LQR Learning Pipelines

between reinforcement learning & adaptive control

Feiran Zhao, ETH Zürich

Coffee talk, ETH Zurich, October 23rd, 2025

Acknowledgments

Keyou You (Tsinghua)

Linbin Huang (Zhejiang)

Xuerui Wang (TU Delft)









papers

Further:

Niklas Persson,

Andres Jürisson,

& Mojtaba Kaheni

Florian Dörfler

Roy Smith

(ETH Zurich)

Pietro Tesi (Florence)

Alessandro Chiuso (Padova)

Claudio de Persis (Groningen)

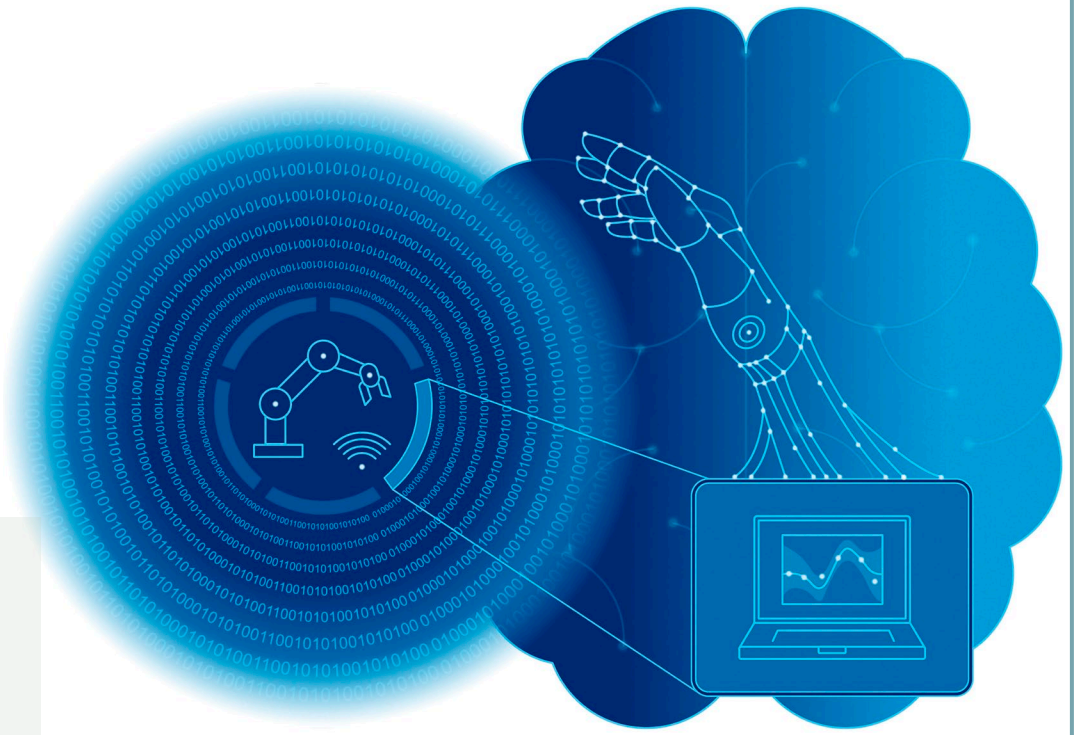
A. V. Papadopoulos (Malardalen)

PERSPECTIVES ON DATA-DRIVEN CONTROL

Scientific landscape

Rich & vast history

fragmented between fields: adaptive control & reinforcement learning (RL)



Culture gaps: adaptive control ↔ RL

- stability *vs* opitimality
- robust *vs* optimism facing uncertainty
- interpretable pen+paper *vs* compute
- theory certificates *vs* empirical studies

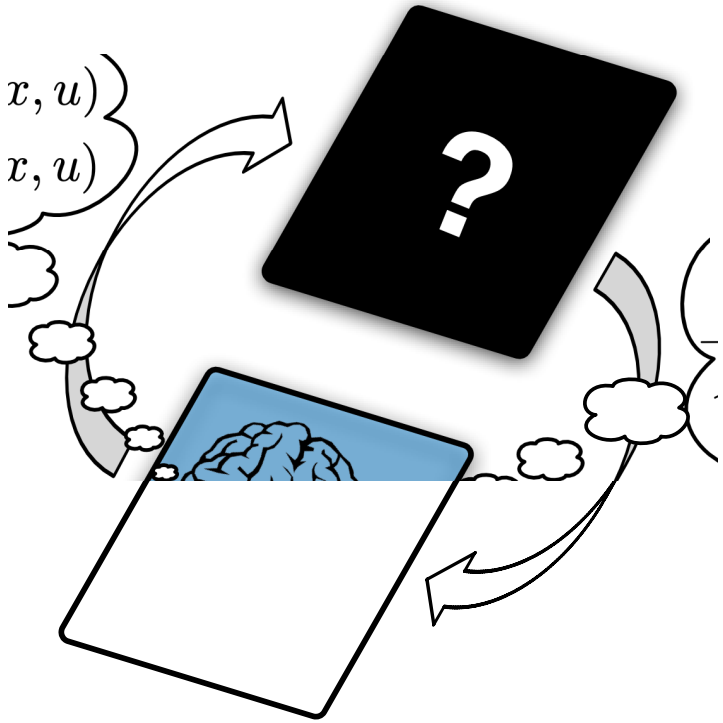
Reinforcement Learning is Direct Adaptive Optimal Control

Richard S. Sutton, Andrew G. Barto, and Ronald J. Williams

- **common root**: dynamic programming
- **early cross-overs**: neuro/adaptive DP
- **today**: cross-fertilization & bridging culture gaps

Data-driven pipelines

- **indirect** (model-based) approach:
  data $\overset{\text{ID}}{\rightarrow}$ model + uncertainty → control
- **direct** (model-free) approach:
  direct MRAC, behavioral, RL…
- **episodic & batch** algorithms:
  collect data batch → design policy
- **online & adaptive** algorithms:
  measure → update policy → act



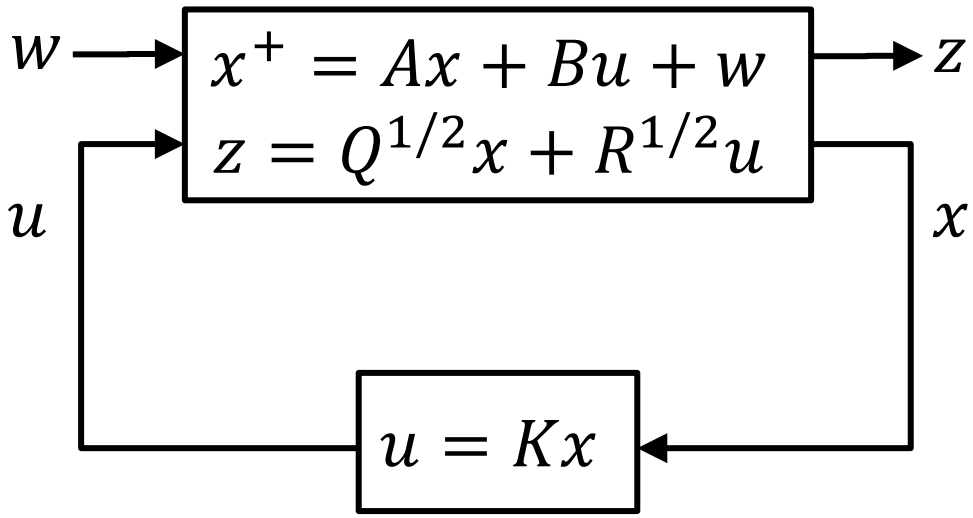
well-documented **trade-offs**

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

→ **gold standard**: adaptive + optimal + robust + cheap + tractable …& direct

Back to basics: LQR

- **cornerstone & <u>the</u> benchmark** of both optimal + adaptive control & RL



$$\underset{K}{\text{minimize}} \quad \mathbb{E}_{w_t} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} x_t^T Q x_t + u_t^T R u_t$$

$$\text{s.t.} \quad x^+ = Ax + Bu + w$$

$$u = K\,x$$

- H_2 **parameterization** of LQR

direct

offline batch | online adaptive

indirect

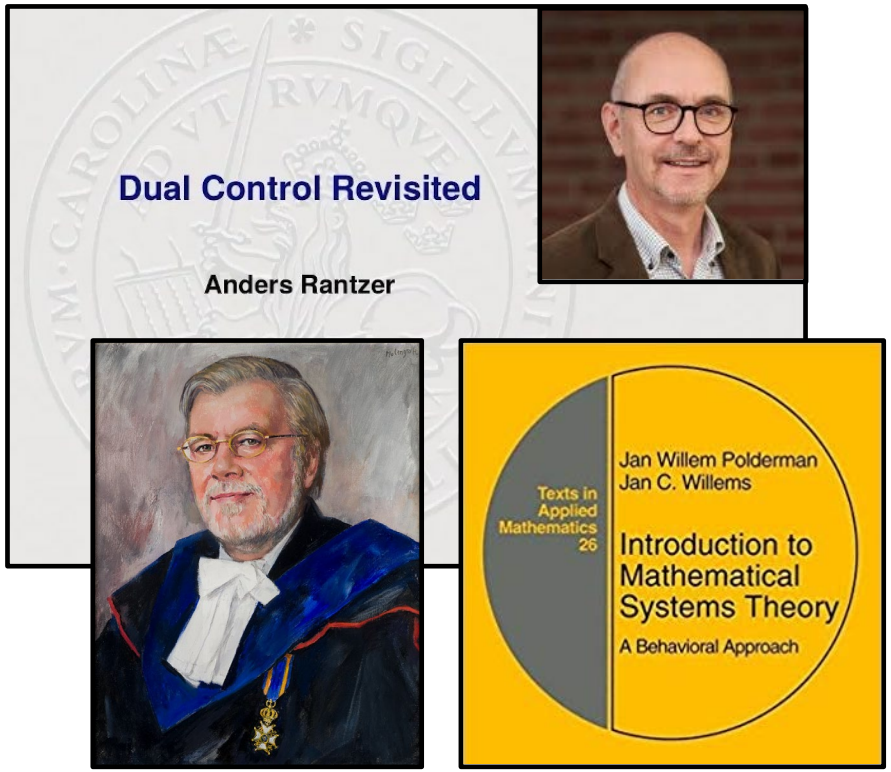
- **research gaps:** no direct + adaptive LQR & no closed-loop certificates of adaptive DP methods

Today: revisit old problems with new perspectives

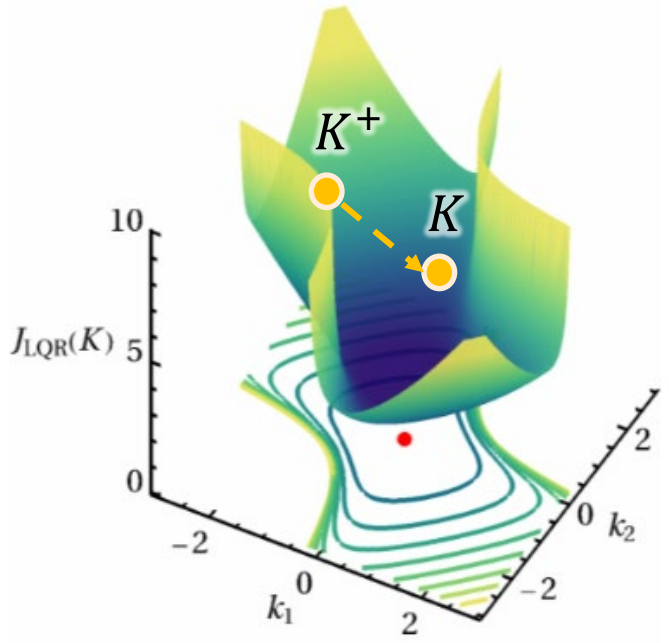
① **Behavior**: a linear system is a sub-space of trajectories that can be re-presented by models or data: trajectory matrices or **sample covariances**

② **adaptation** of optimal control following RL-style **policy gradient descent**

$$K^+ = K - \eta \nabla J(K)$$



x



[Watanabe & Zheng, 25]

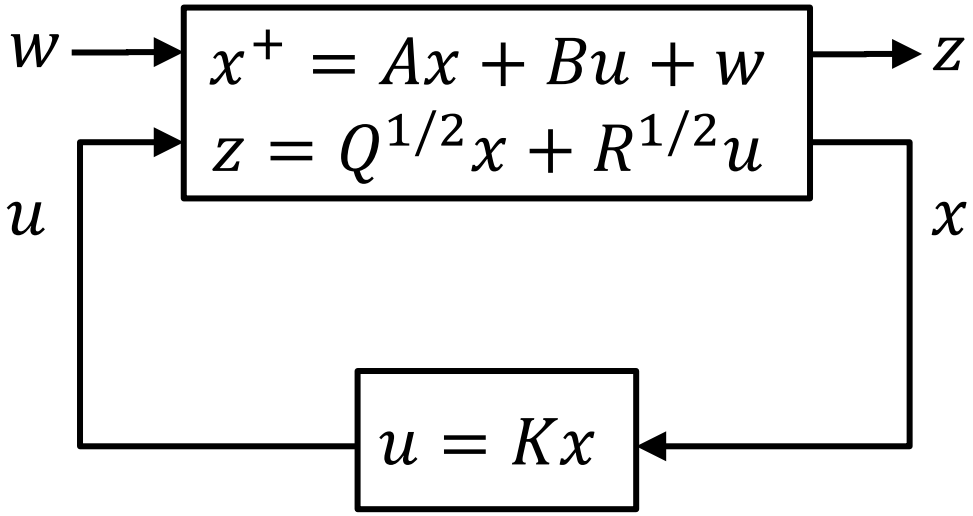
Contents

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

PROBLEM SETUP FOR ADAPTIVE LQR VIA POLICY GRADIENT

LQR parameterization

$$\min_{K} \ \mathbb{E}_{w_t} \ \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} x_t^T Q x_t + u_t^T R u_t$$
$$\text{s.t.} \quad x^+ = Ax + Bu + w$$
$$u = K\,x$$



→ with **controllability Gramian** or

state covariance $\Sigma = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} x_t x_t^T}{T}$

$$\min_{K,\Sigma\succ 0} \ Tr\big(Q\Sigma\big) + Tr\big(K^T R K \Sigma\big)$$
$$\text{s.t.} \ \ \Sigma = I + (A+BK)\Sigma(A+BK)^T$$

→ **model-based solution:** 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: rank $\begin{bmatrix} U_0 \\ X_0 \end{bmatrix} = n + m$

$$X_0 = [x_0, \dots, x_{t-1}]$$
$$U_0 = [u_0, \dots, u_{t-1}]$$
$$W_0 = [w_0, \dots, w_{t-1}]$$

$$X_1 = AX_0 + BU_0 + W_0 \rightarrow X_1 = [x_1, \dots, x_t]$$

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

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

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

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

certainty-equivalent LQR

least squares SysID

Towards an online & adaptive implementation

- **shortcoming** of separating learning and control
  → cannot improve policy **online** & cheaply / rapidly **adapt**



Adaptive Control:
Towards a Complexity-Based General Theory*

G. ZAMES-

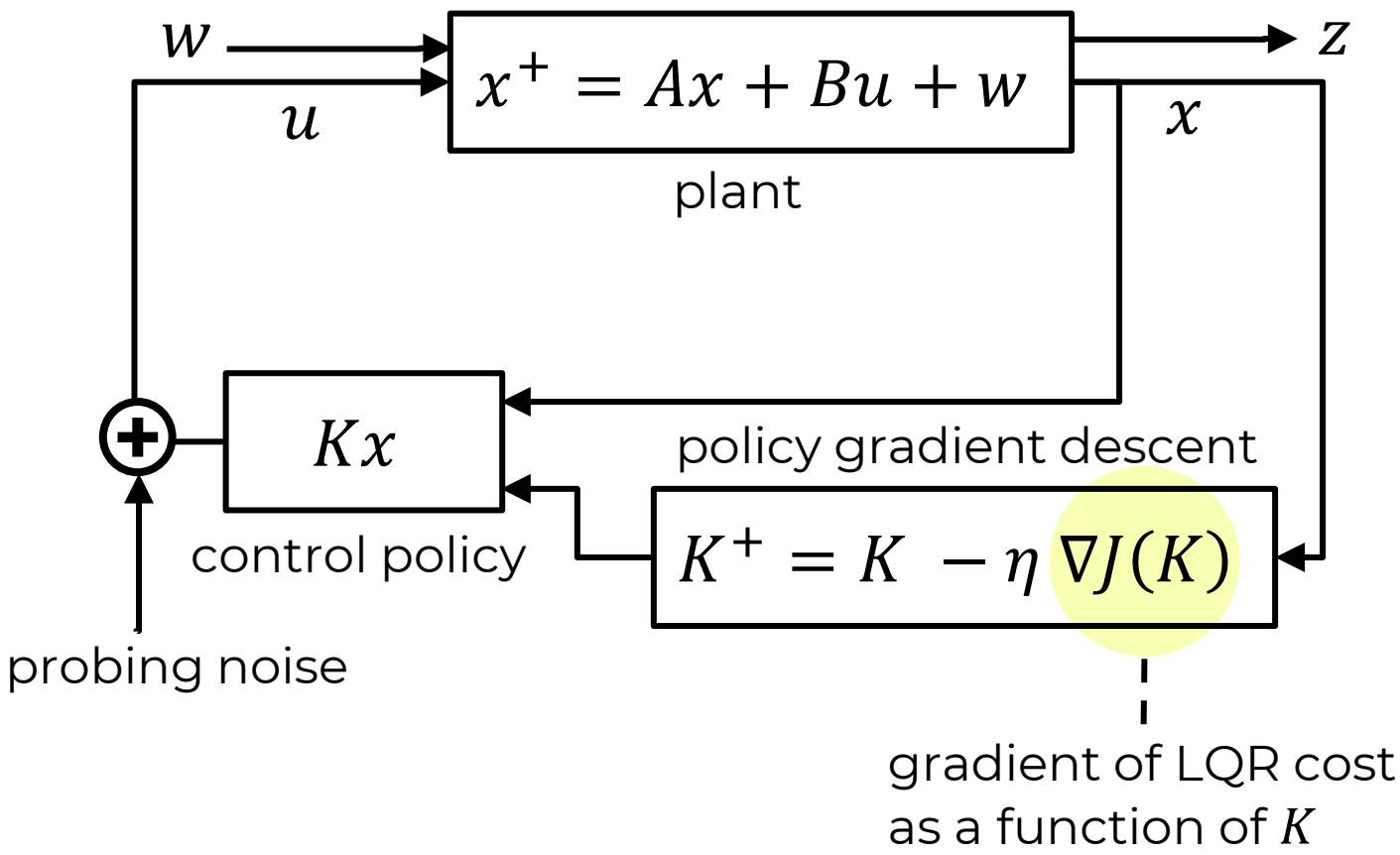
- **monotonicity principles** of adaptive control: acquire information & improve control performance over time

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

- how to "best" improve policy online → **go down the gradient !**

* 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 ?

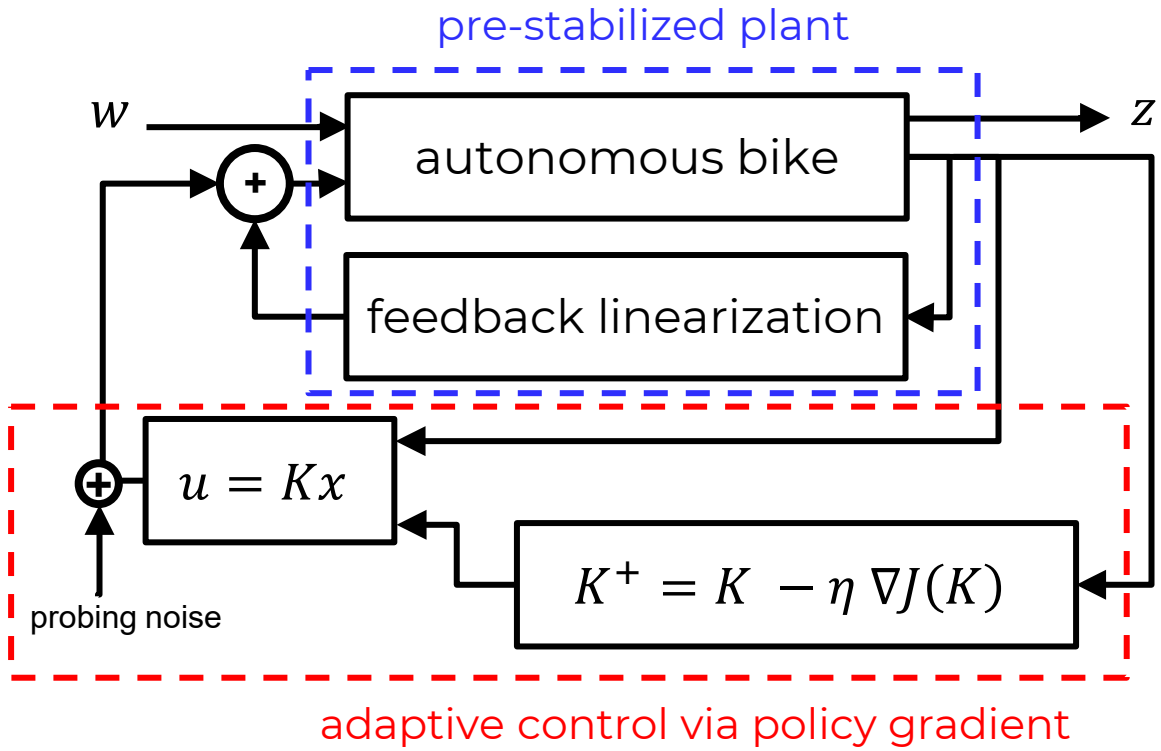
→ **closed loop**

- stability ?
- robustness ?
- optimality ?

Preview: does it work on an autonomous bike ?

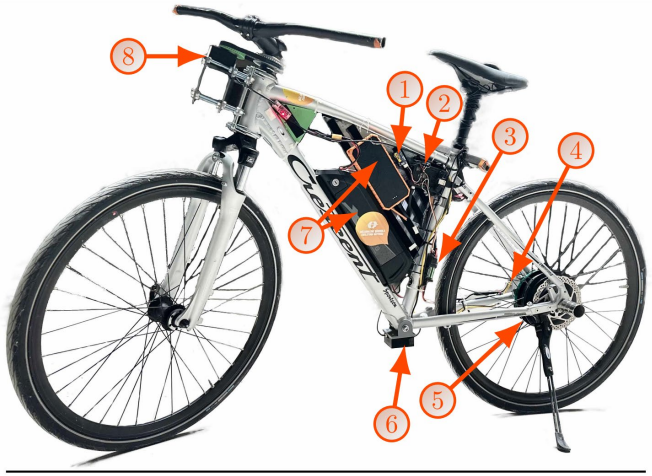


Hardware			
1	RC receiver	5	Bafang RM G040.250.DC
2	Raspberry Pi 4b	6	Xsens MTi-7
3	ESC	7	Batteries
4	Hall sensor	8	Dynamixel XH540-W270-T



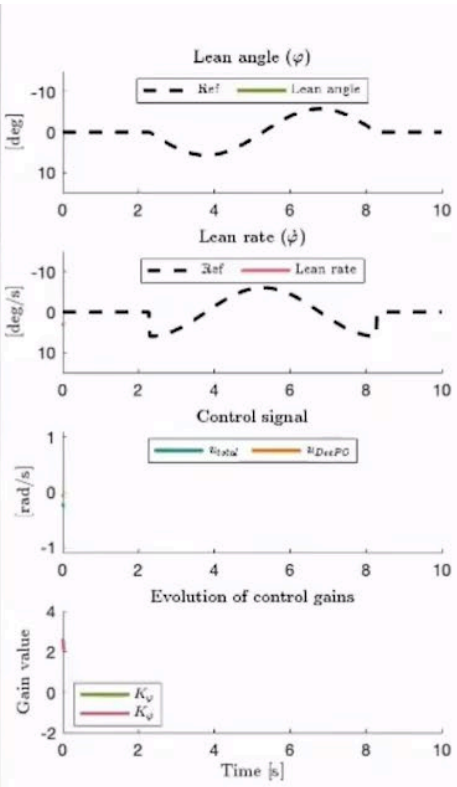
Setup: **autonomous bicycle** with **coarse inner control** (2d dynamics stabilized by feedback linearization) & **outer adaptive policy gradient**

Preview: does it work on an autonomous bike ?



Hardware			
1	RC receiver	5	Bafang RM G040.250.DC
2	Raspberry Pi 4b	6	Xsens MTi-7
3	ESC	7	Batteries
4	Hall sensor	8	Dynamixel XH540-W270-T

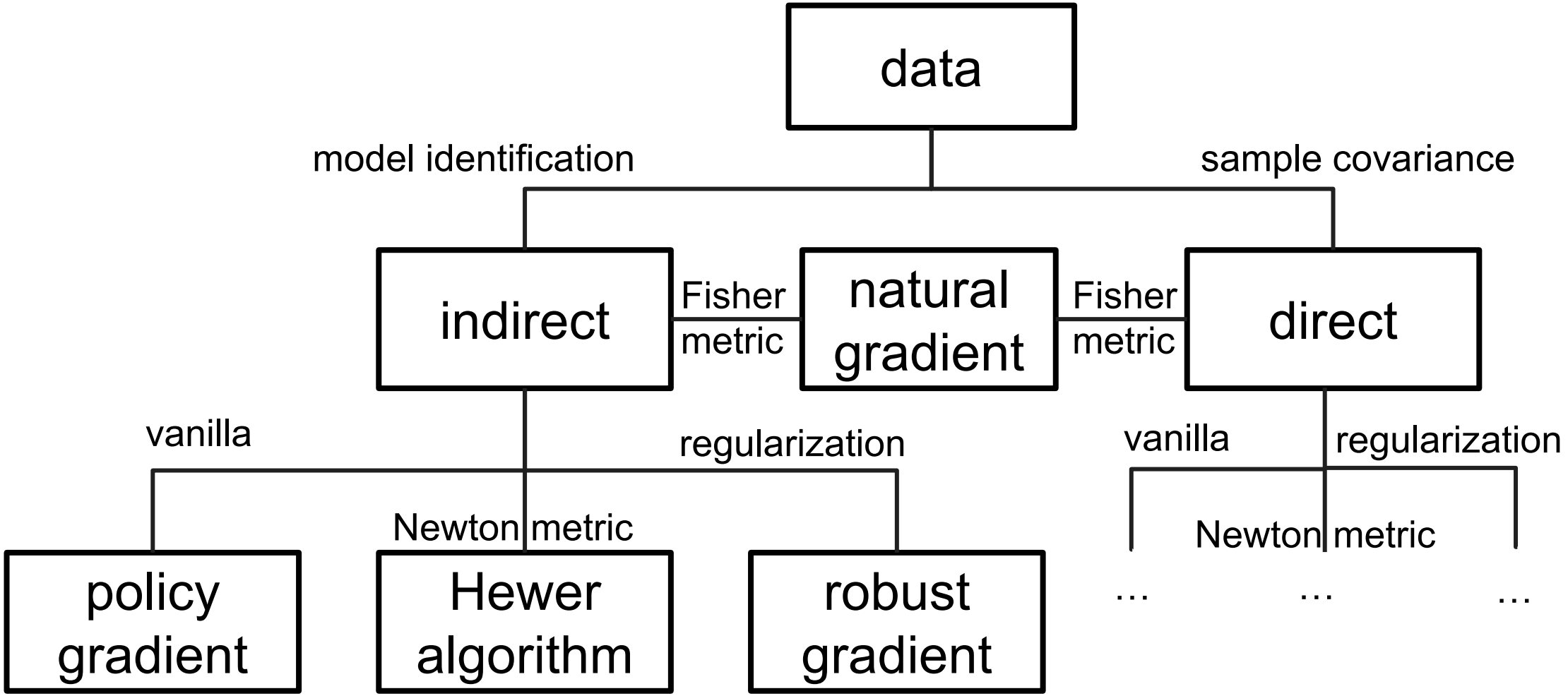




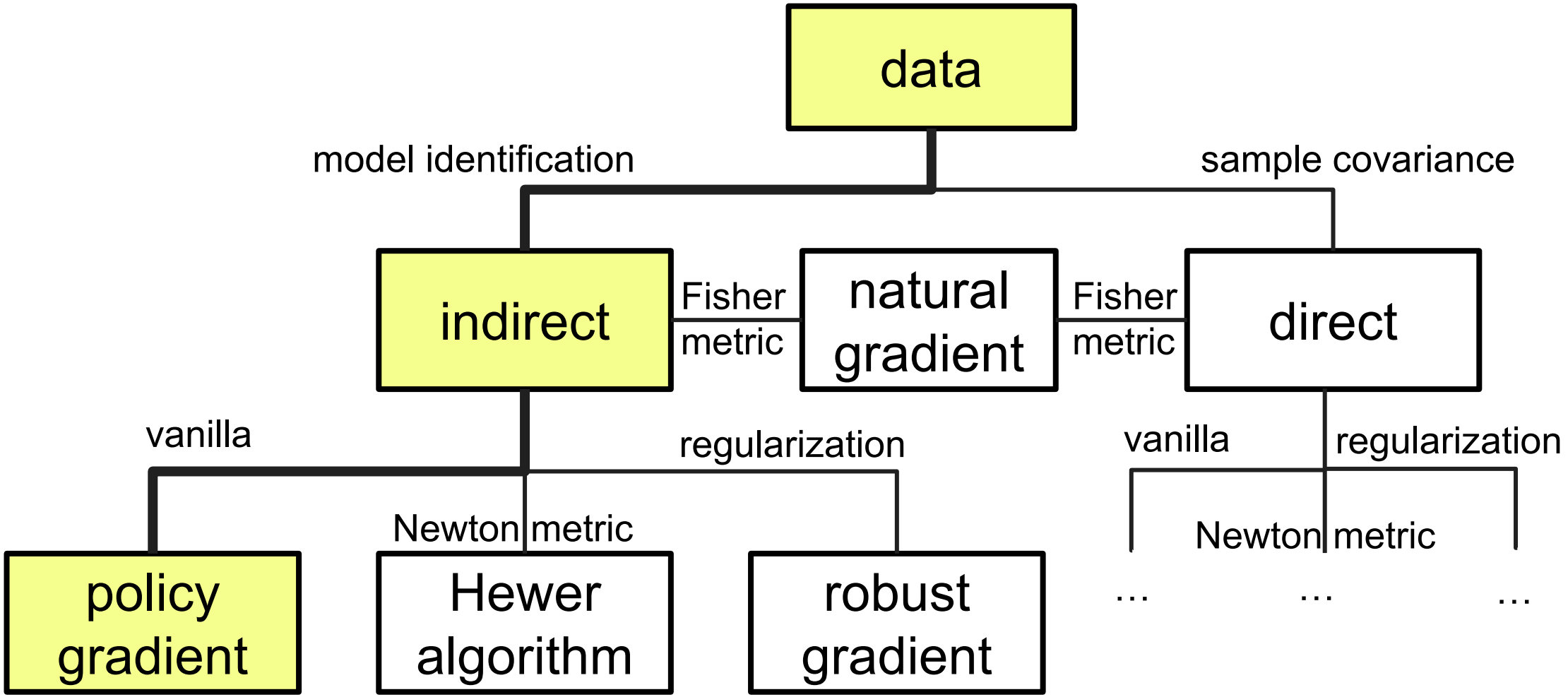
Setup: **autonomous bicycle** with **coarse inner control** (2d dynamics stabilized by feedback linearization) & **outer adaptive policy gradient**

LEARNING PIPELINES FOR POLICY GRADIENT

Algorithmic road map



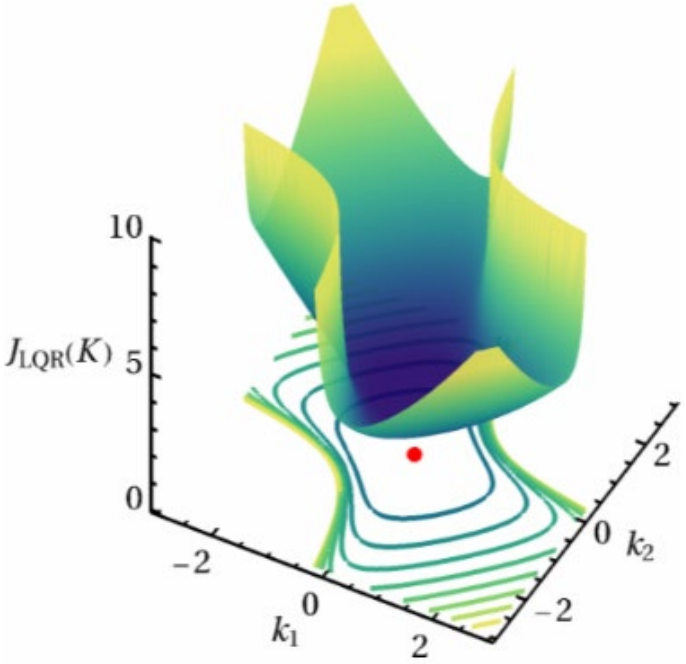
Algorithmic road map



LQR optimization landscape

$$\min_{K,\Sigma\succ 0} \; \boldsymbol{J(K)} = Tr\big(Q\Sigma\big) + Tr\big(K^T R K \Sigma\big)$$
$$\text{s.t.} \;\; \Sigma = I + (A+BK)\Sigma(A+BK)^T$$

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 $\nabla J(K) = 2\Big(\big(R + B^T P\, B\big)K - B^T P A\Big)\Sigma$ [Watanabe & Zheng, 25]

  where $P = Q + K^T R K + (A+BK)^T P (A+BK)$

  & Σ are closed-loop obs. + contr. Gramians

Gradient methods for iterative distributed control synthesis

Karl Mårtensson and Anders Rantzer

- **coercive** with compact sublevel sets
- smooth with locally **bounded Hessian**
- **gradient dominance**: $J(K) \;\leq\; J^* + \; const. \cdot \|\nabla J(K)\|^2$

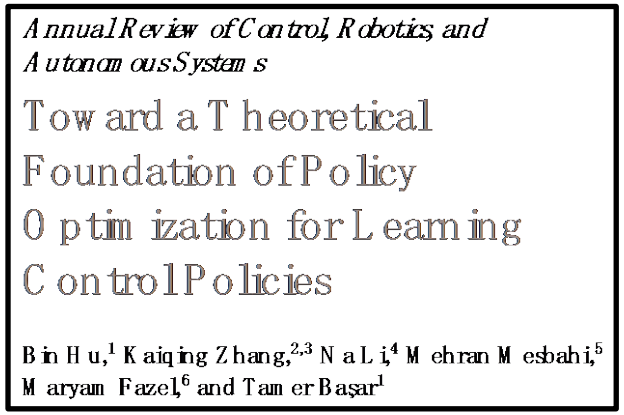
Model-based policy gradient

$$\min_K J(K) = Tr\big(Q\Sigma\big) + Tr\big(K^T R K \Sigma\big)$$

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^*.



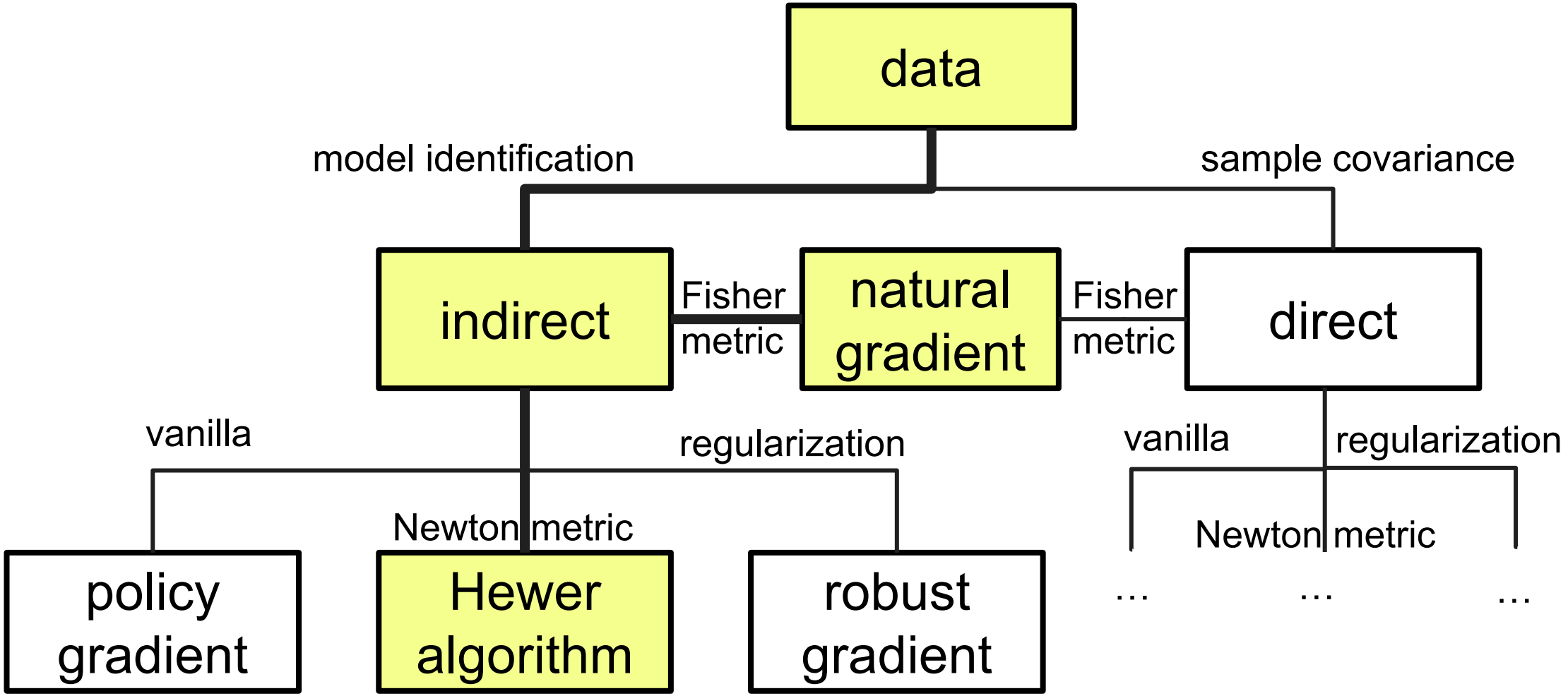


Algorithm: model-based adaptive control via policy gradient

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

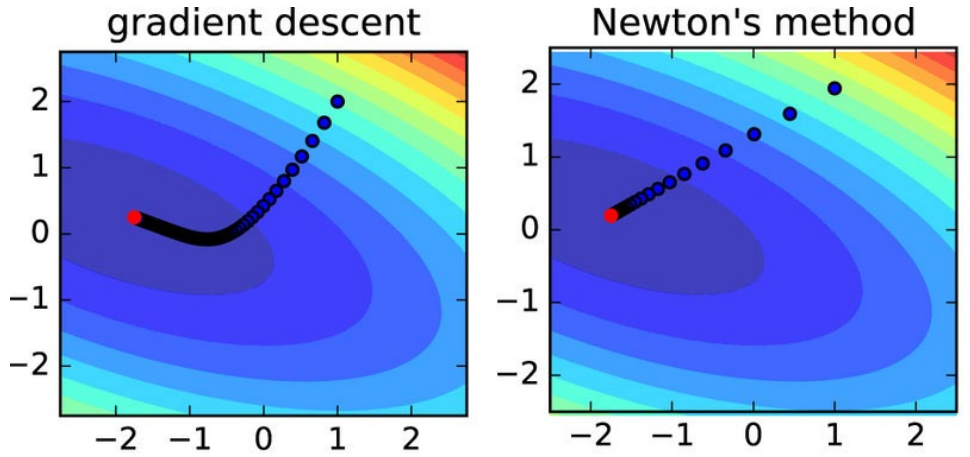
actuate & repeat

Algorithmic road map



Pre-scaled policy gradient

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



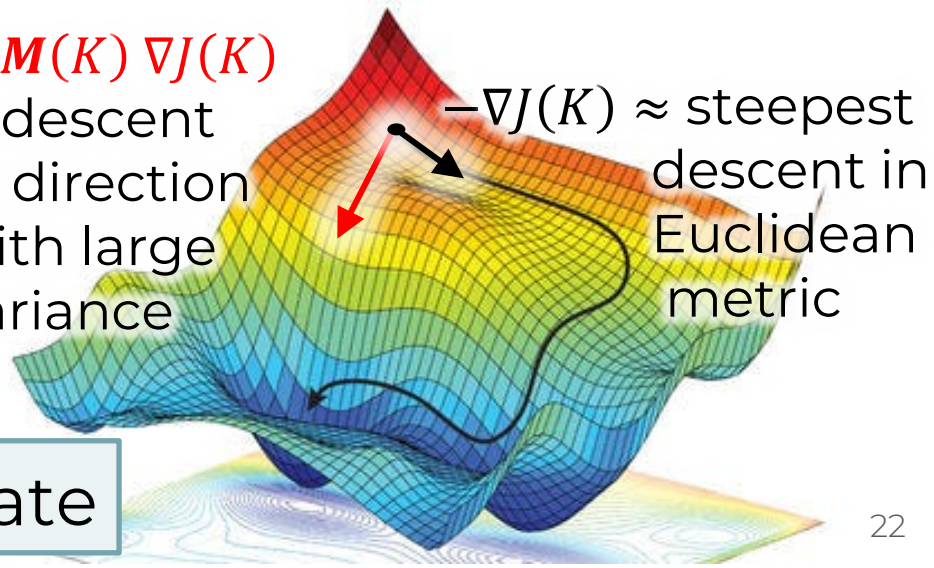
→ **Gauss-Newton**: $\boldsymbol{M(K)}$ cheaply approximates inverse Hessian $\nabla^2 J(K)$

Fact: this equals Hewer's algorithm: policy evaluation ⇆ improvement

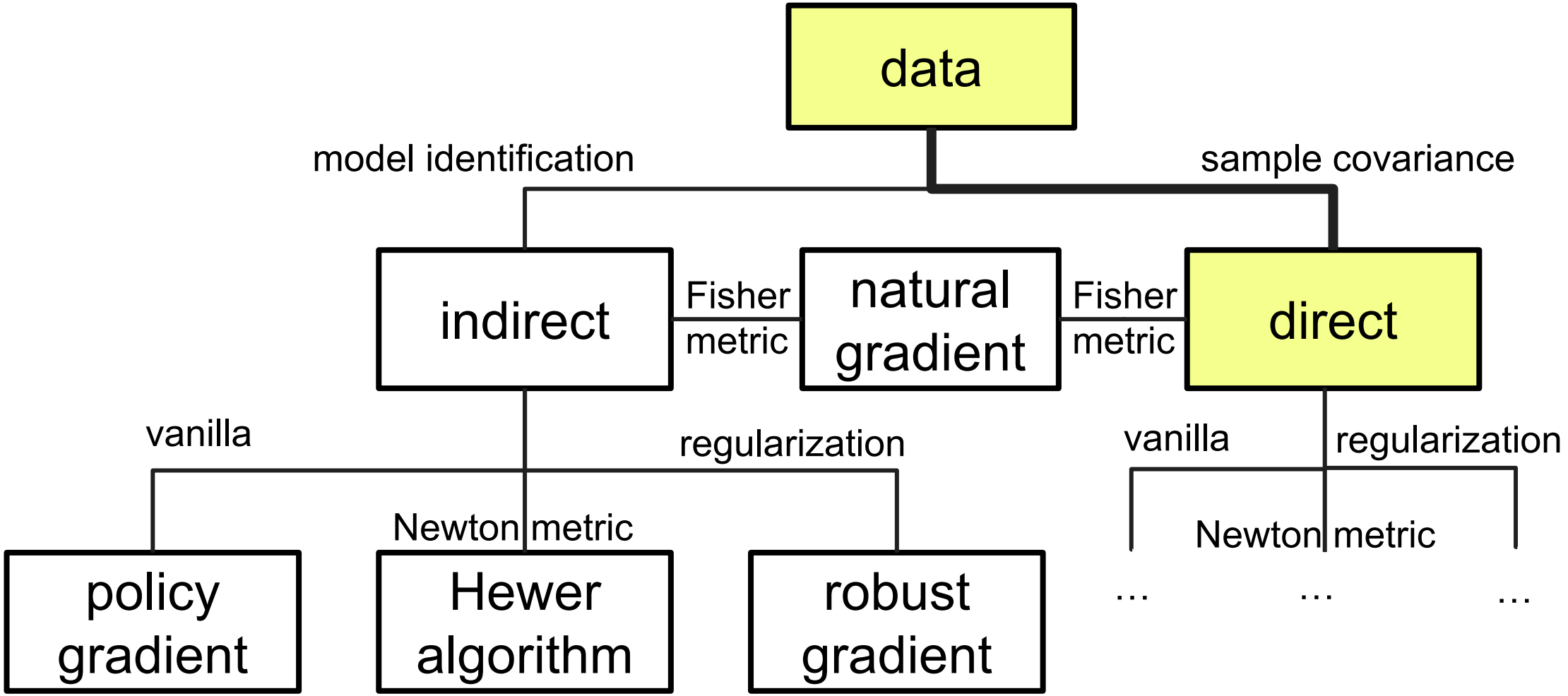
→ **Natural policy gradient** method:

$\boldsymbol{M(K)}$ is inverse Fisher information

Fact: $\boldsymbol{M(K)}\, \nabla J(K) = \nabla J(K) \Sigma^{-1}$ 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

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

from uniform perturbation U & numerous + very long trajectories

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

~ 10^7 samples for 4th order system

- direct policy gradient is inefficient, episodic, & **practically useless ?**

→ sample covariance parameterization to the rescue !

Sample covariance parametrization of the LQR (no noise)

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

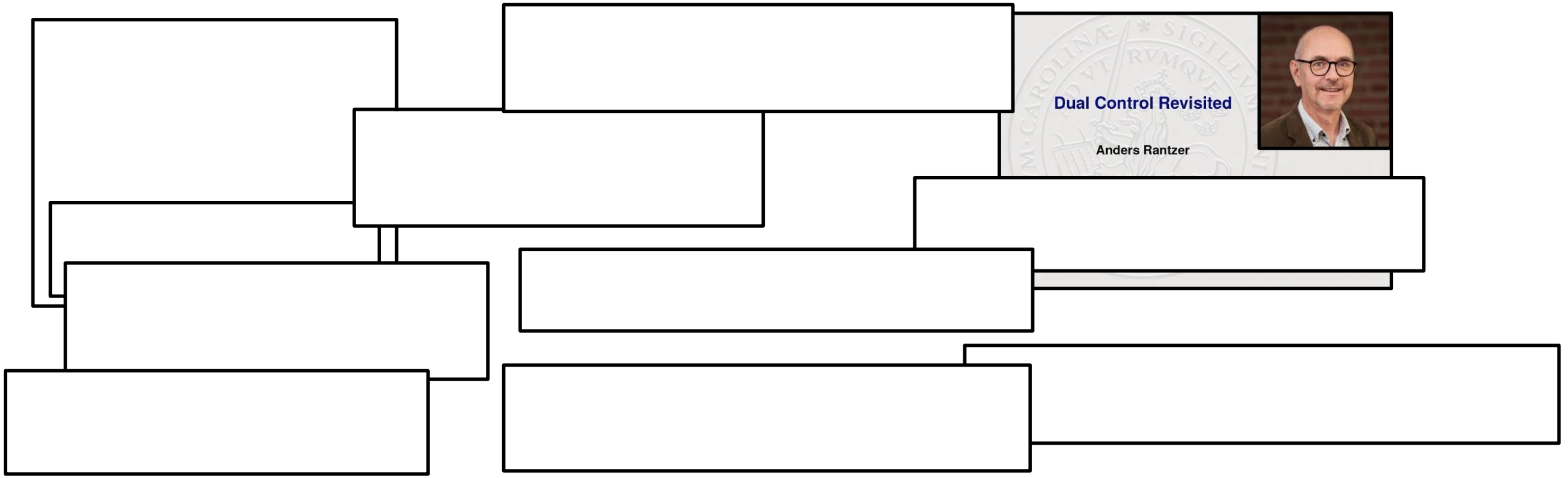
- **sample covariances:** $\Phi = \frac{1}{t}\begin{bmatrix} U_0 \\ X_0 \end{bmatrix}\begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top \succ 0$ (due to PE) & $\Phi^+ = \frac{1}{t} X_1 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top$

$$\Longrightarrow \forall K \;\; \exists V \;\; \text{s.t.} \;\; \boxed{\begin{bmatrix} K \\ I \end{bmatrix} = \frac{1}{t}\begin{bmatrix} U_0 \\ X_0 \end{bmatrix}\begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top V} \quad (\star)$$

- **closed loop:** $A + BK = [B \;\; A]\begin{bmatrix} K \\ I \end{bmatrix} = [B \;\; A]\frac{1}{t}\begin{bmatrix} U_0 \\ X_0 \end{bmatrix}\begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top V = \frac{1}{t} X_1 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top V = \Phi^+ V$

- **direct data-driven** formulation by substituting $A + BK = \Phi^+ V$ & $(\star)$

Aside: covariance parametrization on the rise



Related to trajectory matrices, but

- matrices independent of data size
- uniqueness & no regularization
- recursive rank-1 updates

Covariance parametrization of policy gradient

- **covariance parameterization:**

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

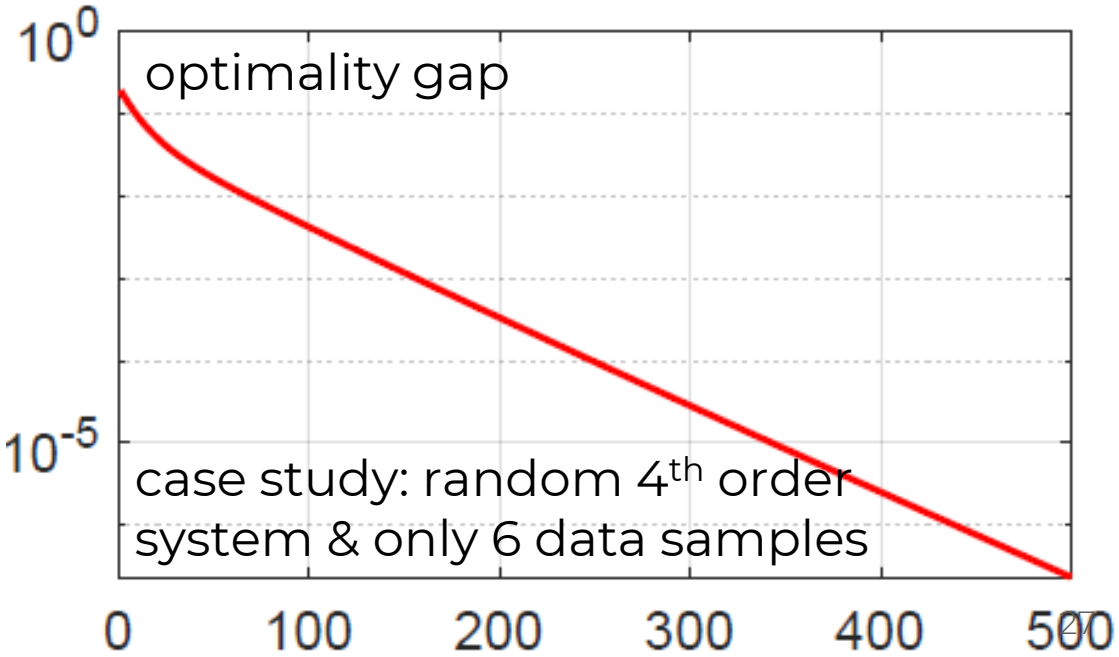
  linear constraint $\begin{bmatrix} K \\ I \end{bmatrix} = \Phi\, V$

$$\min_{K,\Sigma \succ 0, V} \; Tr(Q\Sigma) + Tr(K^T R K \Sigma)$$
$$\text{s.t.} \;\; \Sigma = I + (\Phi^+ V)\Sigma(\Phi^+ V)^T$$
$$\text{s.t.} \;\; \Sigma = I + (A + BK)\Sigma(A + BK)^T$$
$$\begin{bmatrix} K \\ I \end{bmatrix} = \Phi\, V$$

- analogous optimization problem in V-coordinates with linear constraint, where K can be eliminated

- **direct & projected policy gradient**

$$V^+ = V - \eta\, \Pi\left(\nabla J(V)\right)$$



Facts on direct policy gradient

- **covariance parameterization:** substitute $A + BK = \Phi^+ V$ with $\begin{bmatrix} K \\ I \end{bmatrix} = \Phi V$

- **direct & projected policy gradient descent** $V^+ = V - \eta \, \Pi \left(\nabla J(V) \right)$

Fact 1: in original K –coordinates reads as **scaled gradient descent**

$$K^+ = K - \eta \, \boldsymbol{M_t} \, \nabla J(K) \, ,$$

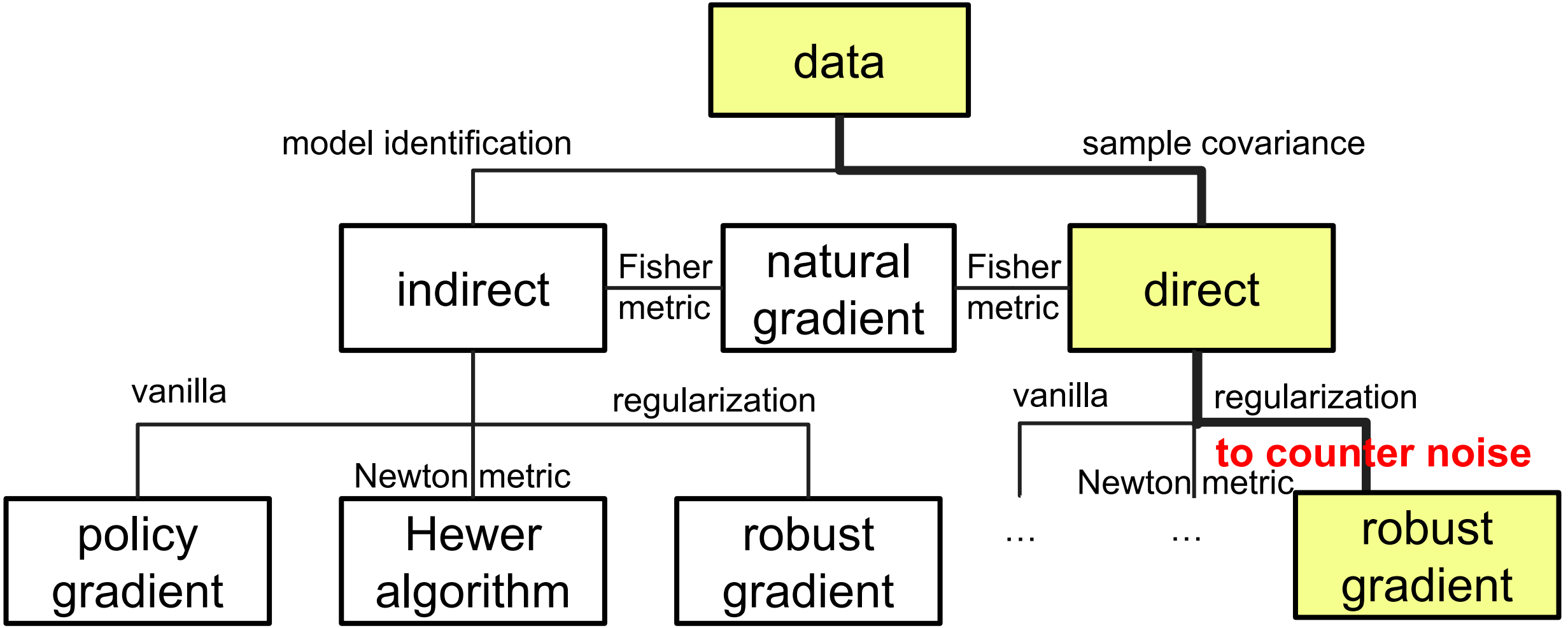
where $\boldsymbol{M_t} = \frac{1}{t^2} \, U_0 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top \Pi \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} \, U_0^T \succ 0 \, .$

$\longrightarrow$ retain strong convergence result

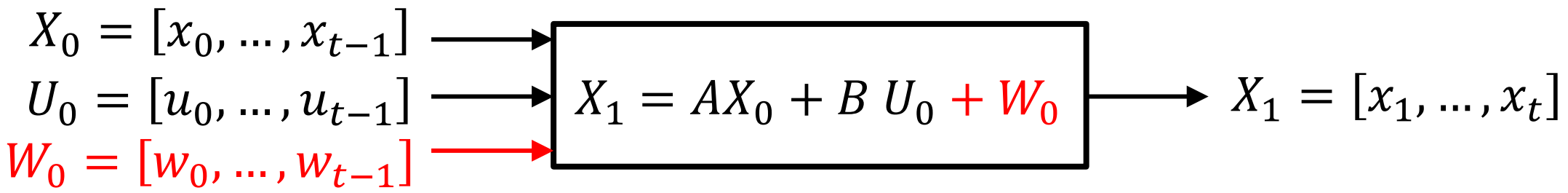
Fact 2: the corresponding **natural gradient descent is invariant**, i.e., equal to the model-based natural gradient descent.

$\longrightarrow$ recover original result

Algorithmic road map



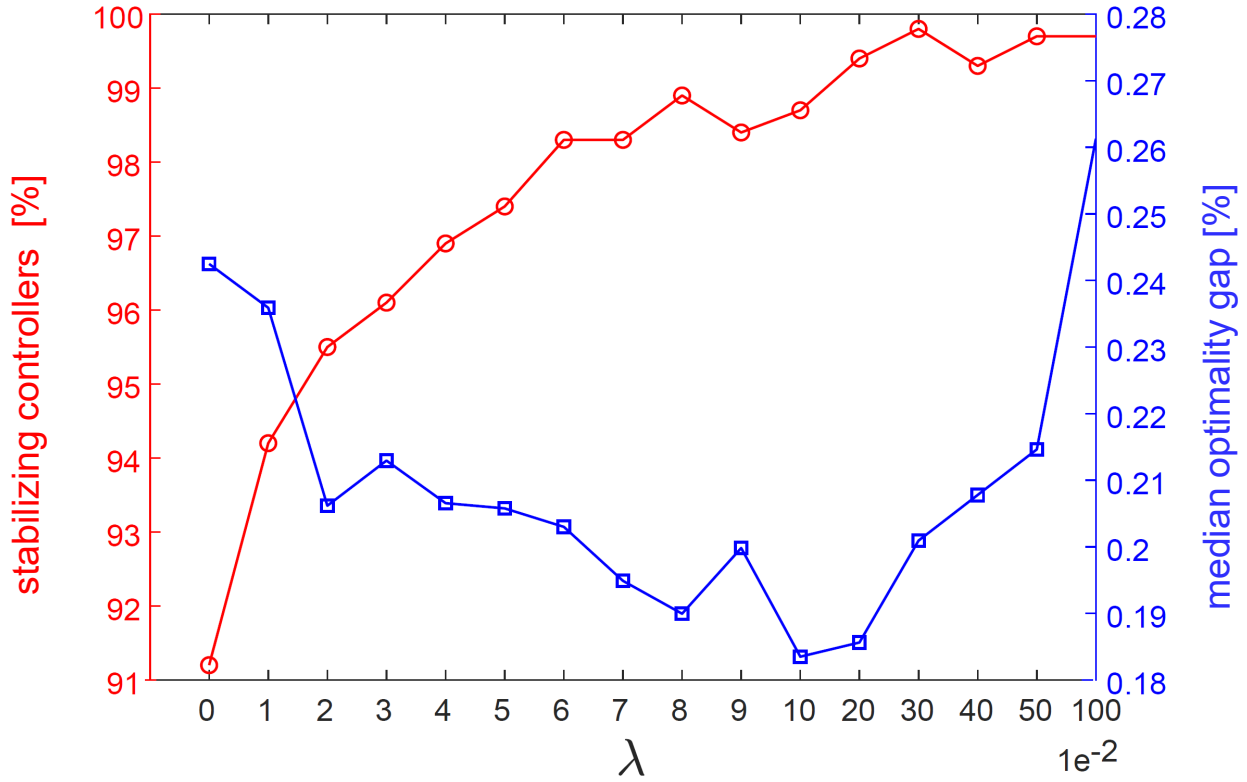
Robustifying covariance regularization of the LQR

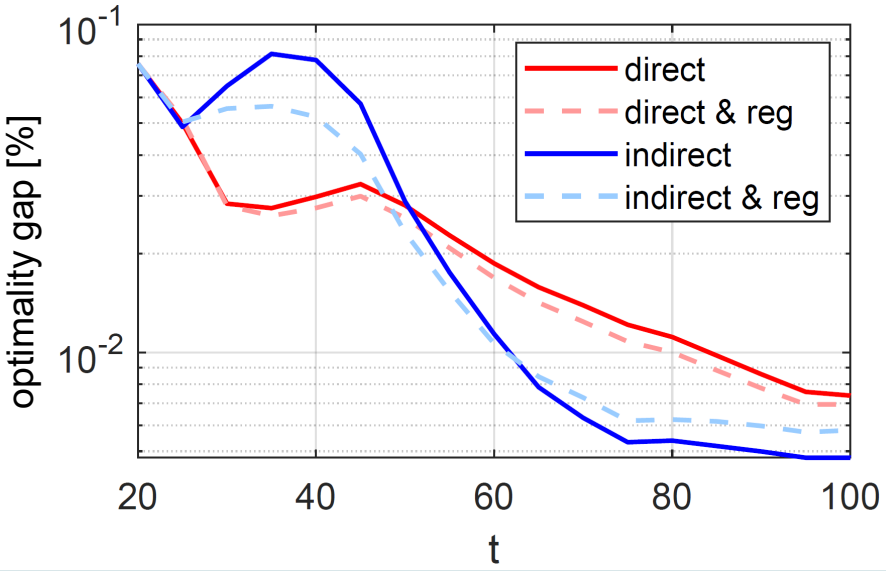


- **closed loop:** $A + BK = (\Phi^+ - \widetilde{W}_0)V$ where $\widetilde{W}_0 = \frac{1}{t} W_0 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top$ (neglected before)
- **difference in Lyapunov equn's** with/without noise $\sim \boldsymbol{V\Sigma V^T \Phi}$
- **regularized LQR**: $J(V) + \lambda \cdot Tr\left(\boldsymbol{V\Sigma V^T}\Phi\right) = J(K) + \lambda \cdot Tr\left(\boldsymbol{\Phi^{-1}} \begin{bmatrix} K \\ I \end{bmatrix} \boldsymbol{\Sigma} \begin{bmatrix} K \\ I \end{bmatrix}^T\right)$
- optimization class does not change → **retain convergence result**

Effect of regularization

- regularizing with λ robustifies & gives better performance
- improves algorithmic stability for indirect & direct methods
- decrease λ as data grows



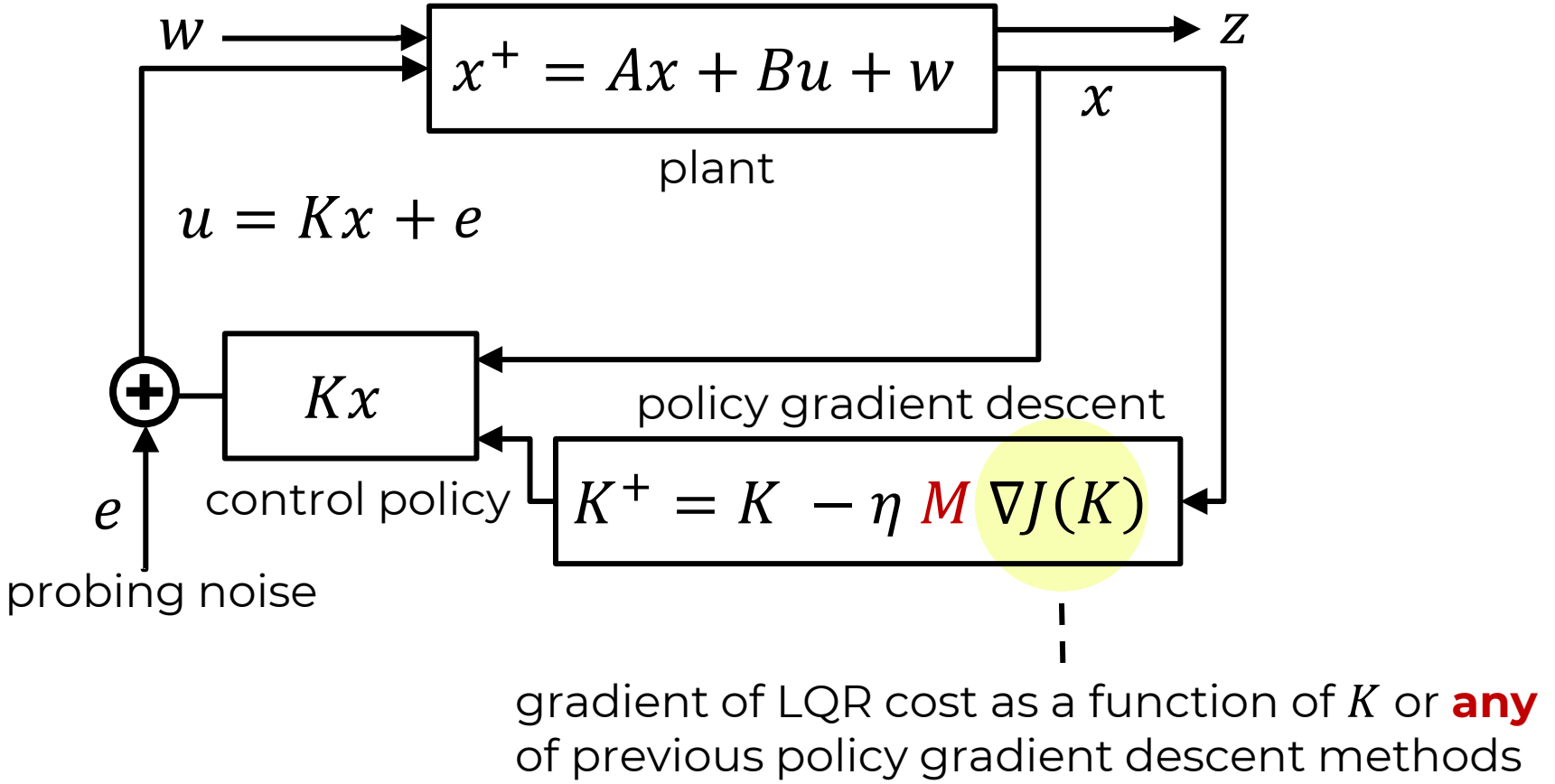


Comparison of stability % P & median optimality gap $\mathcal{M}$

	$\lambda_t = 0$	$\lambda_t = 1/(10\sqrt{t - t_0})$	one-shot-based
direct	$\mathcal{P} = 81\%$ $\mathcal{M} = 0.0014$	$\mathcal{P} = 98\%$ $\mathcal{M} = 0.0011$	$\mathcal{P} = 82\%$ $\mathcal{M} = 0.0013$
indirect	$\mathcal{P} = 83\%$ $\mathcal{M} = 0.0014$	$\mathcal{P} = 99\%$ $\mathcal{M} = 0.0014$	$\mathcal{P} = 82\%$ $\mathcal{M} = 0.0013$

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: it's a switched system, ... so no ?

Information metric

- **persistency of excitation** due to probing:

  $\underline{\sigma}(\Phi_t) \geq \gamma_t$ for some $\gamma_t > 0$



Adaptive Control:
Towards a Complexity-Based General Theory*

G. ZAMES-

- **noise & data covariance**:

  $\delta_t := \left\| \frac{1}{t} W_0 D_0^T \right\|$

- information metric = **signal-to-noise ratio** $SNR_t := \gamma_t / \delta_t$

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

Uniform $\delta_t \sim O(1)$	Constant $\gamma_t \sim O(1)$	$SNR_t \sim O(1)$
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 $\nu_i > 0, i \in \{1,2,3,4,5\}$, depending on (A, B, Q, R, K_0) with $\nu_4 < 1$, so that, if $SNR_t \geq \nu_1 \, \forall t$, $\eta \leq \nu_2$, & for stable K_0

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

$$|x_t| \leq \nu_3 \nu_4^t |x_0| + \nu_5 \max_{0 \leq i < t} |Be_i + w_i|.$$

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

$$C(K_t) - C^* \leq \left(1 - \frac{\eta}{\nu_6}\right)^t \left(C(K_{t_0}) - C^*\right) + O(SNR_t^{-1})$$

SNR & step-size requirements

stable initialization

nominal exponential convergence

bias due to noise

Notes on stability & convergence statement

- **Key to the analysis:** small enough step size to control learning rate & assure **sequential stability**

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

  → Zames' 2nd monotonicity principle: improving performance

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

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

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

- **direct methods:** $SNR_t \geq \frac{const.\|M_t\|}{\underline{\sigma}(M_t)}$, $\eta_t \leq \frac{const.}{\|M_t\|}$, & convergence rate depend on data-dependent matrix $M_t = \frac{1}{t^2} U_0 \begin{bmatrix} U_0 \\ X_0 \end{bmatrix}^\top \Pi \begin{bmatrix} U_0 \\ X_0 \end{bmatrix} U_0^T$

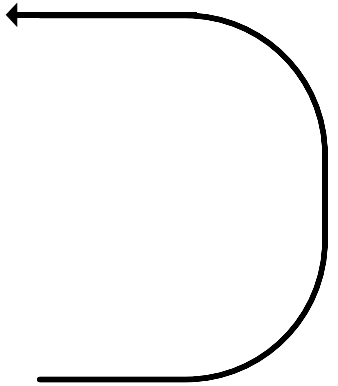
Notes on stability & convergence statement

- all results also hold in **regularized setting** under proper choice of regularization coefficient $\lambda_t \leq O(\delta_t)$ (const. for bounded noise)

- Indirect Gauss-Newton method = adaptive version of **Hewer's algorithm**, which needs additionally K_0 sufficiently close to K^*

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:** $P^+ = \text{Lyapynov}\,(\hat{B}, \hat{A}, K)$
4. **policy improvement:** $K^+ = (\dots)^{-1} P^+ \hat{B} \hat{A}$



actuate & repeat

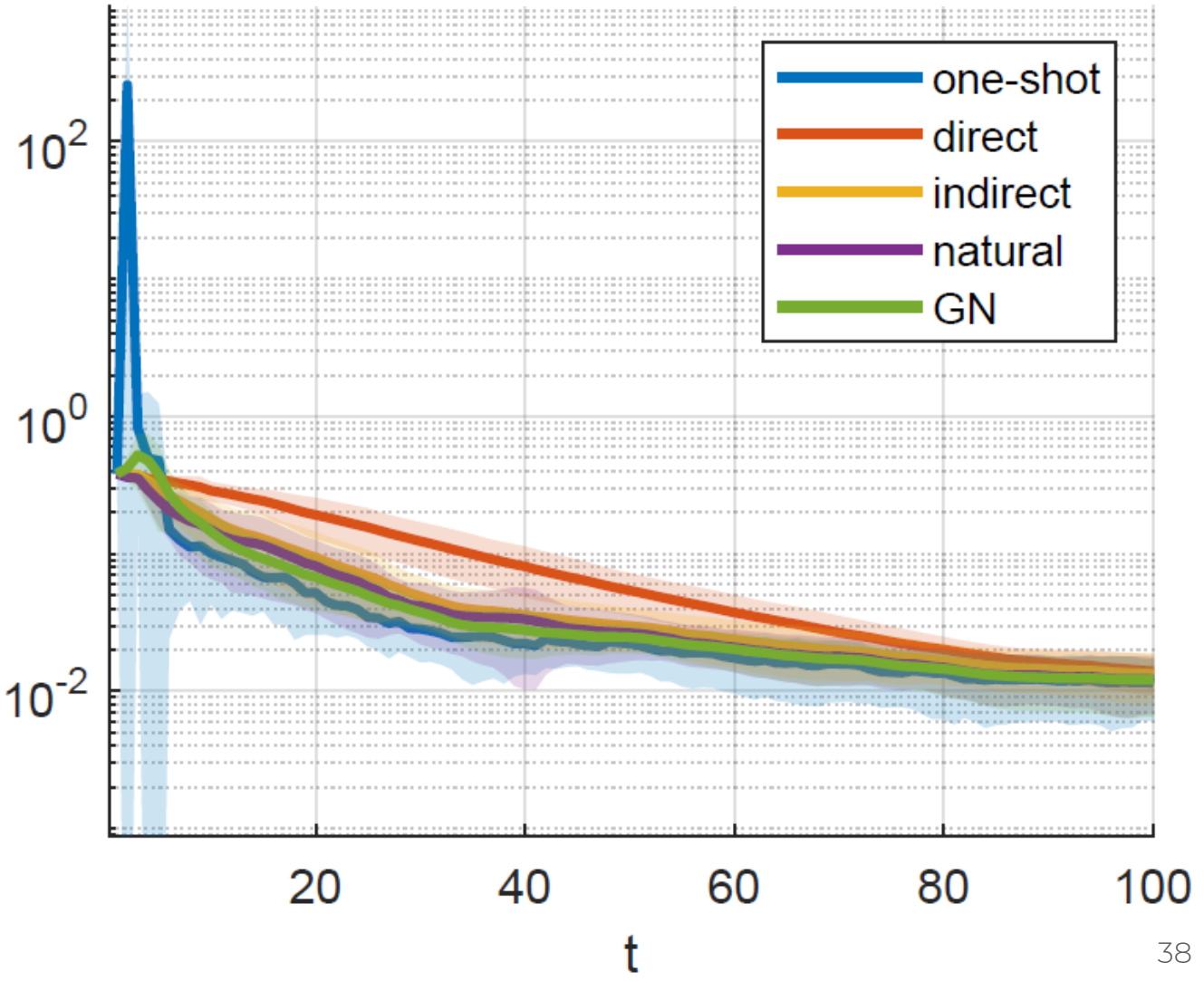
Numerics: convergence to optimality

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

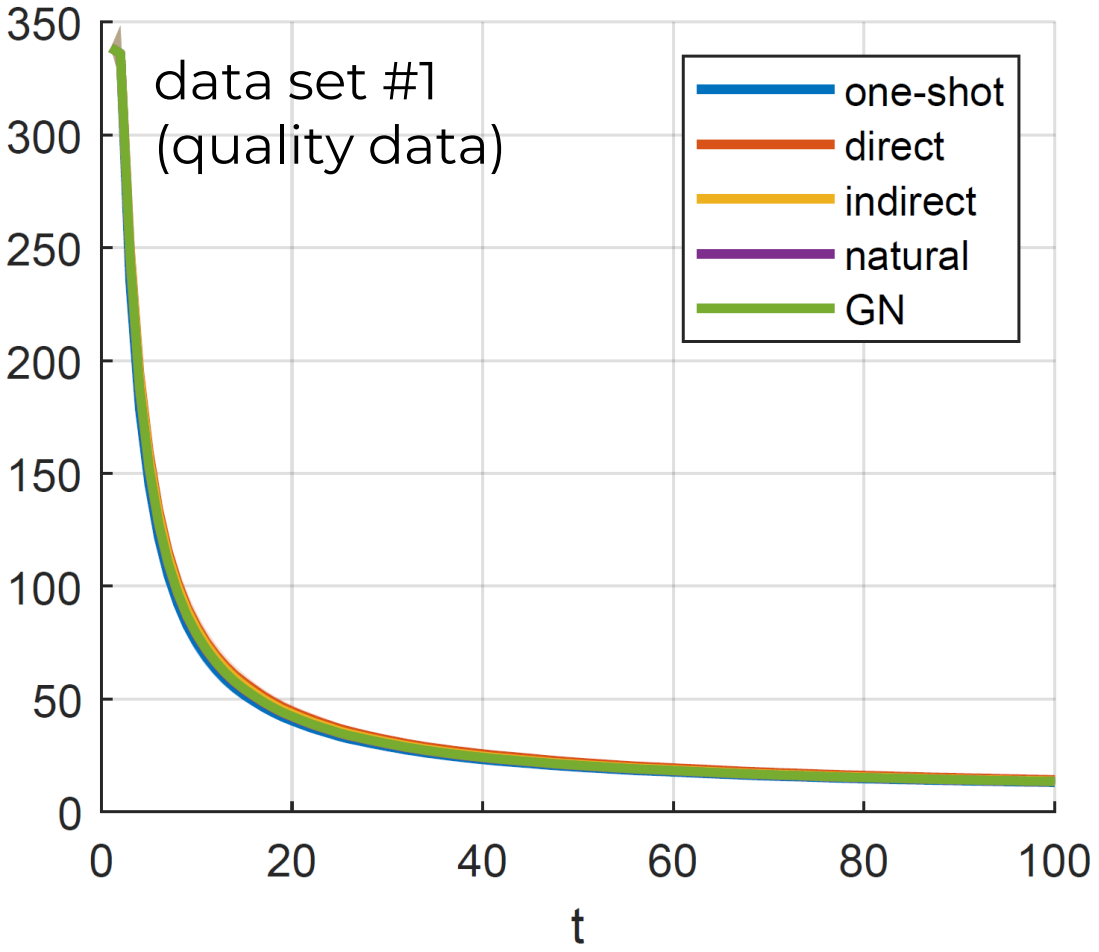
$$A = \begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix}, \quad B = I$$

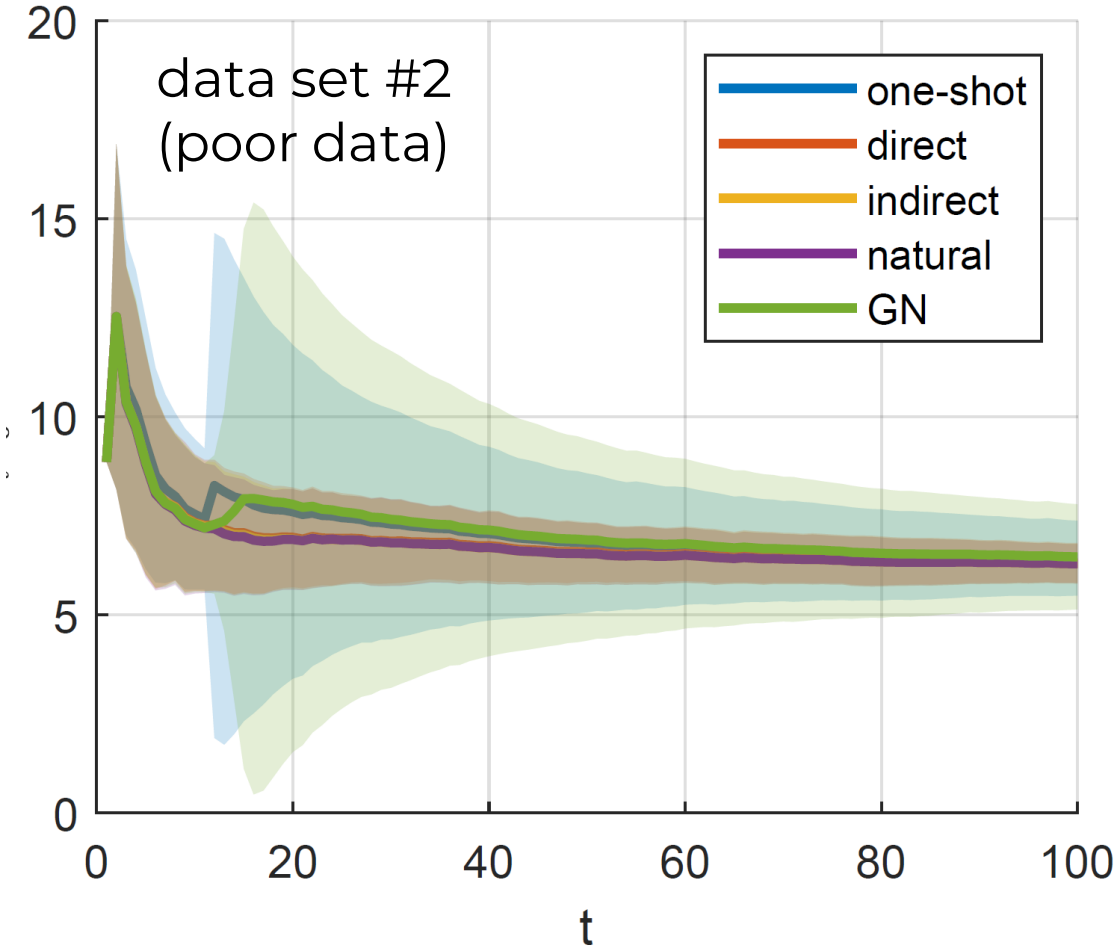
- policy gradient methods are more **robust** to noise versus sensitive one-shot method
- **empirically** observe tighter optimality gap $\sim O(SNR^{-2})$ than our certificate $O(SNR^{-1})$

optimality gap: mean $\pm$ std



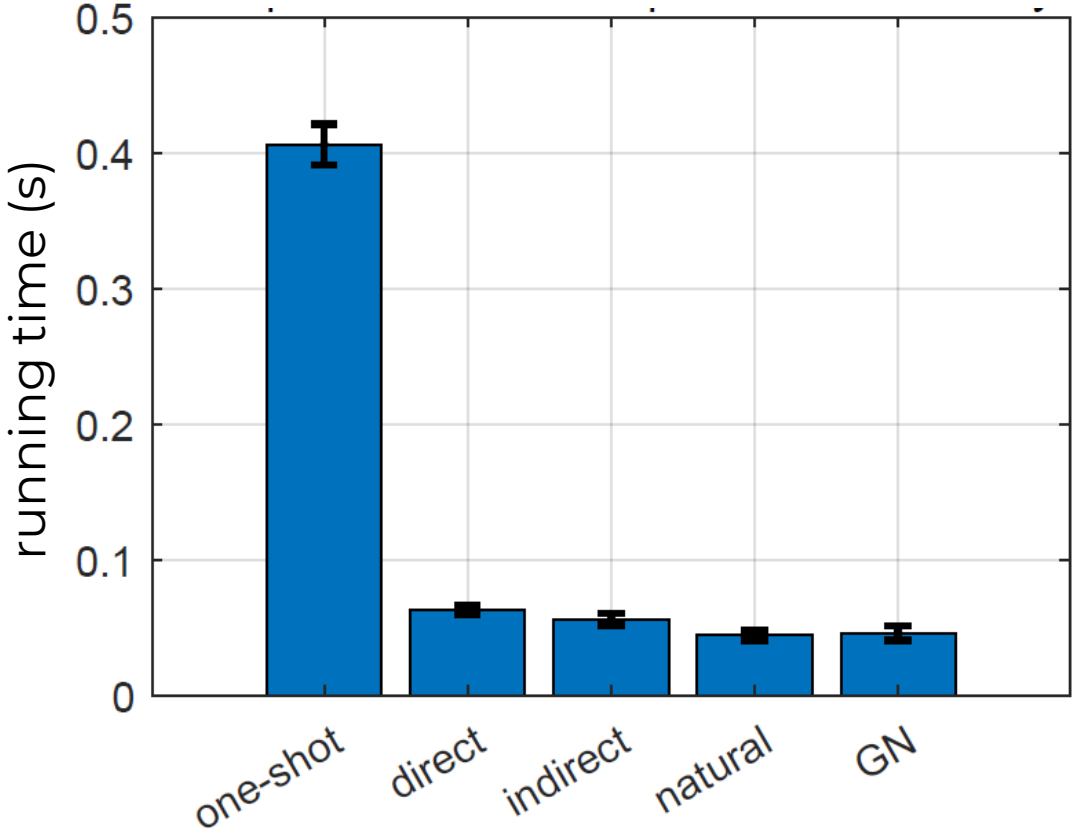
Numerics: mean ± std of closed-loop realized cost

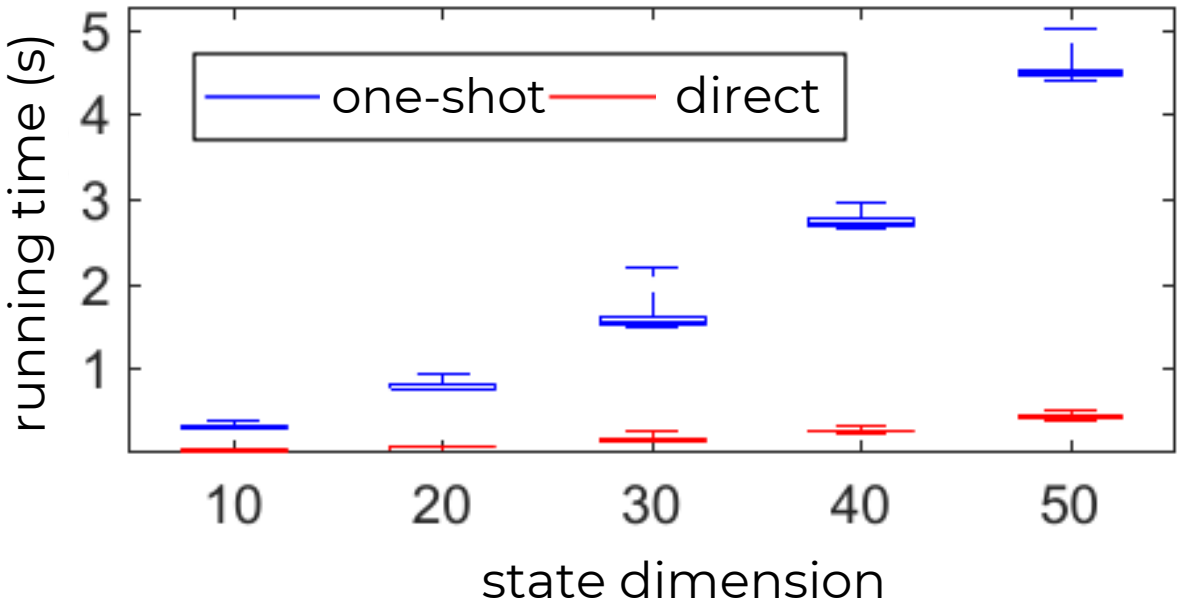




→ all converge with a bias, but one-shot & Gauss-Newton less robust

Numerics: computational efficiency

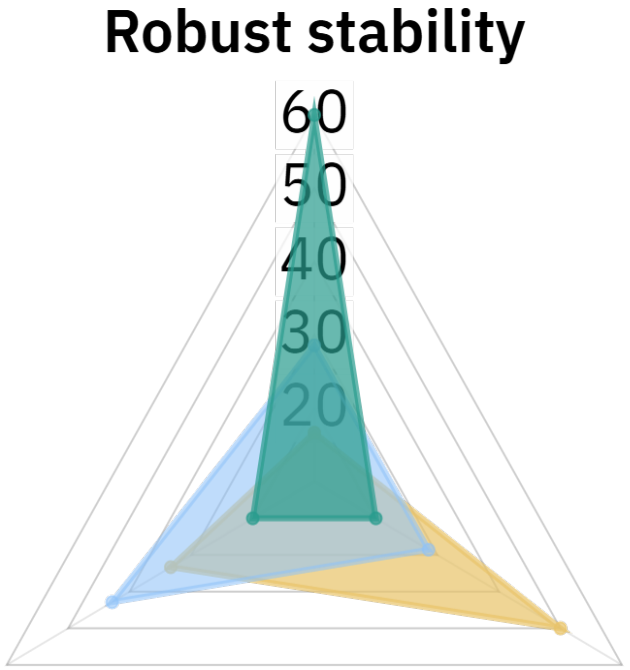




→ all policy gradient methods significantly outperform one-shot-based method in **computational effort**

Comparison among different gradients

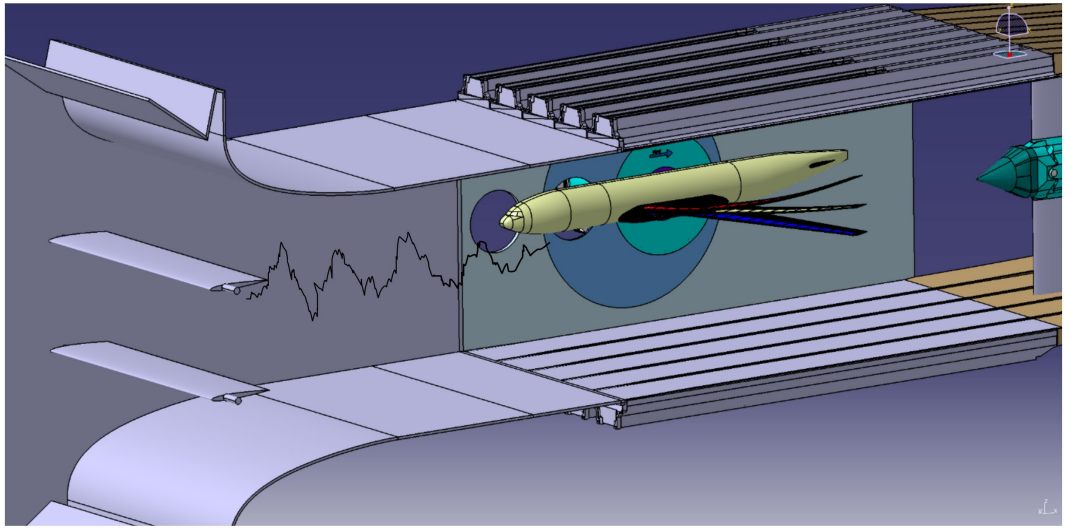
- Vanilla gradient
- Natural gradient
- Gauss-Newton gradient

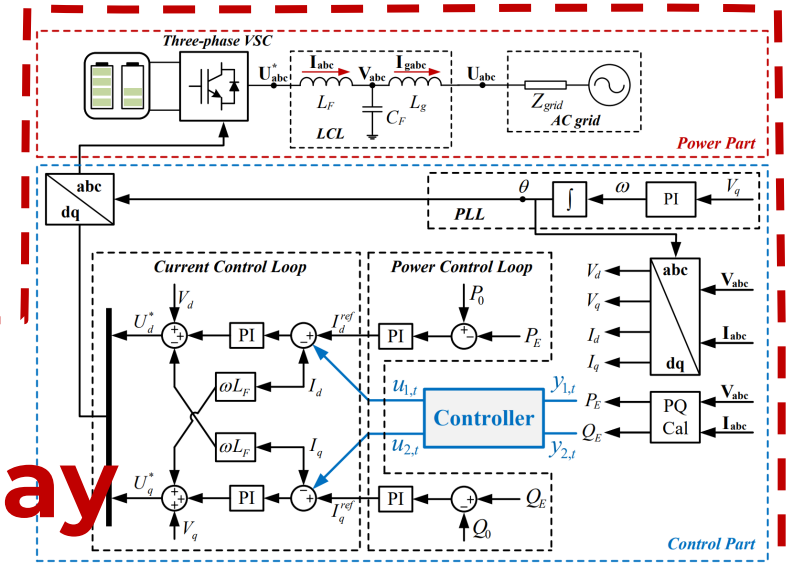


→ Best option: natural gradient (fast computing & convergence)+ regularization (promoting robustness)

Implementations

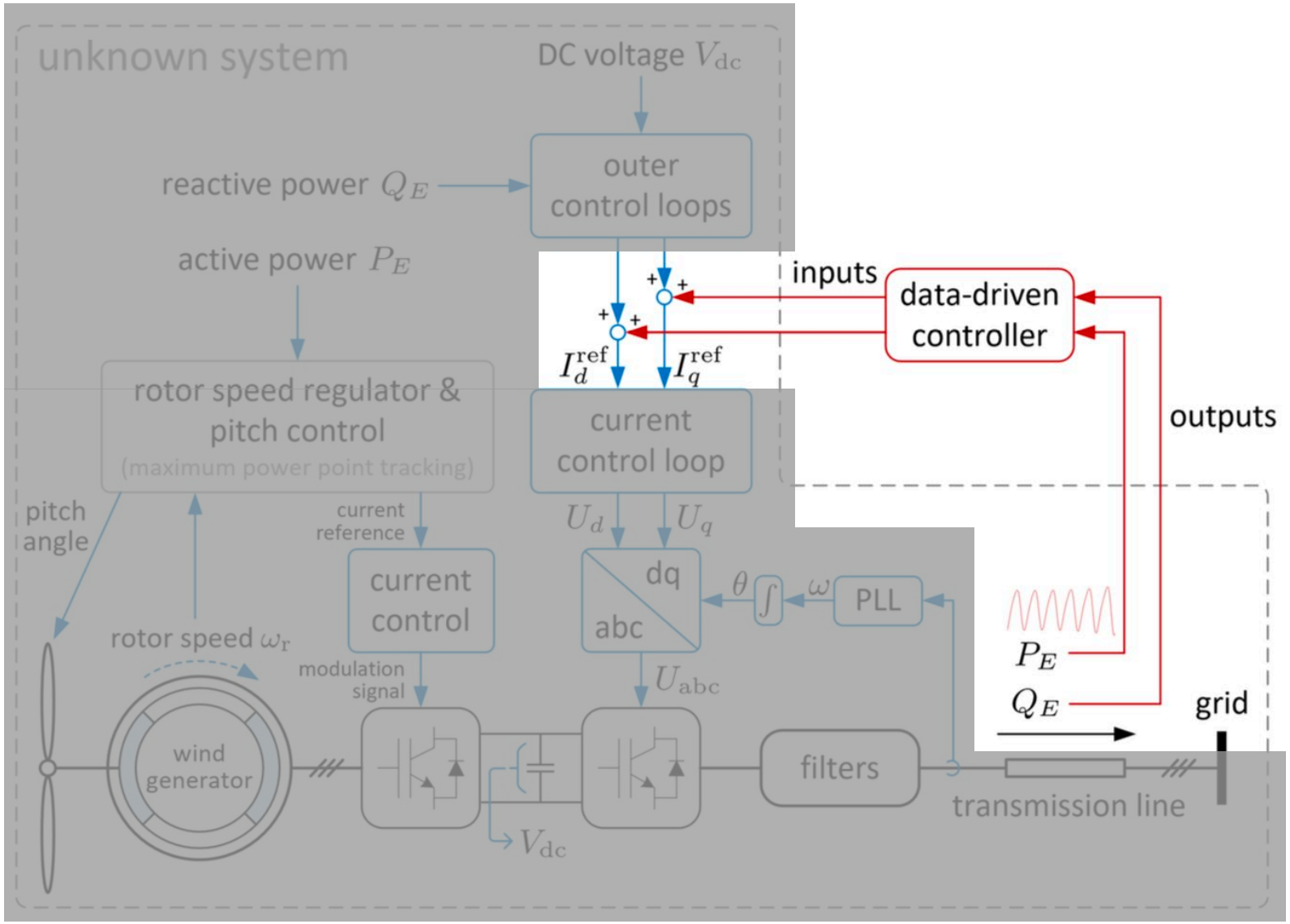






today

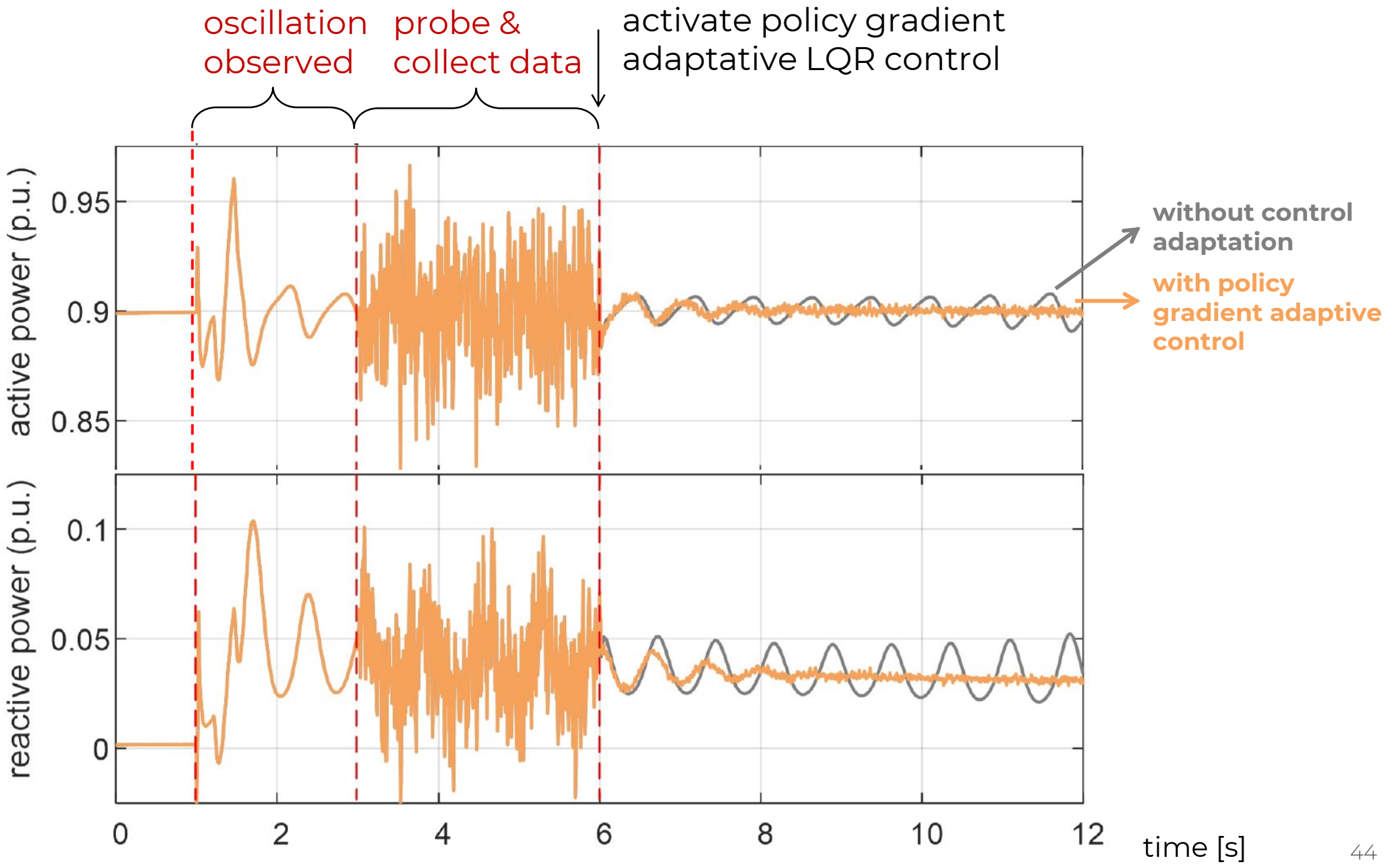
Power systems / electronics case study

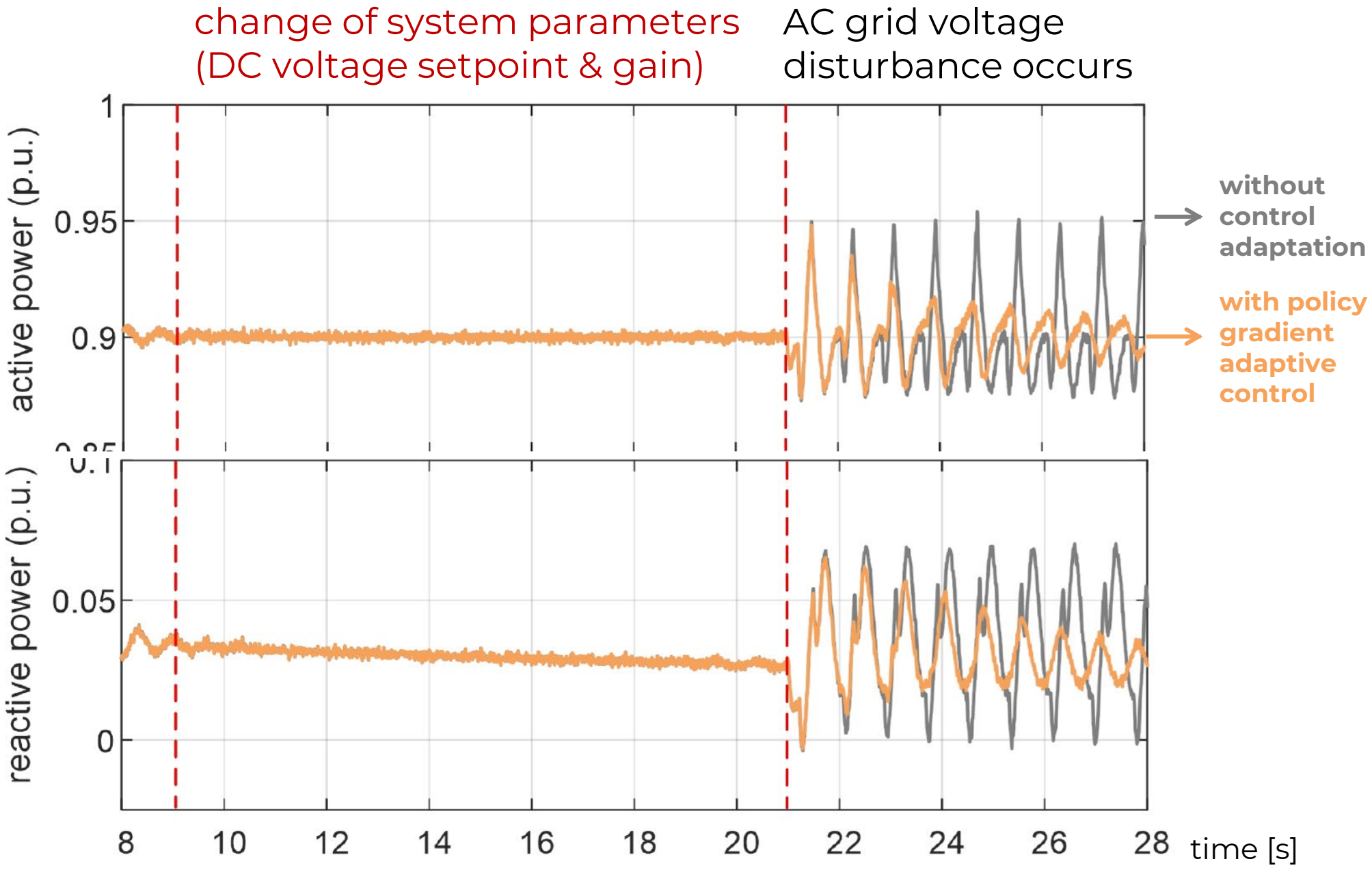


synchronous generator & full-scale converter

- 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

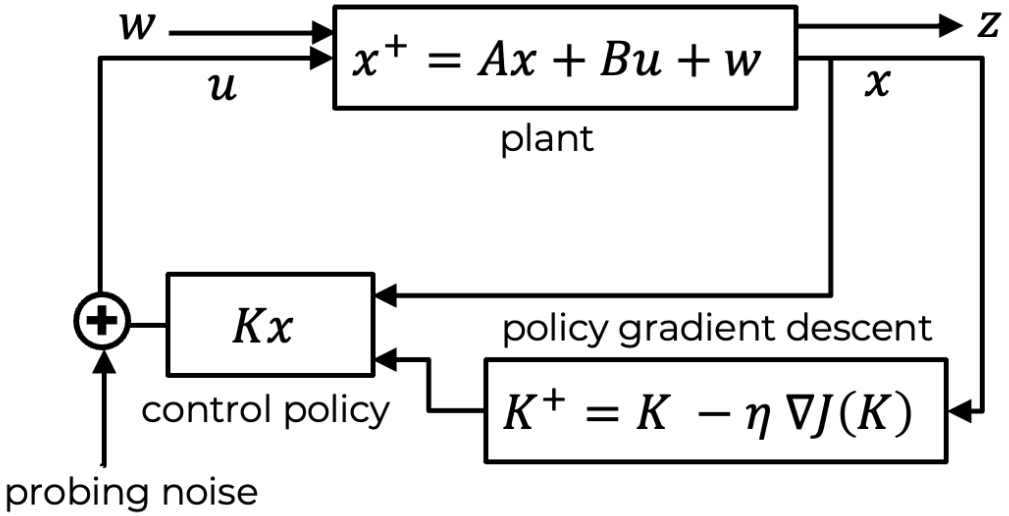
→ **direct policy gradient**


oscillation observed
probe & collect data
activate policy gradient adaptative LQR control
active power (p.u.)
0.95
0.9
0.85
reactive power (p.u.)
0.1
0.05
0
0
2
4
6
8
10
12
time [s]
without control adaptation
with policy gradient adaptive control


change of system parameters
(DC voltage setpoint & gain)
AC grid voltage
disturbance occurs
active power (p.u.)
1
0.95
0.9
reactive power (p.u.)
0.05
0
8
10
12
14
16
18
20
22
24
26
28
time [s]
without
control
adaptation
with policy
gradient
adaptive
control

CONCLUSIONS

Conclusions



Summary

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



Future work

- **technicalities**: weaken assumptions & improve rates
- **extend problem setting**: output feedback, other objectives & complex system classes: stochastic, time-varying, & nonlinear
- **persistency of excitation**: how to best design probing noise

THANKS