Sample Complexity of Model-Free Policy Iteration for the Linear Quadratic Regulator

Vijayanand Digge and Gianluca Bianchin

***Abstract*— This paper provides a simple and explicit finite-time analysis of the sample complexity of the policy iteration (PI) algorithm for the linear quadratic regulator (LQR) problem in discrete-time linear time-invariant systems. In particular, we study a least-squares variant of PI and characterize its data efficiency when the data are collected from a single trajectory and reused across iterations.**

I. INTRODUCTION

At the core of the optimal control problem lies dynamic programming (DP), which provides a general and powerful framework for determining optimal policies. In recent years, reinforcement learning (RL) has emerged as a complementary paradigm for control systems, enabling the learning of optimal control policies directly from data or experience rather than relying on explicit models of system dynamics [1].

In the context of RL, least-squares temporal difference (LSTD) learning has been extensively studied, with strong asymptotic convergence guarantees. However, more recent research has shifted toward finite-time analysis, providing explicit bounds on convergence rates and sample efficiency. Authors in [2] have provided rigorous theoretical bounds, but often involve intricate derivations and assumptions. In this paper, we revisit this problem in a simplified setting, focusing on a clear and explicit finite-time analysis of a least-squares policy iteration (LSPI) scheme.

Notation: We let $\mathrm{svec}(M) \in \mathbb{R}^{n(n+1)/2}$ denote the vectorization of the upper triangular part of a symmetric matrix $M \in \mathbb{S}^n$, where off-diagonal entries are scaled by $\sqrt{2}$ so that $\|M\|_F^2 = \langle \mathrm{svec}(M), \mathrm{svec}(M) \rangle$. The notation $\mathrm{smat}(\cdot)$ denote the inverse operation of $\mathrm{svec}(\cdot)$. The symbol $\otimes$ denotes the Kronecker product.

II. PROBLEM FORMULATION

Consider the linear discrete-time system

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

where, $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^m$ is the input to the system. The state feedback controller is given by

$$u(k) = -Kx(k). \tag{2}$$

Consider the one-step cost defined by:

$$c(x(k), u(k)) = x(k)^\top Q x(k) + u(k)^\top R u(k)$$

where $Q, R \succ 0$. We consider LQR control problem described by the minimization of the infinite-horizon cost:

$$J(x(k), u(k)) = \sum_{k=0}^{\infty} c(x(k), u(k)). \tag{3}$$

Our goal is to determine the control sequence $\{u(k)\}_{k=0}^{k=\infty}$ such that (3) is minimized. As is well known [3], the LQR problem is described by the solution $u(k) = -K^* x(k)$, where

$$K^* = (R + B^\top P B)^{-1} B^\top P A, \tag{4}$$

and $P \succ 0$ is the solution of the discrete-time algebraic Riccati equation:

$$Q + A^\top P A - P - A^\top P B (R + B^\top P B)^{-1} B^\top P A = 0.$$

The objective of this work is to study model-free algorithms for the LQR problem by analyzing the policy iteration (PI) algorithm described in the following section. In particular, we establish lower bounds on the number of policy improvement steps required to achieve a prescribed accuracy with respect to the optimal feedback gain in (4).

III. LEAST SQUARES POLICY ITERATION (LSPI)

Given a candidate control policy $u(k) = -Kx(k)$, interpreted as an approximation of the optimal policy, its quality can be assessed through the so-called *state–action value function*, or Q-function, associated with K. The Q-function is defined as

$$Q^K(x(k), u(k)) := c(x(k), u(k)) + Q^K(x(k), -Kx(k)). \tag{5}$$

Starting from the candidate policy K, an improved policy can then be obtained by minimizing the Q-function:

$$u(k) = \arg\min_u Q^K(x(k), u(k)), \tag{6}$$

which results in a better policy [4]. Policy iteration iteratively applies policy evaluation and policy improvement to determine the optimal policy. This is accomplished by parameterizing the Q-function using a linear architecture as:

$$Q^\theta(x(k), u(k)) := \theta^\top \psi(x(k), u(k)), \tag{7}$$

where $\theta \in \mathbb{R}^d$ is a parameter vector that is shared across states, and $\psi(x(k), u(k)) \in \mathbb{R}^d$ are feature vectors, possibly a quadratic basis functions over elements of $x(k)$ and $u(k)$, i.e., $z(k) \otimes z(k)$ where $z(k) = \begin{bmatrix} x(k)^\top & u(k)^\top \end{bmatrix}$.

A. Minimization of Sum of Mean Square Error

Define the *temporal difference (TD) errors* as:

$$\begin{aligned}\varepsilon_k := &-Q^\theta(x(k), u(k)) + c(x(k), u(k)) \\ &+ Q^\theta(x(k+1), -Kx(k+1)).\end{aligned}$$

For (7) to be a good approximation of (5), the parameter θ should be chosen so that the sum of the squared TD errors are minimized:

$$\theta^* = \arg\min_\theta \frac{1}{T} \|\Gamma - \Phi_K^\top \theta\|^2. \tag{8}$$

Here, the matrices $\Gamma \in \mathbb{R}^{T \times 1}$ and $\Phi_K \in \mathbb{R}^{d \times T}$ are constructed from the data sequence collected over T time steps:

$$\begin{aligned}\Gamma &= \{\, c(x(k), u(k)) \,\}_{k=1}^T, \\ \Phi_K &= \{\, \psi(x(k), u(k)) - \psi(x(k+1), -Kx(k+1)) \,\}_{k=1}^T,\end{aligned}$$

The authors are with ICTEAM Institute and the Department of Mathematical Engineering at UCLouvain, Belgium. {vijayanand.digge, gianluca.bianchin}@uclouvain.be

where, the notation $\{a(k)\}_{k=1}^{T}$ denotes the stacked vector $[a(1)^\top, \ldots, a(T)^\top]^\top$.

The optimization (8) admits the closed-form solution

$$\theta^* = (\Phi_K \Phi_K^\top)^{-1} \Phi_K \Gamma. \tag{9}$$

Define $\Theta^* = \mathrm{smat}(\theta^*)$. Applying the policy-improvement step in (6) yields

$$u(k) = -G(\Theta^*)\, x(k), \tag{10}$$

where, for a matrix Θ partitioned as

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{12}^\top & \Theta_{22} \end{bmatrix},$$

the mapping $G(\Theta)$ is given by $G(\Theta) = -\Theta_{22}^{-1}\Theta_{12}^\top$.

Remark 3.1: For the uniqueness and existence of the solution in the estimation of θ, the matrix $(\Phi_K \Phi_K^\top)$ in (9) must be full rank. Consequently, the number of samples T must be at least equal to the number of feature vectors. This gives:

$$T \geq d = \frac{(n+m)(n+m+1)}{2}.$$

When this condition holds, the invertibility of $(\Phi_K \Phi_K^\top)$ can be guaranteed by introducing a probing noise term η_k in the data generation operation; see line 1 of Algorithm 1. □

We describe the LSPI in Algorithm 1. Here, all the data $\mathcal{D}$ is collected up front using initial stabilizing gain K_0 and is reused in every iteration of LSPI.

Algorithm 1 LSPI for LQR

Input: K_0: initial stabilizing controller,
N: number of policy iterations,
T: length of rollout,
σ_η^2: exploration variance,

1: Collect $\mathcal{D} = \{(x_k, u_k, x_{k+1})\}_{k=1}^{T}$, where $u_k = K_0 x_k + \eta_k, \quad \eta_k \sim \mathcal{N}(0, \sigma_\eta^2 I)$.
2: **for** $t = 0, \ldots, N-1$ **do**
3: $\quad \Theta_t = \mathrm{smat}\big((\Phi_{K_t} \Phi_{K_t}^\top)^{-1} \Phi_{K_t} \Gamma\big)$
4: $\quad K_{t+1} = G(\Theta_t)$.
5: **end for**
6: **return** K_N.

B. Overall Sample complexity

The global exponential convergence of the policy iteration presented in [5] is presented in the following Lemma.

Lemma 1: [5] Given stabilizing gain K_0 and corresponding Θ_0, the state-value function matrix corresponding to optimal gain K^* is Θ^*, we have

$$\|\Theta_{k+1} - \Theta^*\|_2 \leq \frac{\lambda_{\max}(\Theta_\varepsilon^*)}{\lambda_{\min}(\Theta_\varepsilon^*)} \left(\rho(A(K^*)) + \varepsilon\right)^{2k} \|\Theta_0 - \Theta^*\|_2$$

for any $\varepsilon > 0$ such that $\rho(A(K^*)) + \varepsilon < 1$, where Θ_ε^* satisfy the following Lyapunov inequality:

$$A(K^*)^T \Theta_\varepsilon A(K^*) \preceq \left(\rho(A(K^*)) + \varepsilon\right)^2 \Theta_\varepsilon^* \text{ and } \lambda_{\max}(\Theta_\varepsilon^*) < 1,$$

where $A(K^*) = \begin{bmatrix} A & B \\ -K^*A & -K^*B \end{bmatrix}$.

Next, we simplify the convergence rate, these simplification introduces conservatism into the bounds. We have $\Theta_\varepsilon^* \succeq \mu I$ for all $k \geq 0$, where $\mu = \min\{\lambda_{\min}(Q), \lambda_{\min}(R)\}$, from definition of Q-function (5). Then we have

$$\|\Theta_{k+1} - \Theta^*\|_2 \leq C^k \|\Theta_0 - \Theta^*\|_2 \tag{11}$$

where $C = \frac{1}{\mu}\left(\rho(A(K^*)) + \varepsilon\right)^2$. Now we derive the overall sample complexity, that is number of policy iterations

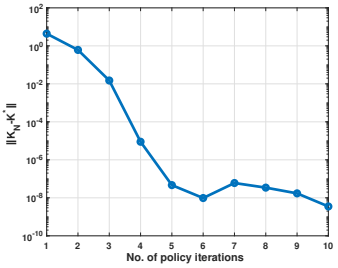

Fig. 1: Convergence of the estimated policy K_N toward the optimal LQR gain K^*, illustrated by the evolution of the error norm $\|K_N - K^*\|$ over policy iteration steps N.

required to achieve a specified level of performance.

Theorem 1: Given $\delta > 0$, Θ_0 corresponds to initial stabilizing gain K_0, and define

$$N_0 = 1 + \frac{\log(\frac{\|\Theta_0 - \Theta^*\|}{\delta})}{\|\log C\|}, \tag{12}$$

where C is given as in (11) . Then, $N > N_0$ policy improvement iteration are sufficient to reach suboptimal controller K_N such that $\|\Theta_N - \Theta^*\| \leq \delta$.

Theorem 1 states that N policy iterations are sufficient to obtain controller gain in δ neighborhood of optimal gain K^*.

IV. Numerical Simulation

We consider the discretized inverted pendulum system described in [6]. The weight matrices are chosen as $Q = I_n$ and $R = 1$. We initialize the algorithm with stabilizing gain K_0, and corresponding Θ_0. Based on this setup, the minimum number of samples required for each policy improvement stage is $T = 6$, The sample data are generated using an exploration noise term $\mathcal{N}(0, I)$ with T-samples. A total of $N_0 = 24$ policy improvement steps are sufficient to guarantee a suboptimal policy accuracy of $\delta = 0.01$. Figure 1 illustrates the convergence behavior of the policy across successive iterations. As observed, the policy quickly converges to the optimal LQR gain within a few iterations.

V. CONCLUSION

This paper presented a finite-time analysis of the policy iteration algorithm implemented via least-squares methods for the LQR problem. We characterized the data efficiency of this approach for discrete-time linear systems and derived explicit finite-time lower bounds when data are collected from a single trajectory and reused across iterations. Future research may extend this analysis to gradient-based methods, investigate the trade-off between gradient and policy improvement steps, and explore computational complexity bounds.

References

[1] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 4. Athena scientific, 2012.

[2] Karl Krauth, Stephen Tu, and Benjamin Recht. "Finite-time analysis of approximate policy iteration for the linear quadratic regulator". In: *Advances in Neural Information Processing Systems* 32 (2019).

[3] Tongwen Chen and Bruce A Francis. *Optimal sampled-data control systems*. Springer Science & Business Media, 2012.

[4] Michail G Lagoudakis and Ronald Parr. "Least-squares policy iteration". In: *Journal of machine learning research* 4.Dec (2003), pp. 1107–1149.

[5] Donghwan Lee. "Convergence of dynamic programming on the semidefinite cone for discrete-time infinite-horizon LQR". In: *IEEE Transactions on Automatic Control* 67.10 (2022).

[6] Andong Liu et al. "New results on stabilization of networked control systems with packet disordering". In: *Automatica* 52 (2015), pp. 255–259.