Oracle Complexity Reduction for Model-free LQR: A Stochastic Variance-Reduced Policy Gradient Approach

Leonardo F. Toso, Han Wang, James Anderson

Department of Electrical Engineering, Columbia University

February 28, 2025

American Control Conference (ACC), 2024

Motivation

Model-free Control

$$x_{t+1} = Ax_t + Bu_t \quad t \geq 0$$

Objective: Design a control sequence $\{u_t\}_t$ that minimizes a cummulative cost $C(x, u) = \mathbf{E}_{x_0} \sum_{t=0}^{\infty} c_t(x_t, u_t)$.

Motivation

Model-free Control

$$x_{t+1} = Ax_t + Bu_t \quad t \geq 0$$

Objective: Design a control sequence $\{u_t\}_t$ that minimizes a cummulative cost $C(x, u) = \mathbf{E}_{x_0} \sum_{t=0}^{\infty} c_t(x_t, u_t)$.

Difficulty: Do not have access to the ground-truth system dynamics (A, B)

Motivation

Model-free Control

$$x_{t+1} = Ax_t + Bu_t \quad t \geq 0$$

Objective: Design a control sequence $\{u_t\}_t$ that minimizes a cummulative cost $C(x,u) = \mathbf{E}_{x_0} \sum_{t=0}^{\infty} c_t(x_t, u_t)$.

Difficulty: Do not have access to the ground-truth system dynamics (A, B)

↓

Policy Gradient (PG)

↓

Collect Data: $\mathcal{D} = \{x_t, u_t\}_t$

↓

Estimation: $\widehat{\nabla}_u C(x,u) = ZO(\mathcal{D})$

↓

Control: PG updates
$\hat{u} \leftarrow \hat{u} - \eta \widehat{\nabla}_u C(x,u)$

↓

Sample Complexity:
$\|\hat{u} - u_\star\| \leq \epsilon$
with # data points $\approx \mathcal{O}(\log(1/\epsilon))$

Mohammadi et al., 2020.

Motivation

Model-free Control

$$x_{t+1} = Ax_t + Bu_t \quad t \geq 0$$

Objective: Design a control sequence $\{u_t\}_t$ that minimizes a cummulative cost $C(x,u) = \mathbf{E}_{x_0} \sum_{t=0}^{\infty} c_t(x_t, u_t)$.

Difficulty: Do not have access to the ground-truth system dynamics (A, B)

Policy Gradient (PG)

Collect Data: $D = \{x_t, u_t\}_t$

Estimation: $\widehat{\nabla}_u C(x,u) = ZO(D)$ → **Rely on cost queries**

Control: PG updates

$\hat{u} \leftarrow \hat{u} - \eta \widehat{\nabla}_u C(x,u)$

Sample Complexity:

$\|\hat{u} - u_\star\| \leq \epsilon$

with # data points $\approx \mathcal{O}(\log(1/\epsilon))$ → **Necessity for cost queries**

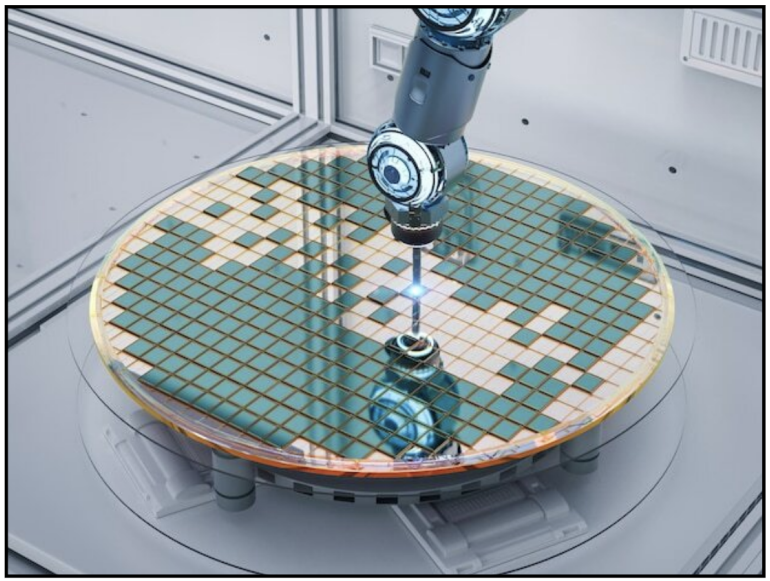
Mohammadi et al., 2020.

Motivation

Cost queries are expensive

What is a cost query?

Precision Robotic Arm



Control: Joint angle, speed

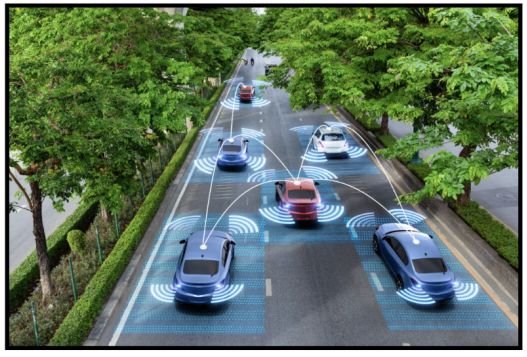
Cost metric:
Positioning error of the silicon wafer

Cost query: Configure the arm and measure the positioning error associated with a specific set of system parameters.

Motivation

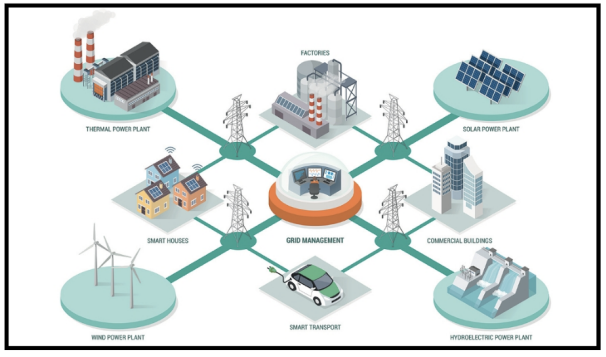
Cost queries are expensive

Autonomous Vehicles



≈ $$$

Smart Grids



≈ $$$$

Industrial Robotics

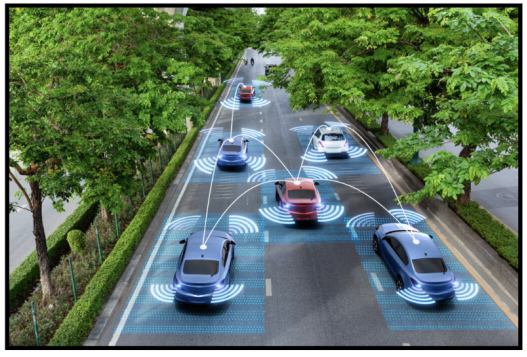
≈ $$$

- Expensive
- Time consuming
- May incur a high risk for human being interaction

Motivation

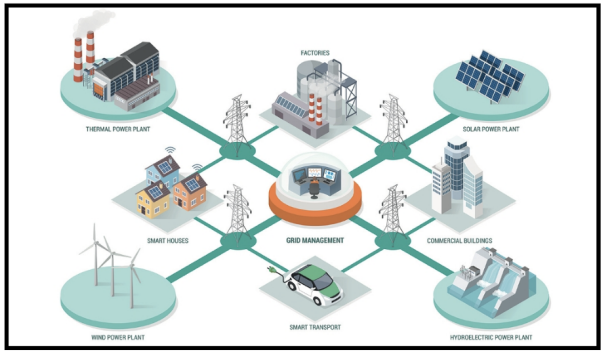
Cost queries are expensive

Autonomous Vehicles



≈ $$$

Smart Grids



≈ $$$$

Industrial Robotics

≈ $$$

We need to reduce the oracle complexity for large scale optimal control via policy gradient methods

Consider the following optimization problem

$$\min_{x \in \mathcal{X}} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

where $f_i(x)$ are convex functions.

Consider the following optimization problem

$$\min_{x \in \mathcal{X}} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

where $f_i(x)$ are convex functions.

Gradient Descent (GD): $x_{k+1} = x_k - \eta \nabla_x f(x_k)$

Stochastic GD: $x_{k+1} = x_k - \eta \nabla_x f_z(x_k)$, where $z \sim \{1, 2, \ldots, n\}$

Consider the following optimization problem

$$\min_{x \in \mathcal{X}} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

where $f_i(x)$ are convex functions.

Gradient Descent (GD): $x_{k+1} = x_k - \eta \nabla_x f(x_k)$

Stochastic GD: $x_{k+1} = x_k - \eta \underbrace{\nabla_x f_z(x_k)}_{\approx \nabla_x f(x_k)?}$, where $z \sim \{1, 2, \ldots, n\}$

Consider the following optimization problem

$$\min_{x \in \mathcal{X}} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

where $f_i(x)$ are convex functions.

Gradient Descent (GD): $x_{k+1} = x_k - \eta \nabla_x f(x_k)$

Stochastic GD: $x_{k+1} = x_k - \eta \underbrace{\nabla_x f_z(x_k)}_{\approx \nabla_x f(x_k)?}$, where $z \sim \{1, 2, \ldots, n\}$

Stochastic Variance-Reduced:

$$x_{k+1} = x_k - \eta(\nabla_x f_z(x_k) \underbrace{- \nabla_x f_z(y) + \nabla_x f(y)}_{\text{zero mean}})$$

where y is generic point.

Johnson and Zhang, 2013, Reddi et al., 2016.

Stochastic Variance-Reduced:

$$x_{k+1} = x_k - \eta \underbrace{\left(\nabla_x f_z(x_k) - \nabla_x f_z(y) + \nabla_x f(y)\right)}_{\text{control variates } v_k}$$

Mean: $\mathbb{E}(v_k) = \mathbb{E}(\nabla_x f_z(x_k))$

Variance: $X = \nabla_x f_z(x_k),\ Y = -\nabla_x f_z(y) + \nabla_x f(y)$

$$\mathbf{var}(v_k) = \mathbf{var}(X) + \mathbf{var}(Y) - 2\mathbf{cov}(X, Y)$$

Johnson and Zhang, 2013, Reddi et al., 2016.

Stochastic Variance-Reduced:

$$x_{k+1} = x_k - \eta \underbrace{(\nabla_x f_z(x_k) - \nabla_x f_z(y) + \nabla_x f(y))}_{\text{control variates } v_k}$$

Mean: $\mathbb{E}(v_k) = \mathbb{E}(\nabla_x f_z(x_k))$

Variance: $X = \nabla_x f_z(x_k),\ Y = -\nabla_x f_z(y) + \nabla_x f(y)$

$$\textbf{var}(v_k) = \textbf{var}(X) + \textbf{var}(Y) - 2\textbf{cov}(X, Y)$$

Question: Can we design an oracle-efficient solution to address the model-free LQR problem by building upon the success of stochastic variance-reduced approaches?

Johnson and Zhang, 2013, Reddi et al., 2016.

Linear Quadratic Regulator (LQR)

Formulation

Consider a discrete LTI dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad t = 0, 1, 2, \ldots \quad \text{(sys-dyn)}$$

where $x_t \in \mathbb{R}^{d_x}$ and $u_t \in \mathbb{R}^{d_u}$.

Linear Quadratic Regulator (LQR)

Formulation

Consider a discrete LTI dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad t = 0, 1, 2, \ldots \quad \text{(sys-dyn)}$$

where $x_t \in \mathbb{R}^{d_x}$ and $u_t \in \mathbb{R}^{d_u}$.

LQR Objective: Design a controller $K^\star$ ($u_t = -K^\star x_t$) that solves

$$K^\star = \text{argmin}_{K \in \mathcal{K}} C(K) = \mathbb{E}\left[\sum_{t=0}^{\infty} x_t \left(Q + K^\top R K\right) x_t\right],$$

subject to (sys-dyn).

Stabilizing set: $\mathcal{K} = \{K \mid \rho(A - BK) < 1\}$.

LQR Objective: Design a controller $K^\star$ ($u_t = -K^\star x_t$) that solves

$$K^\star = \text{argmin}_{K \in \mathcal{K}} C(K) = \mathbb{E}\left[\sum_{t=0}^{\infty} x_t \left(Q + K^\top R K\right) x_t\right],$$

subject to (sys-dyn).

Model-based LQR: Given (A, B, Q, R),

$$K^\star = \texttt{DARE}(A, B, Q, R) \rightarrow \text{Riccati Equation}$$

LQR Objective: Design a controller $K^\star$ ($u_t = -K^\star x_t$) that solves

$$K^\star = \text{argmin}_{K \in \mathcal{K}} C(K) = \mathbb{E}\left[\sum_{t=0}^{\infty} x_t \left(Q + K^\top R K\right) x_t\right],$$

subject to (sys-dyn).

Model-based LQR: Given (A, B, Q, R),

$$K^\star = \texttt{DARE}(A, B, Q, R) \rightarrow \text{Riccati Equation}$$

Q: How to design $K^\star$ when (A, B, Q, R) is unknown?

Linear Quadratic Regulator (LQR)

Policy Gradient LQR

Fazel et al., ICML 2018, proved that despite of the non-convexity* of $C(K)$, PG methods globally converge to $K^\star$, i.e., given

Initial Stabilizing Controller: $K_0 \in \mathcal{K}$

Controllability: (A, B) is controllable

* **Non-convex for** $d_X \geq 3$.

Fazel et al., ICML 2018, proved that despite of the non-convexity* of $C(K)$, PG methods globally converge to $K^\star$, i.e., given

Initial Stabilizing Controller: $K_0 \in \mathcal{K}$

Controllability: (A, B) is controllable

$$K_{n+1} = K_n - \eta \widehat{\nabla} C(K_n), \text{ for } n \in \{0, 1, \ldots, N-1\}$$
$$\downarrow$$
$$C(K_N) - C(K^\star) \leq \epsilon,$$

after $N \geq \mathcal{O}(\log(1/\epsilon))$.

Gradient Dominance: $C(K) - C(K^\star) \leq \lambda \|\nabla C(K)\|_F^2$ for any $K \in \mathcal{K}$.

* **Non-convex for $d_x \geq 3$.**

Other nice properties of the LQR cost:

- Uniform bounds for the gradient
- Lipschitz of the cost
- Lipschitz of the gradient
- Bounded controller difference

Fazel et al. 2018, Bu et al., 2019, Gravell et al. 2020.

Other nice properties of the LQR cost:

- Uniform bounds for the gradient
- Lipschitz of the cost
- Lipschitz of the gradient
- Bounded controller difference

Stabilizing sub-level set: Given (sys-dyn), the stabilizing sub-level set $\mathcal{G} \subseteq \mathcal{K}$ is

$$\mathcal{G} = \{K \mid C(K) - C(K^\star) \leq \gamma(C(K_0) - C(K^\star))\},$$

for some $\gamma > 0$.

Fazel et al. 2018, Bu et al., 2019, Gravell et al. 2020.

$$K_{n+1} = K_n - \eta \widehat{\nabla} C(K_n), \text{ for } n \in \{0, 1, \ldots, N-1\}$$

$$K_{n+1} = K_n - \eta \widehat{\nabla} C(K_n), \text{ for } n \in \{0, 1, \ldots, N-1\}$$

One-point Zeroth-order Estimation (`ZO1P`):

$$(m, r) \to \texttt{ZO1P} : \overline{\nabla} C(K) := \sum_{i=1}^{m} \frac{d_x d_u C(K + U_i) U_i}{m r^2},$$

m (number of trajectories), r (smoothing radius) and $\|U_i\|_F = r$.

$$K_{n+1} = K_n - \eta \widehat{\nabla} C(K_n), \text{ for } n \in \{0, 1, \ldots, N-1\}$$

One-point Zeroth-order Estimation (`ZO1P`):

$$(m, r) \to \texttt{ZO1P} : \overline{\nabla} C(K) := \sum_{i=1}^{m} \frac{d_x d_u C(K + U_i) U_i}{m r^2},$$

m (number of trajectories), r (smoothing radius) and $\|U_i\|_F = r$.

$$K_{n+1} = K_n - \eta \widehat{\nabla} C(K_n), \text{ for } n \in \{0, 1, \ldots, N-1\}$$

One-point Zeroth-order Estimation (`ZO1P`):

$$(m, r) \to \texttt{ZO1P} : \overline{\nabla} C(K) := \sum_{i=1}^{m} \frac{d_x d_u C(K + U_i) U_i}{m r^2},$$

m (number of trajectories), r (smoothing radius) and $\|U_i\|_F = r$.

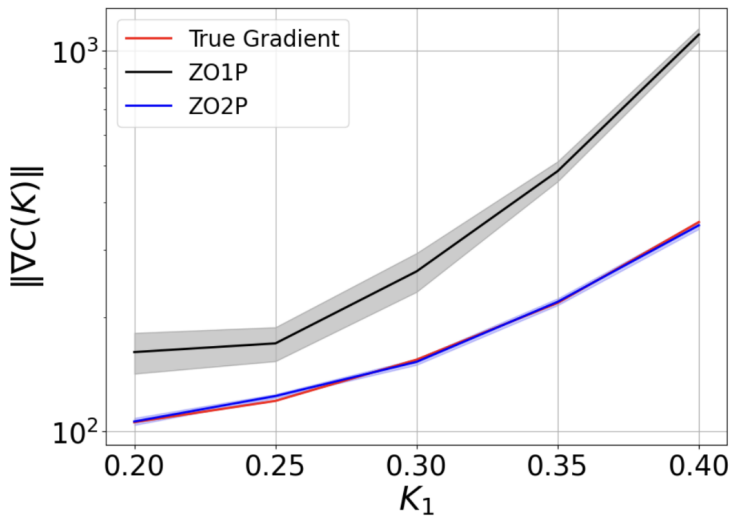
Two-points Zeroth-order Estimation (`ZO2P`):

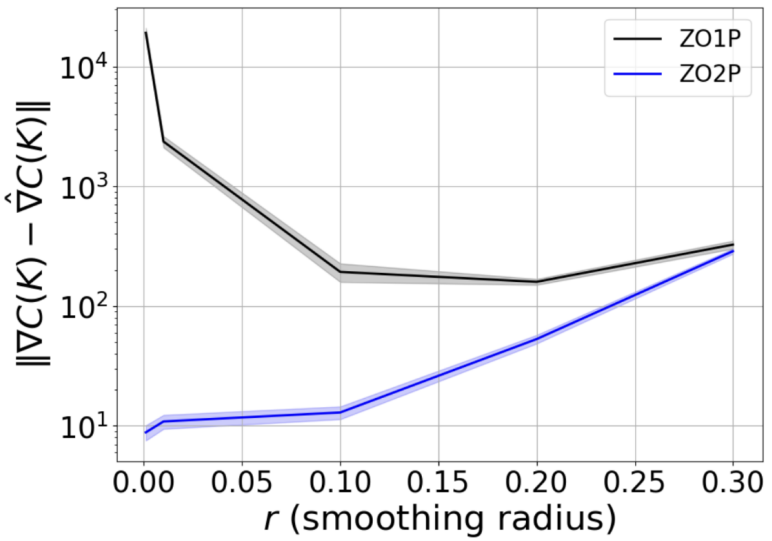
$$(m, r) \to \texttt{ZO2P} : \widetilde{\nabla} C(K) := \sum_{i=1}^{m} \frac{d_x d_u (C(K + U_i) - C(K - U_i)) U_i}{2 m r^2},$$

Policy Gradient LQR

Illustrative Example - ZO Bias and Variance

$$A = \begin{bmatrix} 1.20 & 0.50 & 0.40 \\ 0.01 & 0.75 & 0.30 \\ 0.10 & 0.02 & 1.50 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{bmatrix}, Q = 2I_3, R = \frac{1}{2},$$





$$K = [K_1, K_2, K_3]$$
$$K_2 = -0.45, K_3 = 3.8$$

	Variance	Bias
ZO1P	$\mathcal{O}(1/r^2)$	$\mathcal{O}(r^2)$
ZO2P	$\mathcal{O}(1)$	$\mathcal{O}(r^2)$

Policy Gradient LQR

Improvements on the Oracle Complexity

Comparison on the sample complexity ($\mathbb{S}_c$), and two-point oracle complexity ($\mathcal{N}_{\text{ZO2P}}$) required to achieve $\mathbb{E}\left(C(K_N) - C(K^\star)\right) \leq \epsilon$.

Methods	$\mathbb{S}_c$	$\mathcal{N}_{\text{ZO2P}}$
PG - ZO1P (Fazel et al (2018))	$\mathcal{O}(1/\epsilon^4 \cdot \log(1/\epsilon))$	-
PG - ZO1P (Gravell et al (2019)	$\mathcal{O}(1/\epsilon^4 \cdot \log(1/\epsilon))$	-
PG - ZO1P (Malik et al. (2019)	$\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))$	-
PG - ZO2P (Malik et al. (2019)	$\mathcal{O}(1/\epsilon \cdot \log(1/\epsilon))$	$\mathcal{O}(1/\epsilon \cdot \log(1/\epsilon))$
PG - ZO2P (Mohammadi et al. (2020))	$\mathcal{O}(\log(1/\epsilon))$	$\mathcal{O}(\log(1/\epsilon))$

Policy Gradient LQR

Improvements on the Oracle Complexity

Comparison on the sample complexity ($\mathbb{S}_c$), and two-point oracle complexity ($\mathcal{N}_{\text{ZO2P}}$) required to achieve $\mathbb{E}\left(C(K_N) - C(K^\star)\right) \leq \epsilon$.

Methods	$\mathbb{S}_c$	$\mathcal{N}_{\text{ZO2P}}$
PG - ZO1P (Fazel et al (2018))	$\mathcal{O}(1/\epsilon^4 \cdot \log(1/\epsilon))$	-
PG - ZO1P (Gravell et al (2019)	$\mathcal{O}(1/\epsilon^4 \cdot \log(1/\epsilon))$	-
PG - ZO1P (Malik et al. (2019)	$\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))$	-
PG - ZO2P (Malik et al. (2019)	$\mathcal{O}(1/\epsilon \cdot \log(1/\epsilon))$	$\mathcal{O}(1/\epsilon \cdot \log(1/\epsilon))$
PG - ZO2P (Mohammadi et al. (2020))	$\mathcal{O}(\log(1/\epsilon))$	$\mathcal{O}(\log(1/\epsilon))$

Question: Can we harness synergy between `ZO1P` (i.e., cheap cost queries) and `ZO2P` (i.e., low variance) to reduce two-point oracle complexity for the model-free LQR problem?

Stochastic Variance-Reduced Policy Gradient (SVRPG)

Difficulties

SVRPG and Reinforcement Learning: Xu et al., 2020, demonstrate a sample complexity reduction from $\mathcal{O}(\epsilon^2)$ to $\mathcal{O}(\epsilon^{5/3})$ in the local convergence analysis, i.e., $\|\nabla C(K_N)\|_F^2 \leq \epsilon$

They assume: $\mathbb{E}(\widehat{\nabla} C(K)) = \nabla C(K)$,

Papini et al., 2018.

Stochastic Variance-Reduced Policy Gradient (SVRPG)

Difficulties

SVRPG and Reinforcement Learning: Xu et al., 2020, demonstrate a sample complexity reduction from $\mathcal{O}(\epsilon^2)$ to $\mathcal{O}(\epsilon^{5/3})$ in the local convergence analysis, i.e., $\|\nabla C(K_N)\|_F^2 \leq \epsilon$

They assume: $\mathbb{E}(\widehat{\nabla} C(K)) = \nabla C(K)$,

Difficulties of SVRPG for the model-free LQR:

- K_n needs to stay stabilizing for all iterations
- ZO gradient estimation is biased, i.e., $\mathcal{O}(r^2)$
- ZO2P estimation is cost-query expensive
- ZO1P estimation has a large variance

Papini et al., 2018.

Stochastic Variance-Reduced Policy Gradient (SVRPG)

Difficulties

SVRPG and Reinforcement Learning: Xu et al., 2020, demonstrate a sample complexity reduction from $\mathcal{O}(\epsilon^2)$ to $\mathcal{O}(\epsilon^{5/3})$ in the local convergence analysis, i.e., $\|\nabla C(K_N)\|_F^2 \leq \epsilon$

They assume: $\mathbb{E}(\widehat{\nabla} C(K)) = \nabla C(K)$,

Difficulties of SVRPG for the model-free LQR:

- K_n needs to stay stabilizing for all iterations
- ZO gradient estimation is biased, i.e., $\mathcal{O}(r^2)$
- ZO2P estimation is cost-query expensive
- ZO1P estimation has a large variance

We are the first to combine ZO1P and ZO2P in a SVRPG approach

Papini et al., 2018.

Initialization: $N,\ T,\ \eta,\ n_{\text{out}},\ n_{\text{in}},\ n_{\text{out}},\ r_{\text{in}},\ K_0 \in \mathcal{G}$

- N epochs with length T
- step-size η
- $(n_{\text{out}}, r_{\text{out}})$ ZO2P's parameters
- $(n_{\text{in}}, r_{\text{in}})$ ZO1P's parameters
- K_0 initial stabilizing controller

An SVRPG Approach for the model-free LQR

A Mixed ZO1P and ZO2P Estimation with Control Variates

Initialization: $N,\ T,\ \eta,\ n_{\text{out}},\ n_{\text{in}},\ r_{\text{out}},\ r_{\text{in}},\ K_0 \in \mathcal{G}$

Step 1: For all $n \in \{0, 1, \ldots, N-1\}$ set $K_0^{n+1} = \widetilde{K}^n = K_T^n$ and compute

$$\tilde{\mu} = \widetilde{\nabla} C(\tilde{K}^n) \text{ with } (n_{\text{out}}, r_{\text{out}}) \rightarrow \quad \texttt{ZO2P},$$

Step 2: Within each epoch repeat for $t \in \{0, 1, \ldots, T-1\}$

$$\overline{\nabla} C(K_t^{n+1}), \overline{\nabla} C(\tilde{K}^n) \text{ with } (n_{\text{in}}, r_{\text{in}}) \rightarrow \quad \texttt{ZO1P},$$

$$K_{t+1}^{n+1} = K_t^{n+1} - \eta \left(\overline{\nabla} C(K_t^{n+1}) + \tilde{\mu} - \overline{\nabla} C(\tilde{K}^n) \right),$$

Repeat steps 1 and 2 and return $K_{NT} = K_T^N$.

Step 1: For all $n \in \{0, 1, \ldots, N-1\}$ set $K_0^{n+1} = \widetilde{K}^n = K_T^n$ and compute

$$\tilde{\mu} = \widetilde{\nabla} C(\tilde{K}^n) \text{ with } (n_{\text{out}}, r_{\text{out}}) \rightarrow \quad \texttt{Z02P},$$

Step 2: Within each epoch repeat for $t \in \{0, 1, \ldots, T-1\}$

$$\overline{\nabla} C(K_t^{n+1}), \overline{\nabla} C(\tilde{K}^n) \text{ with } (n_{\text{in}}, r_{\text{in}}) \rightarrow \quad \texttt{Z01P},$$
$$K_{t+1}^{n+1} = K_t^{n+1} - \eta \left(\overline{\nabla} C(K_t^{n+1}) + \tilde{\mu} - \overline{\nabla} C(\tilde{K}^n) \right),$$

Step 1: For all $n \in \{0, 1, \ldots, N-1\}$ set $K_0^{n+1} = \widetilde{K}^n = K_T^n$ and compute

$$\tilde{\mu} = \widetilde{\nabla} C(\tilde{K}^n) \text{ with } (n_{\text{out}}, r_{\text{out}}) \rightarrow \quad \text{ZO2P},$$

Step 2: Within each epoch repeat for $t \in \{0, 1, \ldots, T-1\}$

$$\overline{\nabla} C(K_t^{n+1}), \overline{\nabla} C(\tilde{K}^n) \text{ with } (n_{\text{in}}, r_{\text{in}}) \rightarrow \quad \text{ZO1P},$$
$$K_{t+1}^{n+1} = K_t^{n+1} - \eta \left(\overline{\nabla} C(K_t^{n+1}) + \tilde{\mu} - \overline{\nabla} C(\tilde{K}^n) \right),$$

Idea: Use **ZO2P less often** and control the inner loop gradient variance with **more ZO1P** + control variates.

Convergence: Given $K_0 \in \mathcal{G}$. Suppose that $r_{\text{in}} = r_{\text{out}} = r$,

$$n_{\text{out}} \geq \mathcal{O}(1), \quad n_{\text{in}} \geq \mathcal{O}(T^2), \text{ and } \eta \text{ sufficiently small,}$$

then it holds that

$$\mathbb{E}\left(C(K_{\text{NT}}) - C(K^\star)\right) \leq \Delta_0 \rho^{NT} + \frac{C_{\text{bias}} r^2}{n_{\text{in}}}$$

where $\rho \in (0, 1)$ and $\Delta_0 = C(K_0) - C(K^\star)$.

C_{bias} is provided in the full paper.

Main Results

Convergence Guarantees

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \Delta_0 \rho^{NT} + \frac{C_{\mathsf{bias}} r^2}{n_{\mathsf{in}}}$$

C_{bias} **is provided in the full paper.**

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \Delta_0 \rho^{NT} + \frac{C_{\mathsf{bias}} r^2}{n_{\mathsf{in}}}$$

By selecting $NT \geq \mathcal{O}(\log(1/\epsilon))$ and $r \leq \mathcal{O}(\sqrt{n_{\mathsf{in}}\epsilon})$ we have

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \epsilon$$

for some small tolerance ϵ.

C_{bias} is provided in the full paper.

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \Delta_0 \rho^{NT} + \frac{C_{\mathsf{bias}} r^2}{n_{\mathsf{in}}}$$

By selecting $NT \geq \mathcal{O}(\log(1/\epsilon))$ and $r \leq \mathcal{O}(\sqrt{n_{\mathsf{in}}\epsilon})$ we have

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \epsilon$$

for some small tolerance ϵ.

Sample complexity: $\mathcal{S}_c = 2Nn_{\mathsf{out}} + NTn_{\mathsf{in}}$

C_{bias} **is provided in the full paper.**

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \Delta_0 \rho^{NT} + \frac{C_{\mathsf{bias}} r^2}{n_{\mathsf{in}}}$$

By selecting $NT \geq \mathcal{O}(\log(1/\epsilon))$ and $r \leq \mathcal{O}(\sqrt{n_{\mathsf{in}}\epsilon})$ we have

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \epsilon$$

for some small tolerance ϵ.

Sample complexity: $\mathcal{S}_c = 2Nn_{\mathsf{out}} + NTn_{\mathsf{in}}$

$$NT \geq \mathcal{O}(\log(1/\epsilon)) \rightarrow N = \mathcal{O}(\log(1/\epsilon))^{\beta}, \quad T = \mathcal{O}(\log(1/\epsilon))^{1-\beta}$$
$$n_{\mathsf{out}} = \mathcal{O}(1), \quad n_{\mathsf{in}} = \mathcal{O}(T^2) = \mathcal{O}(\log(1/\epsilon))^{2-2\beta}, \quad \beta \in (0,1),$$

C_{bias} **is provided in the full paper.**

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \Delta_0 \rho^{NT} + \frac{C_{\mathsf{bias}} r^2}{n_{\mathsf{in}}}$$

By selecting $NT \geq \mathcal{O}(\log(1/\epsilon))$ and $r \leq \mathcal{O}(\sqrt{n_{\mathsf{in}}\epsilon})$ we have

$$\mathbb{E}\left(C(K_{\mathsf{NT}}) - C(K^\star)\right) \leq \epsilon$$

for some small tolerance ϵ.

Sample complexity: $\mathcal{S}_c = 2Nn_{\mathsf{out}} + NTn_{\mathsf{in}} = \mathcal{O}(\log(1/\epsilon))^{3-2\beta}$

$$NT \geq \mathcal{O}(\log(1/\epsilon)) \to N = \mathcal{O}(\log(1/\epsilon))^{\beta}, \quad T = \mathcal{O}(\log(1/\epsilon))^{1-\beta}$$
$$n_{\mathsf{out}} = \mathcal{O}(1), \quad n_{\mathsf{in}} = \mathcal{O}(T^2) = \mathcal{O}(\log(1/\epsilon))^{2-2\beta}, \quad \beta \in (0,1),$$

Two-point oracle complexity: $\mathcal{N}_{ZO2P} = 2Nn_{\mathsf{out}} = \mathcal{O}(\log(1/\epsilon))^{\beta}$

Stability: Given $K_0 \in \mathcal{G}$. Suppose that

$$n_{\text{out}}, n_{\text{in}} \text{ sufficiently large, } \textbf{and } r, \eta \text{ sufficiently small}$$

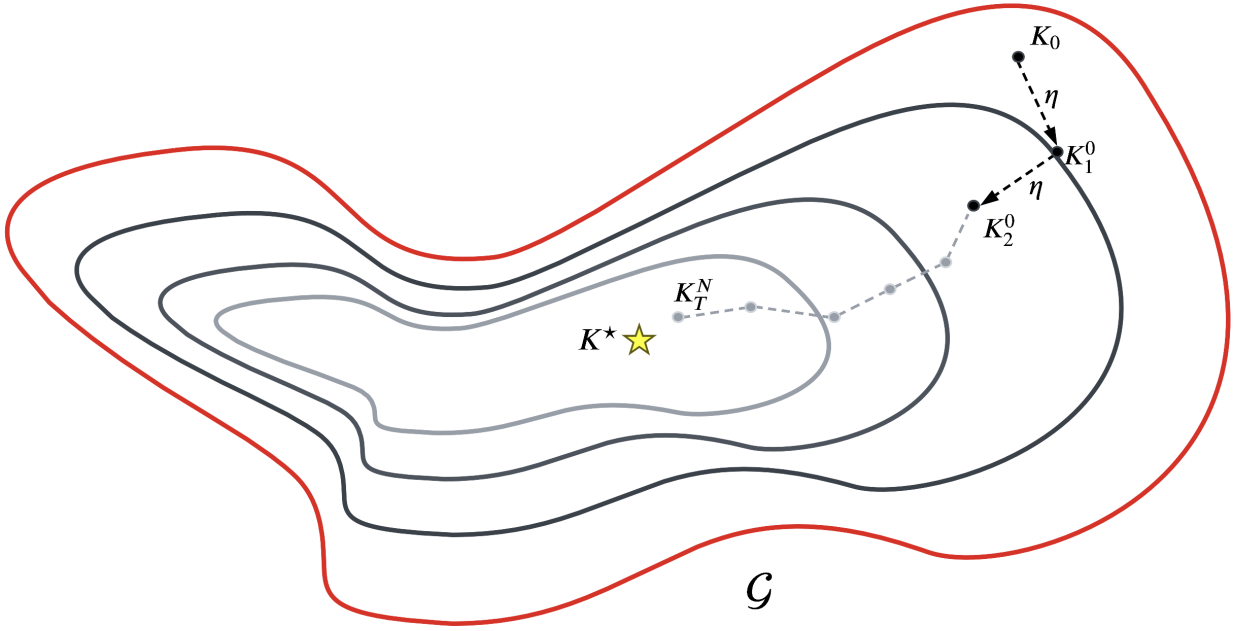
then $K_t^n \in \mathcal{G}$, with high probability, $\forall$ epochs $n \in [N]$ of length $t \in [T]$.

Stability: Given $K_0 \in \mathcal{G}$. Suppose that

$$n_{\text{out}}, n_{\text{in}} \text{ sufficiently large, } \textbf{and } r, \eta \text{ sufficiently small}$$

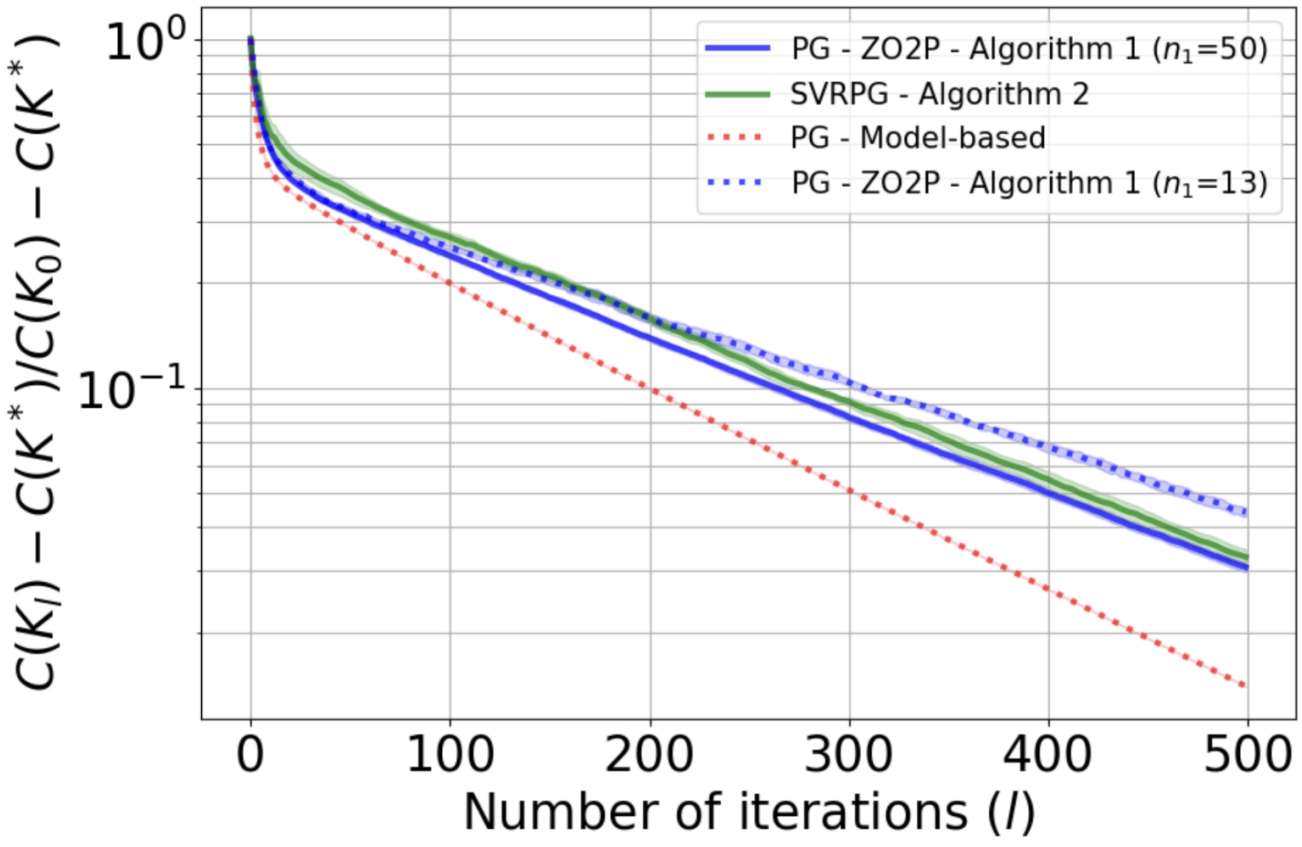
then $K_t^n \in \mathcal{G}$, with high probability, $\forall$ epochs $n \in [N]$ of length $t \in [T]$.

Main takeaway: By carefully controlling the quality of the inner and outer gradient estimations and not taking larger steps the learned controller provably stays within the stabilizing sub-level set.



Numerical Validation

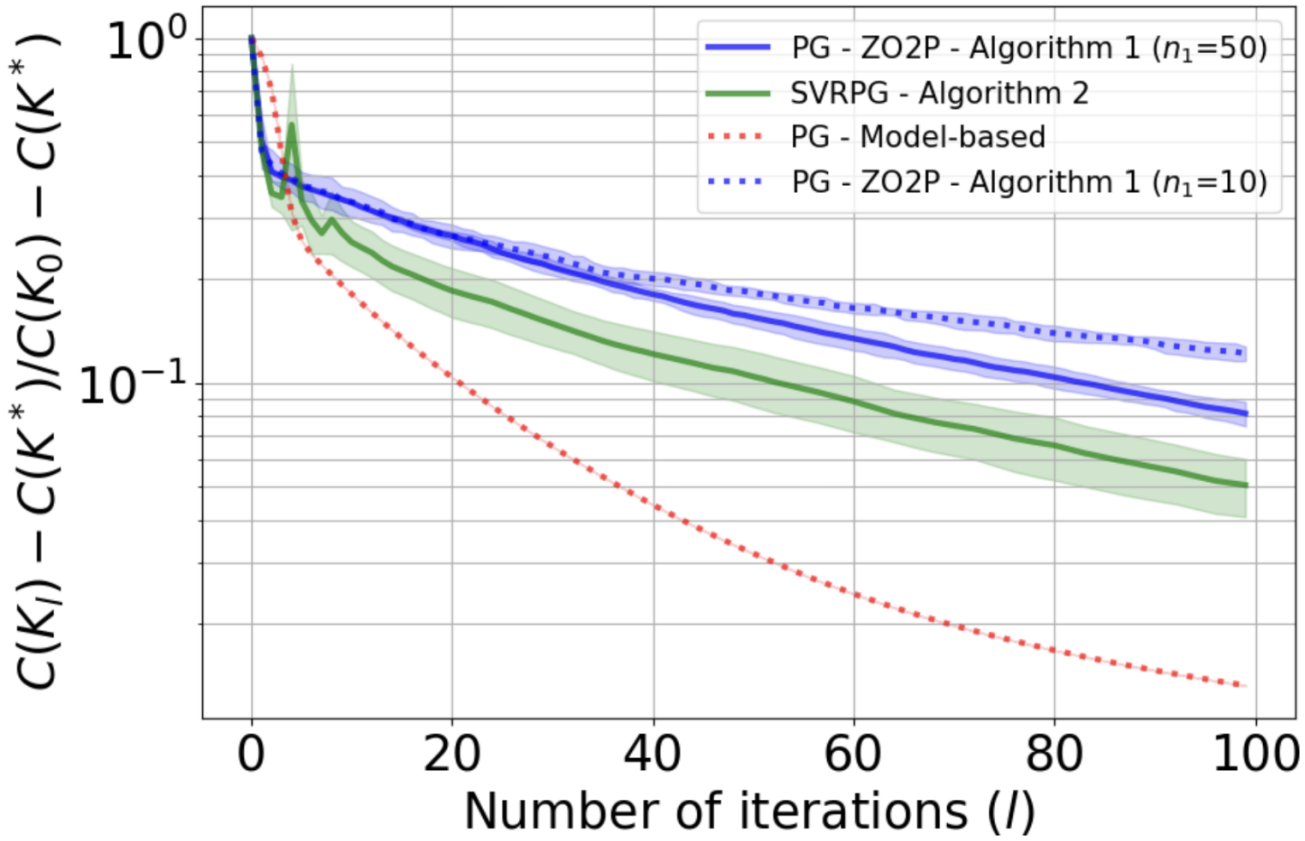
Example 1 - $n_x = 3,\ n_u = 1$



Total # cost queries: ZO1P + ZO2P		**# Two cost queries:** ZO2P	
Algorithm 1 (PG - ZO2P)	**Algorithm 2 (SVRPG)**	**Algorithm 1 (PG - ZO2P)**	**Algorithm 2 (SVRPG)**
50000	37500	25000	6250

Numerical Validation

Example 2 - $n_x = 4$, $n_u = 2$



Total # cost queries: ZO1P + ZO2P		**# Two cost queries:** ZO2P	
Algorithm 1 (PG - ZO2P)	**Algorithm 2 (SVRPG)**	**Algorithm 1 (PG - ZO2P)**	**Algorithm 2 (SVRPG)**
10000	5000	5000	1000

Conclusions

- We propose a stochastic variance-reduced policy gradient approach for the model-free LQR problem.
- Our approach combines the benefits of one-point ZO estimation (i.e., cheap in cost queries) and two-point ZO estimation (i.e., lower variance) with the help of a mixed SVRPG approach.
- We prove that our approach achieves an ϵ-approximate solution with $\mathcal{O}\left(\log\left(1/\epsilon\right)^{3-2\beta}\right)$ queries, with only $\mathcal{O}\left(\log\left(1/\epsilon\right)^{\beta}\right)$ two-point query information for $\beta \in (0, 1)$.
- We prove that (sys-dyn) is stable under the learned controller.

Collaborators

Han Wang

James Anderson

Funding

NSF

COLUMBIA UNIVERSITY
DATA SCIENCE INSTITUTE

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

To find out more:

My website

Full paper

Happy to take questions!