

Learning Pipelines for Adaptive Control

Florian Dörfler, ETH Zürich

IMSI Workshop 2026

Acknowledgments



Feiran Zhao
(ETH Zürich)



Linbin Huang
(Zhejiang Univ.)



Bruce Lee
(ETH Zürich)



Niklas Persson
(Malärdalen Univ.)



Marcell Bartos
(ETH Zürich)



Manuel Klädtke
(TU Dortmund)



Lenart Treven
(ETH Zürich)

Senior co-authors: A. Krause, M. Zeilinger, K. You, X. Wang, A. Chiuso, A. Papadopoulos, ...

PERSPECTIVES ON DATA-DRIVEN CONTROL

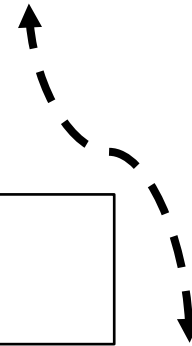
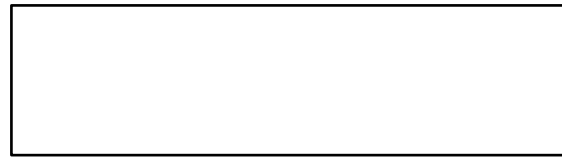
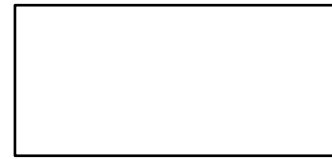
Scientific landscape & culture gaps

in vivo: meant to work in the wild



culture: deploy (with scarce resources) seeking closed-loop robust stability & pen/paper proofs

dynamic programming
in unknown environments



approx/neuro/adaptive
dynamic programming

adaptive control

reinforcement learning

in silico: trained on a simulator

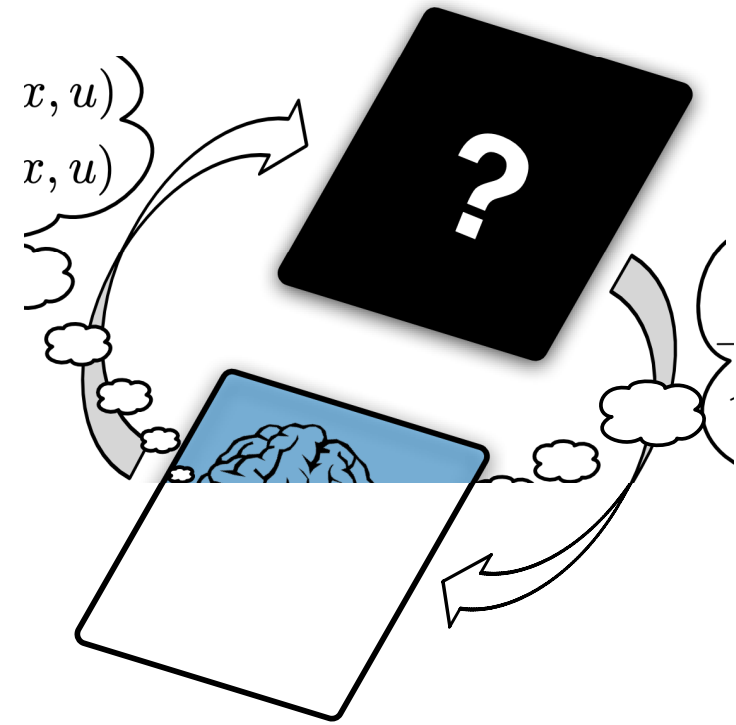


culture: offline training (with many resources) seeking optimality & algorithmic certificates

Data-driven pipelines

- **indirect** (model-based) approach:
data $\xrightarrow{\text{ID}}$ model + uncertainty $\rightarrow$ control
- **direct** (model-free) approach:
data-informativity, behaviors, ...
- **episodic** (offline) algorithms:
collect data batch $\rightarrow$ design policy

- **adaptive** (online) algorithms:
measure $\rightarrow$ update policy $\rightarrow$ act

well-documented **trade-offs**

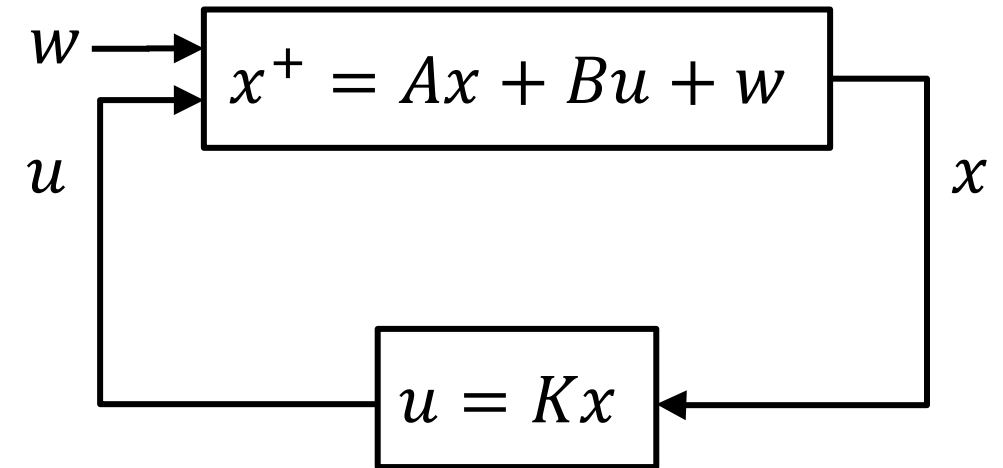
- goal: optimality vs robust stability
- practicality: modular vs end-to-end
- complexity: data, compute, theory

$\rightarrow$ **gold standard**: adaptive + optimal
+ robust + cheap + tractable ...& direct

Back to basics: LQR

- **cornerstone & the benchmark** of both optimal + adaptive control & RL

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & \mathbb{E}_w \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T x_t^T Q x_t + u_t^T R u_t \\ \text{s.t.} \quad & x^+ = Ax + Bu + w \\ & u = Kx \end{aligned}$$



- **research gaps:** no direct + adaptive LQR & no/few closed-loop certificates of RL methods

direct
offline episodic online adaptive
indirect

Today: revisit old problems with new perspectives

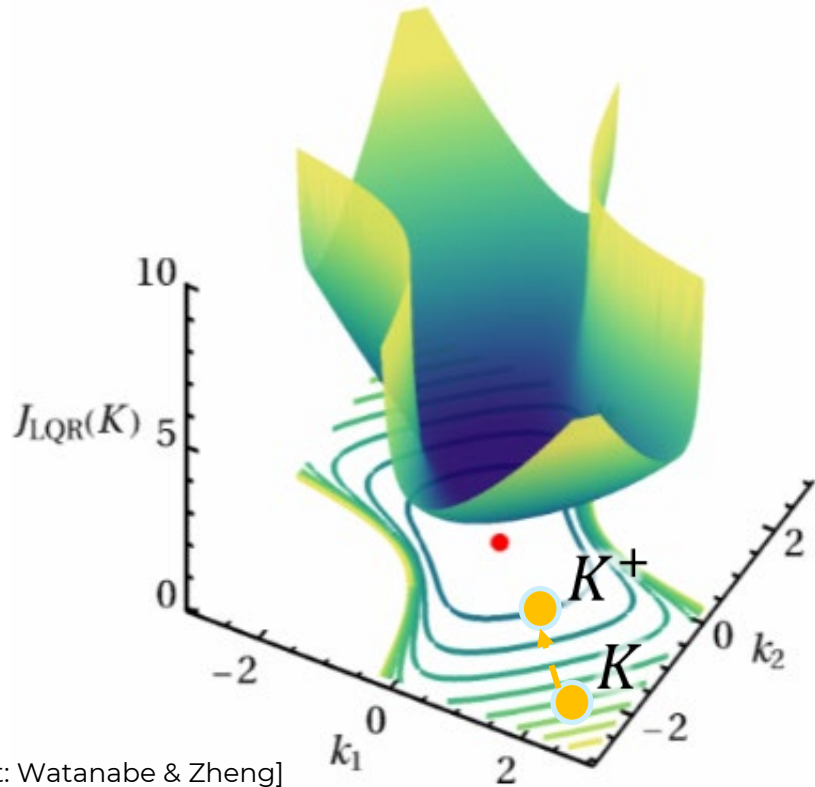
① **adaptation** of control following **policy gradient**

$$K^+ = K - \eta \nabla J_{LQR}(K)$$

② **representation** of trajectory subspace by linear models or **sample covariance**

③ **regularization** of cost promoting prior, robustness, or exploration

$$J_{LQR}(K) \pm \lambda \text{ variance}$$



④ ... & **closed-loop** stability certificates

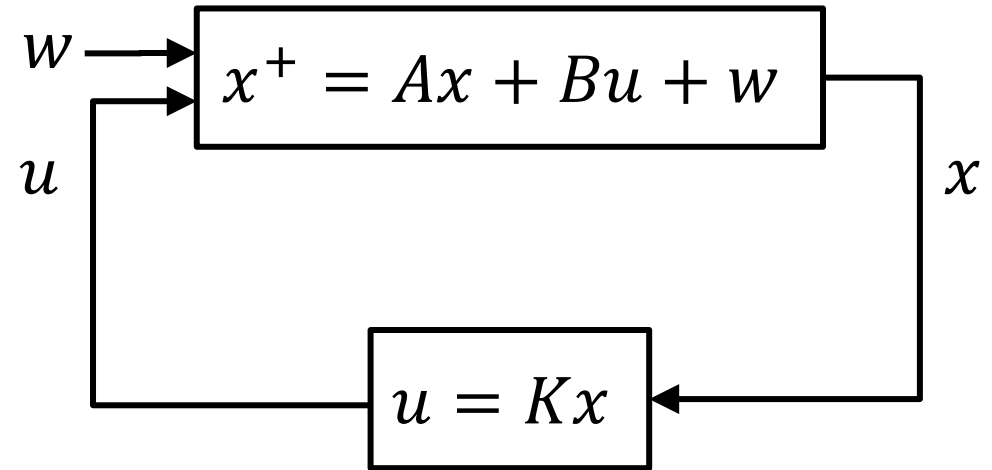
Contents

- 1. problem setup** for adaptive LQR via policy gradient
- 2. learning pipelines** for (in)direct & regularized gradients
- 3. closed loop** stability, optimality, & robustness certificates
- 4. case studies:** robotics, flight, & power system deployments

PROBLEM SETUP FOR ADAPTIVE LQR VIA POLICY GRADIENT

Policy parameterization

$$\begin{aligned} \min_K \quad & \mathbb{E}_w \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T x_t^T Q x_t + u_t^T R u_t \\ \text{s. t.} \quad & x^+ = Ax + Bu + w \\ & u = Kx \end{aligned}$$



→ with **controllability Gramian** or **state covariance** $\Sigma = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T x_t x_t^T}{T}$

$$\begin{aligned} \min_{K, \Sigma > 0} \quad & \text{Tr}(Q\Sigma) + \text{Tr}(K^T R K \Sigma) \\ \text{s. t.} \quad & \Sigma = I + (A + BK)\Sigma(A + BK)^T \end{aligned}$$

→ **algorithmics:** reformulation as SDP or algebraic Riccati equation solved by interior point, contraction, or policy evaluation/improvement

Indirect & certainty-equivalence LQR

- **collect data** (X_0, U_0, X_1) with W_0 unknown & PE: $\text{rank} \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} = n + m$

$$\begin{array}{l} X_0 = [x_0, \dots, x_{t-1}] \\ U_0 = [u_0, \dots, u_{t-1}] \\ W_0 = [w_0, \dots, w_{t-1}] \end{array} \begin{array}{l} \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{array} \boxed{X_1 = AX_0 + BU_0 + W_0} \longrightarrow X_1 = [x_1, \dots, x_t]$$

- **indirect & certainty-equivalence LQR**
(all solved offline)

$$\min_{K, \Sigma \succ 0} \text{Tr}(Q\Sigma) + \text{Tr}(K^T R K \Sigma)$$

$$\text{s. t. } \Sigma = I + (\hat{A} + \hat{B}K)\Sigma(\hat{A} + \hat{B}K)^T$$

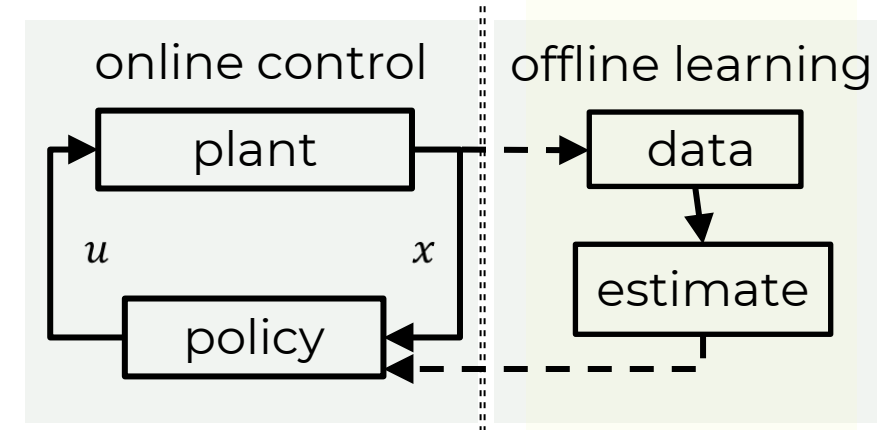
$$[\hat{B}, \hat{A}] = \arg \min_{B, A} \left\| X_1 - [B, A] \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} \right\|_F$$

certainty-
equivalent
LQR

least
squares
SysID

But this is offline!

- **shortcoming** of offline learning
→ cannot improve **online** & **adapt** rapidly



monotonicity principles of adaptive control: acquire information & improve performance over time by interaction



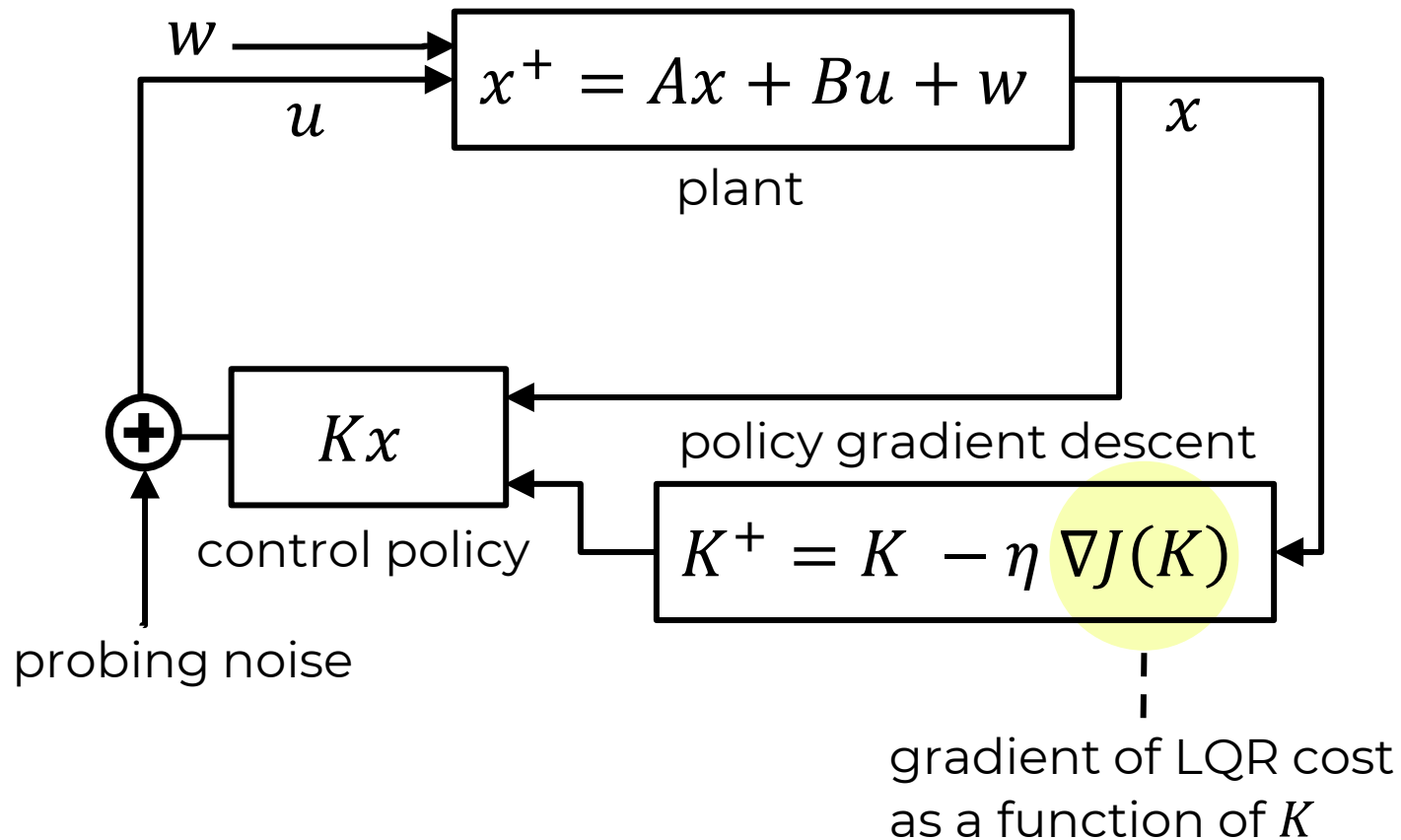
Adaptive Control:
Towards a Complexity-Based General Theory*

G. ZAMES-

- **desired adaptive** solution: online (non-episodic / batch) algorithm, with closed-loop data, & recursive/real-time implementation

* disclaimer: a large part of the adaptive control community focuses on stability & not optimality

Adaptive LQR via policy gradient descent



Seems obvious but...

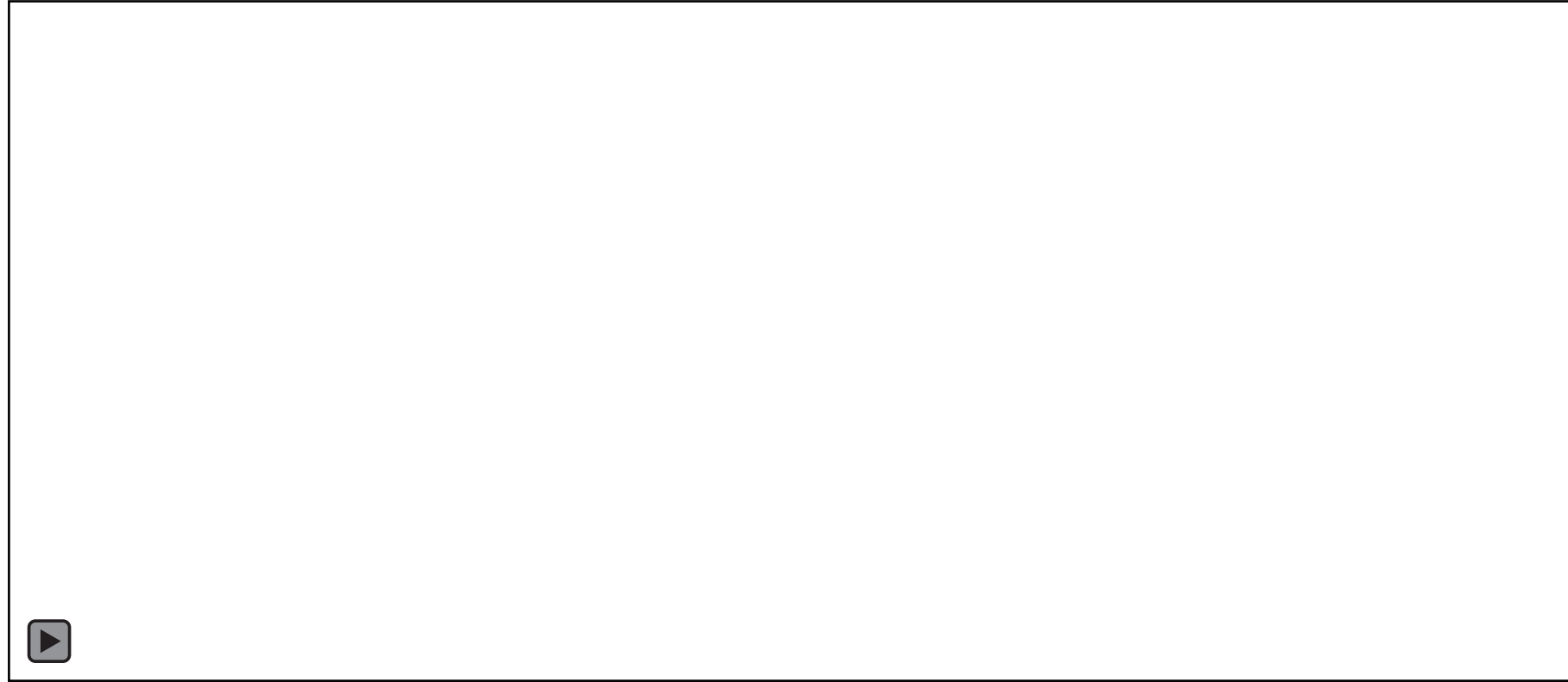
→ **algorithms**

- how to compute $\nabla J(K)$ cheaply & recursively?
- direct or indirect?
- convergence?

→ **closed loop**

- stability?
- robustness?
- optimality?

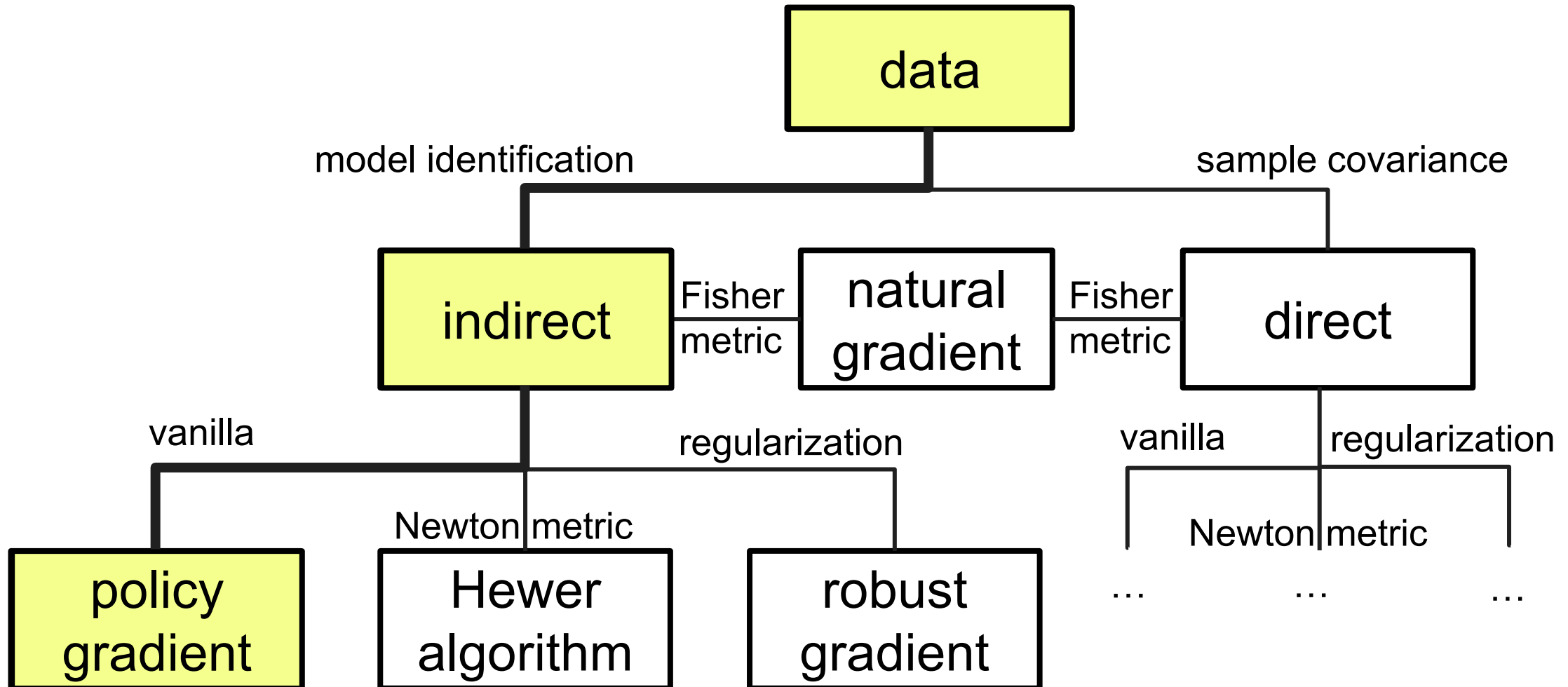
Preview: does it work on an autonomous bike ?



Setup: **autonomous bicycle** with **coarse inner control** (2d dynamics stabilized by feedback linearization) & **outer adaptive policy gradient**

LEARNING PIPELINES FOR POLICY GRADIENT

Algorithmic road map



LQR optimization landscape

$$\begin{aligned} \min_{K, \Sigma \succ 0} \quad & J(K) = \text{Tr}(Q\Sigma) + \text{Tr}(K^T R K \Sigma) \\ \text{s. t.} \quad & \Sigma = I + (A + BK)\Sigma(A + BK)^T \end{aligned}$$

after eliminating
unique $\Sigma \succ 0$,
denote the
objective as $J(K)$

$J(K)$ is usually not convex but for stabilizing K is

- **differentiable** with $\nabla J(K) = 2((R + B^T P B)K - B^T P A)\Sigma$

[plot: Zheng, Pai, Tang]

where $P = Q + K^T R K + (A + BK)^T P (A + BK)$
& Σ are closed-loop obs. + contr. Gramians

- **coercive** & locally **bounded Hessian**

- **gradient dominance:** $J(K) \leq J^* + \text{const.} \cdot \|\nabla J(K)\|^2$

Model-based policy gradient

$$\min_K J(K) = \text{Tr}(Q\Sigma) + \text{Tr}(K^T R K \Sigma)$$

where $\Sigma = I + (A + BK)\Sigma(A + BK)^T$

Fact: For initial K_0 stabilizing & small η , **gradient descent** $K^+ = K - \eta \nabla J(K)$ **converges linearly** to K^* .

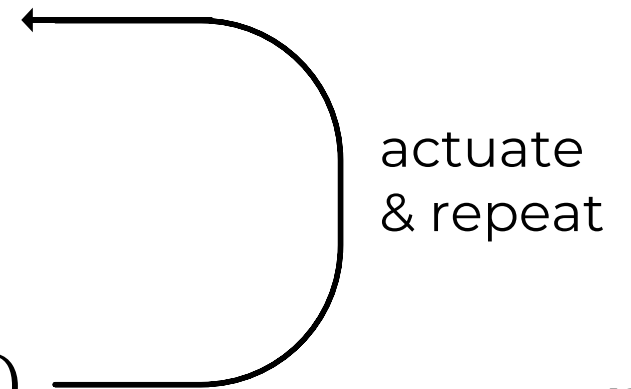
Annual Review of Control, Robotics and Autonomous Systems

Toward a Theoretical
Foundation of Policy
Optimization for Learning
Control Policies

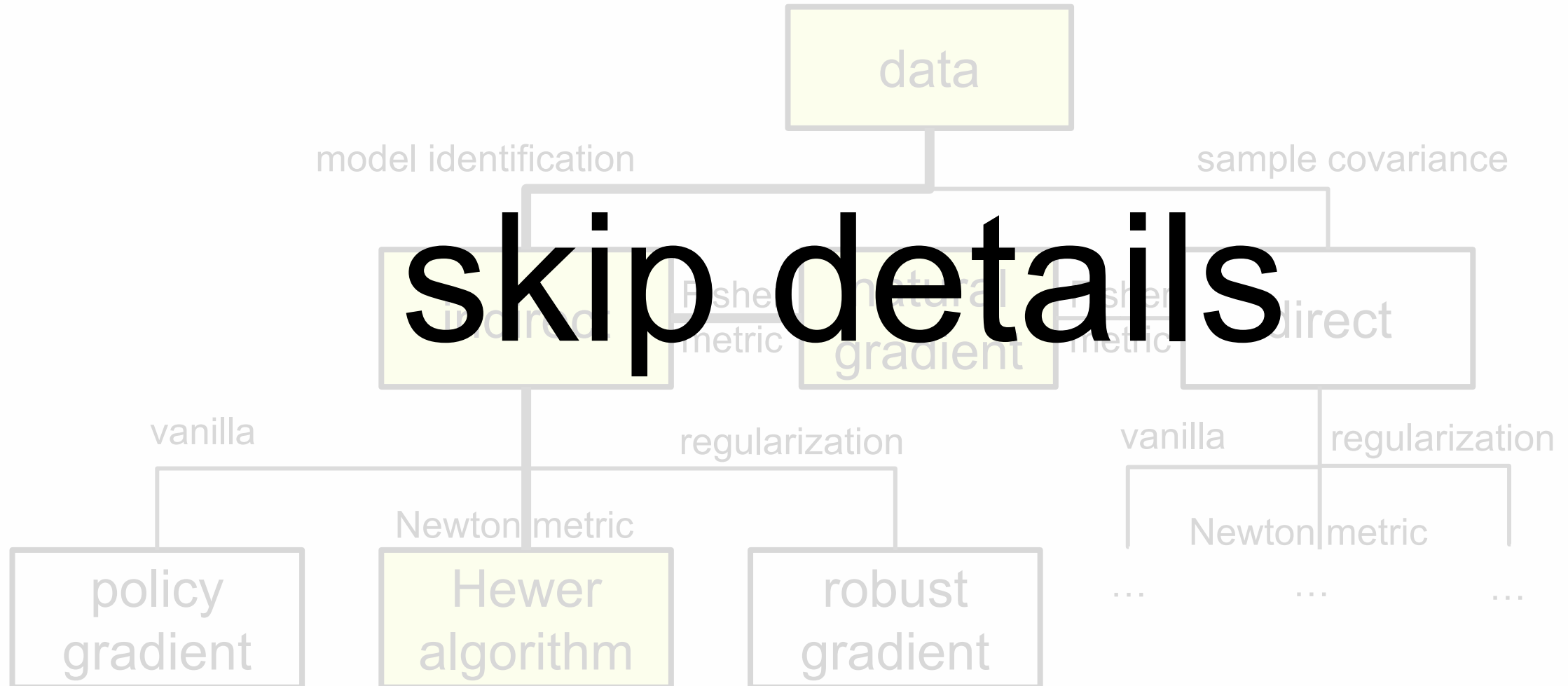
Bin Hu,¹ Kaiqing Zhang,^{2,3} Na Li,⁴ Mehran Mesbahi,⁵
Maryam Fazel,⁶ and Tamer Başar¹

Conceptual algorithm: model-based adaption via policy gradient

1. **data collection:** refresh (X_0, U_0, X_1)
2. **identification** of $(\hat{B}, \hat{A})$ via recursive LS
3. **policy gradient:** $K^+ = K - \eta \nabla J(K)$ using estimates $(\hat{B}, \hat{A})$ & closed-loop Gramians (Σ, P)



Algorithmic road map



Pre-scaled policy gradient

$$K^+ = K - \eta \mathbf{M}(K) \nabla J(K)$$

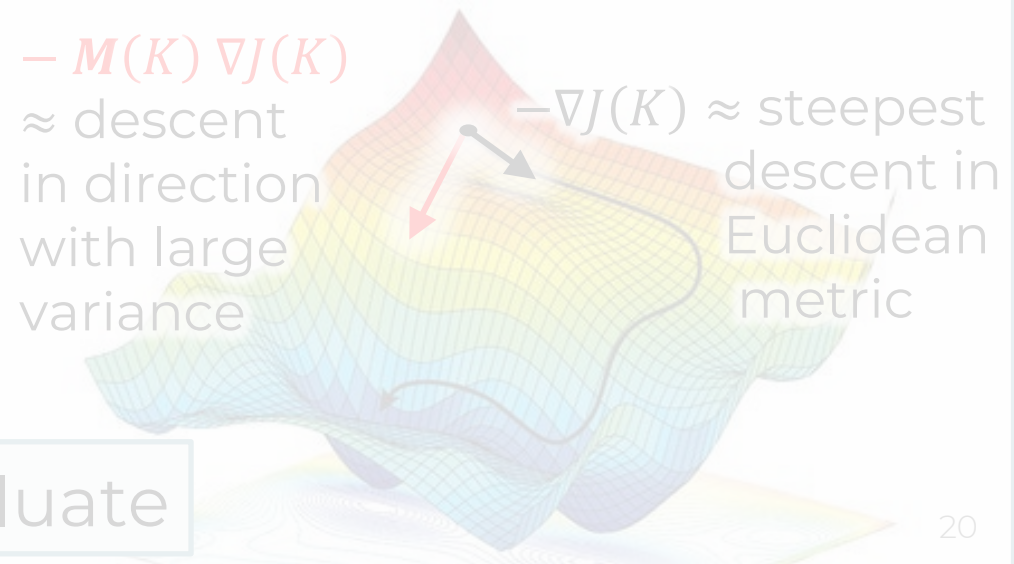
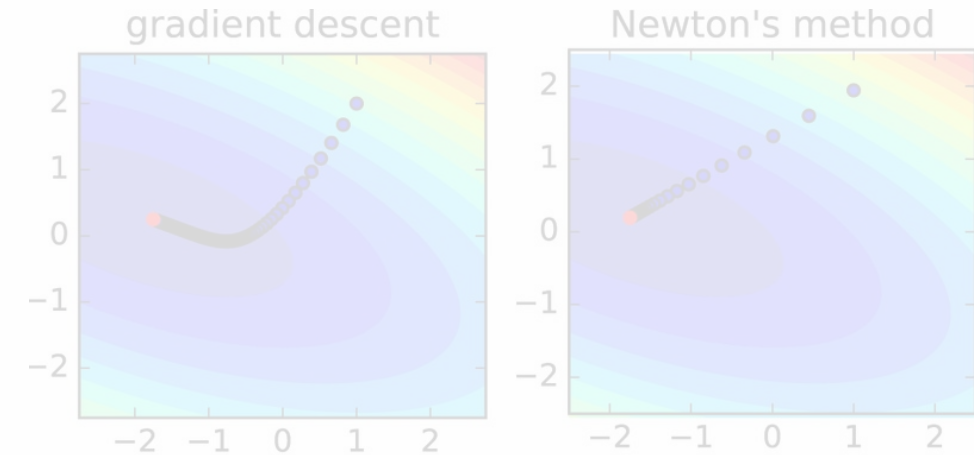
→ **Gauss-Newton**: $\mathbf{M}(K)$ cheaply approximates inverse Hessian $\nabla^2 J(K)$

Fact: this equals Helwig's algorithm: policy evaluation $\Rightarrow$ improvement

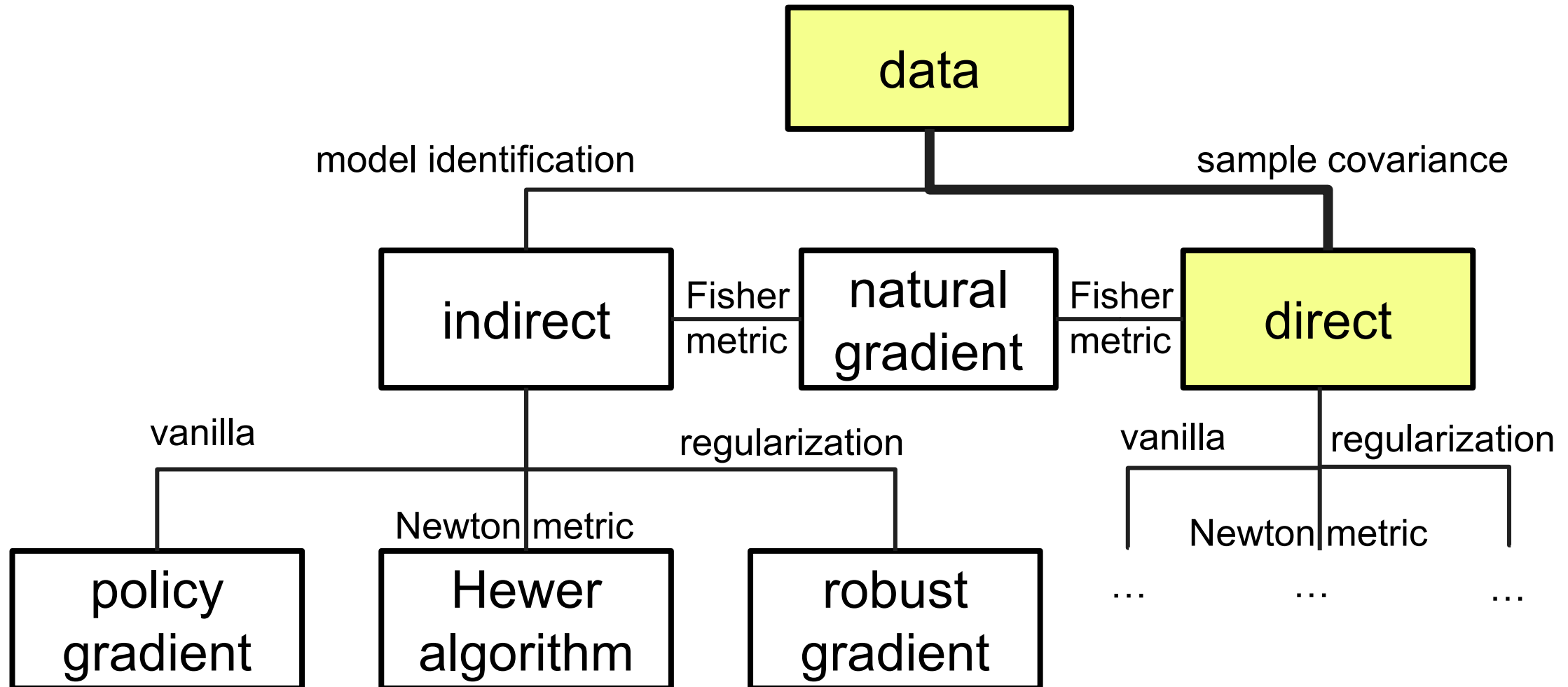
skip details

→ **Natural policy gradient** method:
 $\mathbf{M}(K)$ is inverse Fisher information

Fact: $\mathbf{M}(K) \nabla J(K) = \nabla J(K) \Sigma^{-1}$ is easy to evaluate



Algorithmic road map



Direct (model-free) policy gradient methods

- **issue:** uncertainty propagation is hard in indirect case & outcome suffers from bias error (e.g. model-order in output feedback setup)

- **model-free 0th order methods** constructing two-point estimate

$$\widehat{\nabla J(K)} = (J(K + rU) - J(K - rU)) \cdot \frac{mn}{r^2} U$$

from uniform perturbation U & numerous + very long trajectories

relative performance gap	$\epsilon = 1$	$\epsilon = 0.1$	$\epsilon = 0.01$
# trajectories (100 samples long)	1414	43850	142865

**$\sim 10^7$ samples for
4th order system**

- direct policy gradient is inefficient, episodic, & **practically useless ?**
→ **sample covariance parameterization to the rescue !**

Sample covariance parametrization

(leave out noise for sake of presentation)

$$\begin{aligned} X_0 &= [x_0, \dots, x_{t-1}] \\ U_0 &= [u_0, \dots, u_{t-1}] \end{aligned} \longrightarrow \boxed{X_1 = AX_0 + B U_0} \longrightarrow X_1 = [x_1, \dots, x_t]$$

- sample covariances:** $\Lambda = \frac{1}{t} \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top \succ 0$ & $\Lambda^+ = \frac{1}{t} X_1 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top$
(due to PE)

$$\Rightarrow \forall K \exists V \text{ s.t. } \boxed{\begin{bmatrix} K \\ I \end{bmatrix} = \Lambda V} \quad (*)$$

- closed loop:** $A + BK = [B \ A] \begin{bmatrix} K \\ I \end{bmatrix} \stackrel{\text{green box}}{=} [B \ A] \frac{1}{t} \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top V \stackrel{\text{red box}}{=} \frac{1}{t} X_1 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top V = \Lambda^+ V$

- direct data-driven** formulation by substituting $A + BK = \Lambda^+ V$ & $\boxed{(*)}$

Covariance parametrization of policy gradient

- covariance parameterization:**

substitute $A + BK = \Lambda^+ V$ with

linear constraint $\begin{bmatrix} K \\ I \end{bmatrix} = \Lambda V$

- policy gradient** in V -coordinates

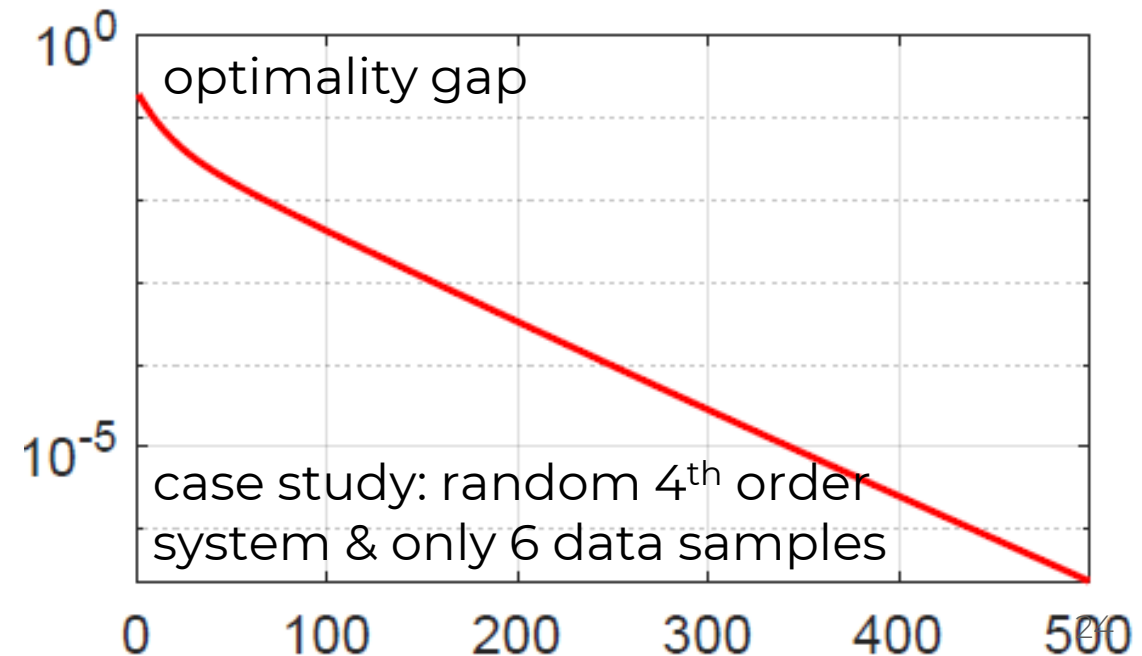
- converges with min. samples
- in original coordinates

$$K^+ = K - \eta \mathbf{M}_t \nabla J(K)$$

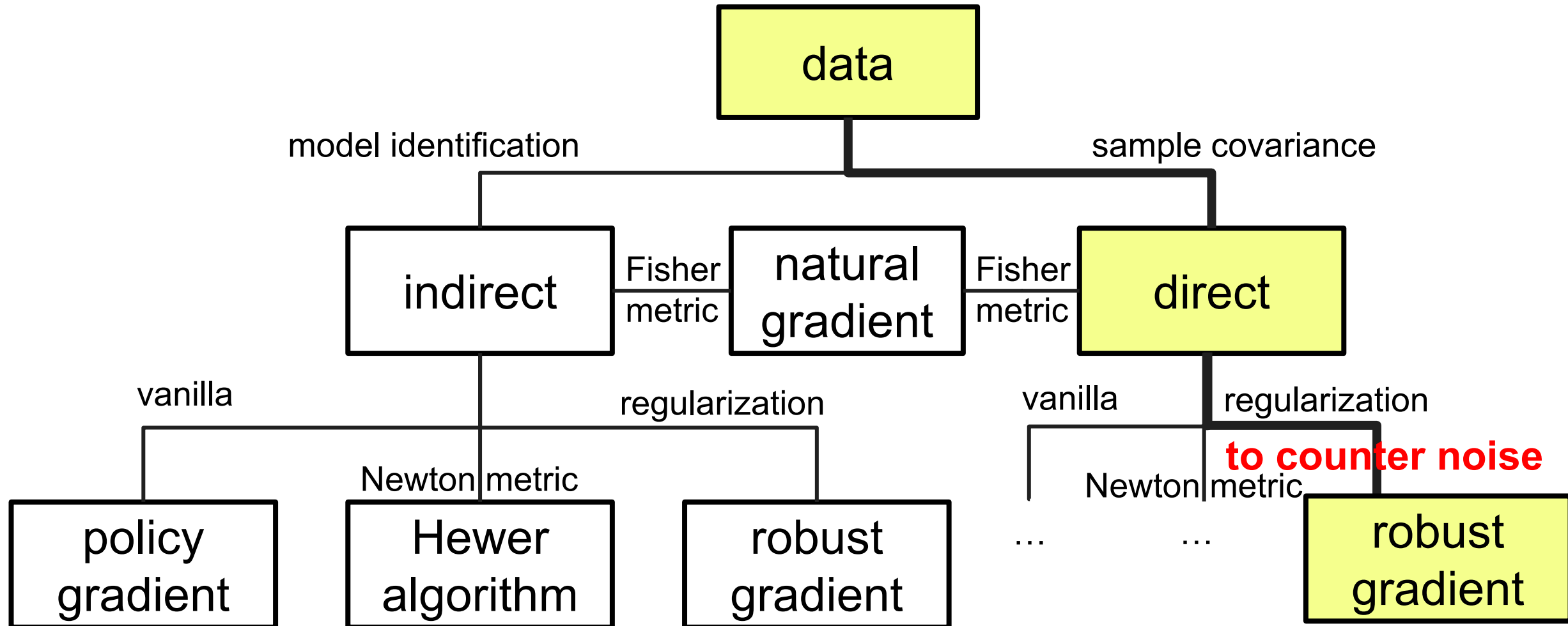
where $\mathbf{M}_t > 0$ depends on data

- natural gradient is coord-invariant

$$\begin{aligned} \min_{K, \Sigma \geq 0} & \quad Tr(Q\Sigma) + Tr(K^T R K \Sigma) \\ \text{s. t.} & \quad \Sigma = I + (\Lambda^+ V) \Sigma (\Lambda^+ V)^T \\ & \quad \begin{bmatrix} K \\ I \end{bmatrix} = \Lambda V \end{aligned}$$



Algorithmic road map



Short history of regularized data-driven LQR

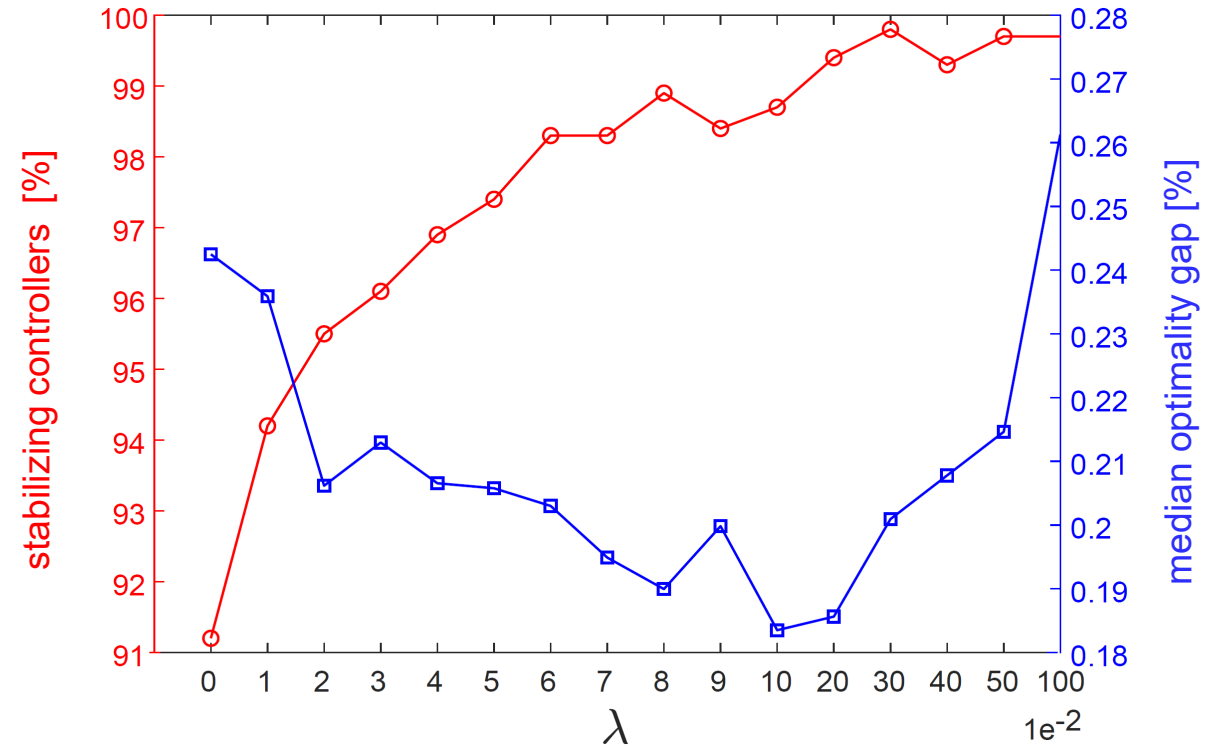
$$J(K) + \lambda \cdot \text{regularizer}(\Sigma, K, \Lambda)$$

incorporating prior, promoting robustness, or encouraging exploration

Few regularizers

- **robust:** $Tr\left(\Lambda^{-1} \begin{bmatrix} K \\ I \end{bmatrix} \Sigma \begin{bmatrix} K \\ I \end{bmatrix}^T\right)$
 - effect: (R, Q) scaled by Λ^{-1}
 - high cost for less explored (u, x)
 - robust stability & performance

- **exploring:** $-Tr(\Lambda^{-1}\Sigma)$
 - promotes directions with high empirical variance
 - achieves exploration with $\sqrt{T}$ regret without probing noise

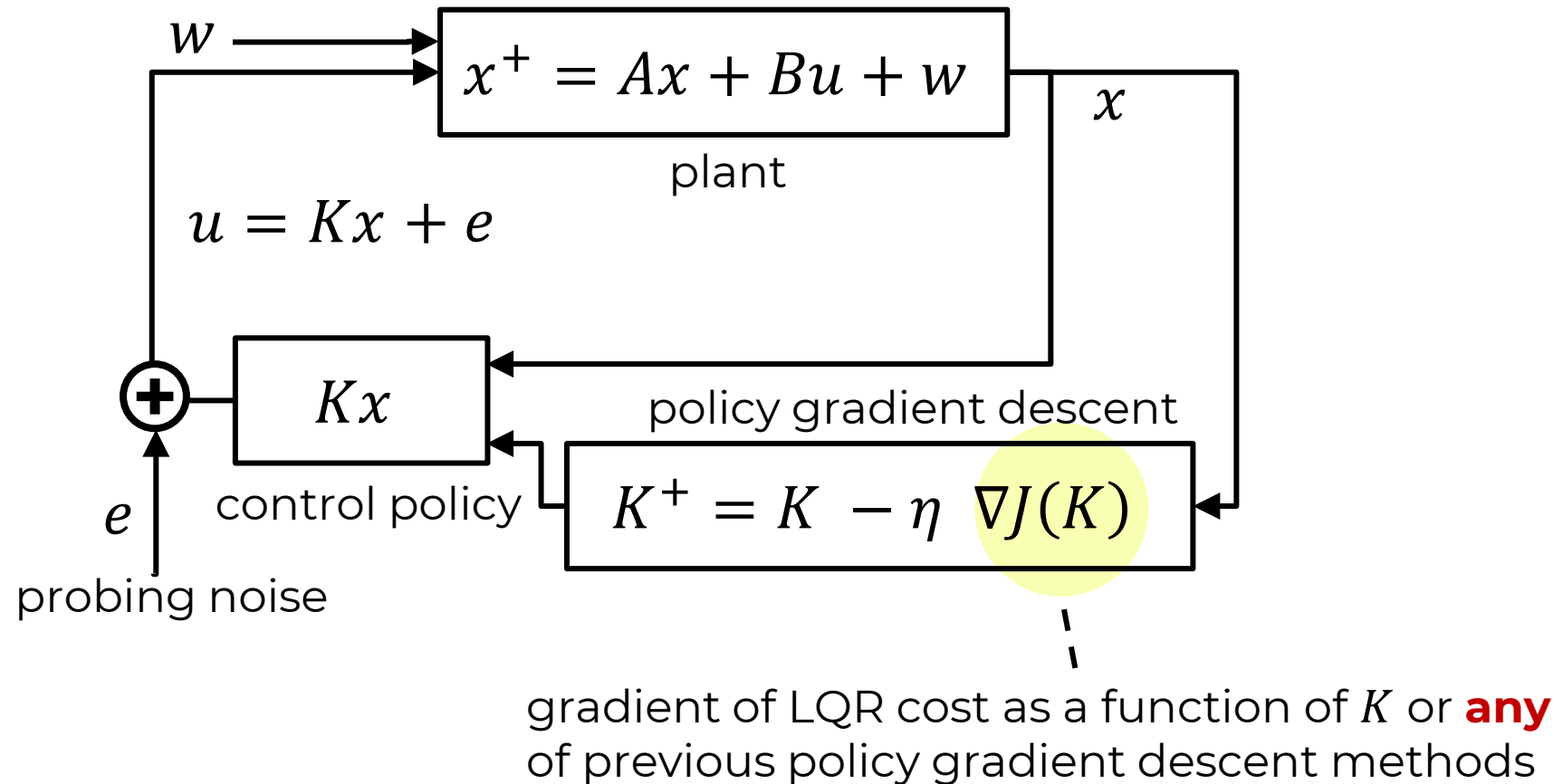


regret achieved by
different methods

by regularizing T

CLOSED-LOOP STABILITY, OPTIMALITY & ROBUSTNESS

Policy gradient descent in closed loop



Q: if each K_t is stabilizing & $J(K_t)$ decreases, we surely get asymptotic stability & optimality ?

A: system is time-varying ... unclear

Information metric

- bounded **noise covariance**:

$$\left\| \frac{1}{t} W_0 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top \right\| \leq \delta_t^2 \text{ for some } \delta_t \geq 0$$

- persistency of excitation** due to probing: $\underline{\sigma}(\Lambda_t) \geq \gamma_t^2$ for some $\gamma_t \geq 0$
- information metric = **signal-to-noise ratio** $SNR_t := \gamma_t / \delta_t$

satisfies Zames' first
monotonicity principle:
information acquisition
= SNR increases



Adaptive Control:
Towards a Complexity-Based General Theory*

G. ZAMES

noise	excitation	SNR
Gaussian $\delta_t \sim O(1/\sqrt{t})$	Constant $\gamma_t \sim O(1)$	$SNR_t \sim O(\sqrt{t})$
	Decay $\gamma_t \sim O(t^{-1/4})$	$SNR_t \sim O(t^{1/4})$

Certificate for **any** of the policy gradient methods

Theorem (simplified): There exist $v_i > 0, i \in \{1,2,3,4,5\}$, depending on (A, B, Q, R, K_0) with $v_4 < 1$, so that, if $SNR_t \geq v_1 \forall t$, $\eta \leq v_2$, & for stable K_0

1. the **closed-loop system is stable** in the sense that

$$|x_t| \leq v_3 \left(1 - \frac{v_4}{2}\right)^t |x_0| + \frac{2v_3}{v_4} \left(\max_{0 \leq i \leq t} |Be_i + w_i|\right).$$

SNR & step-size requirements

2. the **policy converges to optimality** in the sense that

$$C(K_t) - C^* \leq \left(1 - \frac{\eta}{v_5}\right)^t (C(K_{t_0}) - C^*) + O(SNR_t^{-1})$$

stable initialization

nominal exponential convergence

bias due to noise

Notes on stability & convergence statement

- **assumptions:** stable K_0 + large enough SNR + small enough step size to control learning rate & assure **sequential stability**

- **convergence:** nominal exponential + (decreasing) bias term

skip details

→ Zames' 2nd microtendency principle improving performance

→ $O(1/\sqrt{t})$ for Gaussian noise & constant excitation

→ $O(t^{-1/4})$ for Gaussian noise & diminishing excitation

slightly worse
than optimal
known rates
 $O(1/t)$ & $O(1/\sqrt{t})$

- **direct methods:** $SNR_t \geq \frac{\text{const.} \|\mathbf{M}_t\|}{\underline{\sigma}(\mathbf{M}_t)}$, $\eta_t \leq \frac{\text{const.}}{\|\mathbf{M}_t\|}$, & convergence rate

depend on data-dependent matrix $\mathbf{M}_t = \frac{1}{t^2} U_0 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top \Pi \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} U_0^\top$

Notes on stability & convergence statement

- all results also hold in **regularized setting** under proper choice of regularization coefficient $\lambda_t \leq 0$ (γ_t δ_t) (const. for bounded noise)
- Indirect Gauss-Newton method = adaptive version of **Hewer's algorithm**, which needs additionally K_t sufficiently close to K^*

skip details

Algorithm: adaptive Hewer's algorithm

1. **data collection:** refresh (X_0, U_0, X_1)
2. **identification** of $(\hat{B}, \hat{A})$ via recursive LS
3. **policy evaluation:** $P^+ = \text{Lyapunov}(\hat{B}, \hat{A}, K)$
4. **policy improvement:** $K^+ = (\dots)^{-1} P^+ \hat{B} \hat{A}$

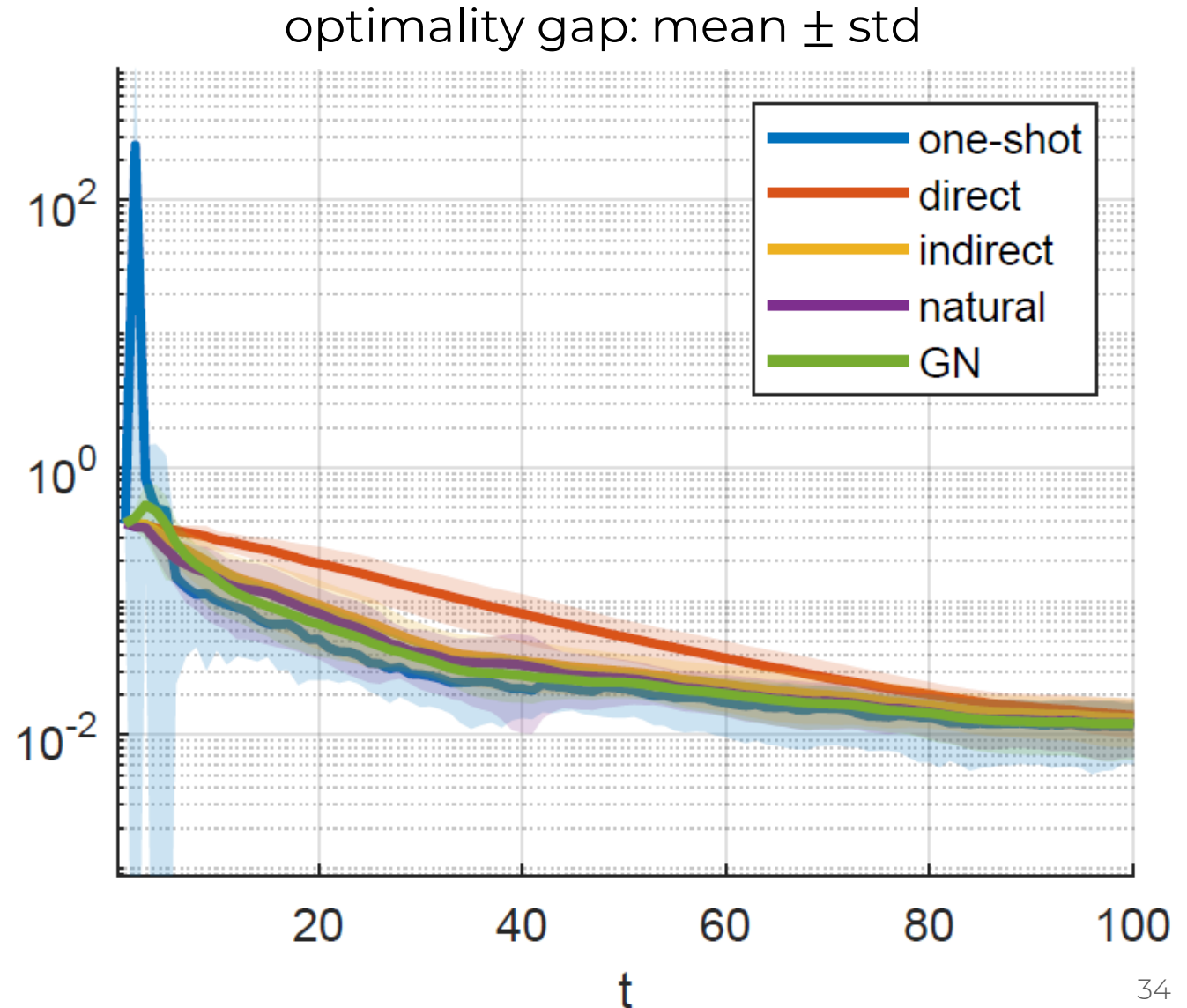


Numerics: convergence to optimality

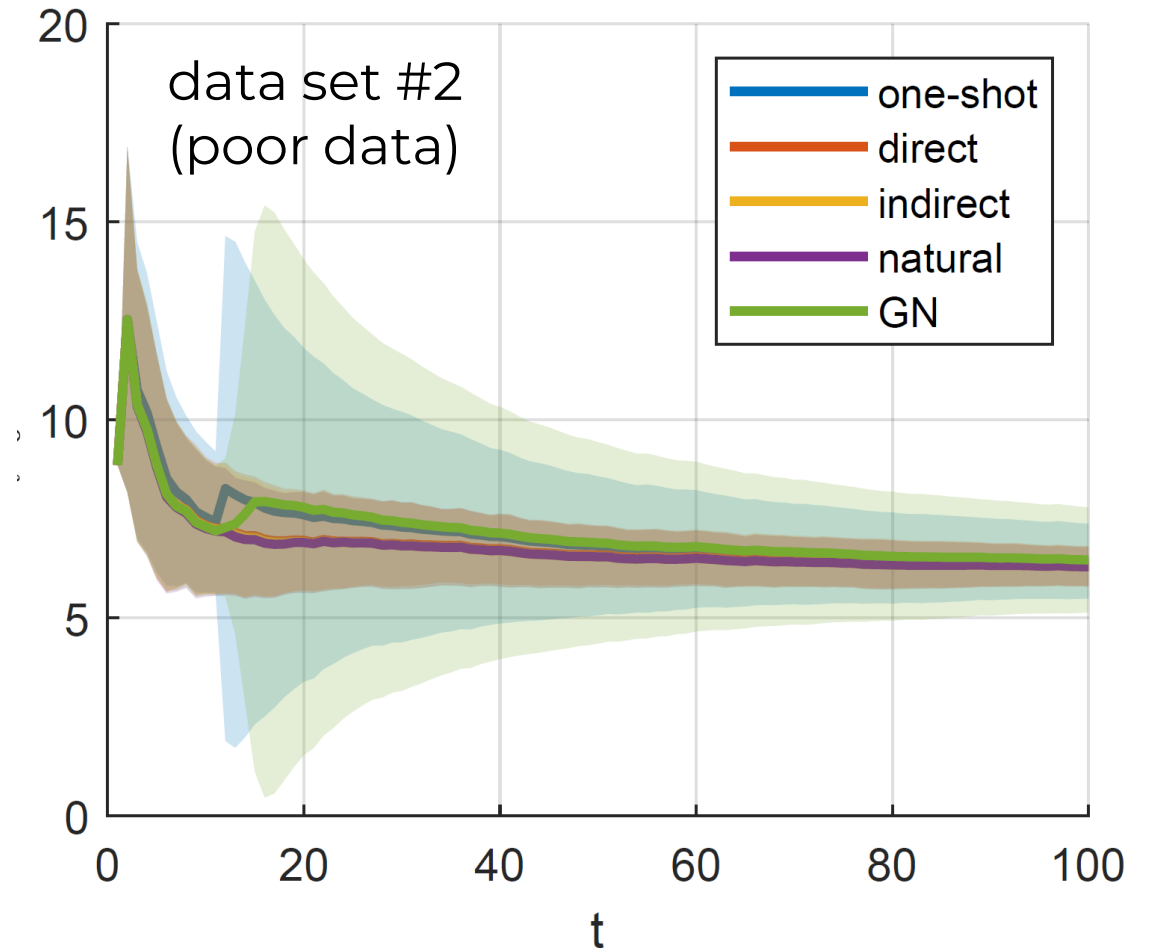
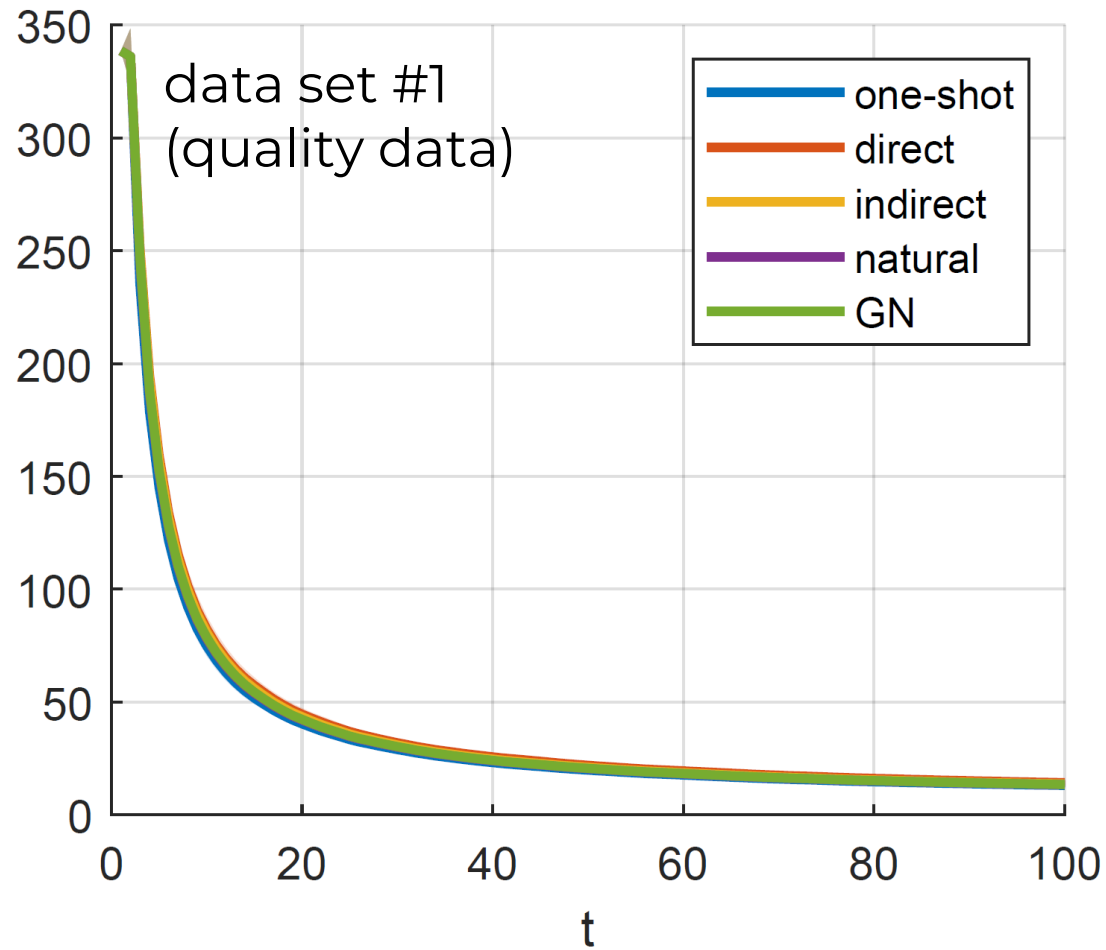
- **case study** [Dean et al. '19]:
discrete-time system with
Gaussian noise $w_t \in \mathcal{N}(0, id)$

$$A = \begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix}, \quad B = I$$

- policy gradient methods are more **robust** to noise versus sensitive one-shot method
- **empirically** observe tighter optimality gap $\sim O(SNR^{-2})$ than our certificate $O(SNR^{-1})$

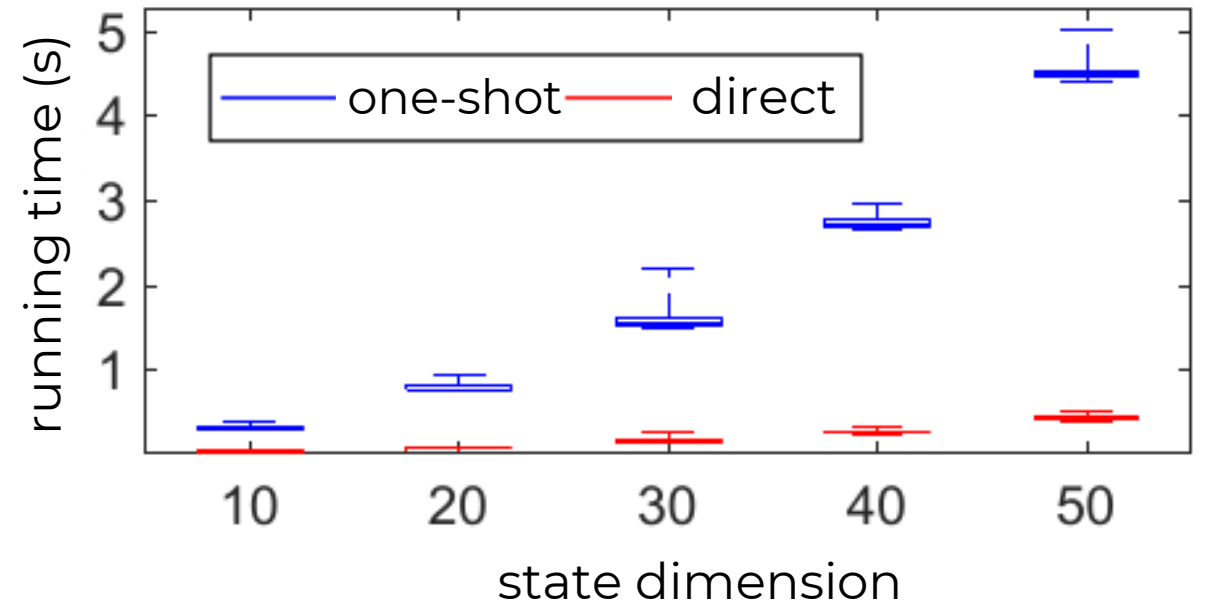
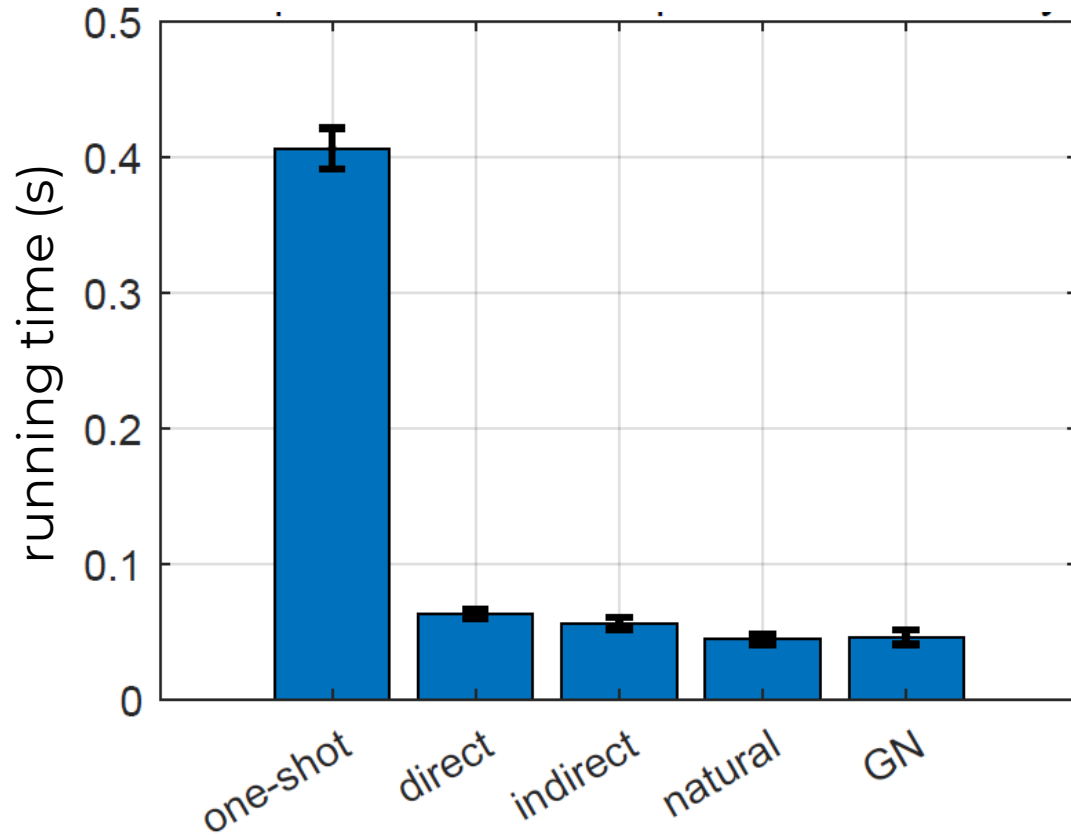


Numerics: mean $\pm$ std of closed-loop realized cost



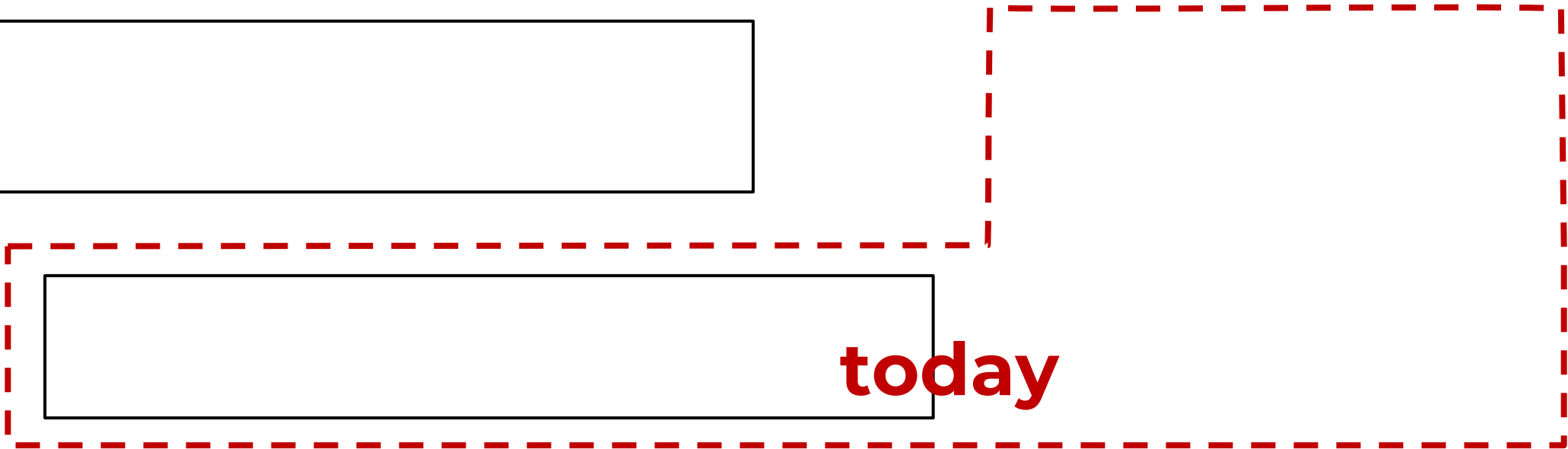
→ all converge with a bias, but one-shot & Gauss-Newton less robust

Numerics: computational efficiency



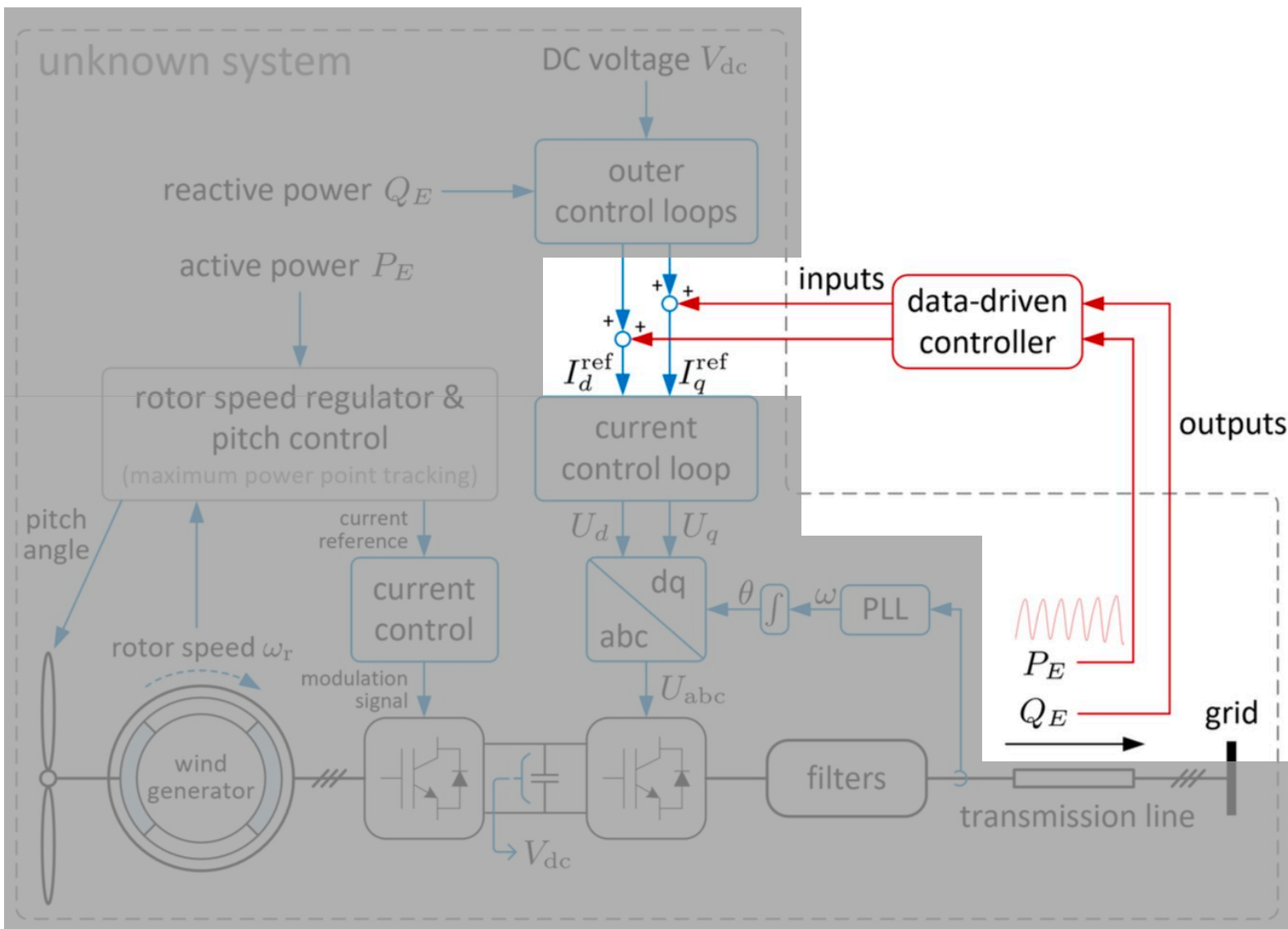
→ all policy gradient methods significantly outperform one-shot-based method in **computational effort**

Implementations



today

Power systems / electronics case study

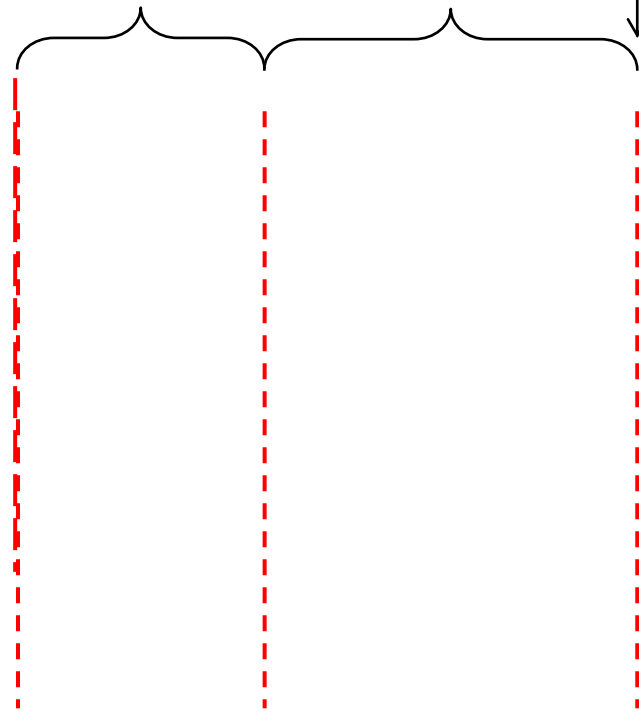


synchronous generator & full-scale converter

- wind turbine becomes **unstable** in weak grid
→ nonlinear oscillations
- converter, turbine, & grid are **black box** for commission engineer
- construct **state space realization** from time shifts (5ms sampling) of inputs & outputs

→ **direct policy gradient**

oscillation
observed probe &
collect data activate policy gradient
adapative LQR control



with iterations) DeePO (100
with iteration) DeePO (1

without DeePO

without control
adaptation
with policy
gradient adaptive
control

time [s]

change of system parameters
(DC voltage setpoint & gain)

AC grid voltage
disturbance occurs

→ without
control
adaptation

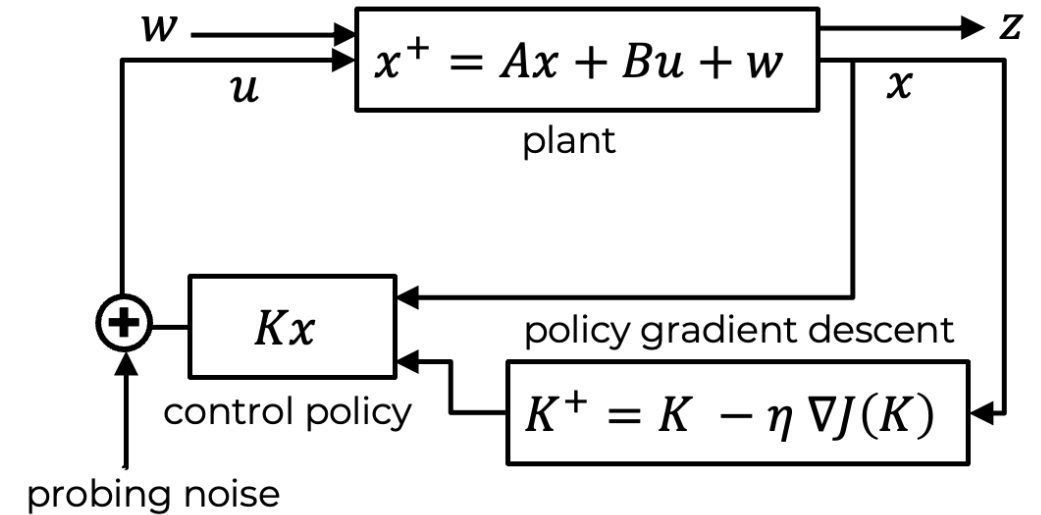
→ with policy
gradient
adaptive
control

time
[s]

Conclusions

Summary

- policy gradient adaptive control
- various algorithmic pipelines
- closed-loop stability & optimality
- academic & real-world case studies



Ongoing & future work

- **technicalities:** improve rates, beyond LTI systems, general SDPs
- **when to adapt?** online vs episodic? "best" batch size? triggered?
- **algorithmic:** break time-scale separation & accelerate adaption

THANKS