Escaping High-order Saddles in Policy Optimization for Linear Quadratic Gaussian (LQG) Control

Yang Zheng*[1], Yue Sun*[2], Maryam Fazel[2], and Na Li[3]

(*equal contribution)

[1] ECE Department, UC San Diego
[2] ECE Department, University of Washington
[3] School of Engineering and Applied Science, Harvard University

61st IEEE Conference on Decision and Control, Cancún, Mexico
December 8, 2022

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

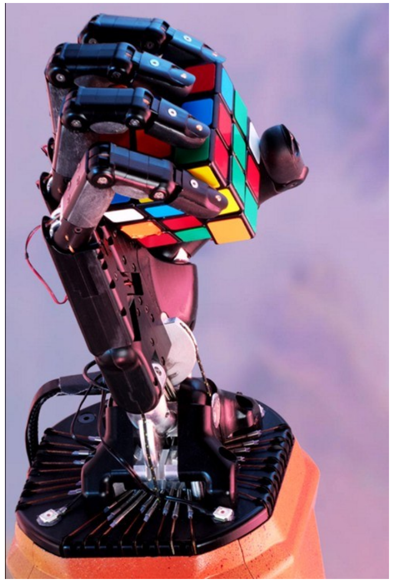
Harvard John A. Paulson School of Engineering and Applied Sciences

Motivation

❑ Model-free methods and data-driven control

- Use direct policy updates
- Become very popular in both academia and practice, from game playing, robotics, and drones, etc.

DeepMind



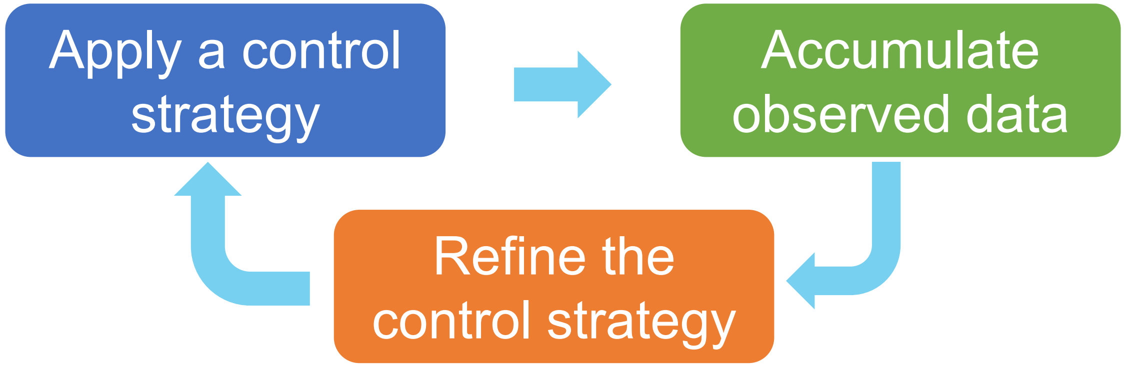
OpenAI

Applications

Duan et al. 2016; Silver et al., 2017; Dean et al., 2019; Tu and Recht, 2019;
Mania et al., 2019; Fazel et al., 2018; Recht, 2019;

Motivation

❑ Model-free methods and data-driven control



Opportunities

- Directly search over a given policy class
- Directly optimize performance on the true system, bypassing the model estimation

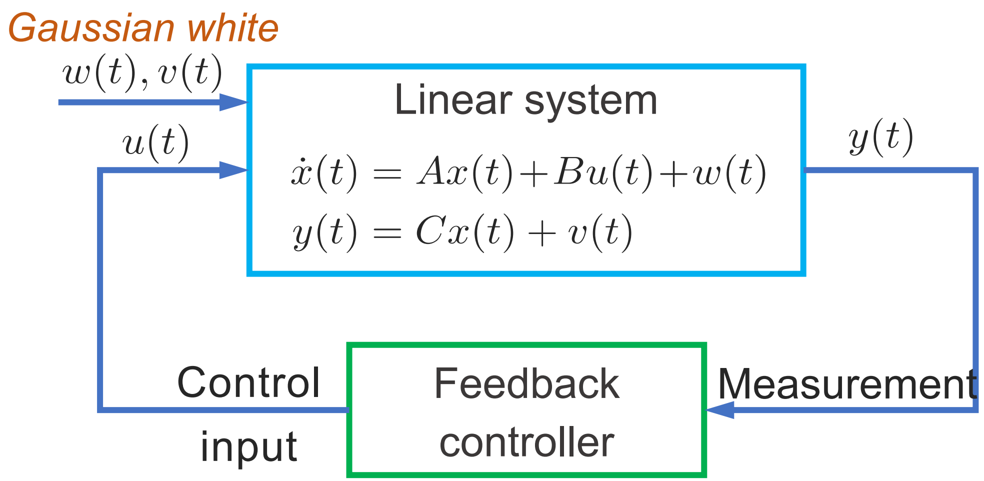
Challenges

- Lack of non-asymptotic performance guarantees
  - ➢ Convergence
  - ➢ Suboptimality
  - ➢ Sample complexity, etc.

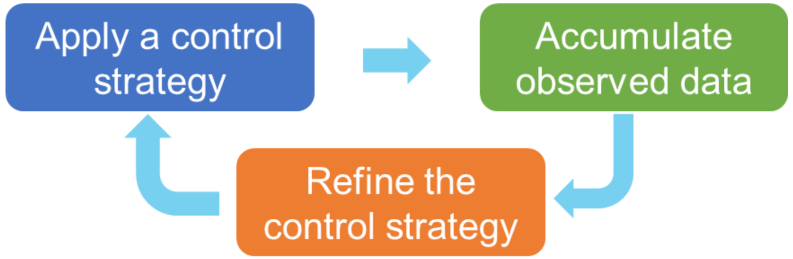
❖ Highly nontrivial even for **linear dynamical systems**

Today's talk

❑ Linear Quadratic Gaussian Control



Direct policy iteration $\mathsf{K}_{i+1} = \mathsf{K}_i - \alpha_i \nabla J(\mathsf{K}_i)$



- ❑ A classical control problem, rich theory in classical control (model-based)
- ❑ Many practical applications
- ❑ Allows **partial observation** of the state
  - Perfect state observation is often not available

LQG as a non-convex optimization problem

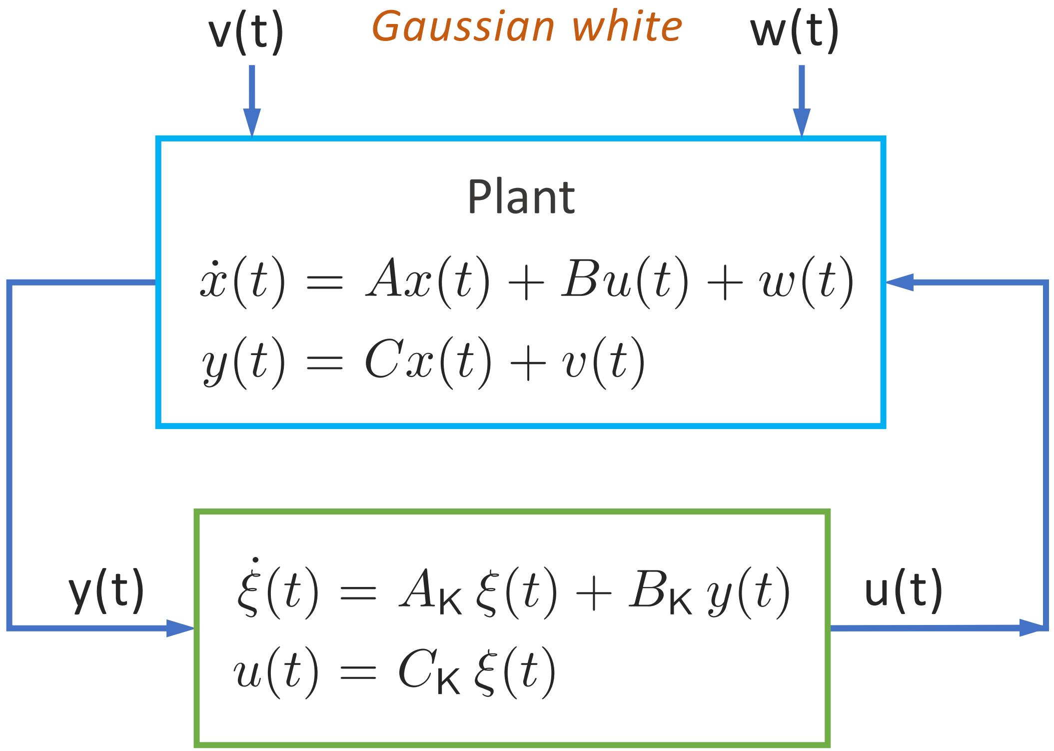
$$\min_{\mathsf{K}} \quad J(\mathsf{K})$$
$$\text{s.t.} \quad \mathsf{K} = (A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}}) \in \mathcal{C}_{\text{full}}$$

- ✓ **Does it converge at all?**
- ✓ **Converge to which point?**
- ✓ **How to escape saddle points?**

Outline

- LQG problem setup
- Stationary points and strict saddle points
- Escaping saddle points via perturbations
- Conclusions

LQG Problem Setup



dynamical controller

$\mathsf{K} = (A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}})$

Standard Assumption	$(A, B),\ (A, W^{1/2})$	Controllable
	$(C, A),\ (Q^{1/2}, A)$	Observable

Objective: The LQG cost

$$\lim_{T \to +\infty} \frac{1}{T} \mathbb{E} \int_0^T (x^\top Q x + u^\top R u)\, dt$$

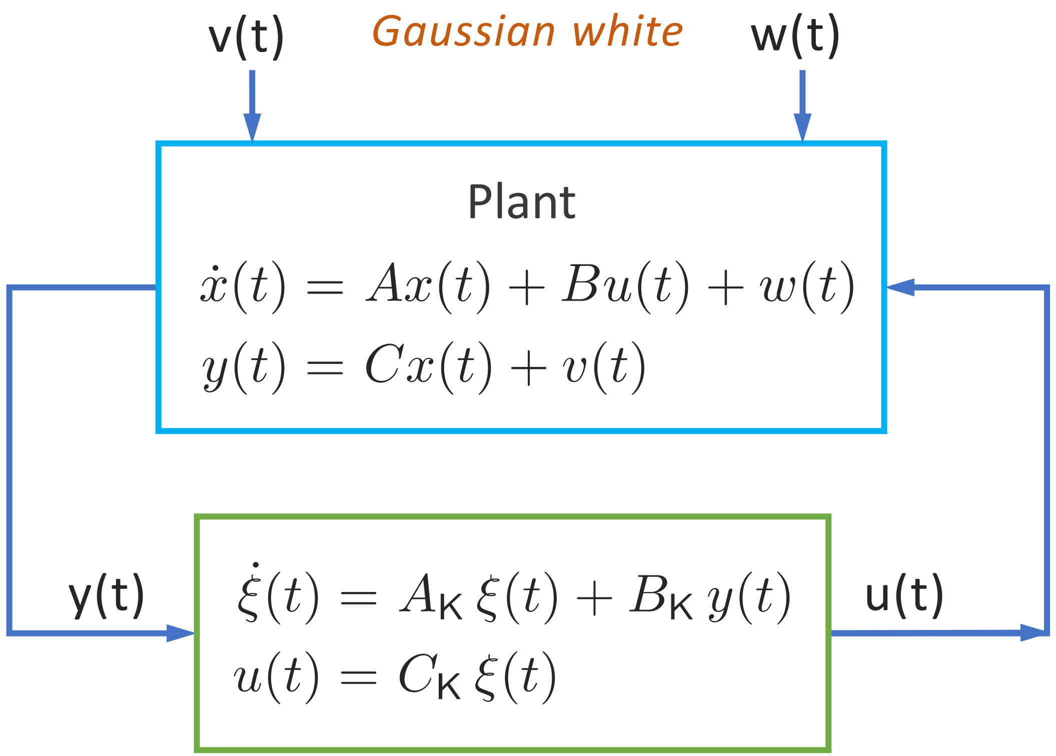
- $\xi(t)$ internal state of the controller
- $\dim \xi(t)$ order of the controller
- $\dim \xi(t) = \dim x(t)$ full-order
- $\dim \xi(t) < \dim x(t)$ reduced-order

Minimal controller

The input-output behavior cannot be replicated by a lower order controller.

* $(A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}})$ controllable and observable

Separation principle



dynamical controller

$$\mathsf{K} = (A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}})$$

Explicit dependence on the dynamics

Objective: The LQG cost

$$\lim_{T\to+\infty} \frac{1}{T}\mathbb{E}\int_0^T (x^\top Q x + u^\top R u)\, dt$$

Solution: Kalman filter for state estimation
+ LQR based on the estimated state

$$\dot{\xi} = (A - BK)\xi + L(y - C\xi),$$
$$u = -K\xi.$$

Two Riccati equations

- **Kalman gain** $L = PC^{\mathsf{T}}V^{-1}$

$$AP + PA^{\mathsf{T}} - PC^{\mathsf{T}}V^{-1}CP + W = 0,$$

- **Feedback gain** $K = R^{-1}B^{\mathsf{T}}S$

$$A^{\mathsf{T}}S + SA - SBR^{-1}B^{\mathsf{T}}S + Q = 0$$

Model-free Optimization formulation

❑ Closed-loop dynamics

$$\frac{d}{dt}\begin{bmatrix} x \\ \xi \end{bmatrix} = \begin{bmatrix} A & BC_{\mathsf{K}} \\ B_{\mathsf{K}}C & A_{\mathsf{K}} \end{bmatrix}\begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & B_{\mathsf{K}} \end{bmatrix}\begin{bmatrix} w \\ v \end{bmatrix},$$

$$\begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C_{\mathsf{K}} \end{bmatrix}\begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} v \\ 0 \end{bmatrix}.$$

❑ Feasible region of the controller parameters

$$\mathcal{C}_{\text{full}} = \left\{\mathsf{K} \,|\mathsf{K} = (A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}}) \text{ is full order} \right.$$
$$\left. \begin{bmatrix} A & BC_{\mathsf{K}} \\ B_{\mathsf{K}}C & A_{\mathsf{K}} \end{bmatrix} \text{ is Hurwitz stable}\right\}$$

❑ Cost function $\lim_{T\to+\infty}\frac{1}{T}\mathbb{E}\int_0^T (x^\top Q x + u^\top R u)\,dt$

$$J(\mathsf{K}) = \operatorname{tr}\left(\begin{bmatrix} Q & 0 \\ 0 & C_{\mathsf{K}}^{\mathsf{T}} R C_{\mathsf{K}} \end{bmatrix} X_{\mathsf{K}}\right) = \operatorname{tr}\left(\begin{bmatrix} W & 0 \\ 0 & B_{\mathsf{K}} V B_{\mathsf{K}}^{\mathsf{T}} \end{bmatrix} Y_{\mathsf{K}}\right)$$

$X_{\mathsf{K}}, Y_{\mathsf{K}}$ Solution to Lyapunov equations

LQG as a non-convex optimization problem

$$\min_{\mathsf{K}} \quad J(\mathsf{K})$$
$$\text{s.t.} \quad \mathsf{K} = (A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}}) \in \mathcal{C}_{\text{full}}$$

Direct policy iteration $\mathsf{K}_{i+1} = \mathsf{K}_i - \alpha_i \nabla J(\mathsf{K}_i)$

$$\nabla J(\mathsf{K}_i) = 0$$

Q1: Structure of stationary points

Q2: How to escape saddle points

Outline

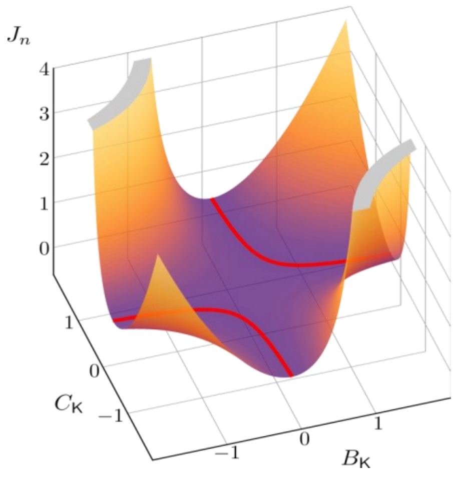
- ❑ LQG problem setup
- ❑ **Stationary points and strict saddle points**
- ❑ Escaping saddle points via perturbations
- ❑ Conclusions

Structure of stationary points

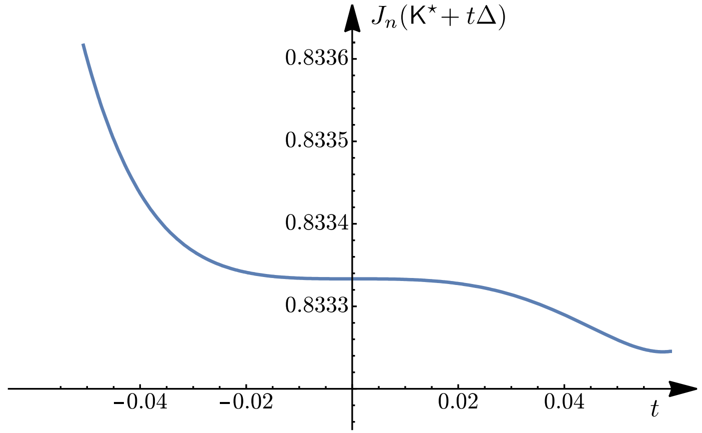
How does the set of Stationary Points look like?

$$\left\{ \mathsf{K} \in \mathcal{C}_{\text{full}} \;\middle|\; \frac{\partial J(\mathsf{K})}{\partial A_{\mathsf{K}}} = 0, \; \frac{\partial J(\mathsf{K})}{\partial B_{\mathsf{K}}} = 0, \; \frac{\partial J(\mathsf{K})}{\partial C_{\mathsf{K}}} = 0, \right\}$$

- ❑ Non-unique, non-isolated



- ❑ Existence of high-order saddle points



Good news: All stationary points corresponding to controllable and observable controllers are globally optimum (Tang, Zheng, Li, 2021).

❑ **Theorem 1 (informal): all bad stationary points are in the same form**

If $\mathsf{K} = \begin{bmatrix} 0 & C_{\mathsf{K}} \\ B_{\mathsf{K}} & A_{\mathsf{K}} \end{bmatrix} \in \mathcal{C}_n$ is a stationary point but not minimal, then $\tilde{\mathsf{K}} = \left[\begin{array}{c|cc} 0 & \hat{C}_{\mathsf{K}} & 0 \\ \hline \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} & 0 \\ 0 & 0 & \Lambda \end{array}\right] \in \mathcal{C}_n$ is also a stationary point with the same LQG cost, where $\hat{\mathsf{K}} = \begin{bmatrix} 0 & \hat{C}_{\mathsf{K}} \\ \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} \end{bmatrix} \in \mathcal{C}_q$ is a minimal realization

Structure of stationary points

❑ **Theorem 1 (informal): all bad stationary points are in the same form**

$$\left\{ \mathsf{K} \in \mathcal{C}_{\text{full}} \;\middle|\; \frac{\partial J(\mathsf{K})}{\partial A_{\mathsf{K}}} = 0,\ \frac{\partial J(\mathsf{K})}{\partial B_{\mathsf{K}}} = 0,\ \frac{\partial J(\mathsf{K})}{\partial C_{\mathsf{K}}} = 0, \right\}$$

a stationary point

$$\mathsf{K} = \begin{bmatrix} 0 & C_{\mathsf{K}} \\ B_{\mathsf{K}} & A_{\mathsf{K}} \end{bmatrix} \in \mathcal{C}_n$$

If it is minimal, then it is globally optimal

If it is not minimal, find a minimal realization

$$\hat{\mathsf{K}} = \begin{bmatrix} 0 & \hat{C}_{\mathsf{K}} \\ \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} \end{bmatrix} \in \mathcal{C}_q$$

Proof idea: K and $\tilde{\mathsf{K}}$ corresponds to the same transfer function in the frequency domain

$$C_{\mathsf{K}}(sI - A_{\mathsf{K}})^{-1}B_{\mathsf{K}} = \hat{C}_{\mathsf{K}}(sI - \hat{A}_{\mathsf{K}})^{-1}\hat{B}_{\mathsf{K}}$$

Thus, they have the same LQG cost, and the new controller remains a stationary point

The following full-order controller with any stable Λ is also a stationary point with the same LQG cost

$$\tilde{\mathsf{K}} = \left[\begin{array}{c|cc} 0 & \hat{C}_{\mathsf{K}} & 0 \\ \hline \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} & 0 \\ 0 & 0 & \Lambda \end{array}\right] \in \mathcal{C}_n$$

Strict saddle points

$$\left\{ \mathsf{K} \in \mathcal{C}_{\text{full}} \;\middle|\; \frac{\partial J(\mathsf{K})}{\partial A_{\mathsf{K}}} = 0, \; \frac{\partial J(\mathsf{K})}{\partial B_{\mathsf{K}}} = 0, \; \frac{\partial J(\mathsf{K})}{\partial C_{\mathsf{K}}} = 0, \right\}$$

a stationary point

$$\mathsf{K} = \begin{bmatrix} 0 & C_{\mathsf{K}} \\ B_{\mathsf{K}} & A_{\mathsf{K}} \end{bmatrix} \in \mathcal{C}_n$$

The same form

$$\tilde{\mathsf{K}} = \left[\begin{array}{c|cc} 0 & \hat{C}_{\mathsf{K}} & 0 \\ \hline \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} & 0 \\ 0 & 0 & \Lambda \end{array}\right] \in \mathcal{C}_n$$

- **Theorem 2 (informal): Under a mild condition, choosing the diagonal stable block Λ randomly leads to a strict saddle point with probability 1**

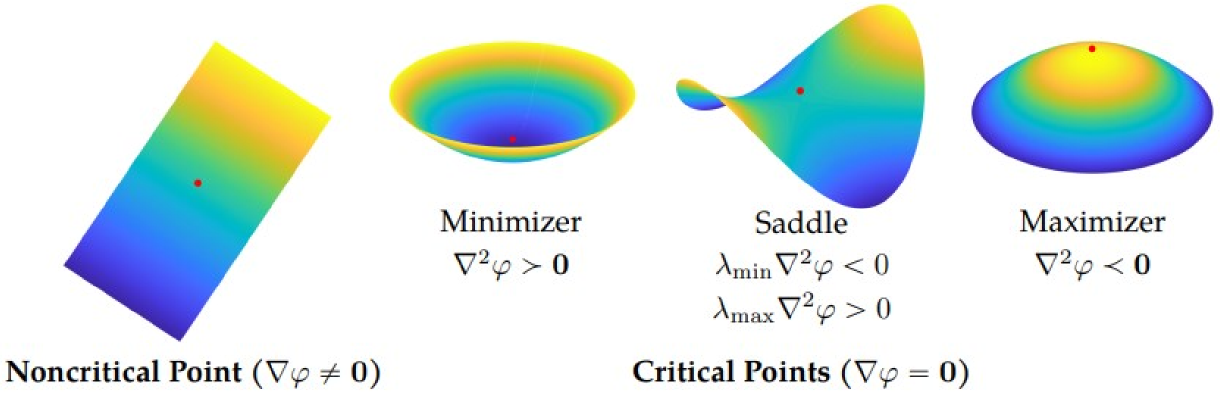


Figure taken from Zhang et al., 2020

- ✓ **Strict saddle points**: the hessian has a strict negative eigenvalue (i.e., escaping direction)
- ✓ **Non-strict (high-order) saddle points**: no such escaping direction, i.e., minimum eigenvalue is zero.
- ✓ **Simple perturbed gradient descent (PGD)** methods can escape strict saddle points efficiently (e.g., Jin et al., 2017)

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., & Jordan, M. I. (2017, July). How to escape saddle points efficiently. In *International Conference on Machine Learning* (pp. 1724-1732). PMLR.

Strict saddle points

Theorem 2: consider a stationary point of the form

$$\tilde{\mathsf{K}} = \left[\begin{array}{c:cc} 0 & \hat{C}_{\mathsf{K}} & 0 \\ \hdashline \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} & 0 \\ 0 & 0 & \Lambda \end{array}\right] \in \mathcal{C}_n$$

We can compute a transfer function $\mathbf{G}(s) := C_{\mathrm{cl}}(sI - A_{\mathrm{cl}}^{\mathsf{T}})^{-1}B_{\mathrm{cl}}$.

- ❑ If $\tilde{\mathsf{K}}$ is globally optimal, then the transfer function above is identically zero for all s.
- ❑ If $\mathbf{G}(s)$ is not identically zero, then choosing the diagonal stable block Λ randomly leads to a strict saddle point with probability 1

Proof idea:

$$\begin{bmatrix} \mathrm{Hess}_{\tilde{\mathsf{K}}}(\Delta^{(1)}, \Delta^{(1)}) & \mathrm{Hess}_{\tilde{\mathsf{K}}}(\Delta^{(1)}, \Delta^{(2)}) \\ \mathrm{Hess}_{\tilde{\mathsf{K}}}(\Delta^{(1)}, \Delta^{(2)}) & \mathrm{Hess}_{\tilde{\mathsf{K}}}(\Delta^{(2)}, \Delta^{(2)}) \end{bmatrix} \in \mathbb{S}^2$$

- ✓ The diagonal entries is always 0,
- ✓ while the off-diagonal entries can be nonzero if $\mathbf{G}(s)$ is not identically zero

Example

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \in \mathbb{S}^2$$

$$\lambda_1 + \lambda_2 = 0, \lambda_1\lambda_2 = -1$$

LQG example

Example: System dynamics

$$A = \begin{bmatrix} -0.5 & 0 \\ 0.5 & -1 \end{bmatrix}, \ B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \ C = \begin{bmatrix} -\frac{1}{6} & \frac{11}{12} \end{bmatrix},$$

Performance weights

$$W = Q = I_2, \ V = R = 1$$

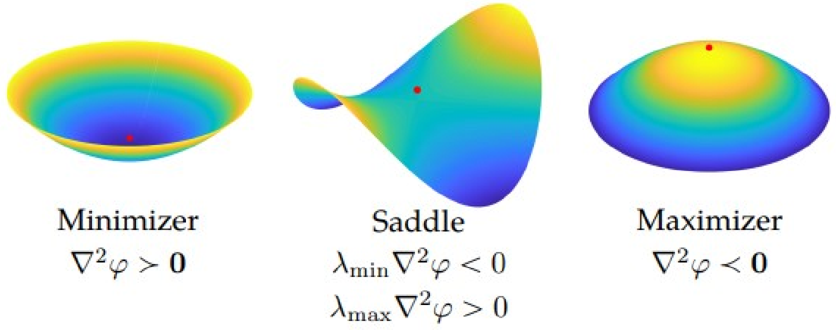
Any zero controller is a stationary point

$$\tilde{\mathsf{K}} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \Lambda \end{array}\right] \in \mathcal{C}_2$$

$$\mathbf{G}(s) = C_{\mathrm{cl}}(sI - A_{\mathrm{cl}}^{\mathsf{T}})^{-1} B_{\mathrm{cl}} = \frac{5(2s-1)}{108(2s^2+3s+1)}.$$

- Choosing a random Λ leads to a strict saddle point.
- For example, when $\Lambda = -\mathrm{diag}(0.5, 0.1)$, the hessian has eigenvalues

$$\lambda_1 = 0.0561, \lambda_2 = -0.0561, \lambda_i = 0, i = 3, \ldots, 8$$



Critical Points $(\nabla\varphi = \mathbf{0})$

Outline

- ❑ LQG problem setup
- ❑ Stationary points and strict saddle points
- ❑ **Escaping saddle points via perturbations**
- ❑ Conclusions

Perturbed Gradient Descent

- Theorem 1 (informal): all bad stationary points are in the same form

$$\tilde{\mathsf{K}} = \left[\begin{array}{c|cc} 0 & \hat{C}_{\mathsf{K}} & 0 \\ \hline \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} & 0 \\ 0 & 0 & \Lambda \end{array}\right] \in \mathcal{C}_n$$

- Theorem 2 (informal): Choosing the diagonal stable block Λ randomly leads to a strict saddle point with probability almost 1

Our idea: a structural perturbation + standard PGD

A high-order saddle → A strict saddle point with the same LQG cost → Standard PGD algorithm (Jin et al., 2017)

Perturbation on Λ

Perturbation on gradients

✓ Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., & Jordan, M. I. (2017, July). How to escape saddle points efficiently. In *International Conference on Machine Learning* (pp. 1724-1732). PMLR.

Numerical examples

Three policy gradient algorithms

1. Vanilla gradient descent $\mathsf{K}_{i+1} = \mathsf{K}_i - \alpha_i \nabla J(\mathsf{K}_i)$
2. Standard PGD algorithm (adding a small random perturbation on iterates; Jin et al., 2017;)
3. **Structural perturbation + standard PGD**

Example: System dynamics

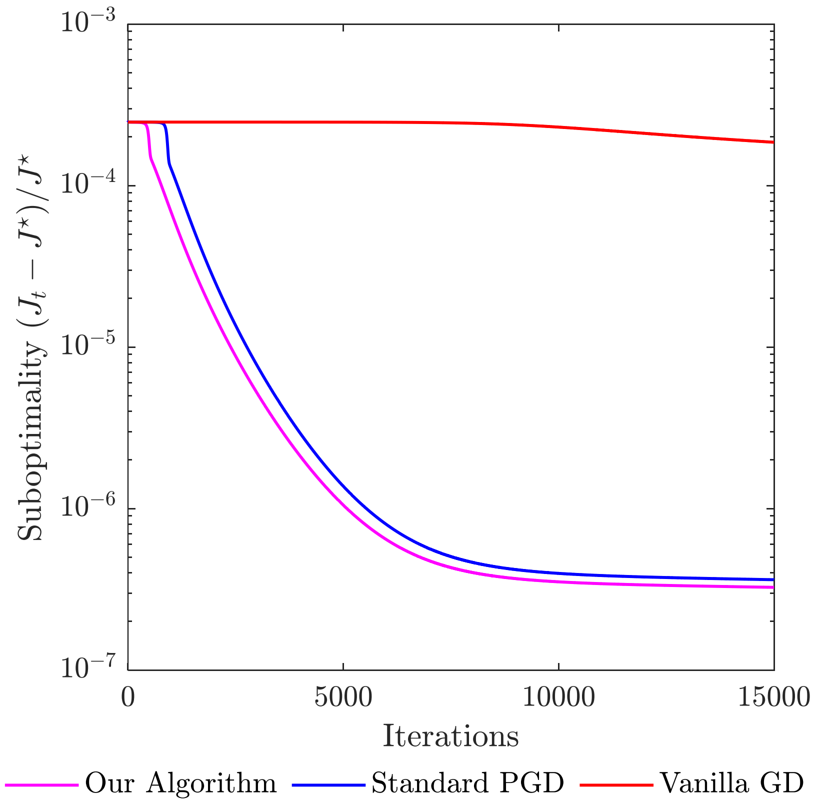
$$A = \begin{bmatrix} -0.5 & 0 \\ 0.5 & -1 \end{bmatrix}, \; B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \; C = \begin{bmatrix} -\frac{1}{6} & \frac{11}{12} \end{bmatrix},$$

Performance weights

$$W = Q = I_2, \; V = R = 1$$

A point that is close to a high-order saddle with zero hessian

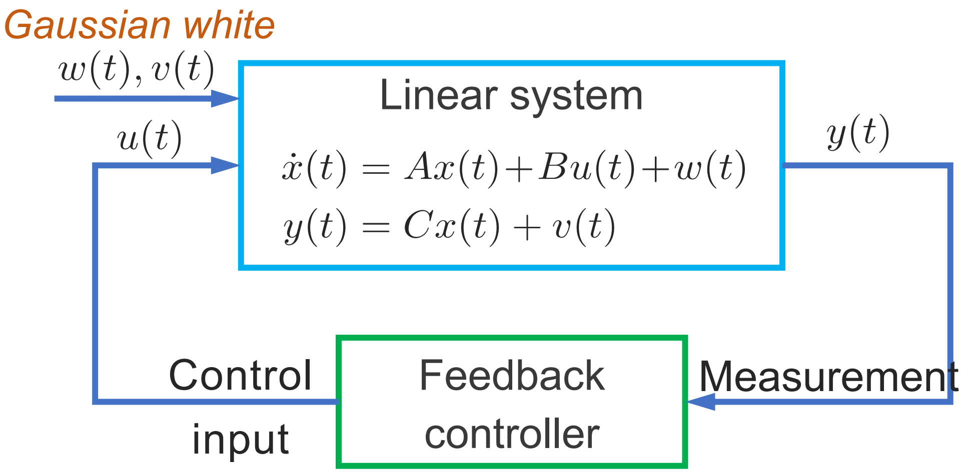
$$A_{\mathsf{K},0} = -0.5 I_2, \; B_{\mathsf{K},0} = \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}, \; C_{\mathsf{K},0} = [0, -0.01]$$



Outline

- ❑ LQG problem setup
- ❑ Stationary points and strict saddle points
- ❑ Escaping saddle points via perturbations
- ❑ **Conclusions**

Summary



LQG as a non-convex optimization problem

$$\min_{\mathsf{K}} \quad J(\mathsf{K})$$
$$\text{s.t.} \quad \mathsf{K} = (A_{\mathsf{K}}, B_{\mathsf{K}}, C_{\mathsf{K}}) \in \mathcal{C}_{\text{full}}$$

- ❑ **Message 1:** all bad stationary points are in the same form

$$\tilde{\mathsf{K}} = \left[\begin{array}{c|cc} 0 & \hat{C}_{\mathsf{K}} & 0 \\ \hline \hat{B}_{\mathsf{K}} & \hat{A}_{\mathsf{K}} & 0 \\ 0 & 0 & \Lambda \end{array}\right] \in \mathcal{C}_n$$

- ❑ **Message 2:** Choosing the diagonal stable block Λ randomly leads to a strict saddle point with probability almost 1

- ❑ **Message 3:** Escaping saddle points for LQG control

A strict saddle point with the same LQG cost + **Standard PGD algorithm (Jin et al., 2017)**

Perturbation on Λ | **Perturbation on gradients**

Ongoing and future work

- ✓ Convergence proof of perturbed policy gradient (PGD)
- ✓ More quantitative analysis of PGD algorithms
- ✓ Alternative model-free parametrization

Escaping High-order Saddles in Policy Optimization for Linear Quadratic Gaussian (LQG) Control

Thank you for your attention!

Q & A