Sample-Efficient Model-Free Policy Gradient Methods for Stochastic LQR via Robust Linear Regression ⋆

Bowen Song * **Sebastien Gros** ** **Andrea Iannelli** *

* *Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany (e-mail: bowen.song,andrea.iannelli@ist.uni-stuttgart.de)*
** *Department of Engineering Cybernetics, Norwegian University of Science and Technology 7491 Trondheim, Norway (e-mail: sebastien.gros@ntnu.no)*

Abstract: Policy gradient algorithms are widely used in reinforcement learning and belong to the class of approximate dynamic programming methods. This paper studies two key policy gradient algorithms, the Natural Policy Gradient and the Gauss–Newton Method, for solving the linear quadratic regulator problem for unknown systems using stochastic data. The main challenge is the inconsistency of estimating random quantities in the policy gradient update due to the resulting errors-in-variables setting. This issue is addressed by proposing a robust primal–dual estimation procedure. Using this improved policy gradient update estimation scheme, this paper delivers a consistent estimator with a convergence rate of order $\mathcal{O}(\epsilon^{-1})$. Theoretical results are further supported by numerical experiments.

Keywords: Policy Gradient Methods, Primal-dual Optimization, Stochastic Systems, Linear Quadratic Regulators

1. INTRODUCTION

Policy gradient methods (PGMs) are fundamental tools in reinforcement learning, as they aim to optimize a parameterized policy with respect to the performance objective. Understanding their convergence properties is important, as it significantly impacts the reliability of deploying PGMs for real-world applications.

The linear quadratic regulator (LQR) problem serves as a canonical benchmark for studying the convergence behavior of PGMs in continuous state and action spaces [Hambly et al. (2021); Ju et al. (2025)]. In the seminal work by Fazel et al. (2018), three representative PGMs were analyzed for solving the LQR problem: policy gradient descent (PGD), natural policy gradient (NPG), and the Gauss–Newton method (GNM). They all require at least the knowledge of the cost gradient, which explicitly depends on the system dynamics. This represents an issue when an exact expression for the system's model is unavailable, which is often the case in complex applications. In this case, one can proceed by estimating the system matrices from data, as in Zhao et al. (2025); Song and Iannelli (2025b). In parallel, there are direct approaches that perform policy optimization without explicitly identifying the system dynamics. Following Fazel et al. (2018), which used a zeroth-order method to estimate gradients from noise-free data for implementing PGD/NPG method, several subsequent studies extended the PGD framework to a stochastic noise setting, such as Hambly et al. (2021). In these analyses, the sample complexity required to obtain an ϵ-optimal policy was initially shown to be $\mathcal{O}(\epsilon^{-4})$. By using stochastic approximation, Malik et al. (2019) reduced the sample complexity of PGD to $\mathcal{O}(\epsilon^{-2})$, and it was further improved to $\mathcal{O}(\epsilon^{-1})$ in Moghaddam et al. (2025).

Beyond the two categories discussed above, there exists another class of methods that directly estimate the specific matrix blocks required to construct a policy gradient update. This idea was first introduced in Tu and Recht (2018). The work Yang et al. (2023) applied this approach to deterministic linear systems using GNM. Later, Yaghmaie et al. (2023) extended it in a stochastic setting without giving convergence guarantees. Recently, Zhou and Lu (2023) and Ju et al. (2025) applied the NPG method to stochastic systems and established convergence guarantees with a sample complexity $\mathcal{O}(\epsilon^{-1}(\ln \epsilon^{-1})^2)$ using ergodic data, and $\mathcal{O}(\epsilon^{-1}(\ln \epsilon^{-1})^7)$ using online data, respectively.

In this work, we apply the natural policy gradient and Gauss–Newton method to solve the linear quadratic regulator problem for an unknown linear system subject to stochastic noise. To construct the quantities required by NPG and GNM updates, we employ a primal–dual estimation scheme that provides consistent estimates from noisy data. This scheme can be viewed as a general framework for analyzing linear regression problems with errors-in-variables. To further accelerate estimation, we introduce a multi-epoch refinement procedure, which improves the statistical rate and achieves a sample complexity of $\mathcal{O}(\epsilon^{-1})$. Once the matrix estimates are obtained, we perform NPG and GNM updates and establish global convergence guar-

⋆ Bowen Song acknowledges the support of the International Max Planck Research School for Intelligent Systems (IMPRS-IS).

antees for both algorithms. Compared with Zhou and Lu (2023); Ju et al. (2025), which apply only the NPG method and require collecting new data at every policy update, our method reuses a single dataset across all iterations, which is key to achieving the better sample complexity. To the best of the authors' knowledge, this is the lowest sample complexity achieved so far for applying NPG and GNM to the LQR problem. Unless otherwise stated, all the technical results presented here are novel and their proofs, for space limitations, are deferred to Song et al. (2025).

Notations

We denote $A \succeq 0$ and $A \succ 0$ as positive semidefinite and positive definite symmetric matrices, respectively. The Kronecker product is represented as $\otimes$, $vec(A) = [a_1^\top, a_2^\top, ..., a_n^\top]^\top$ stacks the columns of matrix A into a vector, $vecv(v) = [v_1^2, v_1v_2, ..., v_1v_n, v_2^2, ..., v_2v_n, ..., v_n^2]^\top$ rearranges the entries of vector v in this pattern, $vecs(P) = [p_{11}, 2p_{12}, ..., 2p_{1n}, p_{22}, ..., 2p_{2n}, ..., p_{nn}]^\top$ stacks the upper-triangular part of matrix $P \succ 0$. For matrices, $\|\cdot\|_F$, $\|\cdot\|$ denote respectively their Frobenius norm, induced 2-norm. I_n is the identity matrix with n rows. The symbol $\lambda_i(A)$ denotes the i-th smallest eigenvalue of the square matrix A. The symbol $\lceil x \rceil$ denotes the ceiling function returning the smallest integer greater or equal than $x \in \mathbb{R}$.

2. PROBLEM SETTING AND PRELIMINARIES

In this work, we consider the following averaged infinite-horizon LQR problem, where the system is subject to additive stochastic noise:

$$\begin{aligned} \min_{u_t} \lim_{T\to+\infty} \frac{1}{T} \mathop{\mathbb{E}}_{x_0,w_t} \sum_{t=0}^{T-1} \left(x_t^\top Q x_t + u_t^\top R u_t\right), \\ \text{s.t. } x_{t+1} = Ax_t + Bu_t + w_t, \\ x_0 \sim \mathcal{N}(0, \Sigma_0),\ w_t \sim \mathcal{N}(0, \Sigma_w), \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ denote the unknown but stabilizable system matrices. The covariance matrices satisfy $\Sigma_0, \Sigma_w \succ 0$ and $Q, R \succ 0$ are the weight matrices. We define the set of stabilizing feedback gains as:

$$\mathcal{S} := \left\{K \in \mathbb{R}^{n_u \times n_x} \,\middle|\, A_K := A + BK \text{ is Schur stable}\right\}.$$

The infinite-horizon average cost under a linear state-feedback policy $u_t = Kx_t$ with $K \in \mathcal{S}$ is defined as:

$$C(K) := \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{x_0,w_t}\left[\sum_{t=0}^{T-1} x_t^\top Q_K x_t\right], \tag{2}$$

with $Q_K := Q + K^\top R K$. For any stabilizing policy $K \in \mathcal{S}$, the gradient of the cost function $C(K)$ is given by:

$$\nabla C(K) = 2E_K \Sigma_K, \tag{3}$$

where $E_K := \left(R + B^\top P_K B\right) K + B^\top P_K A$; P_K is the unique solution to the equation $P_K = A_K^\top P_K A_K + Q_K$, and Σ_K is the average covariance matrix associated with $K \in \mathcal{S}$, defined as

$$\Sigma_K := \lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T-1} \Sigma_t, \text{ with } \Sigma_t := \mathbb{E}_{x_0,w_t}[x_t x_t^\top]. \tag{4}$$

It is a well-known fact (Lewis et al. (2012)) that the optimal K^* minimizing $C(K)$ satisfies

$$K^* = -(R + B^\top P_{K^*} B)^{-1} B^\top P_{K^*} A, \tag{5a}$$

$$\begin{aligned} P_{K^*} = Q + A^\top P_{K^*} A \\ - A^\top P_{K^*} B (R + B^\top P_{K^*} B)^{-1} B^\top P_{K^*} A. \end{aligned} \tag{5b}$$

The average covariance matrix associated with the optimal K^* is denoted as Σ_{K^*}. For later use, we recall the local Lipschitz continuity property of the cost function C.

Lemma 1. (Song and Iannelli, 2025a, Lemma 4) Suppose $K', K \in \mathcal{S}$ are such that $\|K - K'\| \le \min\{h(C(K)), \|K\|\}$ with $h(C(K)) := \frac{\lambda_1(\Sigma_w)\lambda_1(Q)}{4C(K)\|B\|(\|A\|+\|B\|b_K(C(K))+1)}$, it holds:

$$\|C(K) - C(K')\| \le h_C(C(K))\|K - K'\|, \tag{6}$$

where h_C is defined as:

$$\begin{aligned} h_C(C(K)) := 6\Big(\frac{C(K)}{\lambda_1(\Sigma_w)\lambda_1(Q)}\Big)^2 \big(b_K^2(C(K))\|R\|\|B\|(\|A\| \\ +\|B\|b_K(C(K))) + b_K(C(K))\|R\|\big)\mathrm{Tr}(\Sigma_w), \end{aligned}$$

and $b_K(C(K))$ is an upper bound on $\|K\|$, defined as:

$$\begin{aligned} b_K(C(K)) := \frac{1}{\lambda_1(R)}\big(\|B\|\|A\|\frac{C(K)}{\lambda_1(\Sigma_w)} + \\ \sqrt{(C(K) - C(K^*))\big(\|R\| + \|B\|^2\frac{C(K)}{\lambda_1(\Sigma_w)}\big)\lambda_1(\Sigma_w)^{-1}}\big). \end{aligned}$$

The update of the Natural Policy Gradient (NPG) and the Gauss–Newton Method (GNM), $\forall i \in \mathbb{Z}_+$, are given by:

- NPG:
$$\begin{aligned} K_{i+1} &= K_i - \eta \nabla C(K_i)\Sigma_{K_i}^{-1} \\ &= K_i - 2\eta\left[(R + B^\top P_{K_i} B)K_i + B^\top P_{K_i} A\right]; \end{aligned} \tag{7}$$
- GNM:
$$\begin{aligned} K_{i+1} &= K_i - \eta(R + B^\top P_{K_i} B)^{-1}\nabla C(K_i)\Sigma_{K_i}^{-1} \\ &= K_i - 2\eta\left[K_i + (R + B^\top P_{K_i} B)^{-1} B^\top P_{K_i} A\right], \end{aligned} \tag{8}$$

where $\eta > 0$ denotes the step size. The detailed derivations of these update rules and their model-based convergence guarantees are given in Song and Iannelli (2025a). In this work, for the model-free implementations of NPG and GNM, we directly estimate the quantities appearing in the update rules (7) and (8), namely, $B^\top P_{K_i} B$ and $B^\top P_{K_i} A$.

3. FROM DATA TO POLICY GRADIENT VIA THE BELLMAN EQUATION

To estimate the matrices for policy gradient update, we first collect off-policy data $(x^{(k)}, u^{(k)}, x_+^{(k)})_{k=1}^N$ using Algorithm 1. The required sample size N will be specified later.

Algorithm 1 Data Collection

for $k = 1, ..., N$ **do**
  Generate $x^{(k)} \sim \mathcal{N}(0, \Sigma_x)$ and $u^{(k)} \sim \mathcal{N}(0, \Sigma_u)$
  Obtain $x_+^{(k)} = Ax^{(k)} + Bu^{(k)} + w^{(k)}, w^{(k)} \sim \mathcal{N}(0, \Sigma_w)$
end for

Given an arbitrary random triple $(x^{(k)}, u^{(k)}, x_+^{(k)})$, consider the LQR cost in (1), let $u' = Kx$ denote the linear feedback policy, and x_+' the corresponding next state under this policy. The Bellman equation then reads:

$$x^\top P_K x + \mathrm{Tr}(P_K \Sigma_w) = x^\top Q_K x + \mathbb{E}[x_+'^\top P_K x_+' | x, K].$$

When the system is excited using a generic input u from Algorithm 1 that does not follow the linear policy, we define the deviation $\eta := u - Kx$. We have $x_+' = x_+ - B\eta$, then the equation above can be rewritten as

$$\begin{aligned} x^\top P_K x + \mathrm{Tr}(P_K \Sigma_w) = x^\top Q_K x \\ + \mathbb{E}[(x_+ - B\eta)^\top P_K (x_+ - B\eta) | x, K, u]. \end{aligned} \tag{9}$$

Replacing x_+ with dynamics in (1) yields:

$$\begin{aligned} x^\top P_K x + \mathrm{Tr}(P_K \Sigma_w) = x^\top Q_K x + \mathbb{E}[x_+^\top P_K x_+ | x, K, u] \\ + \eta^\top B^\top P_K B \eta - 2\mathbb{E}[(Ax + BKx + B\eta + w)^\top P_K B \eta]. \end{aligned}$$

Rearranging terms, the equation above can be compactly expressed as

$$\Gamma^\top \underbrace{\begin{bmatrix} vec(B^\top P_K A) \\ vecs(B^\top P_K B) \\ vecs(P_K) \end{bmatrix}}_{=:\xi_K} = \underbrace{x^\top (Q + K^\top R K) x}_{=:c}, \tag{10}$$

where $\Gamma := \begin{bmatrix} 2x \otimes (u - Kx) \\ vecv(u) - vecv(Kx) \\ vecv(x) + W - \mathbb{E}[vecv(x_+)|x, K, u] \end{bmatrix}$ and $W := \sum_{k=1}^{n_x} vecv(\sqrt{\lambda_k(\Sigma_w)} v_k)$ with λ_k and v_k denoting the k-th eigenvalue and eigenvector of the covariance matrix Σ_w, respectively. The matrix W is introduced so that the trace term in (10) can be expressed as $\mathrm{Tr}(P_K \Sigma_w) = W vecs(P_K)$. Since the quantity $\mathbb{E}[\mathrm{vecv}(x_+) \mid x, K, u]$ is not analytically computable when A, B are unknown in the model-free setting, we approximate this expectation using the observed sample x_+. Accordingly, we define the data-dependent quantity

$$\hat{\Gamma} := \begin{bmatrix} 2x \otimes (u - Kx) \\ vecv(u) - vecv(Kx) \\ vecv(x) + W - vecv(x_+) \end{bmatrix}. \tag{11}$$

Given a stabilizing feedback gain $K \in \mathcal{S}$ and dataset $(x^{(k)}, u^{(k)}, x_+^{(k)})_{k=1}^N$, we construct the corresponding regression pairs $(c^{(k)}, \hat{\Gamma}^{(k)})_{k=1}^N$ as defined in (10) and (11), respectively. The superscript (k) indicates that these quantities are computed from the sample $(x^{(k)}, u^{(k)}, x_+^{(k)})$. From (11), we have that $\Gamma^{(k)} = \mathbb{E}_w[\hat{\Gamma}^{(k)}]$, $\forall k \in [1, N]$. Using the collected dataset, one option is to formulate the problem as least-squares (LS):

$$\hat{\xi}_K = \arg\min_{\xi} \frac{1}{N} \sum_{k=1}^{N} \|\hat{\Gamma}^{(k)} \xi - c^{(k)}\|^2. \tag{12}$$

When the dataset $(c^{(k)}, \hat{\Gamma}^{(k)})_{k=1}^N$ is sufficiently rich such that $\sum_{k=1}^N \hat{\Gamma}^{(k)} \hat{\Gamma}^{(k)\top} \succ 0$, the estimator $\hat{\xi}_K$ has the closed form expression

$$\hat{\xi}_K = \left(\sum_{k=1}^{N} \hat{\Gamma}^{(k)} \hat{\Gamma}^{(k)\top} \right)^{-1} \left(\sum_{k=1}^{N} \hat{\Gamma}^{(k)} c^{(k)} \right). \tag{13}$$

Note that $vecv(x_+^{(k)})$ enters $\hat{\Gamma}^{(k)}$ in (13), so that the regression is formed using a single realization rather than its conditional expectation, which falls into an errors-in-variables (EIV) setting. Consequently, the least-squares estimator (13) is inconsistent in this setting. To obtain a consistent estimate, we propose using a primal-dual estimation method, as discussed in the following section.

4. PRIMAL-DUAL ESTIMATION

Instead of solving a standard least-squares regression in (12), we consider the following *stochastic saddle-point problem* (Lan, 2020, Section 3.6):

$$\min_{\xi \in X} \max_{y \in Y} \mathbb{E}_{[x,u,x_+]}[y(\hat{\Gamma}\xi - c)], \tag{14}$$

where $\hat{\Gamma}$ and c are constructed using the random data triple $[x, u, x_+]$ generated by Algorithm 1; $Y := \{y \in \mathbb{R} \mid \|y\| \leq 1\}$ and X is a compact convex set containing the true parameter ξ_K. The construction of the set X will be discussed in Subsection 4.2. The min-max formulation admits a worst-case robustification interpretation, where the inner maximization evaluates the largest residual violation over all testing directions of y. We can not directly solve the problem (14) due to the expectation. We approximate it using the available N samples, resulting in the *empirical min–max problem*:

$$\min_{\xi \in X} \max_{y \in Y} \frac{1}{N} \sum_{k=1}^{N} y(\hat{\Gamma}^{(k)} \xi - c^{(k)}). \tag{15}$$

Before presenting an algorithm for solving (15), we introduce an assumption and various lemmas that will play a key role in the estimation error analysis.

Assumption 1. (Informativity). Matrix $\Gamma^{(k)} \in \mathbb{R}^{n_\Gamma}$ with $n_\Gamma := n_x n_u + \frac{n_x(n_x+1)}{2} + \frac{n_u(n_u+1)}{2}$ satisfies:

$$\Gamma^{(k)\top} \Gamma^{(k)} \succeq \alpha, \quad \forall k \in [1, N], \tag{16}$$

for some constant $\alpha > 0$.

From the expression of $\Gamma^{(k)}$, the randomness in $(x^{(k)}, u^{(k)})$ from Algorithm 1 helps ensure that this assumption holds. The reason for introducing this assumption will be explained later. In the stochastic setting, due to the unbounded nature of $(x^{(k)}, u^{(k)}, x_+^{(k)})$, the following lemmas establish high-probability bounds on the collected data.

Lemma 2. Let $\delta \in (0, \frac{1}{e}]$ and define the event

$$\beta^{(k)}(\delta) := \Big\{ \|[x^{(k)}; u^{(k)}; x_+^{(k)}]\|^2 \leq 4\big(\frac{c_2^2}{\sqrt{c_1}} \|\tilde{\Sigma}\| + \frac{c_2^2}{c_1} \mathrm{Tr}(\tilde{\Sigma})\big) \ln \frac{1}{\delta} \Big\}. \tag{17}$$

where $\tilde{\Sigma} := \begin{bmatrix} \Sigma_x & 0 & \Sigma_x A^\top \\ 0 & \Sigma_u & \Sigma_u B^\top \\ A\Sigma_x & B\Sigma_u & \Sigma_{x_+} \end{bmatrix}$ with $\Sigma_{x_+} := A\Sigma_x A^\top + B\Sigma_u B^\top + \Sigma_w$ and c_1, c_2 are two constants introduced in (Song et al., 2025, Lemma 6). Then, the following holds:

$$\mathbb{P}\big[\beta^{(k)}(\delta)\big] \leq 1 - \delta, \forall k \in [1, N]. \tag{18}$$

Consequently, for all $k \in [1, N]$, when $\beta^{(k)}(\delta)$ happens, the associated regression data $\hat{\Gamma}^{(k)}$ and $c^{(k)}$ are also guaranteed to be bounded.

Lemma 3. Suppose the events $\beta^{(k)}(\delta)$ occur for some $\delta \in (0, \frac{1}{e}]$, for all $k \in [1, N]$. Then, the regression data satisfy

$$\|\hat{\Gamma}^{(k)}\| \geq M_\Gamma \left(\ln \frac{1}{\delta} \right), \ \|c^{(k)}\| \geq M_c \left(\ln \frac{1}{\delta} \right), \forall k \in [1, N],$$

with constants M_Γ and M_c (detailed expressions can be found in (Song et al., 2025, (A.1), (A.2))).

Lemma 4. Let $\bar{\Gamma} := \mathbb{E}_{[x,u]}[\Gamma] = \mathbb{E}_{[x,u,x_+]}[\hat{\Gamma}]$, where $[x, u, x_+]$ is a random data triple from Algorithm 1. Then,

$$\|\bar{\Gamma}\| \leq L_\Gamma := 2\|K\|\|\Sigma_x\|_F + \|\Sigma_u\| + (\|K\|^2 + 1)\|\Sigma_x\| + \|\Sigma_{x_+}\| + \|W\|.$$

4.1 Estimation Error Analysis

We now solve the min–max problem (15) using the conditional stochastic primal–dual (CSPD) algorithm presented in Algorithm 2. Before analyzing the convergence properties of the algorithm, we introduce two key definitions: the *Q-gap function* and the *primal–dual gap*. Let $z' := (\xi_K', y')$ and $\tilde{z} := (\tilde{\xi}_K, \tilde{y})$. The Q-gap function is defined as

$$Q(\tilde{z}, z') := \frac{1}{N} \sum_{k=1}^{N} (\Gamma^{(k)} \tilde{\xi}_K - c^{(k)}) y' - (\Gamma^{(k)} \xi_K' - c^{(k)}) \tilde{y},$$

and the primal–dual gap is defined as:

$$g(\tilde{z}) := \max_{z' = (\tilde{\xi}_K, \tilde{y}) \in X \times Y} Q(\tilde{z}, z'). \tag{19}$$

Additionally, we introduce the following function:

Algorithm 2 Conditional Stochastic Primal-dual (CSPD)

Input: Gain matrix $K \in \mathcal{S}; X; \xi_K^{(-1)} = \xi_K^{(0)} \in X; N; y^{(0)} \in Y; \{\eta_k, \lambda_k, \zeta_k\}_{k=1}^N$
for $k = 1, ..., N$ **do**
$G^{(k)} = \xi_K^{(k-1)} + \zeta_k(\xi_K^{(k-1)} - \xi_K^{(k-2)})$
Using $\{x^{(k)}, u^{(k)}, x_+^{(k)}\}$ and K to build $\{\hat{\Gamma}^{(k)}, c^{(k)}\}$.
$y^{(k)} = \arg\min_{\tilde{y}\in Y}\{(c^{(k)} - \hat{\Gamma}^{(k)\top} G^{(k)})\tilde{y} + \frac{\lambda_k}{2}(\tilde{y} - y^{(k-1)})^2\}$
$\xi_K^{(k)} = \arg\min_{\xi\in X}\{y^{(k)}(\hat{\Gamma}^{(k)\top}\xi) + \frac{\eta_k}{2}\|\xi - \xi_K^{(k-1)}\|^2\}$
end for
$(\hat{\xi}_K, \hat{y}) = \frac{2}{N(N+1)}\sum_{k=1}^N k \cdot (\xi_K^{(k)}, y^{(k)})$

$$f(\tilde{\xi}_K) := \max_{y\in Y} \frac{1}{N}\sum_{k=1}^N y(\Gamma^{(k)}\tilde{\xi}_K - c^{(k)}). \tag{20}$$

Since $\Gamma^{(k)}\xi_K = c^{(k)}, \forall k \in [1, N]$, it follows that $g(\hat{z}) \geq f(\hat{\xi}_K)$ where $\hat{z} := (\hat{\xi}_K, \hat{y})$ is the output of Algorithm 2.

Lemma 5. Under Assumption 1:

$$\sqrt{\alpha}\|\hat{\xi}_K - \xi_K\| \leq g(\hat{z}), \tag{21}$$

where α is the constant introduced in Assumption 1.

Lemma 5 allows the estimation error $\|\hat{\xi}_K - \xi_K\|$ to be bounded in terms of the primal-dual gap $g(\hat{z})$. To leverage this, we first recall the following upper bound on the Q-gap function.

Theorem 1. (Lan, 2020, Theorem 3.8) Let $\{\gamma_k, \eta_k, \lambda_k, \zeta_k\}$ be a set of nonnegative reals satisfying $\gamma_{k-1}\eta_{k-1} \leq \gamma_k\eta_k, \gamma_{k-1}\lambda_{k-1} \leq \gamma_k\lambda_k$ and $\gamma_k\zeta_k = \gamma_{k-1}$, and let there exist some $p, q \in (0, 1)$ satisfying $L_\Gamma^2 \leq \frac{q\eta_k}{p\lambda_k}$ for all $k \in [1, N]$ and $K \in \mathcal{S}$. Then

$$\sum_{k=1}^N \gamma_k Q(z^{(k)}, z') \leq \gamma_N\eta_N D_X^2 + \gamma_N\lambda_N D_Y^2 + \sum_{k=1}^N \Lambda_k(z'),$$

where $z^{(k)} := [\xi_K^{(k)}, y^{(k)}]$, $D_X^2 := \max_{x_1,x_2\in X}\|x_1 - x_2\|^2$ and $D_Y^2 := \max_{y_1,y_2\in Y}(y_1 - y_2)^2$ and $\Lambda_k(z') := -\alpha_1(y^{(k)}) - \alpha_2(\xi_K^{(k)}) + \alpha_3(y^{(k)}, y') + \alpha_4(\xi_K^{(k)}, \xi_K')$ with $\alpha_1(y^{(k)}) := \frac{(1-p)\gamma_k\lambda_k}{2}(y^{(k)} - y^{(k-1)})^2$, $\alpha_2(\xi_K^{(k)}) := \frac{(1-q)\gamma_k\eta_k}{2}\|\xi_K^{(k)} - \xi_K^{(k-1)}\|^2$, $\alpha_3(y^{(k)}, y') := \gamma_k[(\hat{\Gamma}^{(k)} - \Gamma^{(k)})G^{(k)}](y^{(k)} - y')$, and $\alpha_4(\xi_K^{(k)}, \xi_K') := \gamma_k[(\hat{\Gamma}^{(k)} - \Gamma^{(k)})^\top y^{(k)}]^\top(\xi_K^{(k)} - \xi_K')$.

By analyzing the terms Λ_k in Theorem 1, we can derive the following theorem, which bounds the primal–dual gap.

Theorem 2. Suppose $\beta(\delta) := \cap_{k\in[1,N]}\beta^{(k)}(\delta)$ occurs for some $\delta \in (0, \frac{1}{e}]$. Under the same assumptions as in Theorem 1, then with probability at least $1 - 2(N+1)\delta$,

$$\left(\sum_{k=1}^N \gamma_k\right) g(\hat{z}) \leq 2(D_X M_X + D_Y M_Y)\sqrt{8\ln\frac{1}{\delta}\sum_{k=1}^N \gamma_k^2}$$
$$+ 2\gamma_N(\eta_N D_X^2 + \lambda_N D_Y^2) + \sum_{k=1}^N\left[\frac{16M_X^2}{\eta_k(1-q)} + \frac{16M_Y^2}{\lambda_k(1-p)}\right],$$

where $\hat{z} := (\hat{\xi}_K, \hat{y})$ is from Algorithm 2; $\Omega_Y := \max_{y\in Y}\{\|y\|\}$; $\Omega_X := \|\xi_K\| + (1+\bar{\zeta})\sqrt{2}D_X$; $\bar{\zeta} := \max_k \zeta_k$; $M_X := M_\Gamma\Omega_Y \ln\frac{1}{\delta}$; $M_Y := M_\Gamma\Omega_X \ln\frac{1}{\delta}$.

By selecting the parameters $\{\eta_k, \zeta_k, \lambda_k\}_{k=1}^N$, we can use Theorem 2 to derive the estimation error upper bound.

Corollary 1. Let $\eta_k = \frac{3\sqrt{2}L_\Gamma D_Y k + 6M_X k^{\frac{3}{2}}}{2\sqrt{2}D_X k}$, $\zeta_k = \frac{k-1}{k}$ and $\lambda_k = \frac{3\sqrt{2}L_\Gamma D_X k + 6M_Y k^{\frac{3}{2}}}{2\sqrt{2}D_Y k}$, $\forall k \in [1, N]$. If $\beta(\delta)$ occurs for some $\delta \in (0, \frac{1}{e}]$, under Assumption 1, then with probability at least $1 - 2(N+1)\delta$:

$$\|\hat{\xi}_K - \xi_K\| \leq \frac{\alpha_5}{N+1} + \frac{\alpha_6}{\sqrt{N}}, \tag{22}$$

with $\alpha_5 := \frac{12L_\Gamma D_X D_Y}{\sqrt{\alpha}}$ and $\alpha_6 := \frac{\alpha_7(D_X M_X + D_Y M_Y)}{\sqrt{\alpha}}$ and $\alpha_7 := 2(48 + 3\sqrt{2} + \frac{16\sqrt{2}}{\sqrt{3}})$.

Corollary 1 shows that the estimator based on the primal–dual optimization method (CSPD) is consistent and yields the sample complexity $\mathcal{O}(\epsilon^{-2} + \epsilon^{-1})$ for accuracy ϵ with high probability. Here we emphasize that these asymptotic convergence properties cannot be obtained using standard least squares (13) due to the errors-in-variables setting.

4.2 Multi-Epoch Scheme for Reducing Sample Complexity

From the constant D_X in (22), which characterizes the size of X, it is clear that if the feasible set X is updated adaptively, the convergence rate can be further improved. Motivated by (Lan, 2020, Lemma 4.5), we propose a multi-epoch algorithm that repeatedly invokes Algorithm 2, using the solution from the previous epoch to warm-start the next epoch. For the initialization of X_0, we can select a larger region containing ξ_K. Then, Algorithm 3 progressively shrinks the feasible set X_s as the epoch s increases, thereby accelerating convergence. The number of samples used in epoch s is denoted by N_s. The procedure is summarized in Algorithm 3.

Algorithm 3 Shrinking Multi-epoch CSPD

Input: $K \in \mathcal{S}$; X; $\tilde{\xi}_0 \in X$; $D_0, S \in \mathbb{R}_{++}$
for $s = 1, ..., S$ **do**
$D_s^2 := 2^{-(s-1)}D_0^2$
$X_s := \{\xi \in X : \|\tilde{\xi}_{s-1} - \xi\| \leq D_s^2\}$
$\tilde{\xi}_s \leftarrow \hat{\xi}_K$ from Algorithm 2 with K; X_s; $\xi_K^{(-1)} = \xi_K^{(0)} = \tilde{\xi}_{s-1}, N_s$ specified by Theorem 3; $y^{(0)} \in Y$; $\{\eta_k, \lambda_k, \zeta_k\}_{k=1}^{N_s}$ specified by Corollary 1
end for
Return $\hat{\xi}_K = \tilde{\xi}_S$

The following theorem characterizes the convergence and sample complexity of Algorithm 3.

Theorem 3. Consider the same assumptions and parameters as in Corollary 1. Suppose $\|\tilde{\xi}_0 - \xi_K\| \leq D_0^2$. Define the number of iterations N_s at the s-th epoch as

$$N_s := \lceil 400\max\{\alpha_8, \alpha_9(\delta)\big(M_X^2 + \frac{D_Y^2 M_Y^2}{D_0^2}2^s\big)\}\rceil, \tag{23}$$

with $\alpha_8 := \frac{L_\Gamma D_Y}{\alpha}$ and $\alpha_9(\delta) := \frac{4000 + 256\ln\frac{1}{\delta}}{\alpha^2}$. Given an arbitrary accuracy ϵ, choose the number of epochs as $S = \lceil\log_2\frac{D_0^2}{\epsilon}\rceil$. Then, with probability at least $1 - 2(N + S)\delta$, the estimate from Algorithm 3 after S epochs satisfies $\|\hat{\xi}_K - \xi_K\| \leq \epsilon$ and the total number of iterates over all epochs is at most:

$$N := 400\left\lceil 2\alpha_8\ln\frac{D_0}{\epsilon} + \alpha_9(\delta)\left(2M_X^2\ln\left(\frac{D_0}{\epsilon}\right) + \frac{D_Y^2 M_Y^2}{\epsilon}\right)\right\rceil.$$

From Theorem 3, we can conclude that, to achieve a desired accuracy ϵ, the required sample complexity scales as $\mathcal{O}(\epsilon^{-1} + \ln(\epsilon^{-1}))$.

5. CONVERGENCE ANALYSIS OF NPG AND GNM

Leveraging the CSPD Algorithms described in Section 4, we are able to estimate the matrices required for implementing the NPG and GNM algorithms to any desired level of accuracy. In this section, we investigate the robustness of the NPG and GNM algorithms with respect to the estimation errors introduced by Algorithm 2 or Algorithm 3. The procedure is summarized in Algorithm 4.

Algorithm 4 Model-free NPG/GNM

Input: $\hat{K}_0 \in \mathcal{S}$
  Run Algorithm 1 to collect data
  for $i = 1, ..., +\infty$ **do**
    Run Algorithm 2 or Algorithm 3 to obtain $\hat{\xi}_{\hat{K}_i}$
    Update $\hat{K}_i$ using NPG (24) or GNM (25)
  end for

The following result provides convergence guarantees for both methods using Algorithm 4:

Theorem 4. Suppose the initial $\hat{K}_0 \in \mathcal{S}$, and consider the natural policy gradient iterates for all $i \in \mathbb{Z}_+$:

$$\hat{K}_{i+1} = \hat{K}_i - 2\eta\big[(R + \widehat{B^\top P_{\hat{K}_i}} B)\hat{K}_i + \widehat{B^\top P_{\hat{K}_i}} A\big], \quad (24)$$

where $\widehat{B^\top P_{\hat{K}_i}} B, \widehat{B^\top P_{\hat{K}_i}} A$ are the estimates from Algorithm 2 or Algorithm 3 and $\eta \le \frac{1}{2\|R+B^\top P_{\hat{K}_0} B\|}$. Given any accuracy $\epsilon > 0$, $\sigma \in (0,1)$, choose the number of iterations n_{N} as

$$n_{\mathrm{N}} \ge \frac{\|\Sigma_{K^*}\|}{2(1-\sigma)\eta\lambda_1(R)\lambda_1(\Sigma_w)} \log \frac{C(\hat{K}_0) - C(K^*)}{\epsilon}.$$

Given any probability $\delta \in (0,1)$ satisfying $\delta n_{\mathrm{N}} \in (0,1)$, assume that the estimation error of $\hat{\xi}_{\hat{K}_i}$ from Algorithm 2 or Algorithm 3 satisfies, $\forall i \in \mathbb{Z}_+$:

$$\mathbb{P}\big\{\|\hat{\xi}_{\hat{K}_i} - \xi_{\hat{K}_i}\| \le \frac{\sigma\epsilon\lambda_1(R)\lambda_1(\Sigma_w)}{h_C(\hat{K}_0)(1 + b_K(C(\hat{K}_0)))\|\Sigma_{K^*}\|}\big\} \ge 1-\delta,$$

where h_C, b_K were introduced in Lemma 1. Then for any $C(\hat{K}_i) \ge C(K^*) + \epsilon$, the following inequality holds:

$$\mathbb{P}\big\{C(\hat{K}_{i+1}) - C(K^*) \le \hat{\gamma}_{\mathrm{N}}(C(\hat{K}_i) - C(K^*))\big\} \ge 1 - \delta,$$

where $\hat{\gamma}_{\mathrm{N}} := 1 - (1-\sigma)\frac{2\eta\lambda_1(R)\lambda_1(\Sigma_w)}{\|\Sigma_{K^*}\|} < 1$. As a result, the NPG method enjoys the following performance bound:

$$\mathbb{P}\big\{\min_{i\in[0,n_{\mathrm{N}}]} C(\hat{K}_i) - C(K^*) \le \epsilon\big\} \ge 1 - \delta n_{\mathrm{N}}.$$

Similarly, we can establish the convergence guarantee for the Gauss–Newton method:

Theorem 5. Suppose the initial $\hat{K}_0 \in \mathcal{S}$, and consider the Gauss-Newton iterates for all $i \in \mathbb{Z}_+$:

$$\hat{K}_{i+1} = \hat{K}_i - 2\eta\left[\hat{K}_i + (R + \widehat{B^\top P_{\hat{K}_i}} B)^{-1}\widehat{B^\top P_{\hat{K}_i}} A\right], \quad (25)$$

where $\widehat{B^\top P_{\hat{K}_i}} B, \widehat{B^\top P_{\hat{K}_i}} A$ are the estimates from Algorithm 2 or Algorithm 3 and $\eta \le \frac{1}{2}$. Given any accuracy $\epsilon > 0$, $\sigma \in (0,1)$, choose the number of iterations n_{G} as

$$n_{\mathrm{G}} \ge \frac{\|\Sigma_{K^*}\|}{2(1-\sigma)\eta\lambda_1(\Sigma_w)} \log \frac{C(\hat{K}_0) - C(K^*)}{\epsilon}.$$

Given any probability $\delta \in (0,1)$ satisfying $\delta n_{\mathrm{G}} \in (0,1)$, assume that the estimation error of $\hat{\xi}_{\hat{K}_i}$ from Algorithm 2 and Algorithm 3 satisfies, $\forall i \in \mathbb{Z}_+$:

$$\mathbb{P}\big\{\|\hat{\xi}_{\hat{K}_i} - \xi_{\hat{K}_i}\| \le \Delta_{GD}\big\} \ge 1 - \delta,$$

where $\Delta_{GD} := \min\big(\frac{\lambda_1(R)}{2}, \frac{\sigma\epsilon\lambda_1(\Sigma_w)}{h_C(C(\hat{K}_0))\|\Sigma_{K^*}\|\bar{\Delta}_G}\big)$ and $\bar{\Delta}_G := \|R^{-1}\| + \frac{\lambda_1(R)}{2} + \frac{\|A\|\|B\|C(\hat{K}_0)}{\lambda_1(\Sigma_w)}$. Then for any $C(\hat{K}_i) \ge C(K^*) + \epsilon$, the following inequality holds:

$$\mathbb{P}\big\{C(\hat{K}_{i+1}) - C(K^*) \le \hat{\gamma}_G(C(\hat{K}_i) - C(K^*))\big\} \ge 1 - \delta,$$

where $\hat{\gamma}_G := 1-(1-\sigma)\frac{2\eta\lambda_1(\Sigma_w)}{\|\Sigma_{K^*}\|} < 1$. As a result, the GNM enjoys the following performance bound:

$$\mathbb{P}\big\{\min_{i\in[0,n_{\mathrm{G}}]} C(\hat{K}_i) - C(K^*) \le \epsilon\big\} \ge 1 - \delta n_{\mathrm{G}}.$$

For both methods, to achieve a desired accuracy on $C(\hat{K}_i) - C(K^*)$, we can use Algorithms 2 or 3 to ensure that the required precision is met. In particular, to reach a target optimality gap ϵ, Theorems 4 and 5 imply that the corresponding estimation error must satisfy $\|\hat{\xi}_{\hat{K}_i} - \xi_{\hat{K}_i}\| = \mathcal{O}(\epsilon)$. Since $\|\xi_{\hat{K}_i}\| \le (\|B\|^2 + \|A\|\|B\| + 1)\frac{C(\hat{K}_0)}{\lambda_1(\Sigma_w)}$ and $\|\hat{K}_i\| \le b_K(C(\hat{K}_0)), \forall i \in \mathbb{Z}_+$, a unified choice of Algorithm 2 parameters can be made to ensure the estimation error requirement is satisfied at every iteration, according to Corollary 1. Then, applying Theorem 3 and Corollary 1, the corresponding sample complexity using Algorithm 3 to achieve this accuracy is therefore $\mathcal{O}(\epsilon^{-1})$. The improvement compared with Zhou and Lu (2023); Ju et al. (2025) was achieved thanks to the fact that we reused the collected data in Algorithm 4 instead of having to collect them at each iteration.

6. NUMERICAL RESULTS

In this section, we present simulation results [1] to illustrate the effectiveness and advantages of Algorithm 4 discussed in the previous sections. We consider the following system from [Yaghmaie et al. (2023); Ju et al. (2025)]:

$$x_{t+1} = \underbrace{\begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{B} u_t + w_t. \quad (26)$$

The weight matrices Q and R are chosen as $0.001I_3$ and I_3. The initial gain $\hat{K}_0$ is selected as the optimal LQR solution for $(A, B, 100Q, R)$. For data collection, we set $\Sigma_x = \Sigma_u = I_3$ and $\Sigma_w = 0.1I_3$. A total of 100 data triples are collected. The primal-dual optimization (Algorithm 2) is configured with $X := \{\xi \in \mathbb{R}^{21} \mid \|\xi\| \le 1\}$, $\zeta_k = \frac{k-1}{k}$, $\lambda_k = \eta_k = 0.001\sqrt{k}$. For Algorithm 3, we set $S = 4$ with epoch sample sizes $N_s = [8, 16, 24, 52]$ and $D_0 = 1$. For all policy gradient update estimation schemes used to implement the model-free NPG and GNM methods, the step size in (24) and (25) is fixed at 0.05. The simulation results are obtained from a Monte Carlo simulation over 30 data samples.

[1] The Matlab codes used to generate these results are accessible from the repository: https://github.com/col-tasas/2025-MFPGM-RobustLR

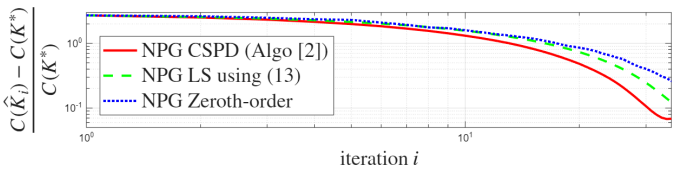

Fig. 1. Convergence comparison of model-free NPG using different policy gradient update estimation schemes

In Figure 1, we illustrate the convergence behavior of three model-free NPG methods. The red solid curve corresponds to the NPG update using the quantities estimated via CSPD method in Algorithm 2 proposed in Section 4. We observe that the algorithm converges to the expected suboptimal value. Using the *same dataset*, we also apply the standard least-squares estimator in (13), shown as the black dashed curve. As the plot indicates, the convergence based on the NPG update estimation via (13) is significantly slower than that obtained using the CSPD method due to the inconsistency introduced by least-squares estimation in the errors-in-variables setting. In addition, we compare against the zeroth-order framework by Song and Iannelli (2025a), plotted as the blue dotted curve. The zeroth-order algorithm (Song and Iannelli, 2025a, Algorithm 1) is set to $r = 0.4,\ n = 1000,\ l = 80$. Implementing this algorithm requires a total of $80 * 100 * 35$ samples, substantially more than needed for the CSPD method.

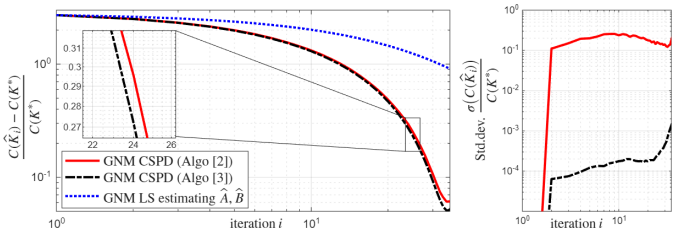

Fig. 2. Convergence comparison of model-free GNM using different policy gradient update estimation schemes

In Figure 2, we compare the convergence behavior of three model-free GNM methods. The red solid curve corresponds to the GNM update estimated using the CSPD method (Algorithm 2), while the black dash-dotted curve shows the GNM update obtained using the multi-epoch CSPD method (Algorithm 3). Both methods use the same dataset, yet the multi-epoch variant achieves faster convergence and a smaller suboptimality gap from the left subplot. The right subplot reports the normalized standard deviation over the Monte Carlo trials, showing that multi-epoch Algorithm 3 also yields smaller variance compared with Algorithm 2. The blue dotted curve represents the GNM update formulated by first identifying the system matrices $\hat{A}, \hat{B}$ from the same data and then applying a model-based update, which results in the slowest convergence among the three methods.

7. CONCLUSION

In this work, we employ a primal–dual framework to estimate the required matrices to formulate model-free natural policy gradient and Gauss–Newton methods for the LQR problem using stochastic data. By adopting a multi-epoch scheme and reusing previously collected data, we achieve a sample complexity of $\mathcal{O}(\epsilon^{-1})$ for the convergence of the algorithms, which improves upon the existing results in the literature. Future directions include investigating whether similar convergence guarantees can be established in an online setting, and developing a closed-loop analysis of the overall system that jointly characterizes the dynamics and the algorithmic updates.

REFERENCES

Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *Proc. of the 37th Machine Learning Research*, volume 80, 1467–1476. PMLR.

Hambly, B., Xu, R., and Yang, H. (2021). Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM Journal on Control and Optimization*, 59(5), 3359–3391.

Ju, C., Kotsalis, G., and Lan, G. (2025). A model-free first-order method for linear quadratic regulator with $\tilde{o}(1/\varepsilon)$ sampling complexity. *SIAM Journal on Control and Optimization*, 63(3), 2098–2123.

Lan, G. (2020). *First-order and Stochastic Optimization Methods for Machine Learning*. Springer.

Lewis, F.L., Vrabie, D., and Syrmos, V.L. (2012). *Optimal control*. John Wiley & Sons.

Malik, D., Pananjady, A., Bhatia, K., Khamaru, K., Bartlett, P., and Wainwright, M. (2019). Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. In *Proc. of the 22nd Int. Conf. on Artificial Intelligence and Statistics*, 2916–2925. PMLR.

Moghaddam, A.N., Olshevsky, A., and Gharesifard, B. (2025). Sample complexity of the linear quadratic regulator: A reinforcement learning lens. *Journal of Machine Learning Research*, 26(151), 1–50.

Song, B., Gros, S., and Iannelli, A. (2025). Sample-efficient model-free policy gradient methods for stochastic LQR via robust linear regression. arXiv preprint arXiv:2512.03764.

Song, B. and Iannelli, A. (2025a). Convergence guarantees of model-free policy gradient methods for LQR with stochastic data. arXiv preprint arXiv:2502.19977.

Song, B. and Iannelli, A. (2025b). Robustness of online identification-based policy iteration to noisy data. *at - Automatisierungstechnik*, 73(6), 398–412.

Tu, S. and Recht, B. (2018). Least-squares temporal difference learning for the linear quadratic regulator. In *Proc. of the 35th Int. Conf. on Machine Learning*, 5005–5014. PMLR.

Yaghmaie, F.A., Gustafsson, F., and Ljung, L. (2023). Linear quadratic control using model-free reinforcement learning. *IEEE Transactions on Automatic Control*, 68(2), 737–752.

Yang, Y., Kiumarsi, B., Modares, H., and Xu, C. (2023). Model-free λ-policy iteration for discrete-time linear quadratic regulation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2), 635–649.

Zhao, F., Chiuso, A., and Dörfler, F. (2025). Policy gradient adaptive control for the LQR: Indirect and direct approaches. arXiv preprint arXiv:2505.03706.

Zhou, M. and Lu, J. (2023). Single timescale actor-critic method to solve the linear quadratic regulator with convergence guarantees. *Journal of Machine Learning Research*, 24(222), 1–34.