}\right)\right)$$
$$\leq \frac{\delta}{10\Delta_0}\left(\Delta_0 + \log\left(\frac{4\Delta_0}{\varsigma \delta}\right)\frac{\varsigma \delta}{20} + \log\left(\frac{4\delta_0}{\varsigma \delta}\right)\frac{\varsigma \delta}{5}\right)$$

Imposing the condition that $\varsigma \log(\frac{4\Delta_0}{\varsigma \delta}) \leq 16\delta^{-1}\Delta_0$, we can prove that $P\left(\max_{t=1,\cdots,T_h} Y_t \geq \frac{10\Delta_0}{\delta}\right) \leq \frac{\delta \mathbb{E}[Y_0]}{10\Delta_0} \leq \frac{\delta}{2}$. We can now conclude that $\mathbb{E}[\Delta_{T_h} 1_{\tau > T_h}] \leq \frac{\delta \varsigma}{2}$ and the event τ occurs after T_h with probability at least $1 - \frac{\delta}{2}$. As a result,

$$P(\Delta_{T_h} \geq \varsigma) \leq P(\Delta_{T_h} 1_{\tau > T_h} \geq \varsigma) + P(1_{\tau \leq T_h})$$
$$\leq \frac{\mathbb{E}[\Delta_{T_h} 1_{\tau > T_h}]}{\varsigma} + P(1_{\tau \leq T_h}) \leq \frac{\delta}{2} + \frac{\delta}{2} = \delta,$$

where we have used Markov's inequality. This verifies (C.22), and thus $\|K_{h,T_h} - \widetilde{K}_h^*\| \leq \epsilon$ is satisfied with a probability of at least $1 - \delta$. Lastly, we analyze the constant G_2 following Corollary 10 of [4] and [21], where

$$G_2 = \sup_{K_h \in \Phi_h} \mathbb{E}\left[\left\|\frac{mn}{2r_h}[J(K_h + r_hU) - J(K_h - r_hU)]U\right\|_F^2\right]$$
$$\Phi_h := \{K_h \mid J(K_h) - J(\widetilde{K}_h^*) \leq 10\delta^{-1}\Delta_0\}.$$

By Corollary 10 of [4], it holds almost surely that $G_2 \leq (mn)\lambda^2$, where $\lambda := \max_h \lambda_h$ and λ_h is the Lipschitz continuity constant of J_h taken over the compact domain Φ_h. In summary, for $\|K_{h,T_h} - \widetilde{K}_h^*\| \leq \epsilon$ to hold with a probability of at least $1 - \delta$, we need to choose the (constant) algorithmic parameters according to $\eta_h \sim \mathcal{O}(\epsilon^2)$, and $r_h \sim \mathcal{O}(\epsilon)$. Then, the iteration complexity for the convergence of the zeroth-order PG method is $T_h \sim \mathcal{O}(\frac{1}{\epsilon^2} \log(\frac{1}{\delta \epsilon^2}))$.

D. Proof of Lemma 1.1

The proof of this lemma mostly follows the steps in Section 4.1.1 of [4]. First, define the smoothed version of J_h as $J_h^{r_h}(K_h) := \mathbb{E}[J_h(K_h + r_hU)]$, where the expectation is taken over U that is uniformly drawn from the surface of a unit sphere. Then, we use $J(K_h; x_0)$ to denote an instantiation of the objective value $J_h(K_h)$ given x_0, and define the two-point zeroth-order estimate of $\nabla J_h^{r_h}$ as

$$g(K_h) := \frac{mn}{2r_h}\left[J(K_h + r_hU; x_0) - J(K_h - r_hU; x_0)\right]U,$$

where mn is the dimension of the policy space. Subsequently, we invoke the L_h-smoothness property to derive

$$\mathbb{E}^t[J_h(K_{h,t+1}) - J_h(K_{h,t})]$$
$$\leq \mathbb{E}^t\left[\langle \nabla J_h(K_{h,t}), K_{h,t+1} - K_{h,t}\rangle + \frac{L_h}{2}\|K_{h,t+1} - K_{h,t}\|_F^2\right]$$
$$= -\langle \eta_h \nabla J_h(K_{h,t}), \nabla J_h^{r_h}(K_{h,t})\rangle + \frac{L_h \eta_h^2}{2}\mathbb{E}^t\left[\|g(K_{h,t})\|_F^2\right]$$
$$= -\eta_h \|\nabla J_h(K_{h,t})\|_F^2 + \eta_h L_h r_h \|\nabla J_h(K_{h,t})\|_F$$
$$\quad + \frac{L_h \eta_h^2}{2}\mathbb{E}^t\left[\|g(K_{h,t})\|_F^2\right],$$

where the inequalities are due to Lemma 14 of [4]. Moreover,

$$\mathbb{E}^t\left[\|g(K_{h,t})\|_F^2\right] = \mathrm{Var}(g(K_{h,t})) + \|\nabla J_h^{r_h}(K_{h,t})\|_F^2$$
$$\leq \mathrm{Var}(g(K_{h,t})) + 2\|\nabla J_h(K_{h,t})\|_F^2$$
$$\quad + 2\|\nabla J_h^{r_h}(K_{h,t}) - \nabla J_h(K_{h,t})\|_F^2$$
$$\leq G_2 + 2\|\nabla J_h(K_{h,t})\|_F^2 + 2L_h^2 r_h^2.$$

Again by the L_h-smoothness property, we have

$$J_h(K_{h,t} - \eta_h \nabla J_h(K_{h,t}))$$
$$\leq J_h(K_{h,t}) - (\eta_h - \frac{\eta_h^2 L_h}{2})\|\nabla J_h(K_{h,t})\|_F^2$$
$$\Longrightarrow (\eta_h - \frac{\eta_h^2 L_h}{2})\|\nabla J_h(K_{h,t})\|_F^2$$
$$\leq J_h(K_{h,t}) - J_h(K_{h,t} - \eta_h \nabla J_h(K_{h,t}))$$
$$\leq J_h(K_{h,t}) - J_h(\widetilde{K}_h^*) = \Delta_t.$$

Then, letting $\eta_h \in (0, \frac{1}{2L_h}]$, we can derive

$$\mathbb{E}^t[\Delta_{t+1} - \Delta_t]$$
$$\leq -\eta_h \|\nabla J_h(K_{h,t})\|_F^2 + 2\eta_h L_h r_h \Delta_t^{1/2} + \frac{L_h \eta_h^2}{2} G_2$$
$$\quad + L_h \eta_h^2 \|\nabla J_h(K_{h,t})\|_F^2 + \eta_h^2 L_h^3 r_h^2$$
$$\leq -\frac{\eta_h \alpha_h}{2}\Delta_t + \frac{\eta_h \alpha_h}{4}\Delta_t + \frac{4\eta_h L_h^2 r_h^2}{\alpha_h} + \frac{L_h \eta_h^2}{2} G_2 + \eta_h^2 L_h^3 r_h^2,$$

where the second inequality is due to that the α_h strong-convexity implies the α_h gradient domination property, the choice of stepsize $\eta_h \leq \frac{1}{2L_h}$, and $2ab \leq a^2 + b^2$ for any a, b. Recall the choices of algorithmic parameters as follows:

$$\eta_h \leq \frac{1}{2L_h}, \quad r_h \leq \min\left\{\frac{\alpha_h}{4L_h}\sqrt{\frac{\varsigma \delta}{10}}, \frac{1}{2L_h}\sqrt{\frac{\alpha_h \varsigma \delta}{5}}\right\}.$$

Then, using the bounds on algorithmic parameters and rearranging terms lead to

$$\mathbb{E}^t[\Delta_{t+1}] \leq \left(1 - \frac{\eta_h \alpha_h}{4}\right)\Delta_t + \frac{L_h \eta_h^2}{2} G_2 + \eta_h \alpha_h \frac{\varsigma \delta}{20},$$

which completes the proof.