Nonstochastic Control

Controlling Dynamics Online

Max Simchowitz (CMU) & Elad Hazan (Princeton)

Motivation: ML as Improper Learning

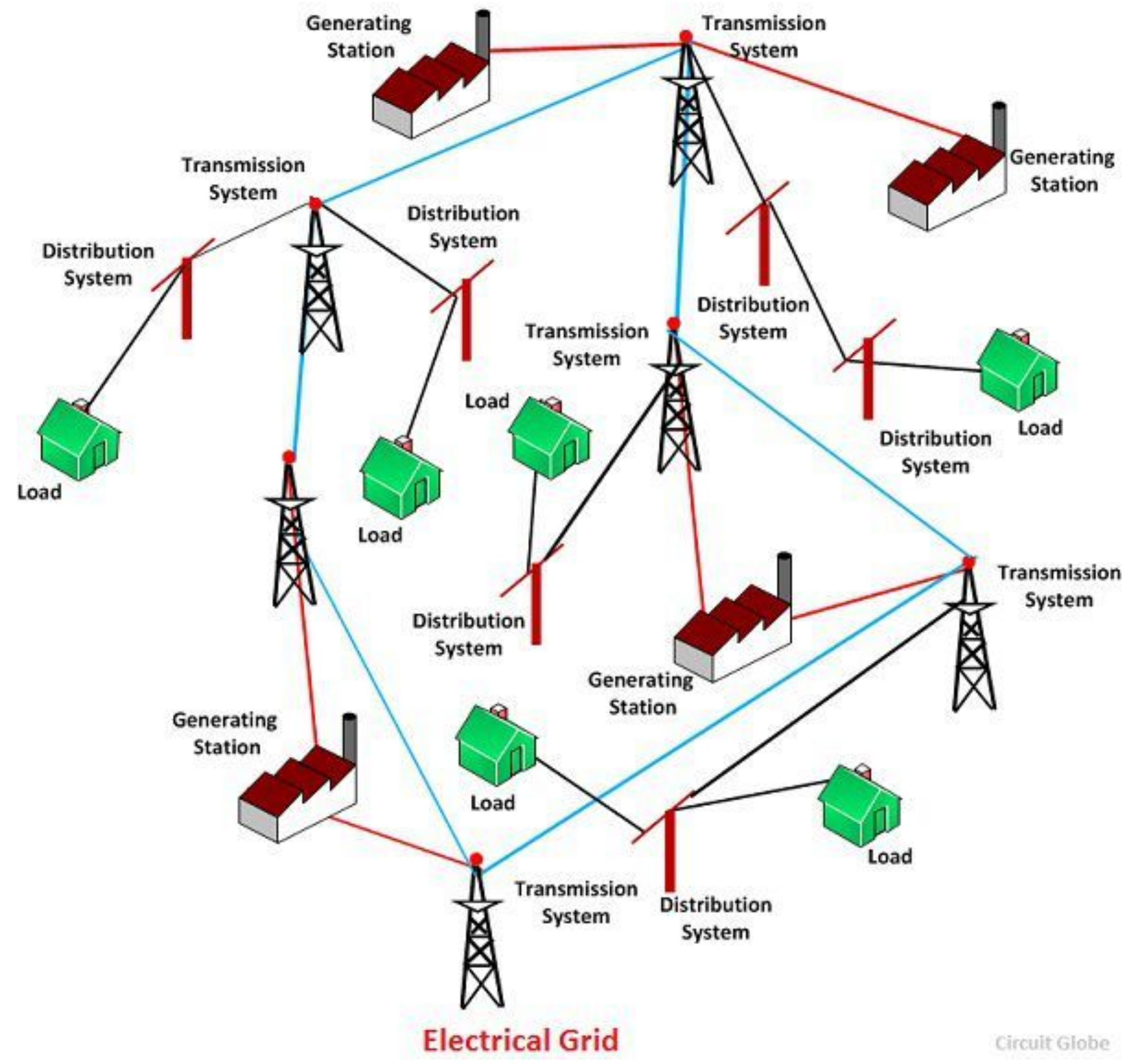
The World is Full of Dynamical Systems

The World is Full of Dynamical Systems

robotics

The World is Full of Dynamical Systems

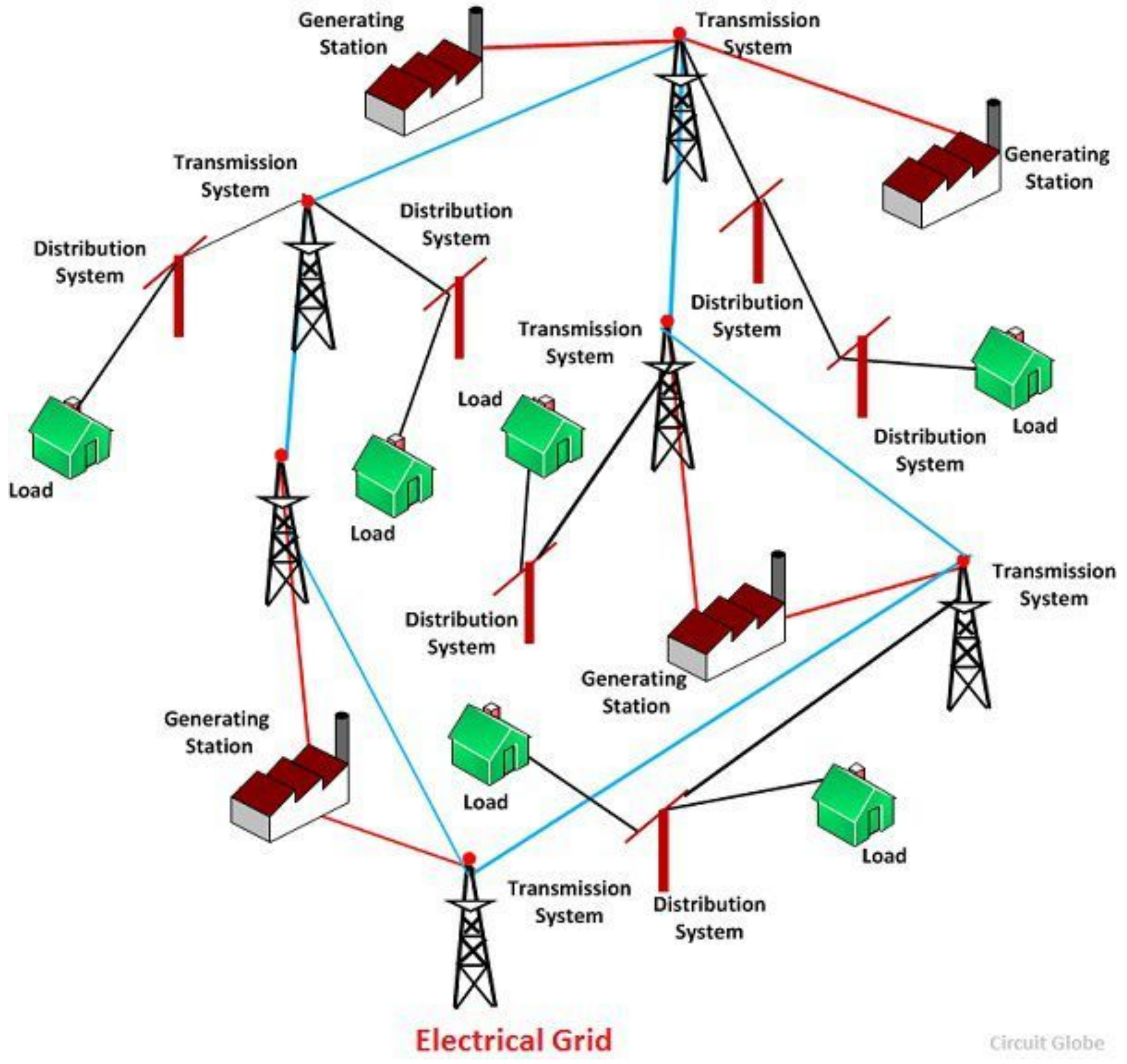
robotics



power grid

The World is Full of Dynamical Systems

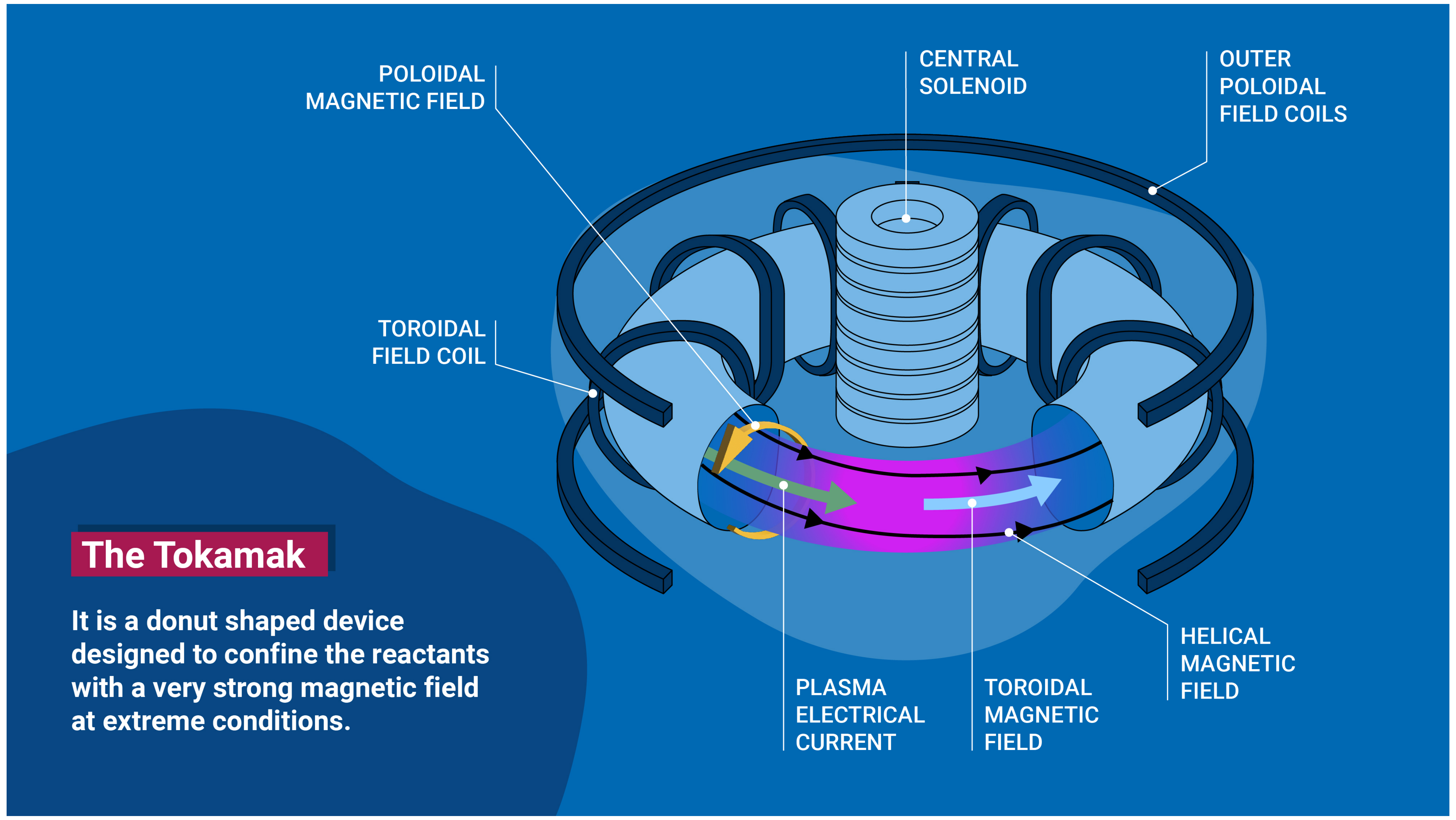
robotics



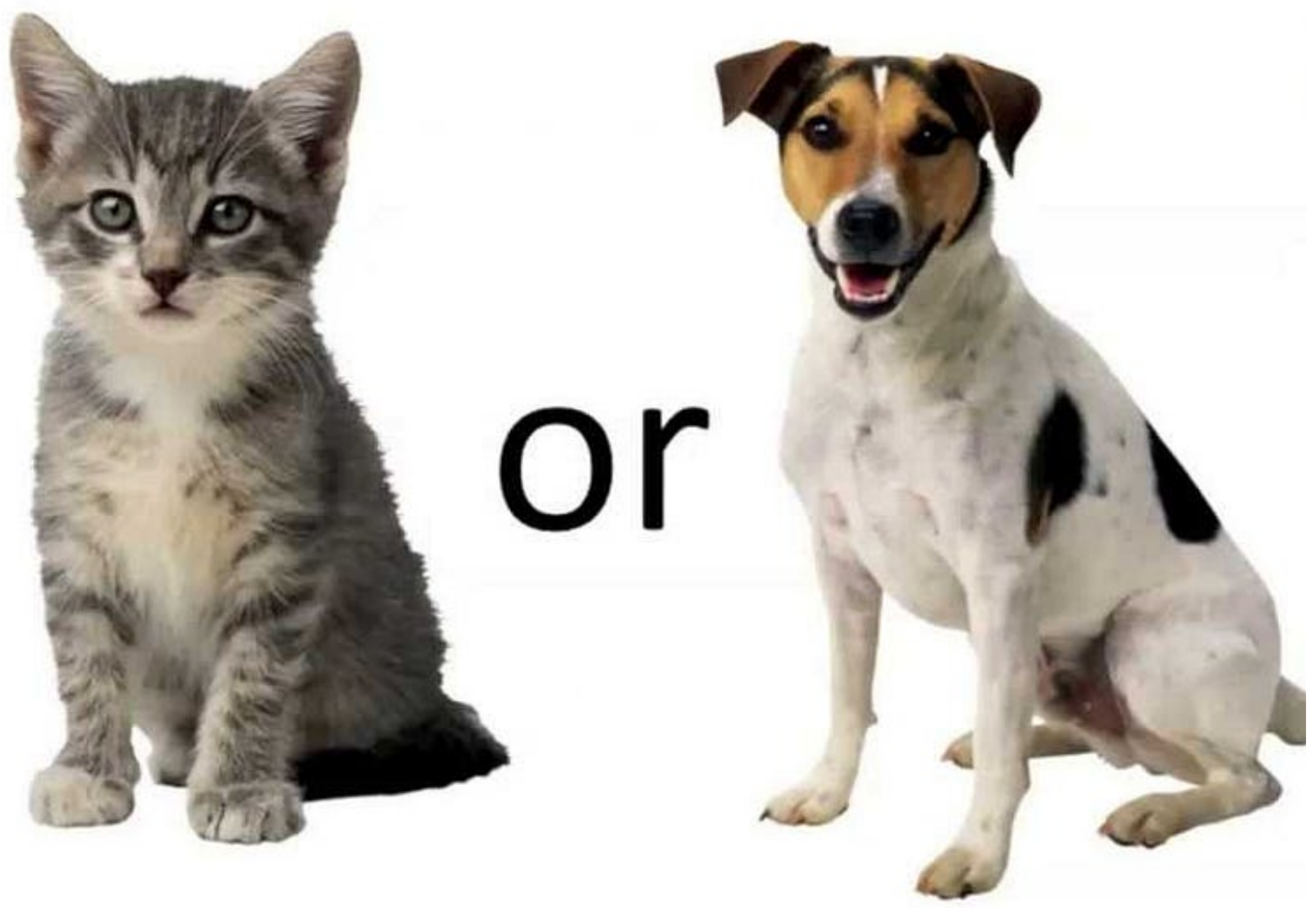
power grid

chemical plants

What about dynamics that are hard to model?



The golden rule of modern machine learning



"If computer vision researchers spent all their time searching for the **correct definition of a "cat"** in 2015, they would have made zero progress"
— **Terry Suh**

***this perspective comes with numerous drawbacks, e.g. robustness**

Applying the golden rule to control

Applying the **golden rule** to control

1. **Learning:** quantities which are unknown can be estimated statistically

Applying the golden rule to control

1. **Learning**: quantities which are unknown can be estimated statistically

2. **Relaxation/"Improperness:"** learn surrogate models which do not share the same functional form as the ground-truth (e.g. neural dynamics)

Applying the golden rule to control

1. **Learning:** quantities which are unknown can be estimated statistically

2. **Relaxation/"Improperness:"** learn surrogate models which do not share the same functional form as the ground-truth (e.g. neural dynamics)

3. **Adaptation:** we can adapt our actions to a changing world.

Applying the golden rule to control

This Tutorial: A Mathematical Formalism for Control that combines **learning**, **improperness**, and **adaption**.

Applying the golden rule to control

This Tutorial: A Mathematical Formalism for Control that combines **learning**, **improperness**, and **adaption**.

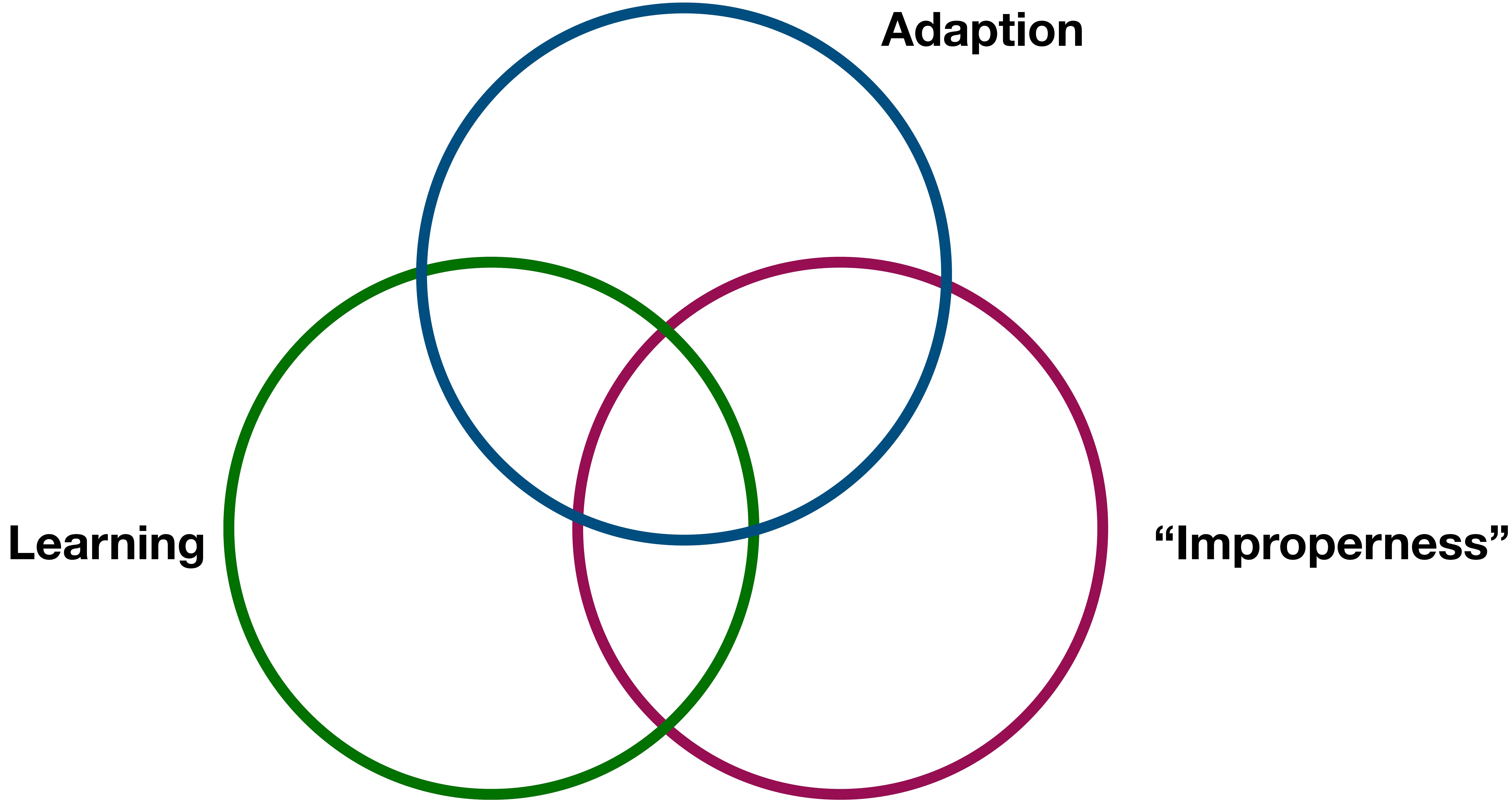
Applying the golden rule to control

This Tutorial: A Mathematical Formalism for Control that combines **learning**, **improperness**, and **adaption**.

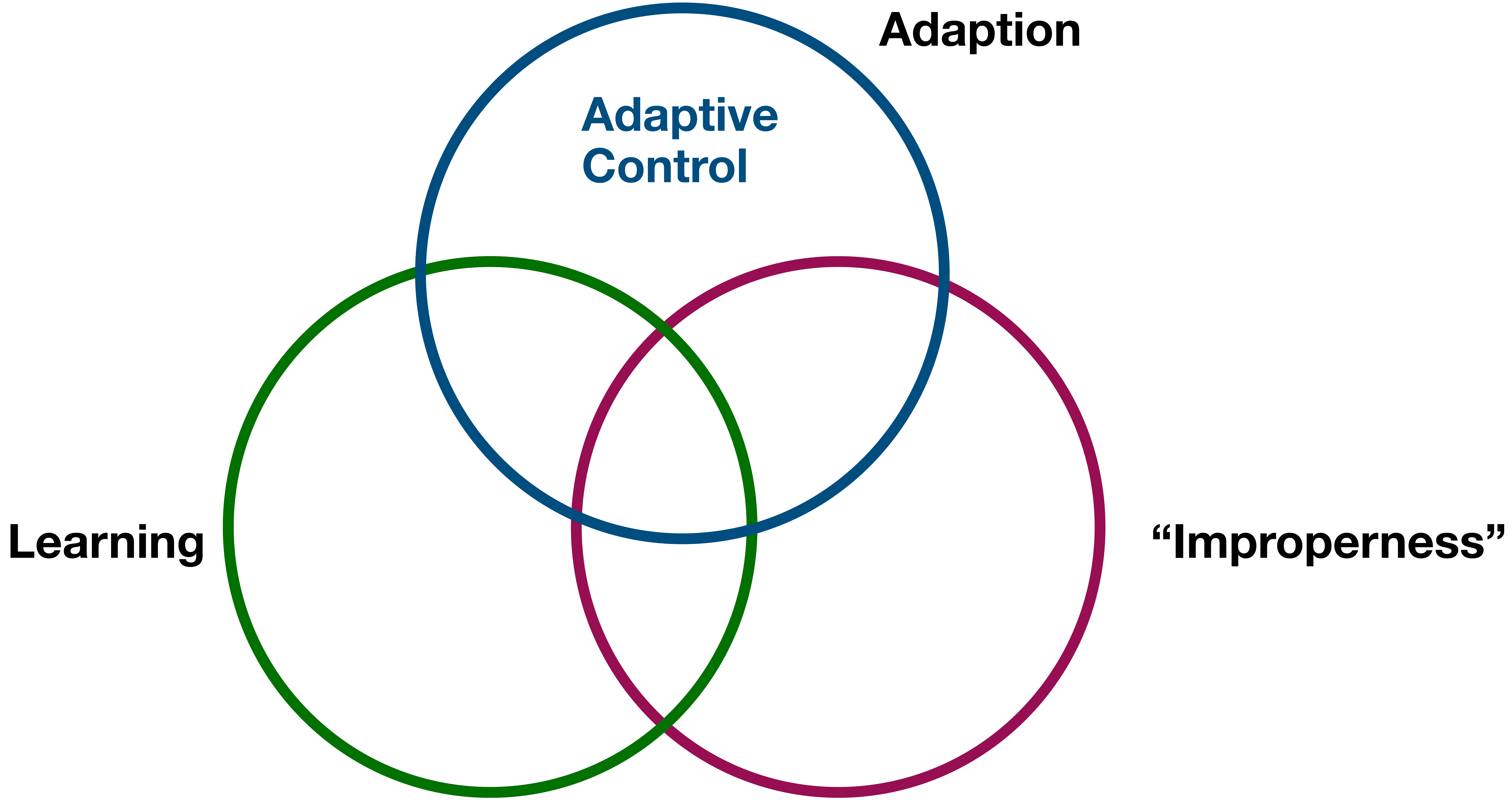
Applying the golden rule to control

This Tutorial: A Mathematical Formalism for Control that combines **learning**, **improperness**, and **adaption**.

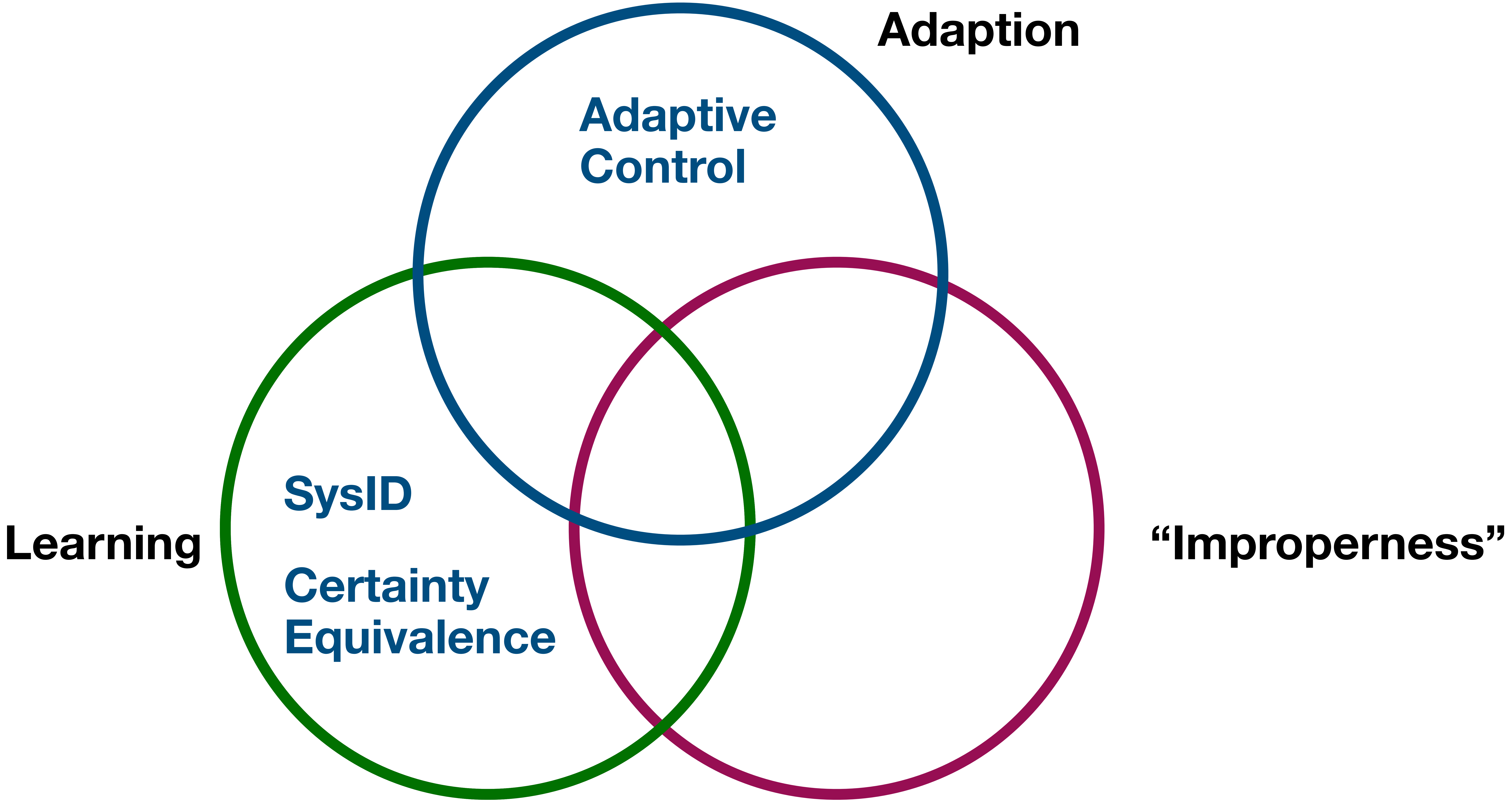
Applying the golden rule to control



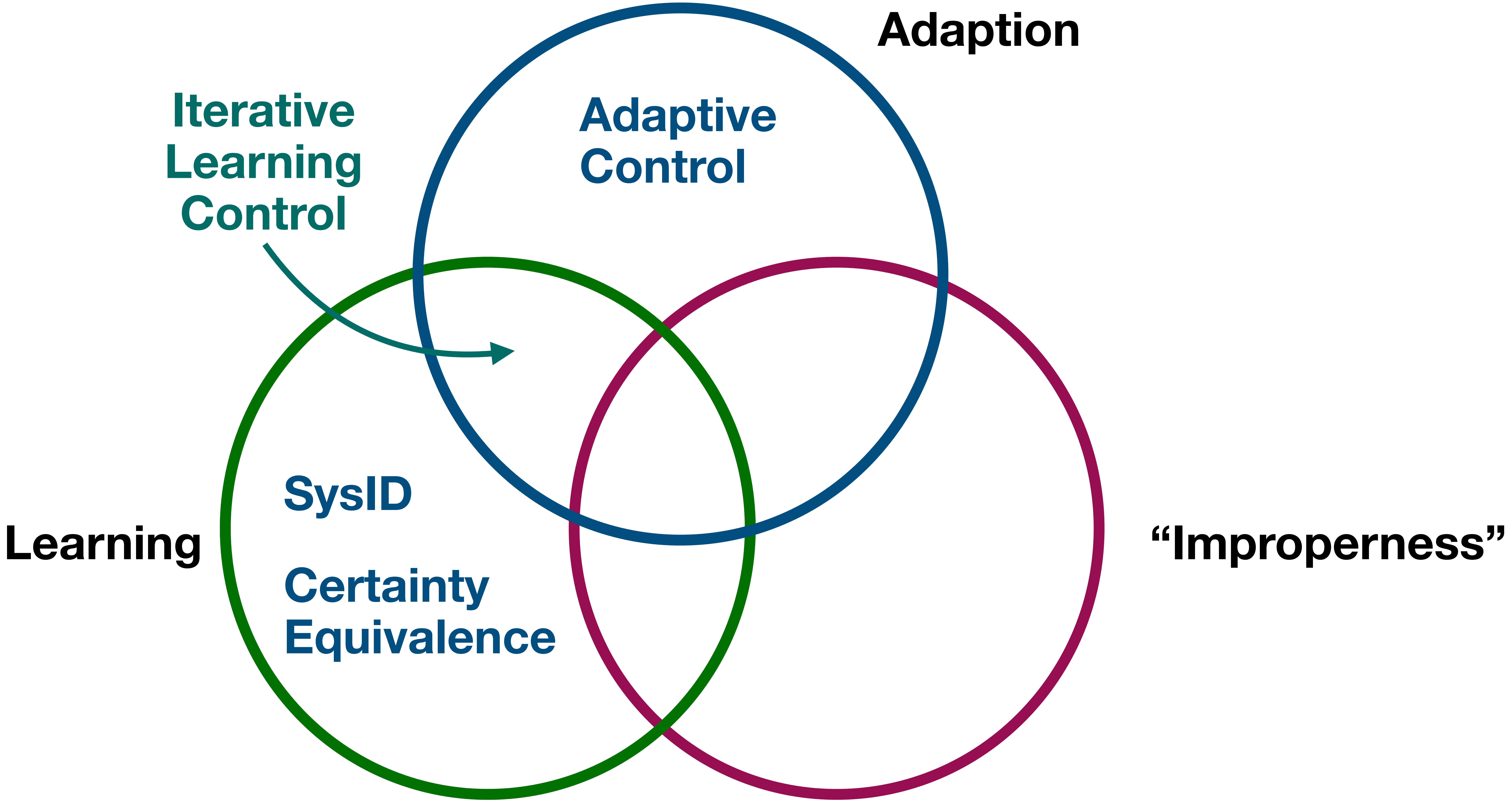
Applying the golden rule to control



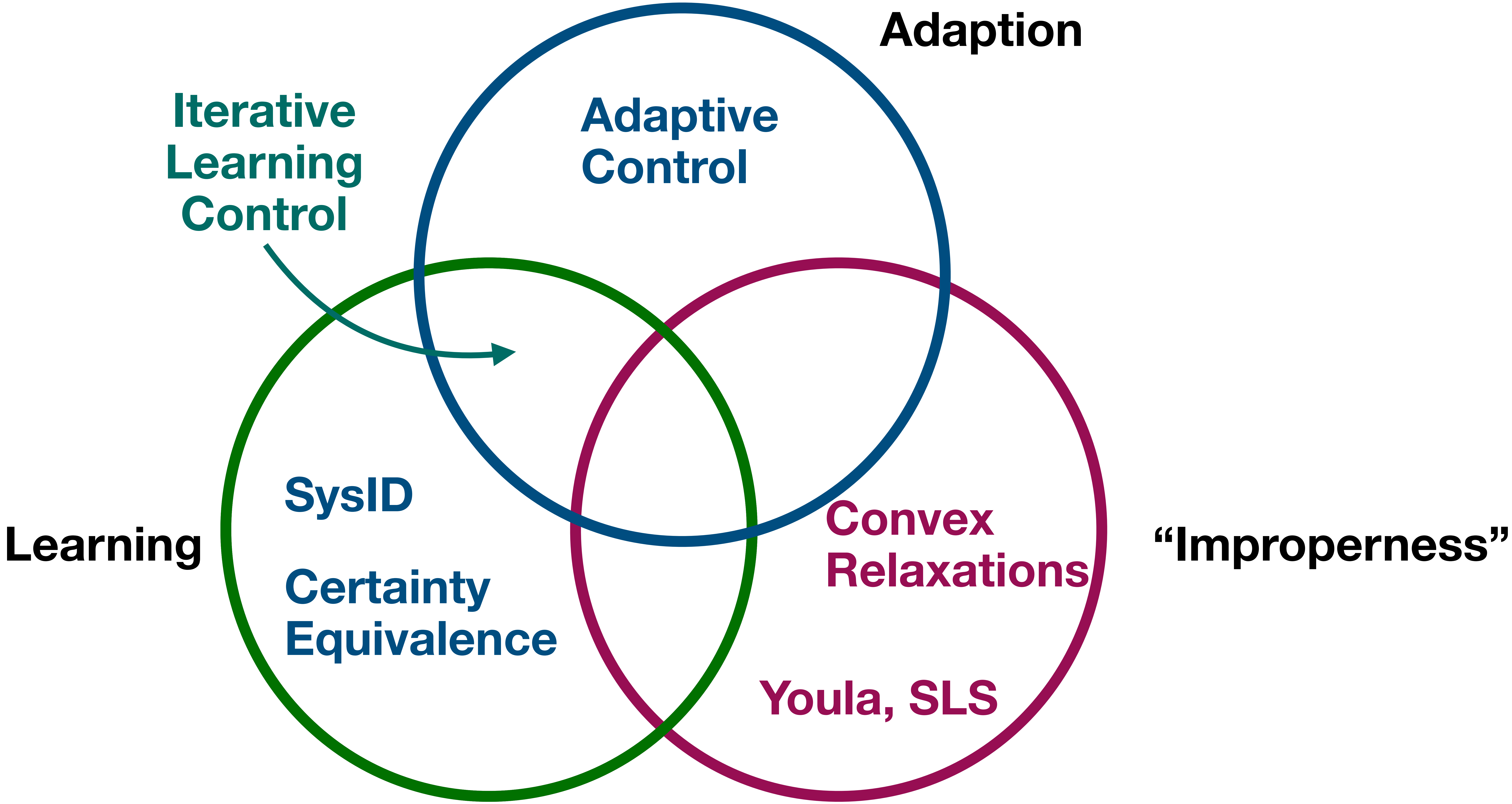
Applying the golden rule to control



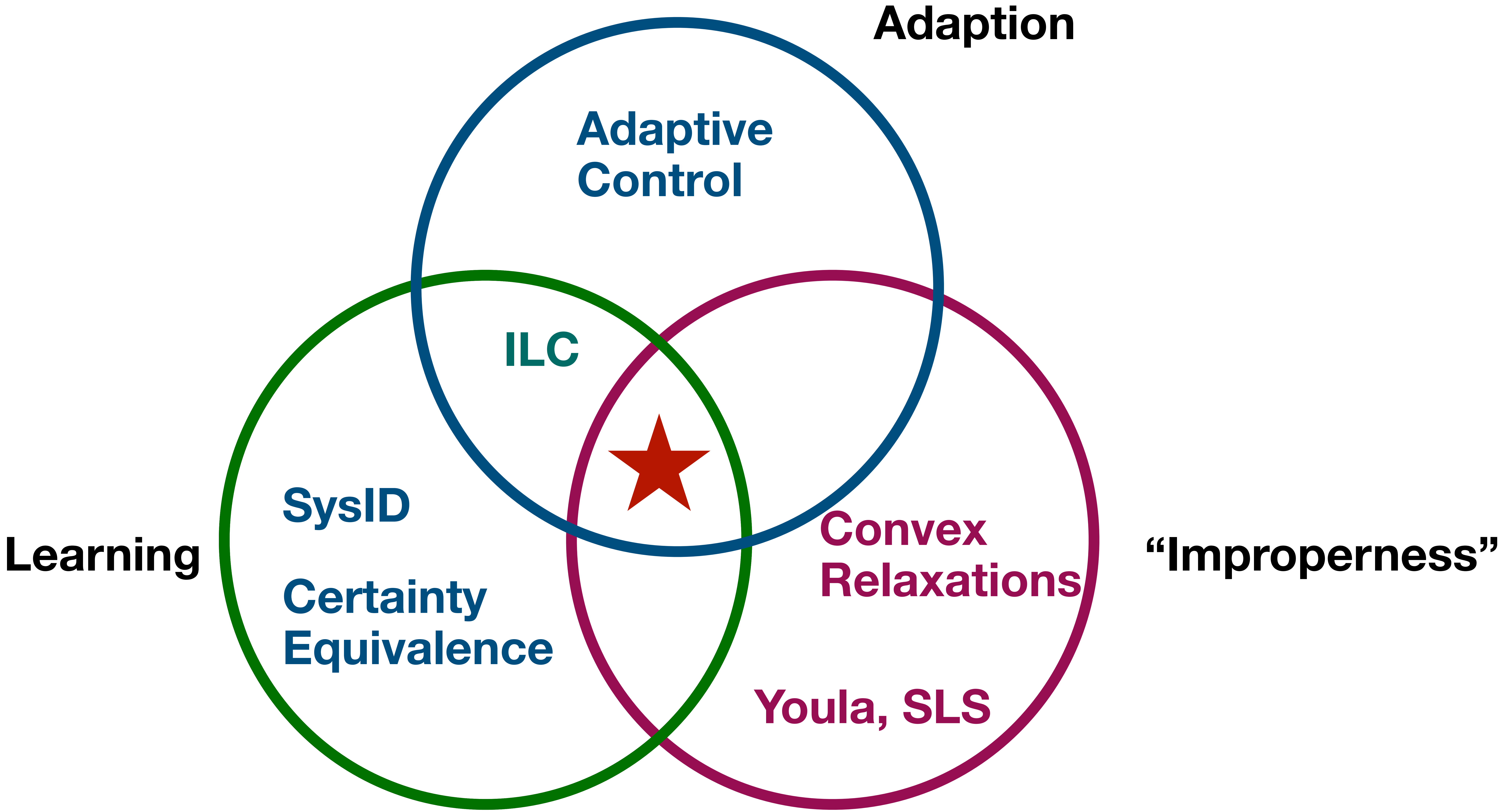
Applying the golden rule to control



Applying the golden rule to control



Non-stochastic control at the intersection



Core Concepts:

Core Concepts:

1. From optimal/robust control to regret

Core Concepts:

1. From optimal/robust control to regret

2. From “proper controller” to convex relaxation

Core Concepts:

1. From optimal/robust control to regret

2. From “proper controller” to convex relaxation

3. Combine statistical learning with online optimization

Basics of Classical Control

Background: Dynamical Systems

Recall: A **dynamical system** is

$$\boxed{x_{t+1} = f(x_t, u_t) + w_t}$$

state **control input** **disturbance**

Background: Dynamical Systems

Recall: A **dynamical system** is

$$x_{t+1} = f(x_t, u_t) + w_t$$

$$y_t = g(x_t) + e_t$$

dynamics model

observation model

observation noise

Control As an Interactive Protocol

For each time t,

Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t)

Control As an Interactive Protocol

For each time t,

1. **Nature picks noise** (w_t, e_t)
2. **Dynamics reveal** $y_t = g(x_t) + e_t$

Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t

Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, w_t) + w_t$

Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, w_t) + w_t$

Goal: For a given cost $c(\cdot\,,\cdot)$, make $J_T = \sum_{t=1}^{T} c(y_t, u_t)$ as small as possible.

Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, w_t) + w_t$

Goal: For a given cost $c(\cdot,\cdot)$, make $J_T = \sum_{t=1}^{T} c(y_t, u_t)$ as small as possible.

what does this mean?

Agent's 'Strategy': A Control Policy

If **dynamics** and $W := (w_{1:T}, e_{1:T})$ known beforehand, can directly* optimize

$$\boxed{J_T = \sum_{t=1}^{T} c(y_t, u_t)}$$

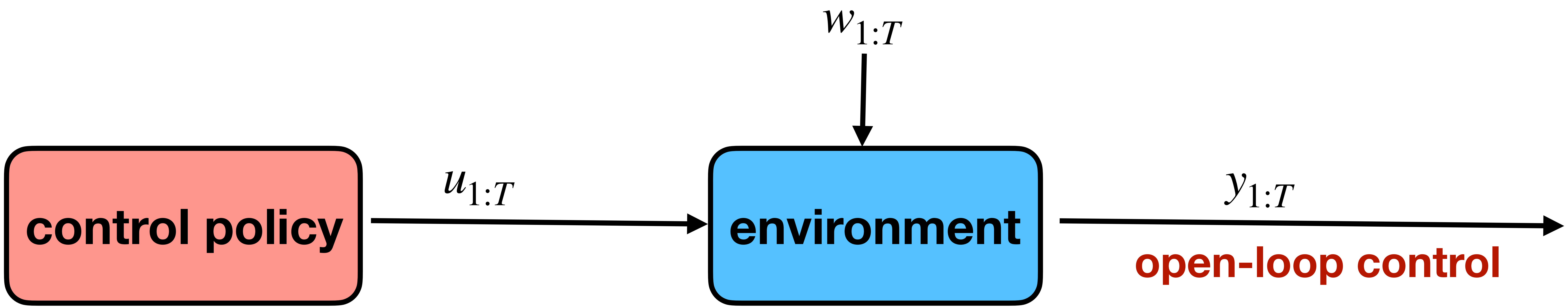
might be hard computationally

Agent's 'Strategy': A Control Policy

If **dynamics** and $W := (w_{1:T}, e_{1:T})$ known beforehand, can directly* optimize

$$\boxed{J_T = \sum_{t=1}^{T} c(y_t, u_t)}$$

*might be hard computationally

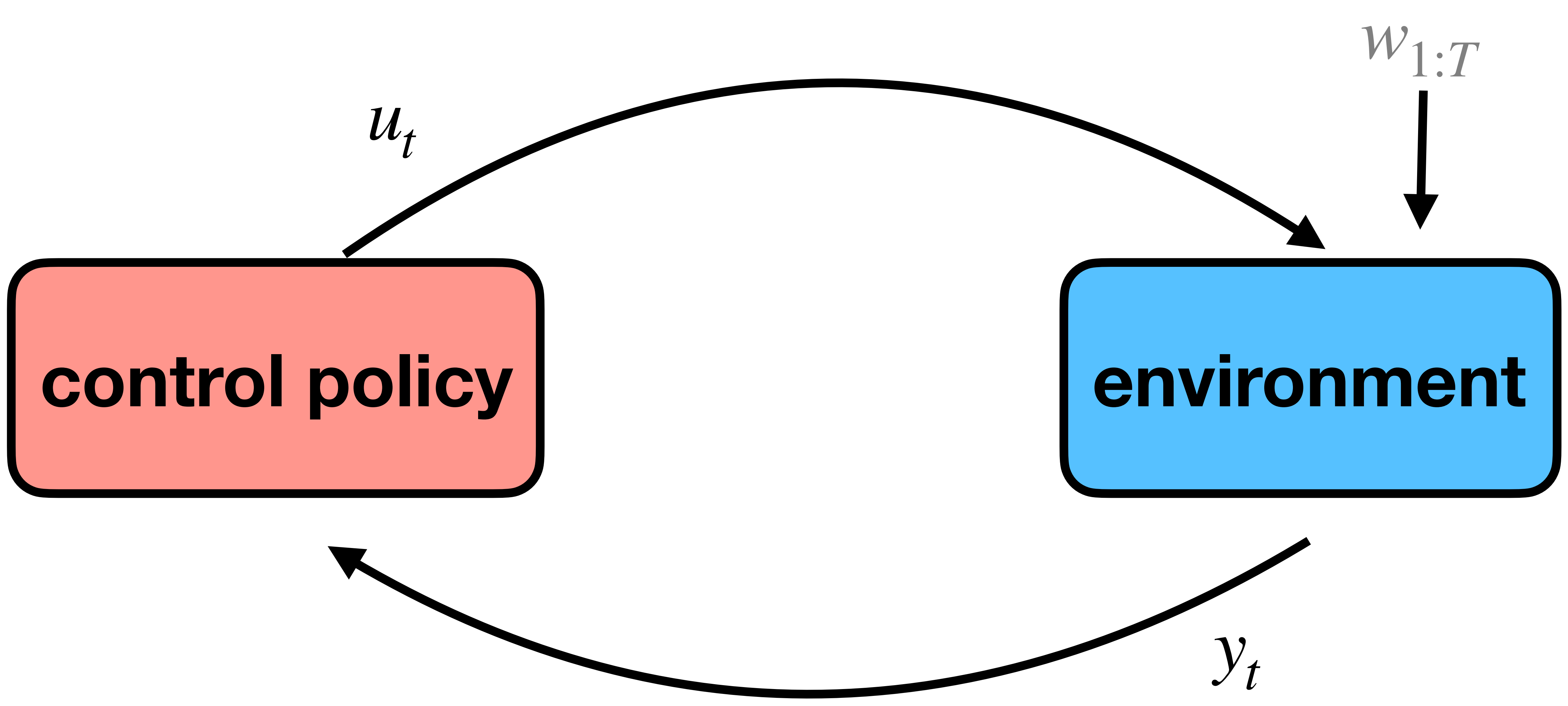


Agent's 'Strategy': A Control Policy

Otherwise: need **control policy** π mapping **past observations to current input,** to hedge **over future uncertainty** (and handle **partial observation**)

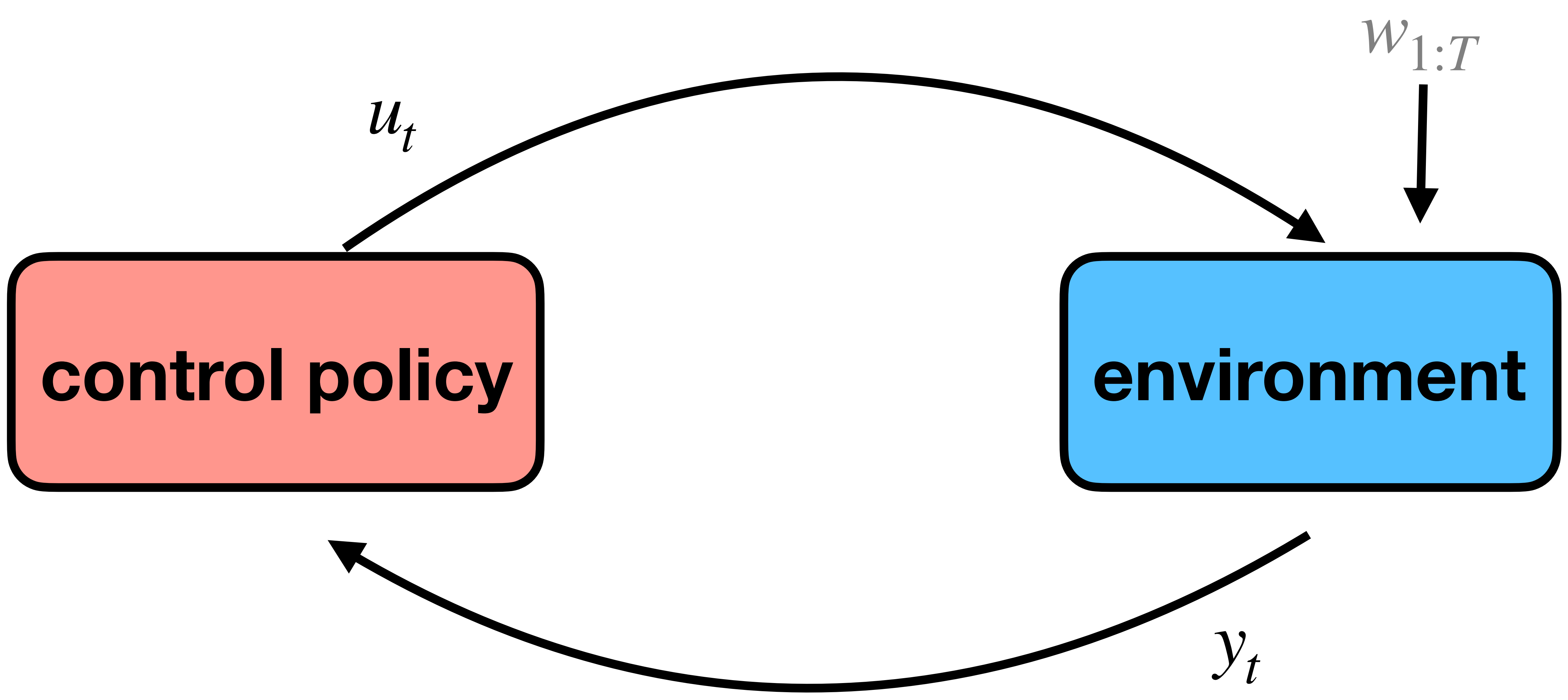
Agent's 'Strategy': A Control Policy

Otherwise: need **control policy** π mapping **past observations to current input,** to hedge **over future uncertainty** (and handle **partial observation**)



Agent's 'Strategy': A Control Policy

Otherwise: need **control policy** π mapping **past observations to current input,** to hedge **over future uncertainty** (and handle **partial observation**)



1. **History Dependent:** $\pi : (y_{1:t}, u_{1:t-1}) \to u_t$
2. **State-Based** $\pi : (x_{1:t}, u_{1:t-1}) \to u_t$
3. **State-Feedback** $\pi : x_t \to u_t$

Background: Control Cost

Recall: For a fixed dynamical system, the **control cost** of a policy π is

$$\boxed{J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)}$$

control-cost

Background: Control Cost

Recall: For a fixed dynamical system, the **control cost** of a policy π is

$$J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$$

control-cost

1. $x_{t+1} = f(x_t, u_t) + w_t$ $\quad$ $y_t = g(x_t) + e_t$

Background: Control Cost

Recall: For a fixed dynamical system, the **control cost** of a policy π is

$$J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$$

1. $x_{t+1} = f(x_t, u_t) + w_t$ $\quad$ $y_t = g(x_t) + e_t$

2. $u_t = \pi(y_{1:t}, u_{1:t-1})$

Background: Control Cost

Recall: For a fixed dynamical system, the **control cost** of a policy π is

$$J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$$

1. $x_{t+1} = f(x_t, u_t) + w_t$ $y_t = g(x_t) + e_t$
2. $u_t = \pi(y_{1:t}, u_{1:t-1})$
3. $W = (w_{1:T}, e_{1:T})$

Background: The Optimal Control Problem

The optimal control problem is

$$\min_{\pi} \mathbb{O}[J_T(\pi; W)]$$

Background: The Optimal Control Problem

The optimal control problem is

$$\min_{\pi} \mathbb{O}[J_T(\pi; W)]$$

describes the W

Background: The Optimal Control Problem

The optimal control problem is

$$\min_{\pi} \mathbb{O}[J_T(\pi; W)]$$

Background: The Optimal Control Problem

The optimal control problem is

$$\min_{\pi} \mathbb{O}[J_T(\pi; W)]$$

1. **fixed** W

(open-loop planning/trajectory opt.)

Background: The Optimal Control Problem

The optimal control problem is

$$\min_{\pi} \mathbb{O}[J_T(\pi; W)]$$

1. **fixed** W — **(open-loop planning/trajectory opt.)**
2. **random** $\mathbb{E}_W$ — **(stochastic optimal control)**

Background: The Optimal Control Problem

The optimal control problem is

$$\min_{\pi} \mathbb{O}[J_T(\pi; W)]$$

1. **fixed** W **(open-loop planning/trajectory opt.)**
2. **random** $\mathbb{E}_W$ **(stochastic optimal control)**
3. **worst-case** $\sup_{W \in \ldots}$ **(robust control**, e.g. the work of John Doyle**)**

Summary

Summary

1. We introduced **dynamical systems** $\boxed{x_{t+1} = f(x_t, u_t) + w_t}$ $\boxed{y_t = g(x_t) + e_t}$

Summary

1. We introduced **dynamical systems** $x_{t+1} = f(x_t, u_t) + w_t$ $y_t = g(x_t) + e_t$

2. We formulated control as an **interactive protocol,** and described open- and closed-loop policies are agent strategies

Summary

1. We introduced **dynamical systems** $\boxed{x_{t+1} = f(x_t, u_t) + w_t}$ $\boxed{y_t = g(x_t) + e_t}$

2. We formulated control as an **interactive protocol,** and described open- and closed-loop policies are agent strategies

3. We introduced the noise-dependent **cost functional** $J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$

Summary

1. We introduced **dynamical systems** $\boxed{x_{t+1} = f(x_t, u_t) + w_t}$ $\boxed{y_t = g(x_t) + e_t}$

2. We formulated control as an **interactive protocol,** and described open- and closed-loop policies are agent strategies

3. We introduced the noise-dependent **cost functional** $J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$

4. We briefly described **classical noise models** (fixed, random, worst-case).

Basics of Linear Control

Background: Linear Dynamical System

Recall: A **linear dynamical system** is

$$x_{t+1} = Ax_t + Bu_t + w_t$$

$$y_t = Cx_t + e_t$$

Background: Linear Dynamical System

Recall: A **linear dynamical system** is

$$x_{t+1} = Ax_t + Bu_t + w_t \qquad y_t = Cx_t + e_t$$

dynamics model

Background: Linear Dynamical System

Recall: A **linear dynamical system** is

$$x_{t+1} = Ax_t + Bu_t + w_t \qquad y_t = Cx_t + e_t$$

dynamics model **observation model**

Background: Linear Dynamical System

Recall: A **linear dynamical system** is

$$x_{t+1} = Ax_t + Bu_t + w_t \qquad y_t = Cx_t + e_t$$

dynamics model **observation model**

Rationale: Local Taylor Approximation of Nonlinear Dynamics.

Linear Quadratic Optimal Control Problems

Linear Quadratic Optimal Control

$$J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$$

$$c(y, u) = y^\top Q y + u^\top R u$$

Linear Quadratic Optimal Control Problems

Linear Quadratic Optimal Control

$$J_T(\pi; W) = \sum_{t=1}^{T} c(y_t, u_t)$$

$$c(y, u) = y^\top Q y + u^\top R u$$

convex quadratic: $Q, R \succeq 0$

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic Optimal Control

Stochastic Control

Robust Control

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

Stochastic Control

Robust Control

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

Stochastic Control

The $\mathscr{H}_2$ control problem: w_t, e_t are i.i.d Gaussian (*Kalman, LQG*)

Robust Control

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|,\|e\|\le 1} [J_T(\pi; W)]$$

Stochastic Control

The $\mathscr{H}_2$ control problem: w_t, e_t are i.i.d Gaussian (*Kalman, LQG*)

Robust Control

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|,\|e\|\le 1} [J_T(\pi; W)]$$

Stochastic Control

The $\mathscr{H}_2$ control problem: w_t, e_t are i.i.d Gaussian (*Kalman, LQG*)

Robust Control

The $\mathscr{H}_\infty$ control problem:
w_t, e_t are worst case (*Doyle*)

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic LQ Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_w[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|\le 1} [J_T(\pi; W)]$$

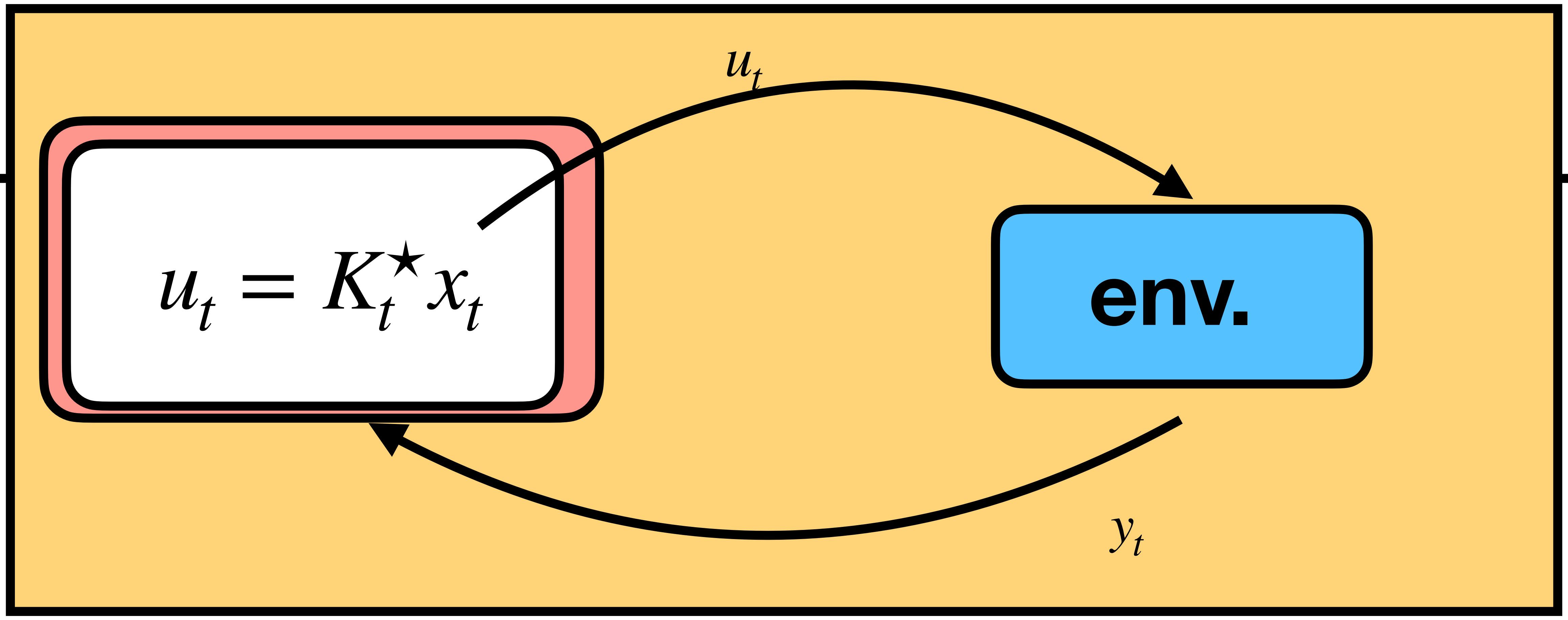
Linear Quadratic Optimal Control Problems

Classical Linear Quadratic LQ Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_w[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|\leq 1} [J_T(\pi; W)]$$

Theorem: If **fully observed** $(y_t \equiv x_t)$, **state-feedback is optimal**



Linear Quadratic Optimal Control Problems

Classical Linear Quadratic LQ Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|,\|e\|\leq 1} [J_T(\pi; W)]$$

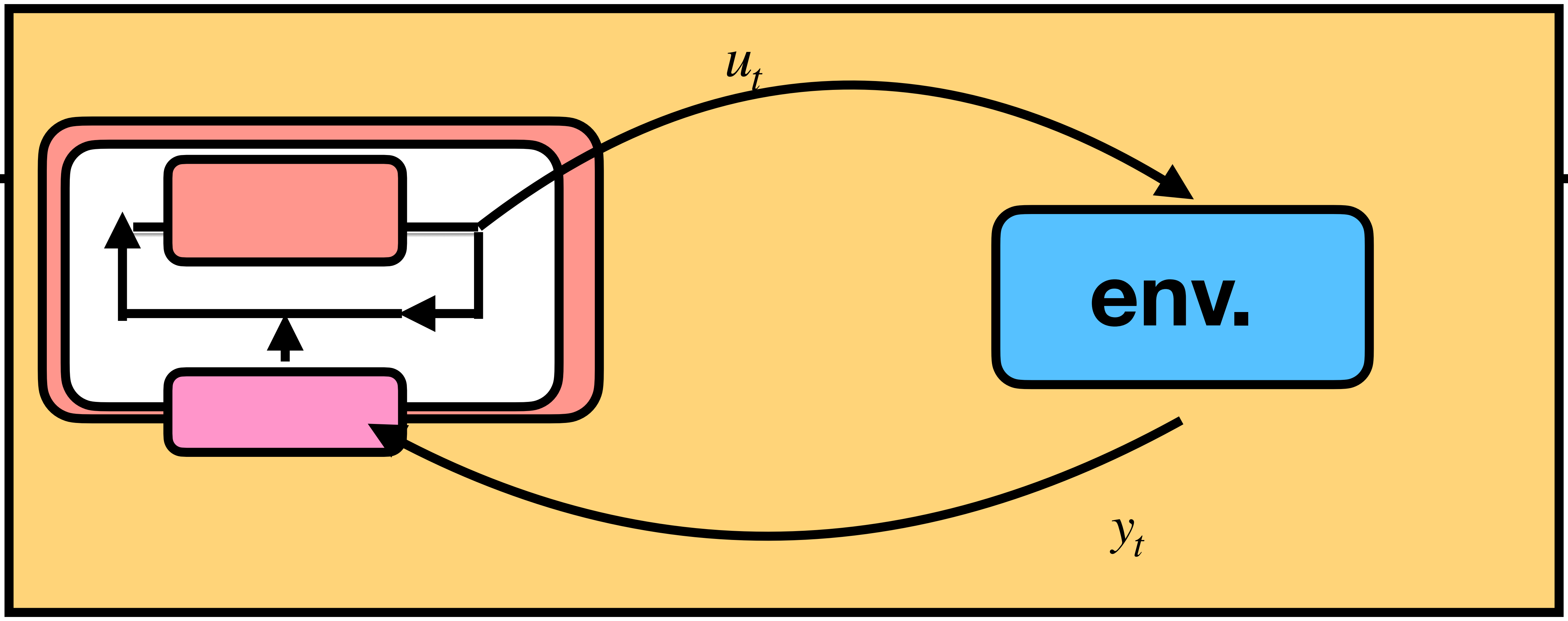
Linear Quadratic Optimal Control Problems

Classical Linear Quadratic LQ Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|,\|e\|\le 1} [J_T(\pi; W)]$$

Theorem: For **general LQ control** are **linear dynamic policies** are optimal**:**



$$z_{t+1} = A_\pi z_t + B_\pi y_t$$

$$u_t = C_\pi z_t + D_\pi y_t$$

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic LQ Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|,\|e\|\leq 1} [J_T(\pi; W)]$$

Linear Quadratic Optimal Control Problems

Classical Linear Quadratic LQ Optimal Control

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_{w,e}[J_T(\pi; W)]$$

$$\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \sup_{\|w\|,\|e\|\leq 1} [J_T(\pi; W)]$$

Important Takeaway: Linear Quadratic Control Problems admit **easy-to-express** controllers.

Beyond LQ Control

Beyond LQ Control

Challenge 1: If costs/constraints are **no longer quadratic**, optimal control is **hard to describe, even if dynamics are linear.**

Beyond LQ Control

Challenge 1: If costs/constraints are **no longer quadratic**, optimal control is **hard to describe, even if dynamics are linear.**

Example (ℓ_1 control, *Borelli '03*): $c(y, u) = \|y\|_1 + \|u\|_2$

Beyond LQ Control

Challenge 1: If costs/constraints are **no longer quadratic**, optimal control is **hard to describe, even if dynamics are linear.**

Example (ℓ_1 control, *Borelli '03*): $c(y, u) = \|y\|_1 + \|u\|_2$

Challenge 2: Optimizing over **feedback** (static or dynamic) is **non-convex** and can be computationally hard:

Beyond LQ Control

Challenge 1: If costs/constraints are **no longer quadratic**, optimal control is **hard to describe, even if dynamics are linear.**

Example (ℓ_1 control, *Borelli '03*): $c(y, u) = \|y\|_1 + \|u\|_2$

Challenge 2: Optimizing over **feedback** (static or dynamic) is **non-convex** and can be computationally hard:

This is because, e.g. in full observation $x_t = \sum_s (A + BK)^{t-s}(Bu_s + w_s)$

Beyond LQ Control

Challenges: Direct optimization over feedback controllers can be hard, and exact optimal control laws can be hard to express.

Beyond LQ Control

Challenges: Direct optimization over feedback controllers can be hard, and exact optimal control laws can be hard to express.

Insight: **Optimization restricted to linear policies can be reparametrized to be convex if costs/constraints are convex**

Beyond LQ Control

Challenges: Direct optimization over feedback controllers can be hard, and exact optimal control laws can be hard to express.

Insight: **Optimization restricted to linear policies can be reparametrized to be convex if costs/constraints are convex**

Powerful Observation: Youla-Kućera ’76, Zames ’81 (IO), Anderson et al. ’19 (SLS)

Summary

Summary

1. We introduced **linear dynamical systems**

$$x_{t+1} = Ax_t + Bu_t + w_t$$

$$y_t = Cx_t + e_t$$

Summary

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + e_t$$

1. We introduced **linear dynamical systems**

2. We described the **optimal control laws** for linear-quadratic **(LQ)** control

Summary

1. We introduced **linear dynamical systems**

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + e_t$$

2. We described the **optimal control laws** for linear-quadratic **(LQ)** control

3. We described **computational difficulties** beyond the LQ regime

Summary

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + e_t$$

1. We introduced **linear dynamical systems**

2. We described the **optimal control laws** for linear-quadratic **(LQ)** control

3. We described **computational difficulties** beyond the LQ regime

4. We hinted at **convex relaxations** as a tool for efficient optimization.

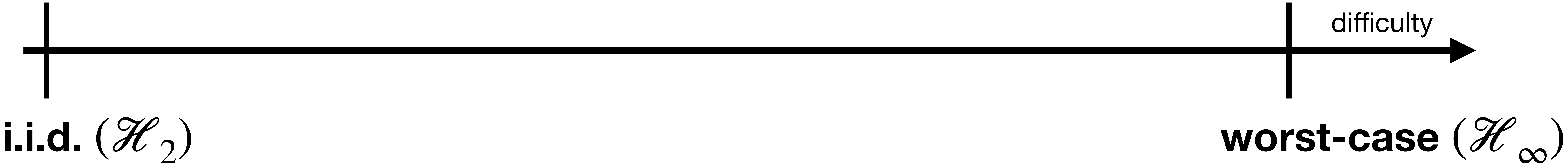
The Non-Stochastic Control Problem

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?



The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

Naively: $\min_\pi J_T(\pi; W)$ **for every** W

i.i.d. **worst-case**

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

Naively: $\min_{\pi} J_T(\pi; W)$ **for every** W

i.i.d.

"What π would we pick if we knew noise W in hindsight"

worst-case

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

Naively: $\min_\pi J_T(\pi; W)$ **for every** W

i.i.d.

worst-case

"What π would we pick if we knew noise W in hindsight"

But of course: impossible, and leads to open loop control

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

Naively: $\min_\pi J_T(\pi; W)$ **for every** W

i.i.d. **worst-case**

The Non-Stochastic Control Problem

Motivating Question: What lies between i.i.d. and worst case?

Naively: $\min_\pi J_T(\pi; W)$ **for every** W

i.i.d.

worst-case

We will allow adversarial noise, but introduce **regret** to measure performance

Solution Concept: Regret

“i’ve had a few”

Solution Concept: Regret

"i've had a few"

$$J_T(\mathbb{A}; W) = \sum_{t=1}^{T} c(y_t^{\mathbb{A}}, u_t^{\mathbb{A}})$$

Solution Concept: Regret

"i've had a few"

$$J_T(\mathbb{A}; W) = \sum_{t=1}^{T} c(y_t^{\mathbb{A}}, u_t^{\mathbb{A}})$$

$\mathbb{A}$ **for algorithm**

Solution Concept: Regret

"i've had a few"

$$J_T(\mathbb{A}; W) = \sum_{t=1}^{T} c(y_t^{\mathbb{A}}, u_t^{\mathbb{A}})$$

$\mathbb{A}$ for algorithm

also called 'learner' or 'agent'

Solution Concept: Regret

"i've had a few"

Fix a class of comparator policies $\pi \in \Pi$

$$J_T(\mathbb{A}; W) = \sum_{t=1}^{T} c(y_t^{\mathbb{A}}, u_t^{\mathbb{A}})$$

$\mathbb{A}$ **for algorithm**

also called 'learner' or 'agent'

Solution Concept: Regret

"i've had a few"

Fix a class of comparator policies $\pi \in \Pi$

$$J_T(\mathbb{A}; W) = \sum_{t=1}^{T} c(y_t^{\mathbb{A}}, u_t^{\mathbb{A}})$$

$\mathbb{A}$ **for algorithm**

also called 'learner' or 'agent'

$$J_T(\pi; W) = \sum_{t=1}^{T} c(y_t^{\pi}, u_t^{\pi})$$

counterfactual cost under policy $\pi \in \Pi$

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

$$\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W)$$

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = \quad J_T(\mathbb{A};W) \; - \; \min_{\pi\in\Pi} \; J_T(\pi;W)$$

excess cost of algorithm

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

$$\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W)$$

excess cost of algorithm

best-in-hindsight

(with full knowledge of disturbances)

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

$$\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W)$$

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$$

Goal: $\mathrm{Reg}_T = o(T)$ **(vanishing regret as fraction of horizon) for all** W

Solution Concept: Regret

Fix a class of comparator policies $\pi \in \Pi$

$$\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W)$$

Goal: $\mathrm{Reg}_T = o(T)$ **(vanishing regret as fraction of horizon) for all** W

"competing with Π**"**

Solution Concept: Regret

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$$

Why a restricted comparator class?

Solution Concept: Regret

$$\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W)$$

Why a restricted comparator class?

1. If Π is **unrestricted,** comparator cost is **open-loop optimal plan.**

Solution Concept: Regret

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$$

Why a restricted comparator class?

1. If Π is **unrestricted,** comparator cost is **open-loop optimal plan.**

we can embed a **prediction problem** where comparator has zero cost (perfect knowledge), but learner has $\Omega(T)$ cost.

Solution Concept: Regret

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$$

Why a restricted comparator class?

1. If Π is **unrestricted,** comparator cost is **open-loop optimal plan.**

Solution Concept: Regret

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$$

Why a restricted comparator class?

1. If Π is **unrestricted,** comparator cost is **open-loop optimal plan.**

2. We can restrict Π to make optimization **computationally efficient.**

Solution Concept: Regret

$$\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$$

Why a restricted comparator class?

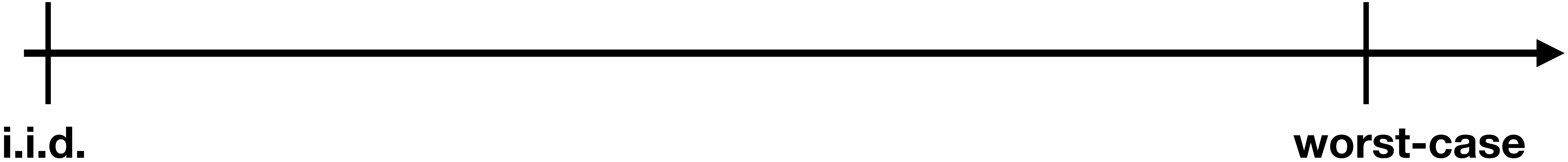
1. If Π is **unrestricted,** comparator cost is **open-loop optimal plan.**

2. We can restrict Π to make optimization **computationally efficient.**

Key Idea: Optimizing over linear policies can **efficient, even when optimal control is not.**

Compared to What? For linear dynamics.

What class of comparator policies $\pi \in \Pi$**?**



Compared to What? For linear dynamics.

What class of comparator policies $\pi \in \Pi$**?**

i.i.d.

worst-case

Informally: Π **is the set of all linear policies**
that stabilize the dynamics

Compared to What? For linear dynamics.

What class of comparator policies $\pi \in \Pi$**?**

i.i.d.

Informally: Π **is the set of all linear policies** that stabilize the dynamics

worst-case

For LQ control, these are all linear policies that are **stable** with exponential decay

Compared to What? For linear dynamics.

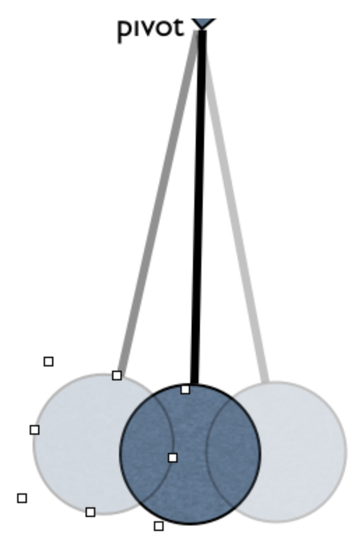
What class of comparator policies $\pi \in \Pi$?

i.i.d.

Informally: Π is the set of all linear policies
that stabilize the dynamics

worst-case

For LQ control, these are all linear policies that are **stable** with exponential decay



Pendulum

Nonstochastic Control As an Interactive Protocol

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. **Nature picks noise (w_t, e_t)**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. **Nature picks noise (w_t, e_t)** **(adversarially!)**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. **Nature picks noise (w_t, e_t)** (adversarially!)
2. **Dynamics reveal $y_t = g(x_t) + e_t$**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t) (adversarially!)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t) (adversarially!)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, u_t) + w_t$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t) (adversarially!)
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, u_t) + w_t$

Goal: make $\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W) = o(T)$.

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Nonstochastic Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t) and a cost c_t
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, u_t) + w_t$, suffer $c_t(y_t, u_t)$

Goal: make $\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W) = o(T)$.

Nonstochastic Control As an Interactive Protocol

For each time t,

1. Nature picks noise (w_t, e_t) and a cost c_t
2. Dynamics reveal $y_t = g(x_t) + e_t$
3. Control agent picks u_t
4. Dynamics evolve $x_{t+1} = f(x_t, u_t) + w_t$, suffer $c_t(y_t, u_t)$

Goal: make $\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W) = o(T)$.

defined with changing costs

Linear Nonstochastic Control: Interactive Protocol

For each time t,

1. **Nature picks noise (w_t, e_t) and a cost c_t**
2. **Dynamics reveal $y_t = Cx_t + e_t$**
3. **Control agent picks u_t**
4. **Dynamics evolve $x_{t+1} = Ax_t + Bu_t + w_t$, suffer $c_t(y_t, u_t)$**

Goal: make $\mathrm{Reg}_T(\mathbb{A}; \Pi) = J_T(\mathbb{A}; W) - \min_{\pi \in \Pi} J_T(\pi; W) \; = o(T)$.

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary

Summary

1. Non-stochastic control is an **intermediate** between stochastic and robust

Summary

1. Non-stochastic control is an **intermediate** between stochastic and robust

2. We define **regret** to a **restricted comparator class** as a performance yardstick when noise is possibly **adversarial**

Summary

1. Non-stochastic control is an **intermediate** between stochastic and robust

2. We define **regret** to a **restricted comparator class** as a performance yardstick when noise is possibly **adversarial**

3. We formulated the **non-stochastic control** protocol, including changing costs.

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$ **small.**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$ **small.**

How to compete benchmark online, despite unknown costs/ disturbances

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$ **small.**

How to compete benchmark online, despite unknown costs/ disturbances

How to efficiently parameterize control policies $\pi \in \Pi$**?**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = \underline{J_T(\mathbb{A};W)} - \underline{\min_{\pi\in\Pi} J_T(\pi;W)}$ **small.**

How to compete benchmark online, despite unknown costs/disturbances

How to efficiently parameterize control policies $\pi \in \Pi$?

Tool: Online Convex Optimization

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = \underline{J_T(\mathbb{A}; W)} - \underline{\min_{\pi\in\Pi} J_T(\pi; W)}$ **small.**

How to compete benchmark online, despite unknown costs/ disturbances

How to efficiently parameterize control policies $\pi \in \Pi$?

Tool: Online Convex Optimization

Tool: Convex Control Parametrization

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Topics Covered

Roadmap: Topics Covered

1. **GPC**: Fully Observed, Known-Dynamics

Roadmap: Topics Covered

1. GPC: Fully Observed, Known-Dynamics
2. Nature’s Y’s: Partially Observed, Known-Dynamics

Roadmap: Topics Covered

1. GPC: Fully Observed, Known-Dynamics
2. Nature's Y's: Partially Observed, Known-Dynamics
3. Unknown Dynamics: System Identification

Roadmap: Topics Covered

1. GPC: Fully Observed, Known-Dynamics

2. Nature's Y's: Partially Observed, Known-Dynamics

3. Unknown Dynamics: System Identification

4. Optimal Regret: Leveraging Curvature

Roadmap: Topics Covered

1. GPC: Fully Observed, Known-Dynamics
2. Nature's Y's: Partially Observed, Known-Dynamics
3. Unknown Dynamics: System Identification
4. Optimal Regret: Leveraging Curvature
5. Open Problems / Hardness Results

Roadmap: Assumptions

Roadmap: Assumptions

Assumption 1: Costs $c_t(x, u)$ are convex, $O(1)$-Lipschitz

Roadmap: Assumptions

Assumption 1: Costs $c_t(x, u)$ are convex, $O(1)$-Lipschitz

Assumption 2: Disturbances are uniformly bounded $\sup_t \|w_t\|, \|e_t\| \leq 1$

Roadmap: Assumptions

Assumption 1: Costs $c_t(x, u)$ are convex, $O(1)$-Lipschitz

(can be relaxed)

Assumption 2: Disturbances are uniformly bounded $\sup_t \|w_t\|, \|e_t\| \leq 1$

(can be relaxed)

The Gradient Perturbation Controller (GPC)

Roadmap

1. GPC: Fully Observed, Known-Dynamics

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Warmup: Known System + Stable Dynamics

1. **Fully Observed:** $y_t \equiv x_t$
2. **Known Dynamics:** $x_{t+1} = Ax_t + Bu_t + w_t$
3. **Stable Dynamics:** $\|A^s\| \leq C\rho^s$

Warmup: Known System + Stable Dynamics

1. **Fully Observed:** $y_t \equiv x_t$

2. **Known Dynamics:** $x_{t+1} = Ax_t + Bu_t + w_t$

3. **Stable Dynamics:** $\|A^s\| \leq C\rho^s$

(Don't worry: all will be relaxed)

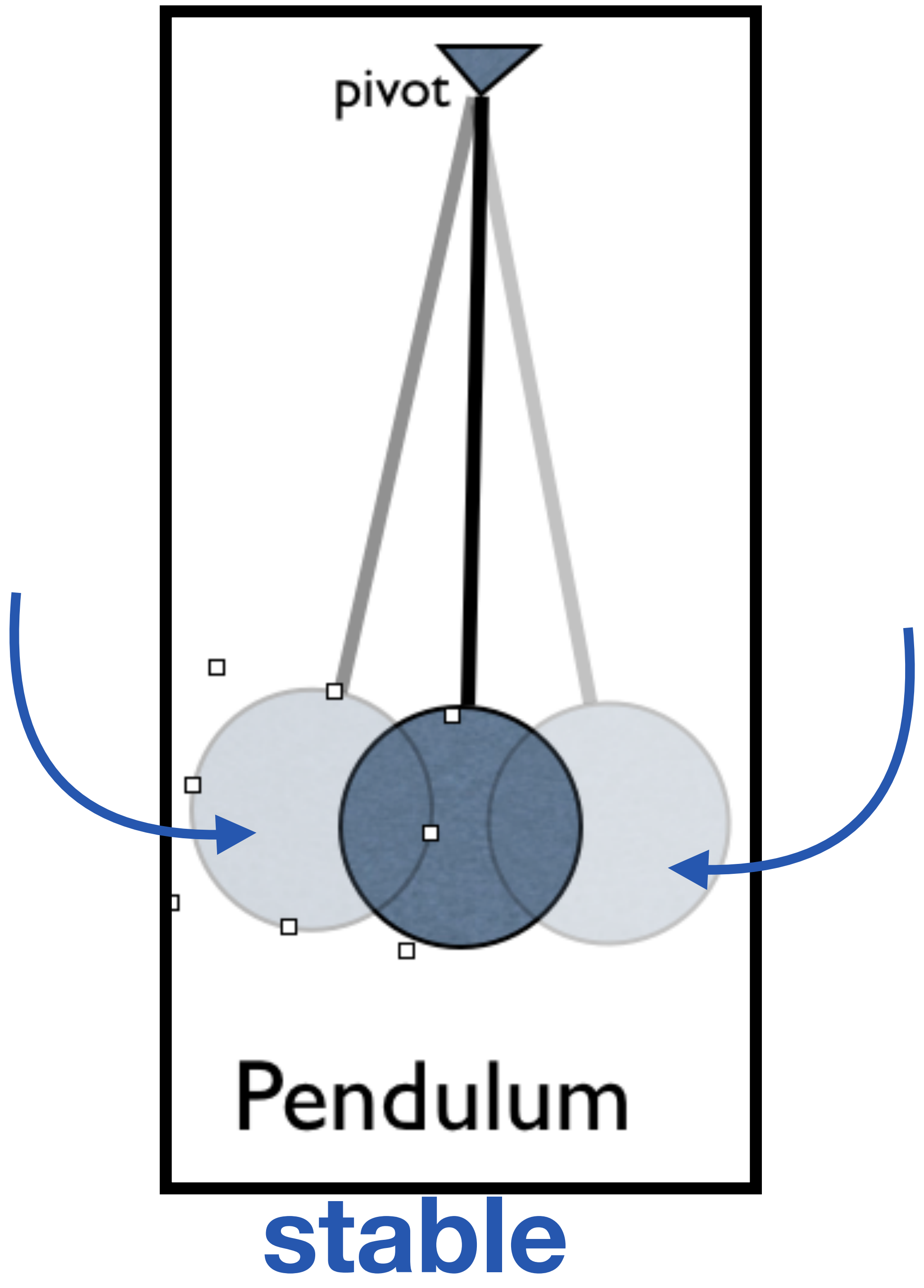
Warmup: Known System + Stable Dynamics

1. **Fully Observed:** $y_t \equiv x_t$

2. **Known Dynamics:** $x_{t+1} = Ax_t + Bu_t + w_t$

3. **Stable Dynamics:** $\|A^s\| \leq C\rho^s$

(Don't worry: all will be relaxed)



Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = J_T(\mathbb{A};W) - \min_{\pi\in\Pi} J_T(\pi;W)$ **small.**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = \underline{J_T(\mathbb{A};W)} - \underline{\min_{\pi\in\Pi} J_T(\pi;W)}$ **small.**

How to compete benchmark online, despite unknown costs/ disturbances

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = \underline{J_T(\mathbb{A};W)} - \underline{\min_{\pi\in\Pi} J_T(\pi;W)}$ **small.**

How to compete benchmark online, despite unknown costs/disturbances

How to efficiently parameterize control policies $\pi \in \Pi$?

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = \underline{J_T(\mathbb{A};W)} - \underline{\min_{\pi\in\Pi} J_T(\pi;W)}$ **small.**

How to compete benchmark online, despite unknown costs/disturbances

How to efficiently parameterize control policies $\pi \in \Pi$?

Tool: Online Convex Optimization

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Roadmap: Core Challenges

Goal: make $\mathrm{Reg}_T(\mathbb{A};\Pi) = \underline{J_T(\mathbb{A};W)} - \underline{\min_{\pi\in\Pi} J_T(\pi;W)}$ **small.**

How to compete benchmark online, despite unknown costs/disturbances

How to efficiently parameterize control policies $\pi \in \Pi$?

Tool: Online Convex Optimization

Tool: Convex Control Parametrization

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

The Gradient Perturbation Controller

For $t = 1,2,\ldots$

The Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ defined in terms of $M = (M^{[0]}, \ldots, M^{[k]})$ (convex parametrization)

The Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ defined in terms of $M = (M^{[0]}, \ldots, M^{[k]})$ **(convex parametrization)**

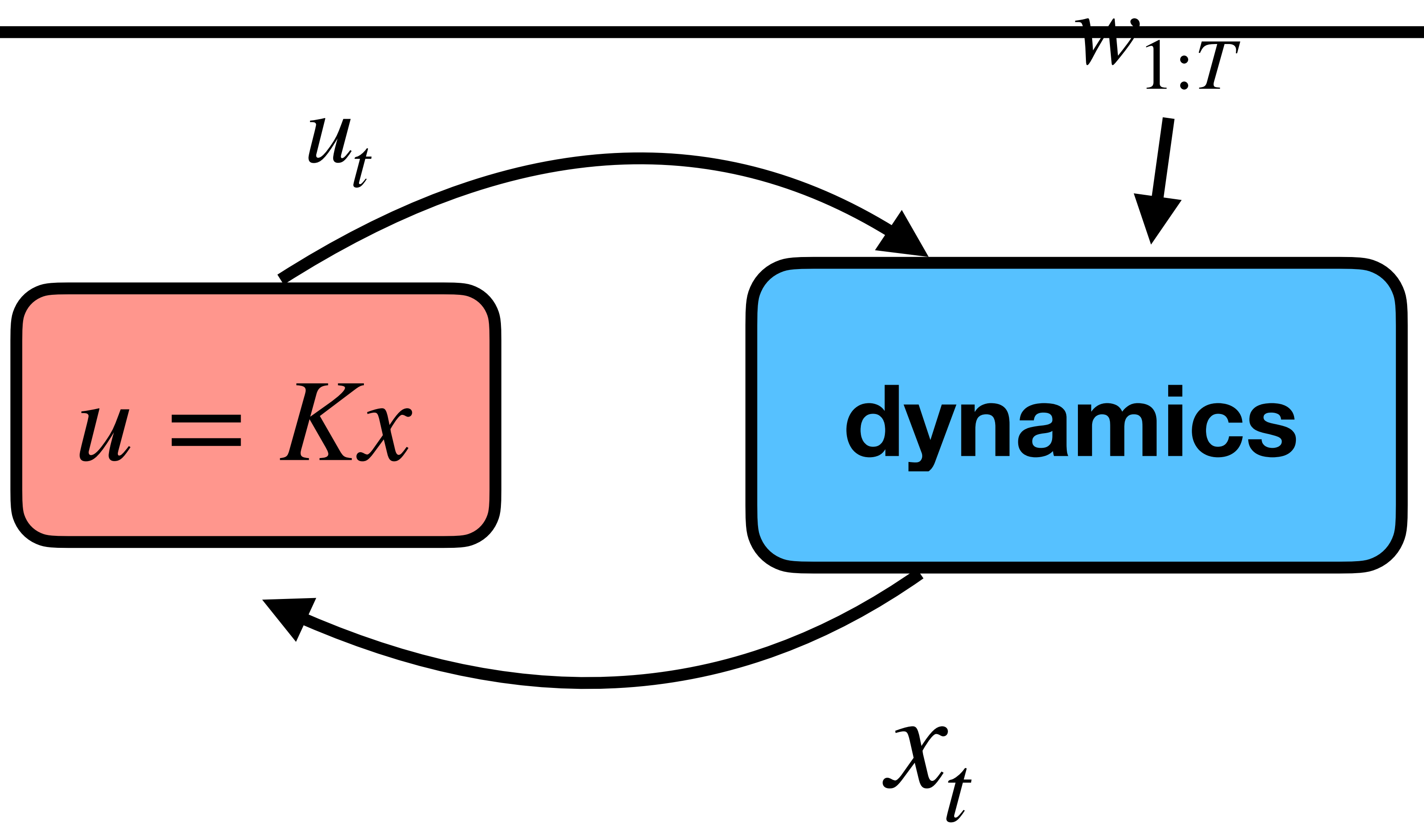
2. $\boxed{M_{t+1} \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$ where $\tilde{F}_t$ is convex **(online gradient descent)**

Goal: Known System + Stable Dynamics

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

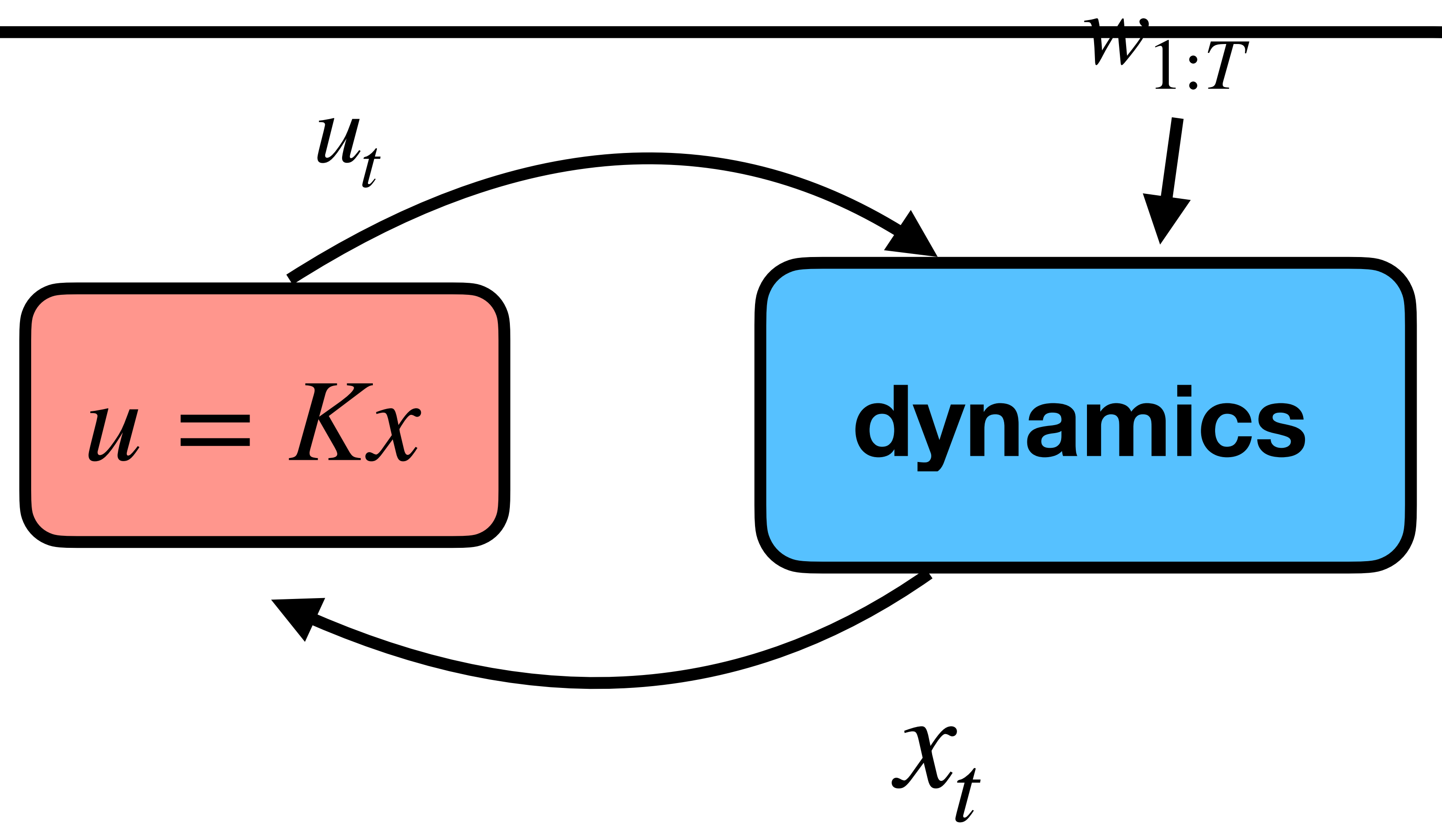
$$\Pi_{\text{feedback}} := \{\pi(x) = Kx : A + BK \text{ is } (C, \rho) \text{ stable})$$



Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

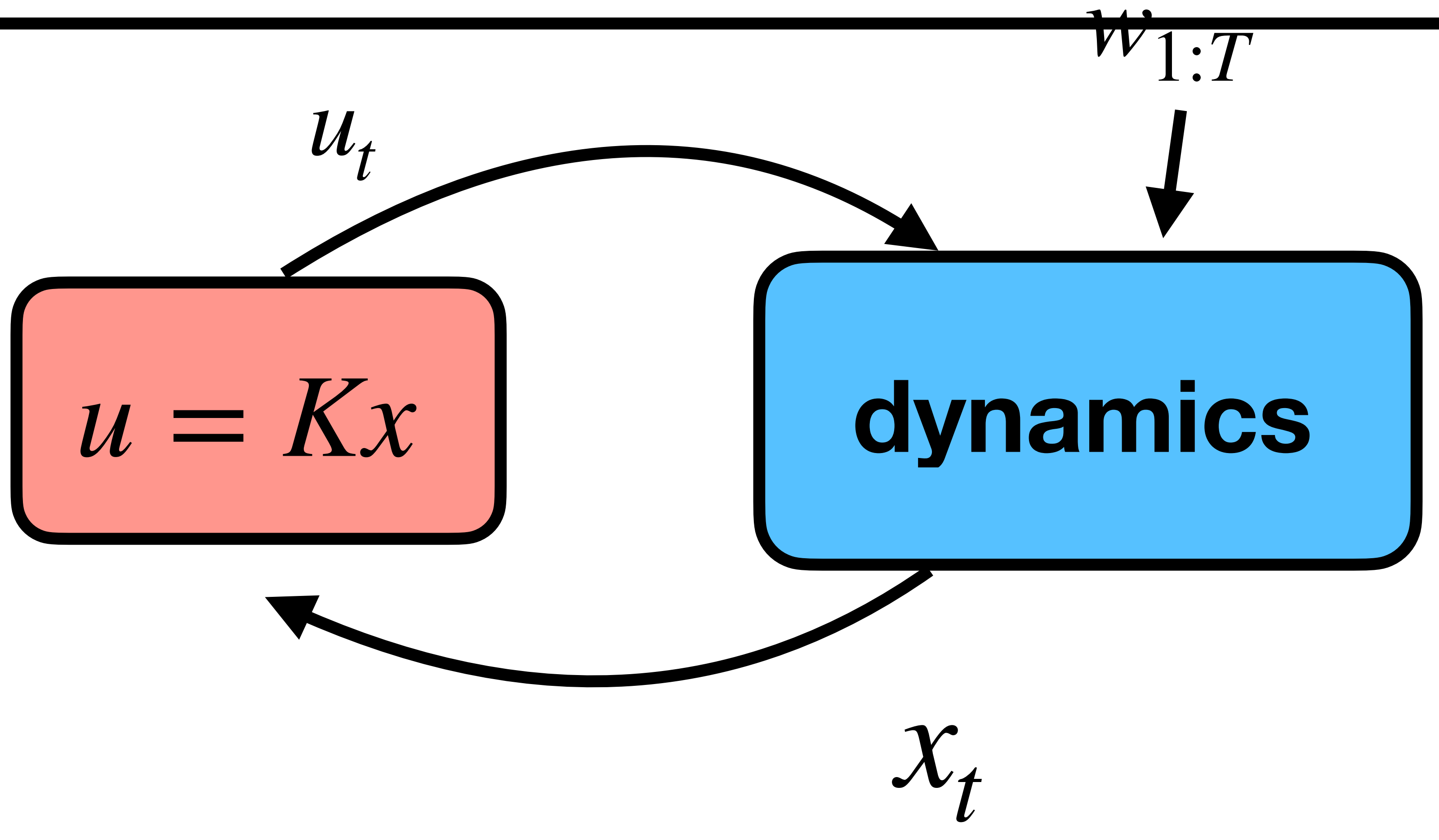
$$\Pi_{\text{feedback}} := \{\pi(x) = Kx : \underbrace{A + BK}_{\text{closed loop dynamics}} \text{ is } (C, \rho) \text{ stable})$$



Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

$$\Pi_{\text{feedback}} := \{\pi(x) = Kx : \underbrace{A + BK}_{\text{closed loop dynamics}} \text{ is } (C, \rho) \text{ stable})$$

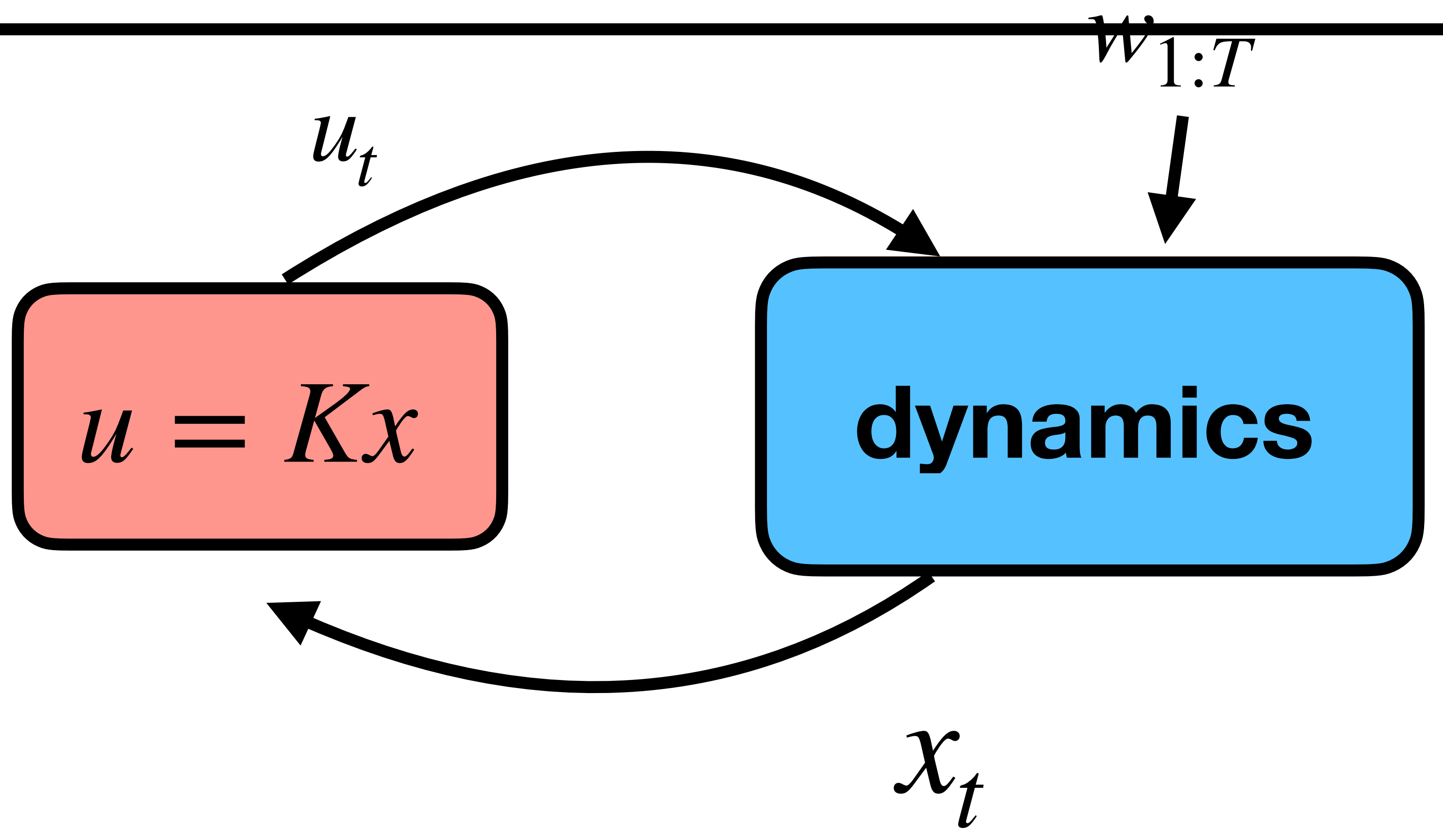


Includes optimal $\mathcal{H}_2, \mathcal{H}_\infty$ controllers

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

$\Pi_{\text{feedback}} := \{\pi(x) = Kx : A + BK$ is (C, ρ) stable$)$



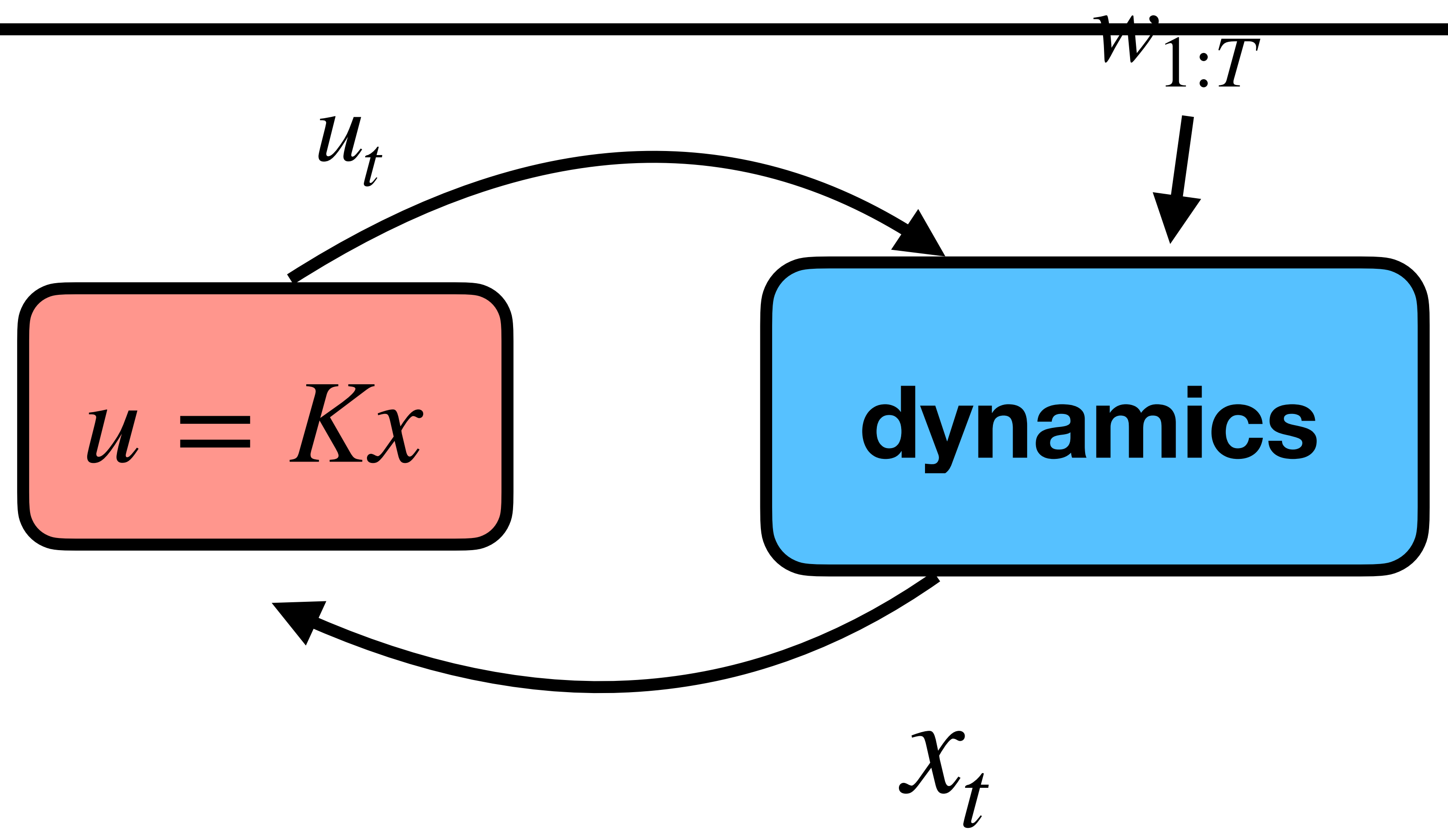
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

Theorem: Gradient Perturbation Control (GPC) attains

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\mathrm{feedback}}} J_T(\pi^K; W) \leq \tilde{O}(\sqrt{T})$$

$\Pi_{\mathrm{feedback}} := \{\pi(x) = Kx : A + BK$ is (C, ρ) stable)



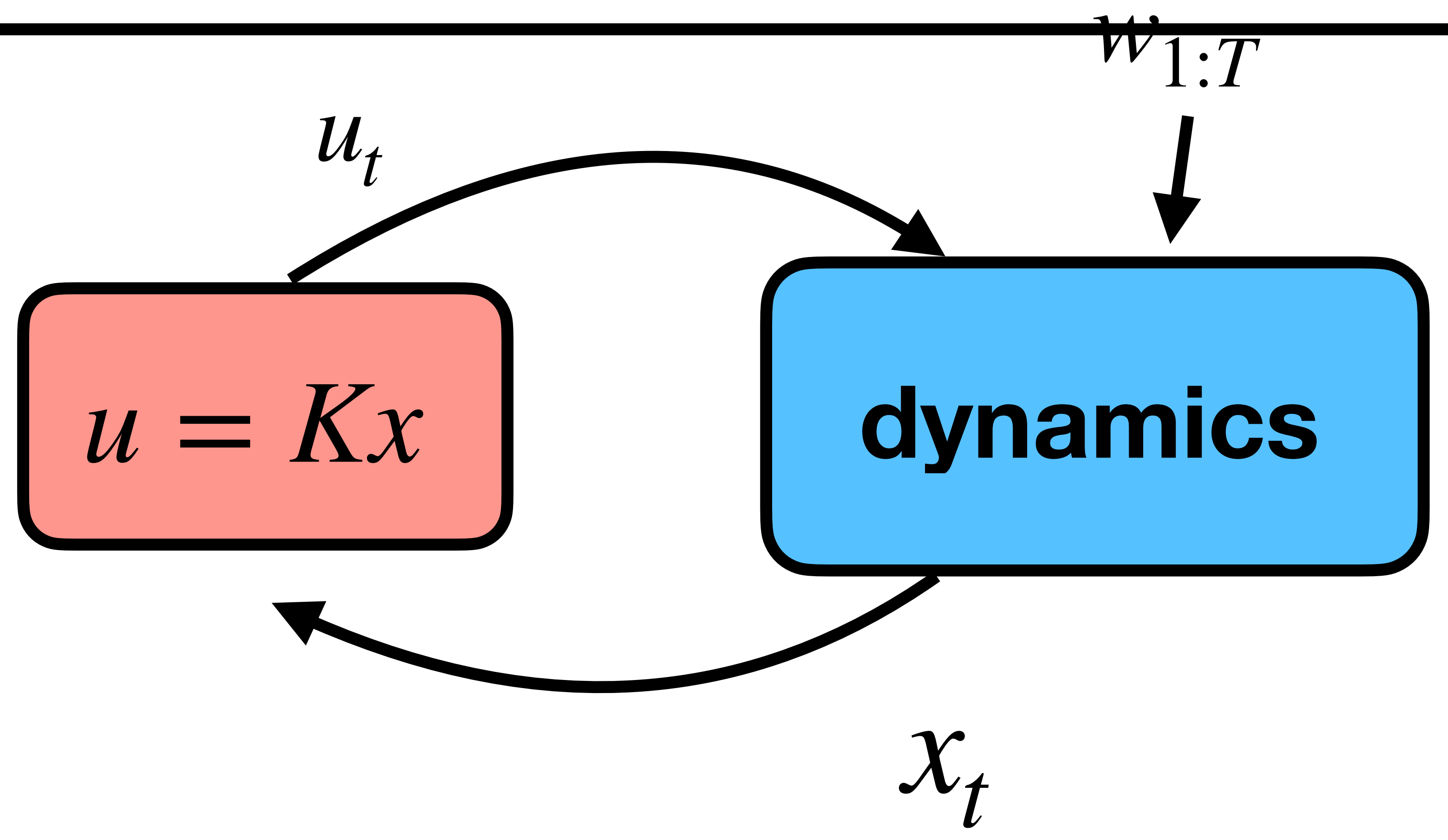
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

Theorem: Gradient Perturbation Control (GPC) attains

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^K; W) \le \tilde{O}(\sqrt{T})$$

$\Pi_{\text{feedback}} := \{\pi(x) = Kx : A + BK$ is (C, ρ) stable)



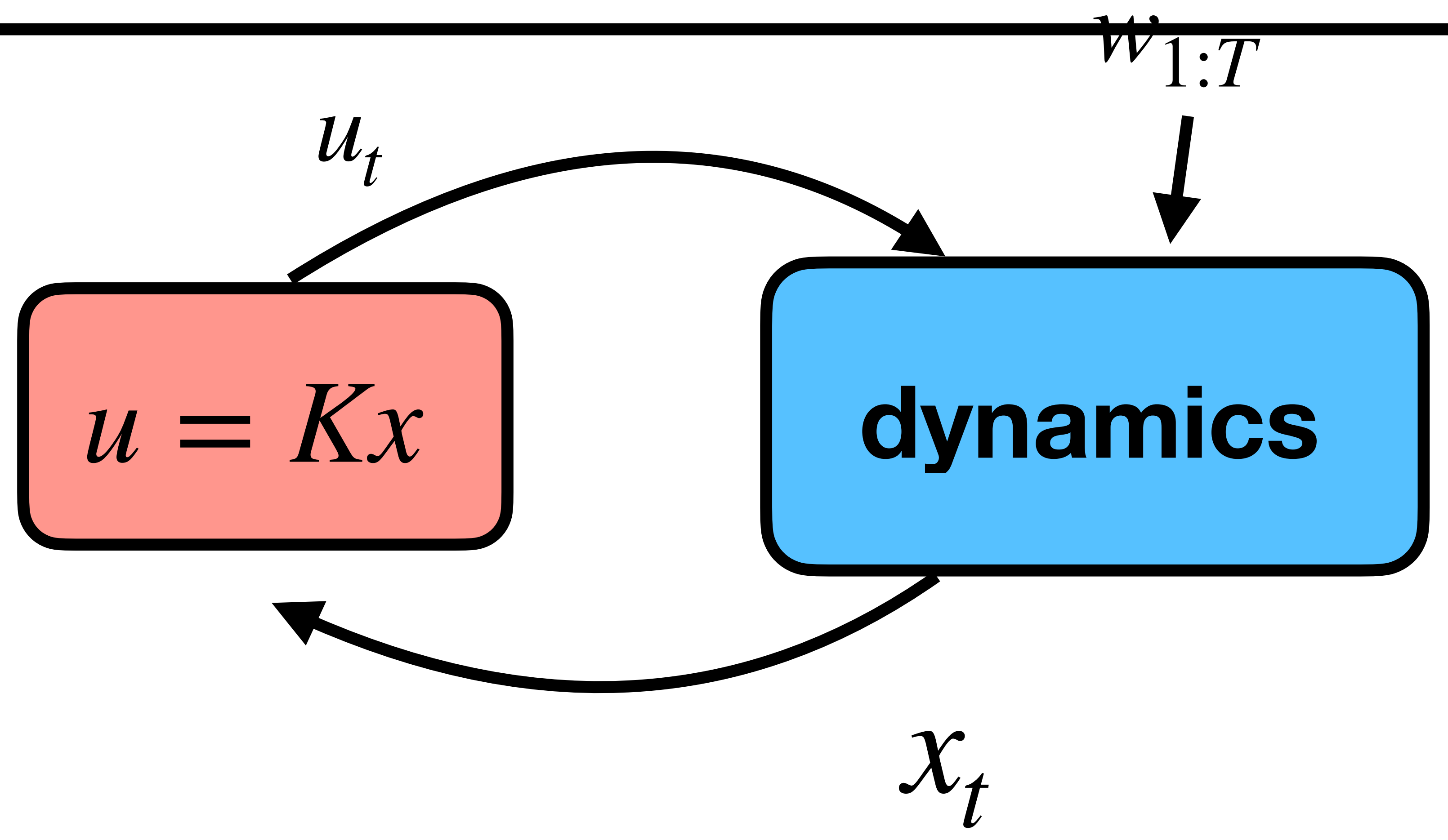
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

Theorem: Gradient Perturbation Control (GPC) attains

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^K; W) \leq \tilde{O}(\sqrt{T})$$

$\Pi_{\text{feedback}} := \{\pi(x) = Kx : A + BK$ is (C, ρ) stable)



Includes optimal $\mathscr{H}_2, \mathscr{H}_\infty$ controllers

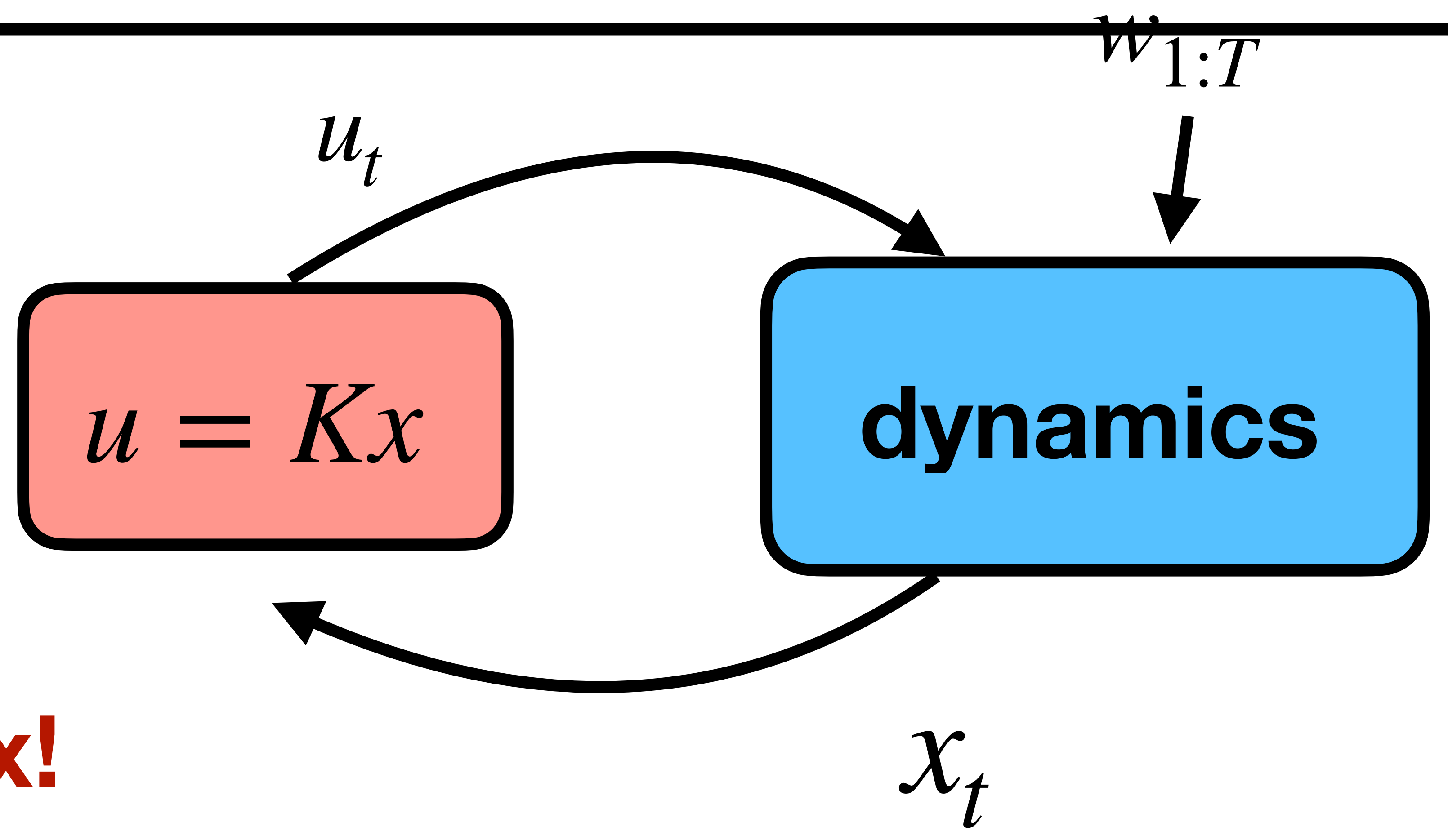
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Goal: Known System + Stable Dynamics

Theorem: Gradient Perturbation Control (GPC) attains

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\mathrm{feedback}}} J_T(\pi^K; W) \le \tilde{O}(\sqrt{T})$$

$\Pi_{\mathrm{feedback}} := \{\pi(x) = Kx : A + BK$ is (C, ρ) stable$\}$



Includes optimal $\mathscr{H}_2, \mathscr{H}_\infty$ controllers **but non-convex!**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

Equivalent to the SLS Parametrization of *(Anderson et al, 2019)*

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

this is implementable **online** with known dynamics: $w_t = x_{t+1} - (Ax_t + Bu_t)$

Equivalent to the SLS Parametrization of *(Anderson et al, 2019)*

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

independent of past control inputs

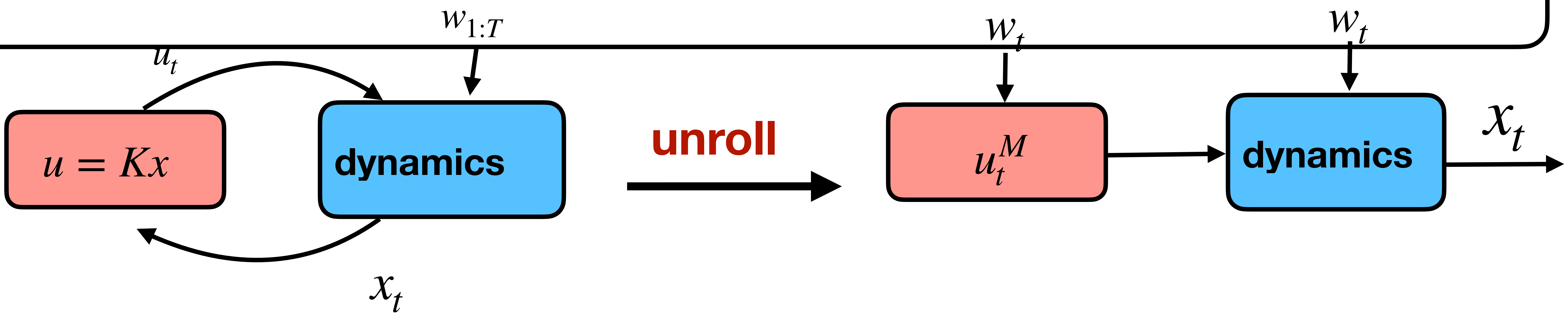
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

independent of past control inputs



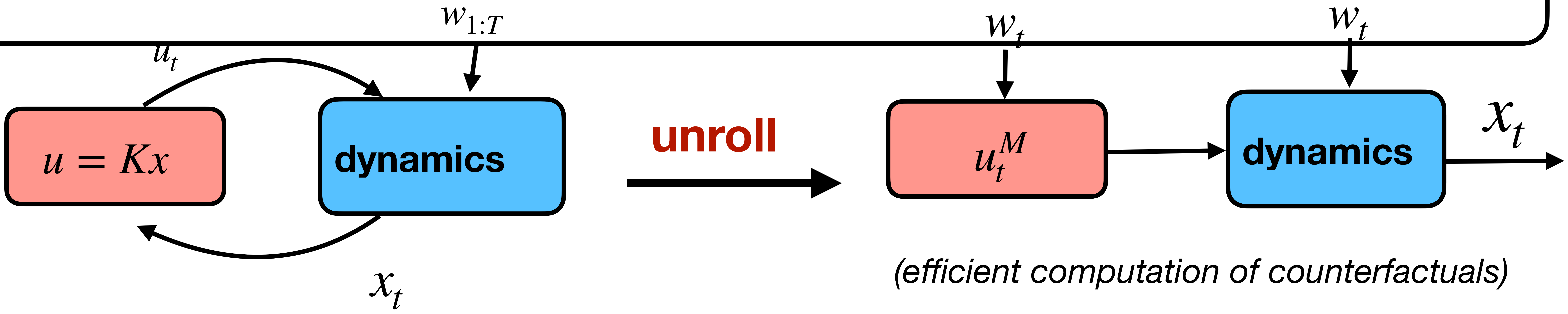
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Define: The **Disturbance Feedback Control (DFC)** parameterization:

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

independent of past control inputs



(efficient computation of counterfactuals)

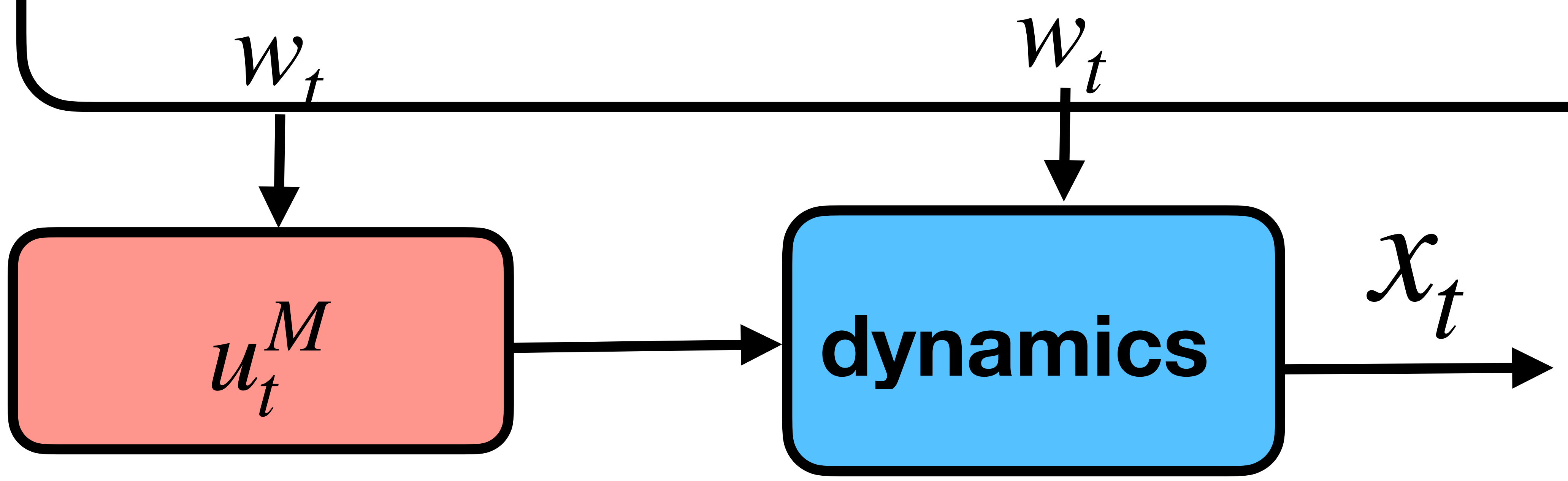
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Observation: The mapping from $M \rightarrow (x_t^M, u_t^M)$ is **linear**

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

independent of past control inputs



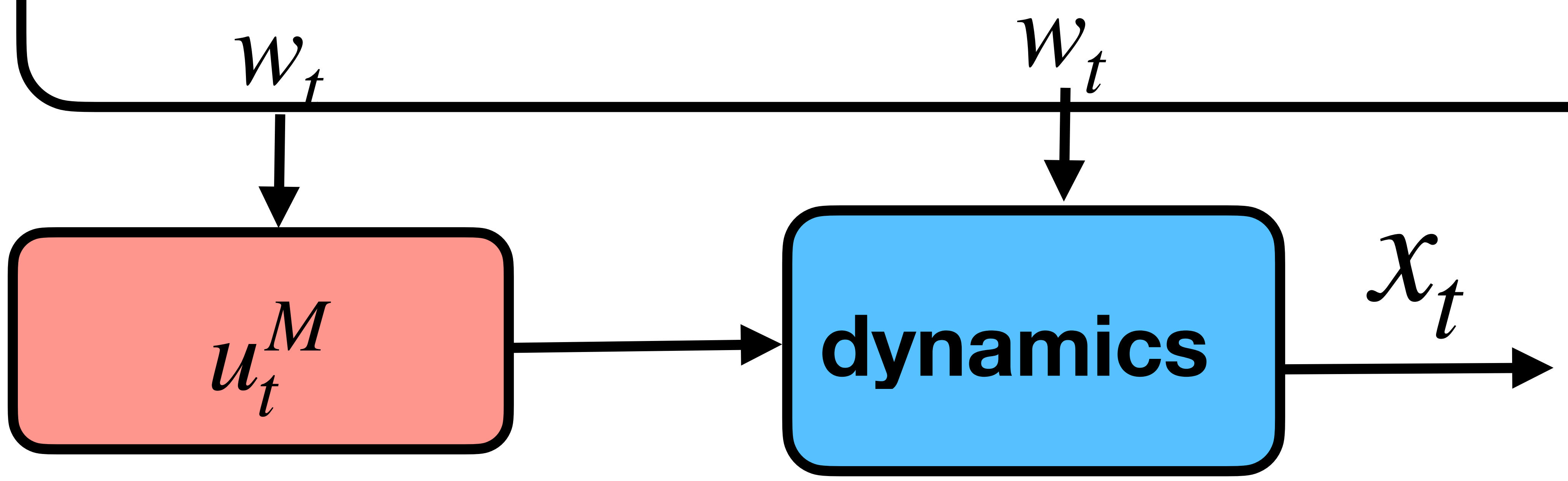
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Observation: The mapping from $M \to (x_t^M, u_t^M)$ is **linear**

$$\boxed{u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}}$$

independent of past control inputs



Corollary: Assuming convex costs, mapping $M \to J_T(\pi^M; W) := \sum_{t=1}^{T} c_t(x_t^M, u_t^M)$ is **convex**

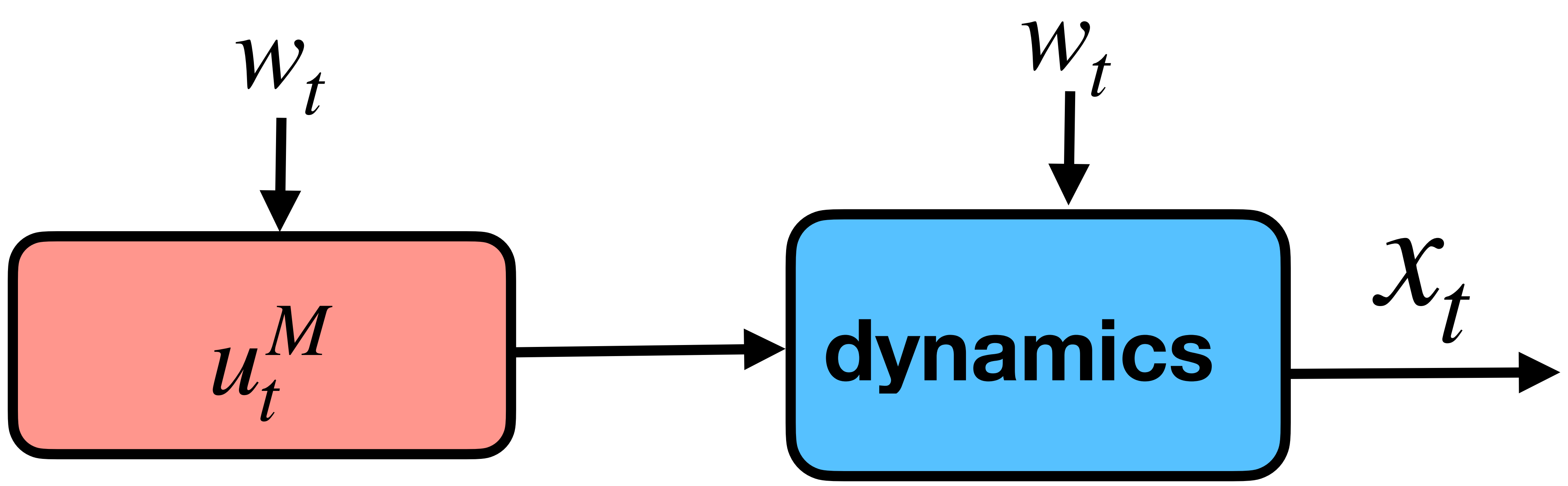
Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Observation: The mapping from $M \rightarrow (x_t^M, u_t^M)$ is **linear**

$$u_t^M = \sum_{i=1}^{k} M^{[i]} w_{t-i}$$

Corollary: By linearity of dynamics, mapping $M \rightarrow J_T(\pi^M; W) := \sum_{t=1}^{T} c_t(x_t^M, u_t^M)$ is **convex**



Therefore, in hindsight, we can **efficiently optimize** over controllers.

In learning theory, we call this **improper learning.**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Theorem: Consider any controller K such that $A + BK$ is (C, ρ) stable.

Then, $\exists$ a DFC controller $u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i}$ with $\|M\| \le O^\star(1)$ s.t.

$\sup_t \|x_t^K - x_t^M\| \le O_\star(\rho^k)$, where $O_\star(1) = \mathrm{poly}(C, (1-\rho)^{-1})$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Theorem: Consider any controller K such that $A + BK$ is (C, ρ) stable.

Then, $\exists$ a DFC controller $u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i}$ with $\|M\| \leq O^\star(1)$ s.t.

$\sup_t \|\underline{x_t^K} - \underline{x_t^M}\| \leq O_\star(\rho^k)$, where $O_\star(1) = \mathrm{poly}(C, (1-\rho)^{-1})$

states under K states under M

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Theorem: Consider any controller K such that $A + BK$ is (C, ρ) stable.

Then, $\exists$ a DFC controller $u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i}$ with $\|M\| \leq O^\star(1)$ s.t.

$\sup_t \|x_t^K - x_t^M\| \leq O_\star(\rho^k)$, where $O_\star(1) = \mathrm{poly}(C, (1-\rho)^{-1})$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Theorem: Consider any controller K such that $A + BK$ is (C, ρ) stable.

Then, $\exists$ a DFC controller $u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i}$ with $\|M\| \leq O^\star(1)$ s.t.

$\sup_t \|x_t^K - x_t^M\| \leq O_\star(\rho^k)$, where $O_\star(1) = \text{poly}(C, (1-\rho)^{-1})$

Informally: DFC Controllers are an **improper relaxation** of static feedback controllers

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Corollary: Let Π_{feedback} denote all policies $\pi(x) = Kx$ makes s.t. $A + BK$ is (C, ρ) stable. Then, the class Π_{gpc} of all memory-k controllers with

$$\boxed{u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i} \qquad \sum_i \|M^{[i]}\| \leq O_\star(1)}$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Corollary: Let Π_{feedback} denote all policies $\pi(x) = Kx$ makes s.t. $A + BK$ is (C, ρ) stable. Then, the class Π_{gpc} of all memory-k controllers with

$$u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i} \qquad \sum_i \|M^{[i]}\| \leq O_\star(1)$$

satisfies

$$\inf_M J_T(\Pi_{\text{gpc}}) - \inf_K J_T(\Pi_{\text{feedback}}) \leq O_\star(T\rho^k)$$

(assuming Lipschitz costs)

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Corollary: Let Π_{feedback} denote all policies $\pi(x) = Kx$ makes s.t. $A + BK$ is (C, ρ) stable. Then, the class Π_{gpc} of all memory-k controllers with

$$u_t^M = \sum_{i=0}^{k} M^{[i]} w_{t-i} \qquad \sum_i \|M^{[i]}\| \le O_\star(1)$$

satisfies

$$\inf_M J_T(\Pi_{\text{gpc}}) - \inf_K J_T(\Pi_{\text{feedback}}) \le O_\star(T\rho^k)$$

(assuming Lipschitz costs)

suffices to optimize over Π_{gpc}

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

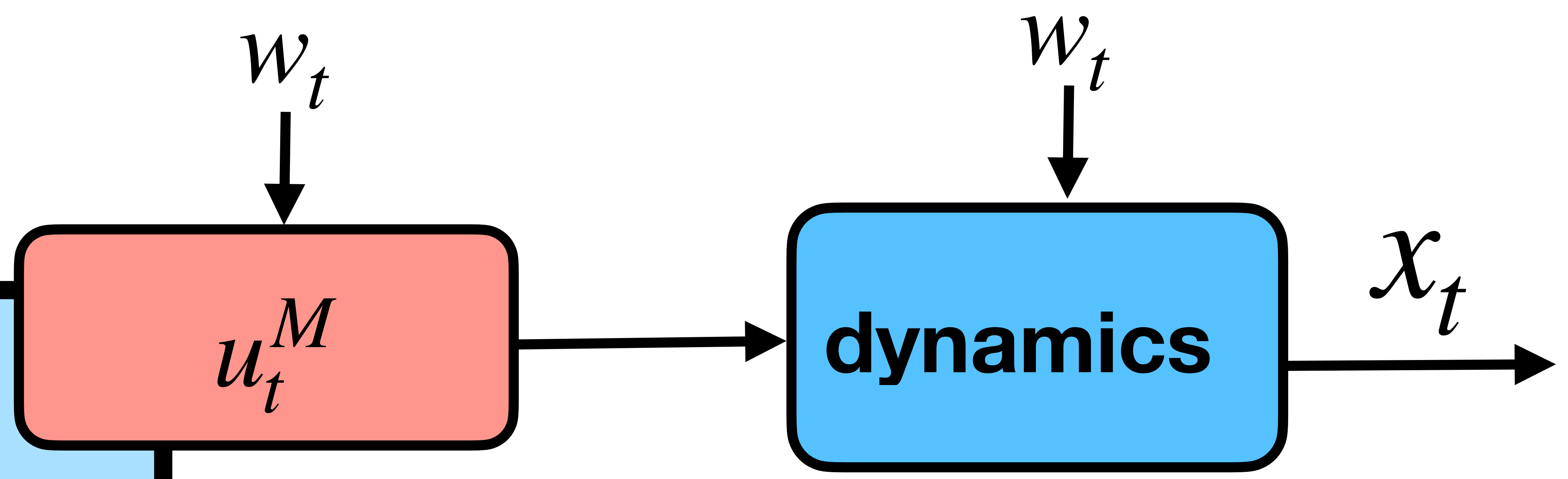
Tool 1: Convex Controller Parametrization

Summary

Agrawal, Bullins, Hazan, Kakade, Singh “Online Control with Adversarial Disturbances”, 2019

Tool 1: Convex Controller Parametrization

Summary

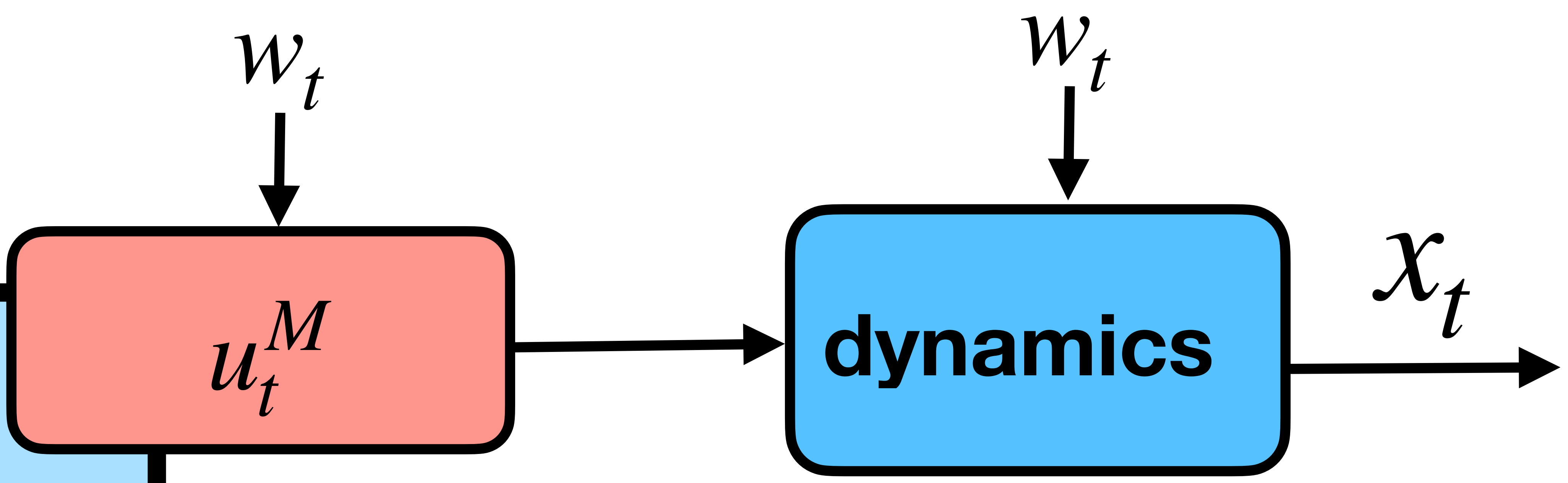


1. Efficient optimization mapping from $M \to J_T(\pi^M; W) := \sum_{t=1}^{T} c_t(x_t^M, u_t^M)$ is **convex**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 1: Convex Controller Parametrization

Summary



1. Efficient optimization mapping from $M \to J_T(\pi^M; W) := \sum_{t=1}^{T} c_t(x_t^M, u_t^M)$ is **convex**

2. For bounded M of memory k: $\inf_M J_T(\Pi_{\text{gpc}}) - \inf_K J_T(\Pi_{\text{feedback}}) \leq O_\star(T\rho^k)$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

The Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$ ✓

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

The Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_{t+1} \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$ **where** $\tilde{F}_t$ **is convex**

(online gradient descent)

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2: Online Convex Optimization

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1, 2, \ldots,$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Nature selects **convex loss function** $f_t : \Theta \to \mathbb{R}$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Nature selects **convex loss function** $f_t : \Theta \to \mathbb{R}$

Goal: Make $\mathrm{OcoReg}_T := \sum_{t=1}^{T} f_t(\theta_t) - \inf_{\theta \in \Theta} \sum_{t=1}^{T} f_t(\theta) \ \leq o(T)$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Nature selects **convex loss function** $f_t : \Theta \to \mathbb{R}$

Goal: Make $\mathrm{OcoReg}_T := \underbrace{\sum_{t=1}^{T} f_t(\theta_t)}_{\text{realized loss}} - \inf_{\theta\in\Theta} \sum_{t=1}^{T} f_t(\theta) \leq o(T)$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Nature selects **convex loss function** $f_t : \Theta \to \mathbb{R}$

Goal: Make $\mathrm{OcoReg}_T := \underbrace{\sum_{t=1}^{T} f_t(\theta_t)}_{\text{realized loss}} - \inf_{\theta\in\Theta} \underbrace{\sum_{t=1}^{T} f_t(\theta)}_{\text{best-in-hindsight}} \leq o(T)$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Nature selects **convex loss function** $f_t : \Theta \to \mathbb{R}$

Intuition: $\mathrm{OcoReg}_T := \sum_{t=1}^{T} f_t(\theta_t) - \inf_{\theta \in \Theta} \sum_{t=1}^{T} f_t(\theta) \;\leq o(T)$

Tool 2: Online Convex Optimization

Protocol: Online Convex Optimization.

For times $t = 1,2,\ldots,$

Learner selects action $\theta_t \in \Theta$

Nature selects **convex loss function** $f_t : \Theta \to \mathbb{R}$

Intuition: $\mathrm{OcoReg}_T := \sum_{t=1}^{T} f_t(\theta_t) - \inf_{\theta \in \Theta} \sum_{t=1}^{T} f_t(\theta) \ \leq o(T)$

forces learning under adversarial uncertainty!

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Optimization.

For times $t = 1, 2, \ldots,$

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Optimization.

For times $t = 1, 2, \ldots,$

Learner updates $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Optimization.

For times $t = 1, 2, \ldots,$

Learner updates $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$

step size

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Optimization.

For times $t = 1,2,\ldots,$

Learner updates $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$

step size **gradient** (or convex subgradient)

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Descent (OGD).

For times $t = 1,2,\ldots,$

Learner updates $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Descent (OGD).

For times $t = 1,2,\ldots,$

Learner updates $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$

Theorem (Zinkevich '03)**:** Suppose that $\mathrm{Diam}(\Theta) \leq D$ and each f_t is G-Lipschitz. Then OGD with step size $\eta_t = (DG) \cdot \frac{1}{\sqrt{t}}$ satisfies

Tool 2: Online Convex Optimization

Algorithm: Online Gradient Descent (OGD).

For times $t = 1, 2, \ldots,$

Learner updates $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$

Theorem (Zinkevich '03)**:** Suppose that $\mathrm{Diam}(\Theta) \leq D$ and each f_t is G-Lipschitz. Then OGD with step size $\eta_t = (DG) \cdot \frac{1}{\sqrt{t}}$ satisfies

$$\mathrm{OcoReg}_T := \sum_{t=1}^{T} f_t(\theta_t) - \inf_{\theta \in \Theta} \sum_{t=1}^{T} f_t(\theta) \leq 2DG\sqrt{T}$$

Tool 2': Reducing Online Control to OCO

Protocol: Online Control over GPC Parameterization

For times $t = 1,2,\ldots,$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

Protocol: Online Control over GPC Parameterization

For times $t = 1,2,\ldots,$

Learner selects action M_t, and executes $u_t^{\mathbb{A}} = \sum_{i=0}^{k} M_t^{[i]} w_{t-i}$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

Protocol: Online Control over GPC Parameterization

For times $t = 1,2,\ldots,$

Learner selects action M_t, and executes $u_t^{\mathbb{A}} = \sum_{i=0}^{k} M_t^{[i]} w_{t-i}$

Nature selects **convex loss function** c_t and **noise** w_t

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

Protocol: Online Control over GPC Parameterization

For times $t = 1,2,\ldots,$

Learner selects action M_t, and executes $u_t^{\mathbb{A}} = \sum_{i=0}^{k} M_t^{[i]} w_{t-i}$

Nature selects **convex loss function** c_t and **noise** w_t

Learner suffers $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

Protocol: Online Control over GPC Parameterization

For times $t = 1,2,\ldots,$

Learner selects action M_t, and executes $u_t^{\mathbb{A}} = \sum_{i=0}^{k} M_t^{[i]} w_{t-i}$

Nature selects **convex loss function** c_t and **noise** w_t

Learner suffers $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$

Dynamics evolve $x_{t+1}^{\mathbb{A}} = Ax_t^{\mathbb{A}} + Bu_t^{\mathbb{A}} + w_t$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2’: Reducing Online Control to OCO

Agrawal, Bullins, Hazan, Kakade, Singh “Online Control with Adversarial Disturbances”, 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

counterfactual cost with memory k

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

counterfactual cost with memory k

Specifically $\tilde{F}_t(M) = F_t(M, \ldots, M) = c_t\left(\sum_{i=1}^{k} A^{i-1} B u_{t-i}^{M}, u_t^{M}\right)$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \le s \le t$

counterfactual cost with memory k

Specifically $\tilde{F}_t(M) = F_t(M, \ldots, M) = c_t\left(\sum_{i=1}^{k} A^{i-1} B u_{t-i}^{M}, u_t^{M}\right)$

This is convex in M!

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

counterfactual cost with memory k

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

counterfactual cost with memory k

Update $M_{t+1} \leftarrow M_t - \eta \nabla \tilde{F}_t(M_t)$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

counterfactual cost with memory k

Update $M_{t+1} \leftarrow M_t - \eta \nabla \tilde{F}_t(M_t)$ **Online Gradient Descent**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t - k \leq s \leq t$

counterfactual cost with memory k

⚠️⚠️**Warning: Technical Part**⚠️⚠️

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**

2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_s \leftarrow u_s^{M_s},\ t-k \leq s \leq t$

counterfactual cost with memory k

⚠️⚠️**Warning: Technical Part**⚠️⚠️

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**
2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow u_{t-\ell}^{M}$ **cost with memory k**

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^M; W)$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**
2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow u_{t-\ell}^M$ **cost with memory k**

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\mathrm{feedback}}} J_T(\pi^M; W)$$

$$= \sum_{t=1}^T c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}}) - \inf_{M \in \Pi_{\mathrm{gpc}}} \sum_{t=1}^T c_t(x_t^M, u_t^M) + O^\star(T\rho^k)$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**
2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow u_{t-\ell}^M$ **cost with memory k**

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^M; W)$$

$$= \sum_{t=1}^T c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}}) - \inf_{M \in \Pi_{\text{gpc}}} \sum_{t=1}^T c_t(x_t^M, u_t^M) + O^\star(T\rho^k)$$

$$\boxed{= \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) + O_\star(T\rho^k)}$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**
2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow u_{t-\ell}^M$ **cost with memory k**

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^M; W)$$

$$= \sum_{t=1}^T c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}}) - \inf_{M \in \Pi_{\text{gpc}}} \sum_{t=1}^T c_t(x_t^M, u_t^M) + O^\star(T\rho^k)$$

$$\boxed{= \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) + O_\star(T\rho^k)}$$

Online Convex Optimization with Memory

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

1. $c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}})$ = **true algorithm cost**
2. $F_t(M_t, \ldots, M_{t-k}) = c_t(x_t, u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow u_{t-\ell}^M$ **cost with memory k**

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^M; W)$$

$$= \sum_{t=1}^T c_t(x_t^{\mathbb{A}}, u_t^{\mathbb{A}}) - \inf_{M \in \Pi_{\text{gpc}}} \sum_{t=1}^T c_t(x_t^M, u_t^M) + O^\star(T\rho^k)$$

$$\boxed{= \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) + O_\star(T\rho^k)}$$

Online Convex Optimization with Memory **stability**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\text{feedback}}) \le \sum_{t=1}^{T} F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^{T} F_t(M,\ldots,M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M,\ldots,M) \qquad u_t \leftarrow u_t^M$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\mathrm{feedback}}) \le \sum_{t=1}^{T} F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^{T} F_t(M,\ldots,M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \quad \tilde{F}_t(M) = F_t(M,\ldots,M) \quad u_t \leftarrow u_t^M$$

1. **Ignore long history:** $F_t(M_t,\ldots,M_{t-k}) = c_t(x_t,u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow \sum_{i=0}^{k} M^{[i]} w_{t-\ell-i}$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\text{feedback}}) \le \sum_{t=1}^{T} F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^{T} F_t(M,\ldots,M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M,\ldots,M) \qquad u_t \leftarrow u_t^M$$

1. **Ignore long history:** $F_t(M_t,\ldots,M_{t-k}) = c_t(x_t,u_t) \mid x_{t-k} \leftarrow 0,\ u_{t-\ell} \leftarrow \sum_{i=0}^{k} M^{[i]} w_{t-\ell-i}$

2. **Take gradient updates as if you M_t was not changing.**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) \le \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) + O_\star(T\rho^k)$$

Algorithm: **Gradient-Perturbation Controller (GPC)**

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M, \ldots, M) \qquad u_t \leftarrow u_t^M$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\text{feedback}}) \leq \sum_{t=1}^{T} F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^{T} F_t(M,\ldots,M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M,\ldots,M) \qquad u_t \leftarrow u_t^M$$

Theorem (OCO with Memory, Anava '13)**:** If $\eta_t = O(1/\sqrt{t})$, then

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\mathrm{feedback}}) \le \sum_{t=1}^T F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M,\ldots,M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M,\ldots,M) \qquad u_t \leftarrow u_t^M$$

Theorem (OCO with Memory, Anava '13)**:** If $\eta_t = O(1/\sqrt{t})$, then

$$\le \sum_{t=1}^T F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M,\ldots,M) \le O(k^2\sqrt{T})$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) \le \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M, \ldots, M) \qquad u_t \leftarrow u_t^M$$

Theorem (OCO with Memory, Anava '13)**:** If $\eta_t = O(1/\sqrt{t})$, then

$$\le \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) \le O(k^2\sqrt{T})$$

Intuition: Combine the standard regret for OCO with bound that

$$|F_t(M_t, \ldots, M_{t-k}) - \tilde{F}_t(M)| \le O_\star(1) \cdot \sum_{1 \le \ell, j, \le k} \eta_{t-i} \le k^2 \eta_{t-k} \lesssim O_\star(k^2\sqrt{T})$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\mathrm{feedback}}) \leq \sum_{t=1}^{T} F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^{T} F_t(M, \ldots, M) + O_\star(T\rho^k)$$

Algorithm: **Gradient-Perturbation Controller (GPC)**

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M, \ldots, M) \qquad u_t \leftarrow u_t^M$$

Corollary: If $\eta_t = O(1/\sqrt{t}), k \gg \log T$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\mathrm{feedback}}) \leq \sum_{t=1}^{T} F_t(M_t,\ldots,M_{t-k}) - \inf_M \sum_{t=1}^{T} F_t(M,\ldots,M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M,\ldots,M) \qquad u_t \leftarrow u_t^M$$

Corollary: If $\eta_t = O(1/\sqrt{t})$, $k \gg \log T$

$$\mathrm{Reg}_T(\mathbb{A};\Pi_{\mathrm{feedback}}) \leq O_\star(k^2\sqrt{T} + T\rho^k) = \tilde{O}(\sqrt{T})$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Tool 2': Reducing Online Control to OCO

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) \leq \sum_{t=1}^T F_t(M_t, \ldots, M_{t-k}) - \inf_M \sum_{t=1}^T F_t(M, \ldots, M) + O_\star(T\rho^k)$$

Algorithm: Gradient-Perturbation Controller (GPC)

$$M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t) \qquad \tilde{F}_t(M) = F_t(M, \ldots, M) \qquad u_t \leftarrow u_t^M$$

Corollary: If $\eta_t = O(1/\sqrt{t}),\, k \gg \log T$

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) \leq O_\star(k^2\sqrt{T} + T\rho^k) = \tilde{O}(\sqrt{T})$$

finally! we are done :)

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary: Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$ ✓

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary: Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ defined in terms of $M = (M^{[0]}, \ldots, M^{[k]})$ ✓

2. $\boxed{M_t \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$ where $\tilde{F}_t$ is convex ✓

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary: Gradient Perturbation Controller

For $t = 1,2,\ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ defined in terms of $M = (M^{[0]}, \ldots, M^{[k]})$ ✓
2. $\boxed{M_t \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$ where $\tilde{F}_t$ is convex ✓

Theorem: **Gradient Perturbation Control** (GPC) attains

$$\boxed{\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) = J_T(\mathbb{A}; W) - \inf_{\pi^K \in \Pi_{\text{feedback}}} J_T(\pi^M; W) \leq \tilde{O}(\sqrt{T})}$$ ✓

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Previously, we assumed **stable dynamics:** $\|A^s\| \leq C\rho^s$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Previously, we assumed **stable dynamics:** $\|A^s\| \leq C\rho^s$

Here, we assume we know **any** K_0 such that: $\|(A + BK_0)^s\| \leq C\rho^s$ **is closed-loop stable**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Previously, we assumed **stable dynamics:** $\|A^s\| \leq C\rho^s$

Here, we assume we know **any** K_0 such that**:** $\|(A + BK_0)^s\| \leq C\rho^s$ **is closed-loop stable**

e.g. f you know the dynamics, you can solve an LQR problem to get K_0

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Previously, we assumed **stable dynamics:** $\|A^s\| \leq C\rho^s$

Here, we assume we know **any** K_0 such that: $\|(A + BK_0)^s\| \leq C\rho^s$ **is closed-loop stable**

e.g. f you know the dynamics, you can solve an LQR problem to get K_0

**stay tuned for if you don't know* K_0

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Assume given **any** K_0 such that: $\|(A + BK_0)^s\| \leq C\rho^s$ **is closed-loop stable**

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Assume given **any** K_0 such that: $\|(A+BK_0)^s\| \leq C\rho^s$ **is closed-loop stable**

Theorem: GPC with $u_t \leftarrow K_0 x_t + \sum_{i=0}^{k} M^{[i]} w_{t-i}$ attains

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Assume given **any** K_0 such that: $\|(A + BK_0)^s\| \leq C\rho^s$ is **closed-loop stable**

Theorem: GPC with $u_t \leftarrow K_0 x_t + \sum_{i=0}^{k} M^{[i]} w_{t-i}$ attains

$$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{feedback}}) \leq \tilde{O}(\sqrt{T})$$

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

From Stable to Stabilized

Assume given **any** K_0 such that: $\|(A + BK_0)^s\| \leq C\rho^s$ **is closed-loop stable**

Theorem: GPC with $u_t \leftarrow K_0 x_t + \sum_{i=0}^{k} M^{[i]} w_{t-i}$ attains

$\mathrm{Reg}_T(\mathbb{A}; \Pi_{\mathrm{feedback}}) \leq \tilde{O}(\sqrt{T})$ **Proof:** Same, but fold K_0 into dynamics

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary

1. We introduce and analyze the **Gradient Perturbation Controller** (GPC)

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Summary

1. We introduce and analyze the **Gradient Perturbation Controller** (GPC)

2. It is built on **Disturbance Feedback Control** (DFC) as convex, “improper” representation of linear controllers (equivalent to SLS, *Anderson et al. ’19*)

Agrawal, Bullins, Hazan, Kakade, Singh “Online Control with Adversarial Disturbances”, 2019

Summary

1. We introduce and analyze the **Gradient Perturbation Controller** (GPC)

2. It is built on **Disturbance Feedback Control** (DFC) as convex, "improper" representation of linear controllers (equivalent to SLS, *Anderson et al. '19*)

3. We build on the **Online Convex Optimization** (OCO) framework to develop a gradient-based controller

Agrawal, Bullins, Hazan, Kakade, Singh "Online Control with Adversarial Disturbances", 2019

Generalizations

Roadmap

2. Nature's Y's: Partially Observed, Known-Dynamics
3. Unknown Dynamics: System Identification
4. Optimal Regret: Leveraging Curvature

Roadmap

2. Nature’s Y’s: Partially Observed, Known-Dynamics

From Full Observation to Nature's Y's

From Full Observation to Nature's Y's

Goal: Compete the linear controllers for **partially observed** $y_t = Cx_t + e_t$

From Full Observation to Nature’s Y’s

Goal: Compete the linear controllers for **partially observed** $y_t = Cx_t + e_t$

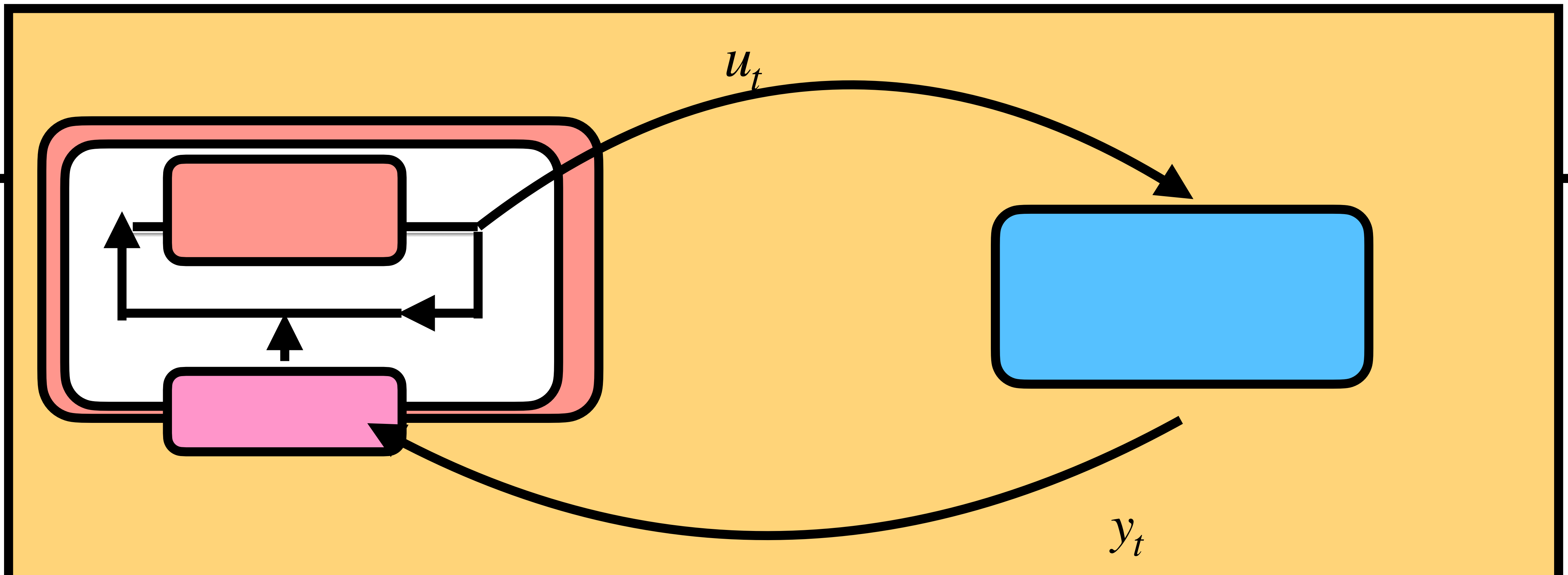
Challenge 1: GPC controller needed to “see” w_t, which are now **hidden**

From Full Observation to Nature's Y's

Goal: Compete the linear controllers for **partially observed** $y_t = Cx_t + e_t$

Challenge 1: GPC controller needed to "see" w_t, which are now **hidden**

Challenge 2: Static feedback on y_t, $u_t = Ky_t$, is **suboptimal** for partial observation.



$$z_{t+1} = A_\pi z_t + B_\pi y_t$$

$$u_t = C_\pi z_t + D_\pi y_t$$

From Full Observation to Nature's Y's

Idea: Convex parametrization (control lang.) or improperness (learning lang.)

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Idea: Convex parametrization (control lang.) or improperness (learning lang.)

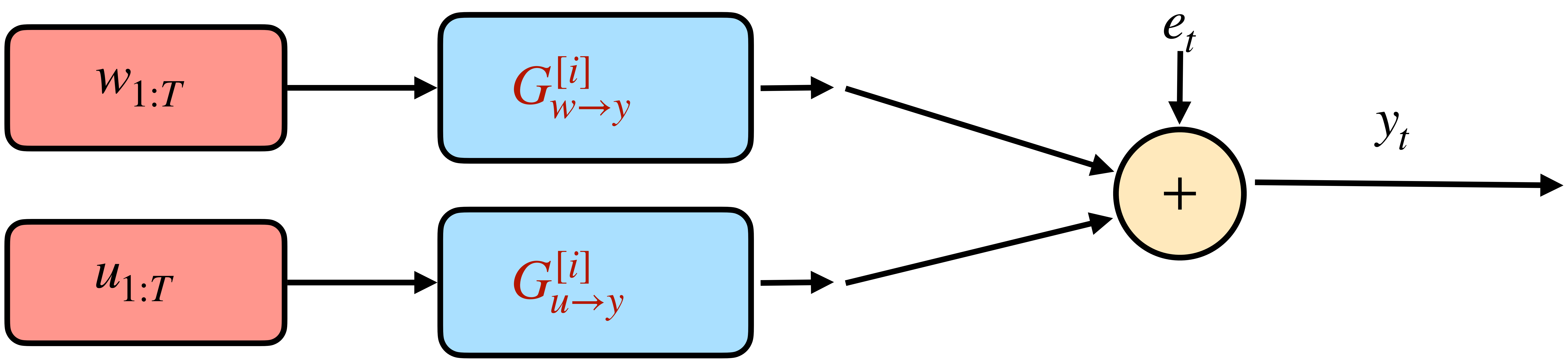
Define the **Markov Operators** $y_t = e_t + \sum_{i=0}^{t} G_{w \to y}^{[i]} w_{t-i} + G_{u \to y}^{[i]} u_{t-i}$

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Idea: Convex parametrization (control lang.) or improperness (learning lang.)

Define the **Markov Operators** $y_t = e_t + \sum_{i=0}^{t} G^{[i]}_{w\to y} w_{t-i} + G^{[i]}_{u\to y} u_{t-i}$

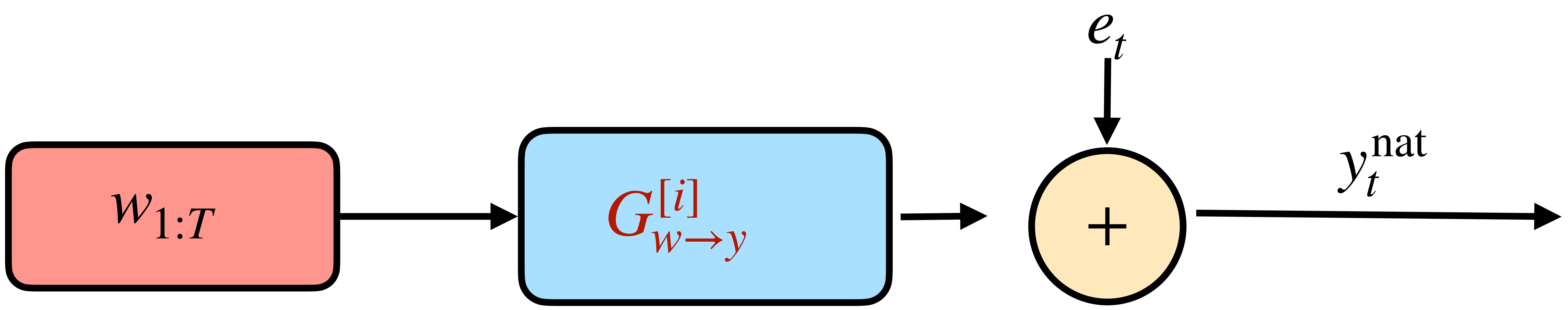


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Idea: Convex parametrization (control lang.) or improperness (learning lang.)

Define the **Markov Operators** $y_t = e_t + \sum_{i=0}^{t} G_{w\to y}^{[i]} w_{t-i} + G_{u\to y}^{[i]} u_{t-i}$

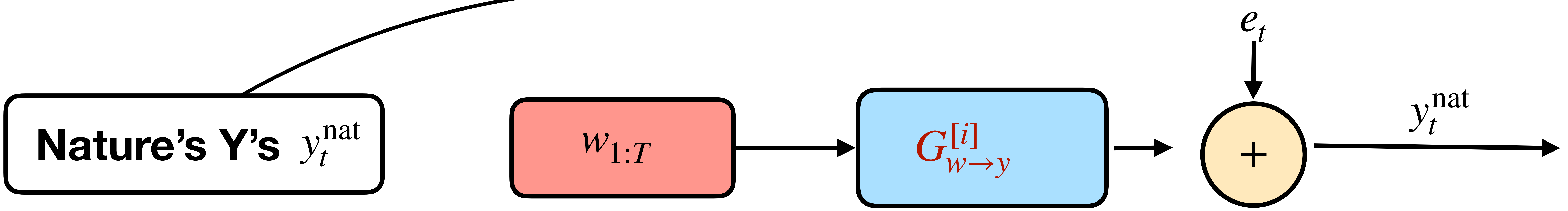


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

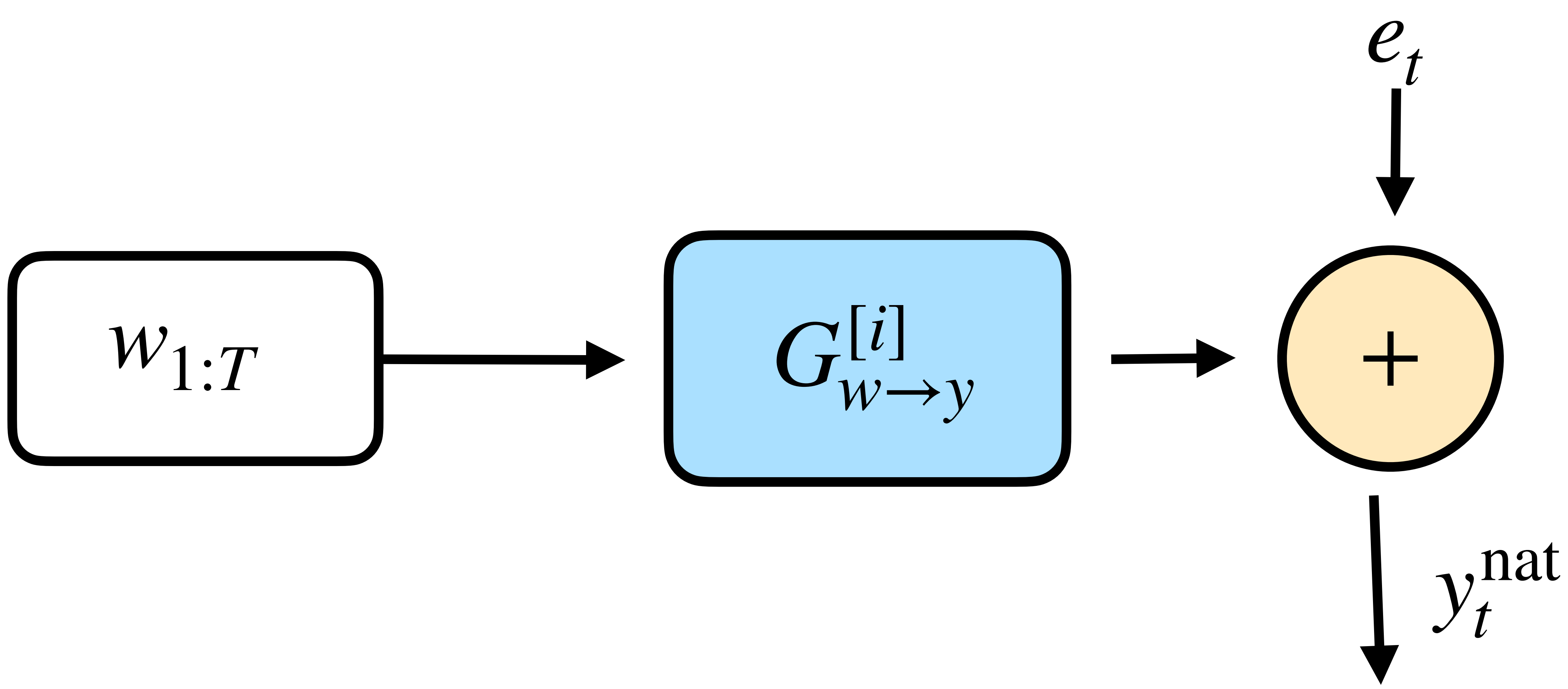
Idea: Convex parametrization (control lang.) or improperness (learning lang.)

Define the **Markov Operators** $\underline{y_t = e_t + \sum_{i=0}^{t} G^{[i]}_{w\to y} w_{t-i}} + \cancel{G^{[i]}_{u\to y} u_{t-i}}$



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

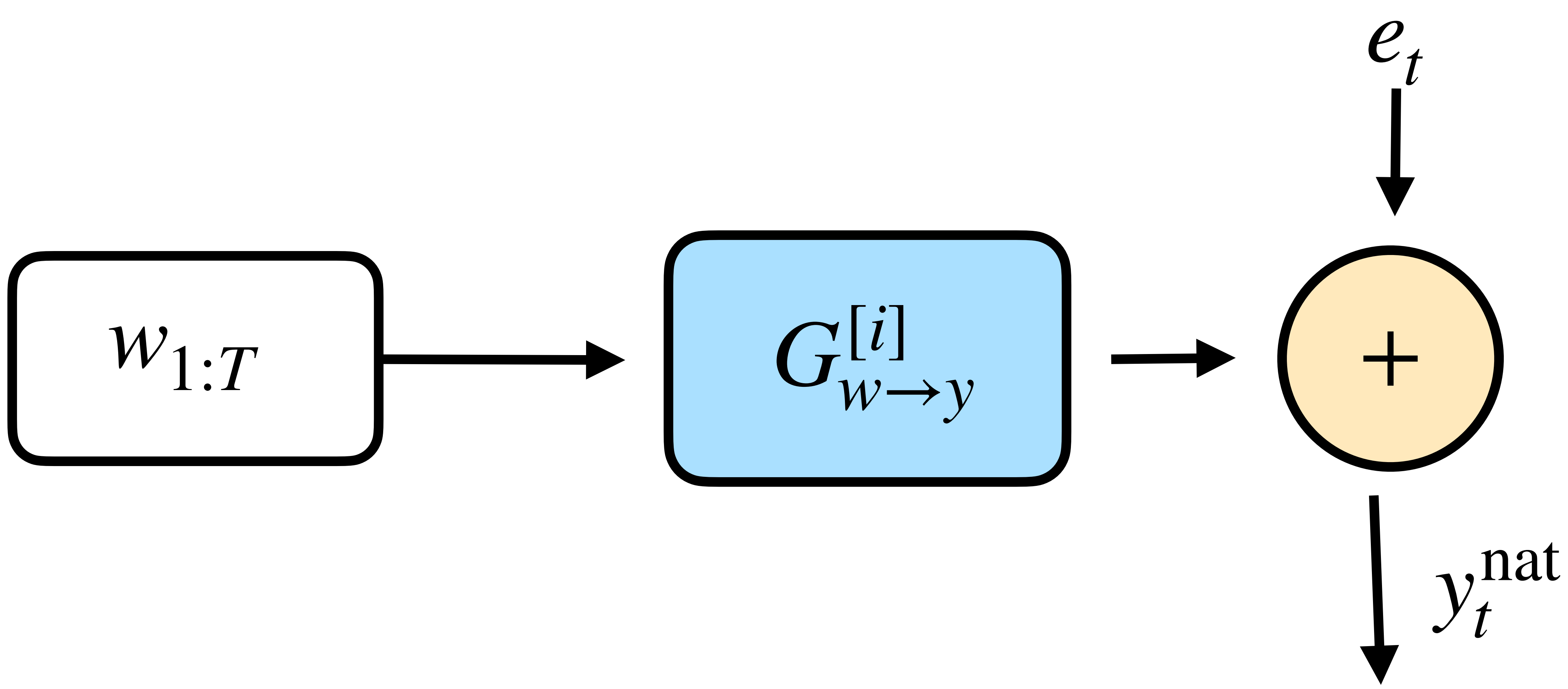
From Full Observation to Nature's Y's



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

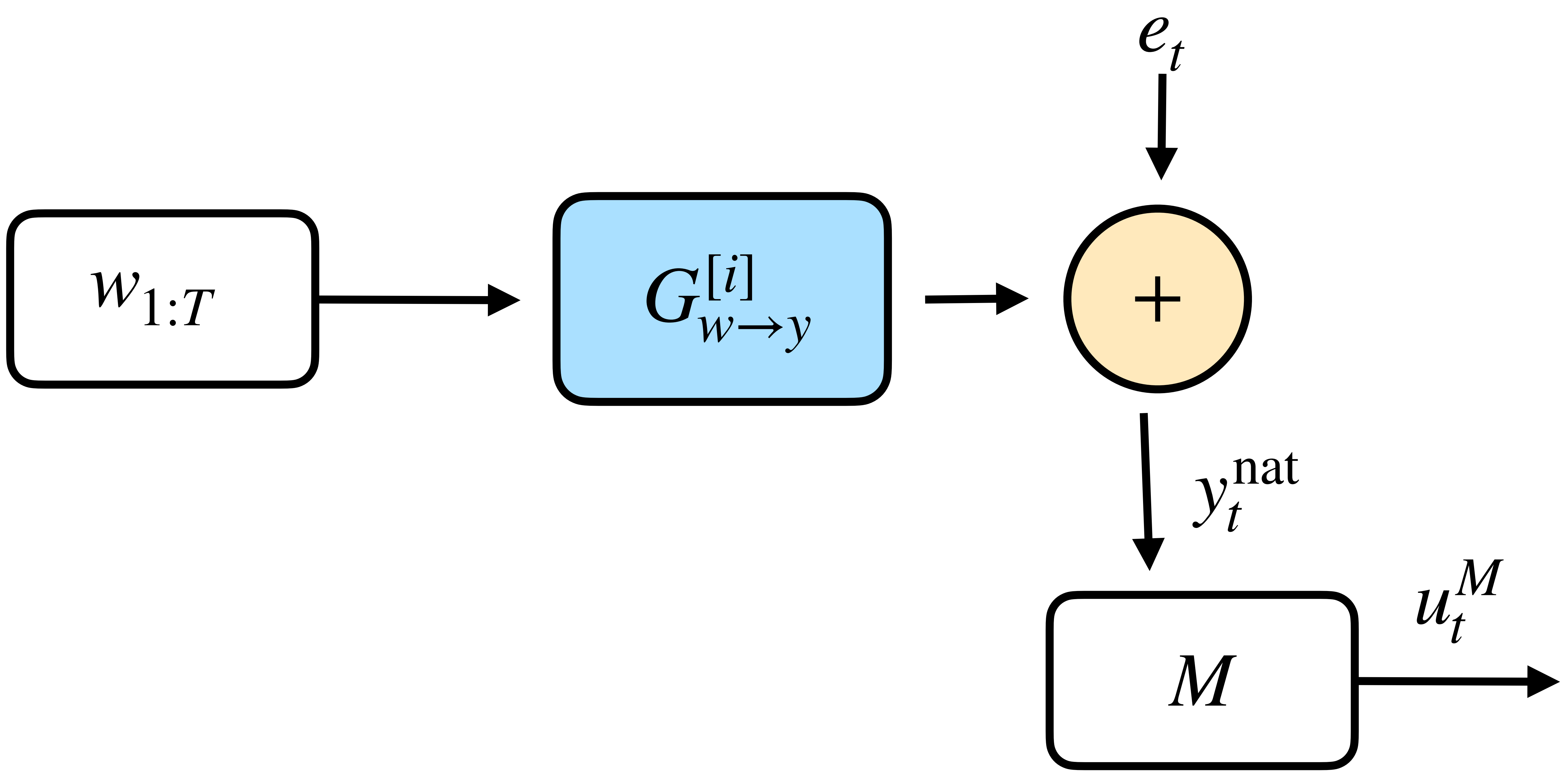
Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]} y_{t-i}^{\text{nat}}$



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

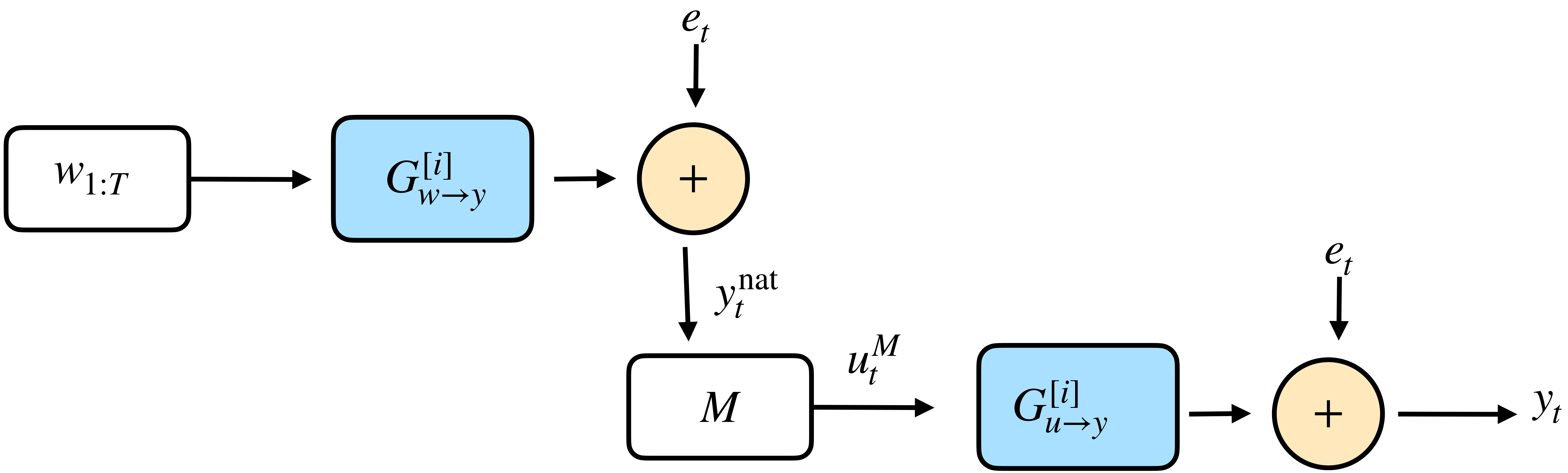
Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]} y_{t-i}^{\text{nat}}$



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

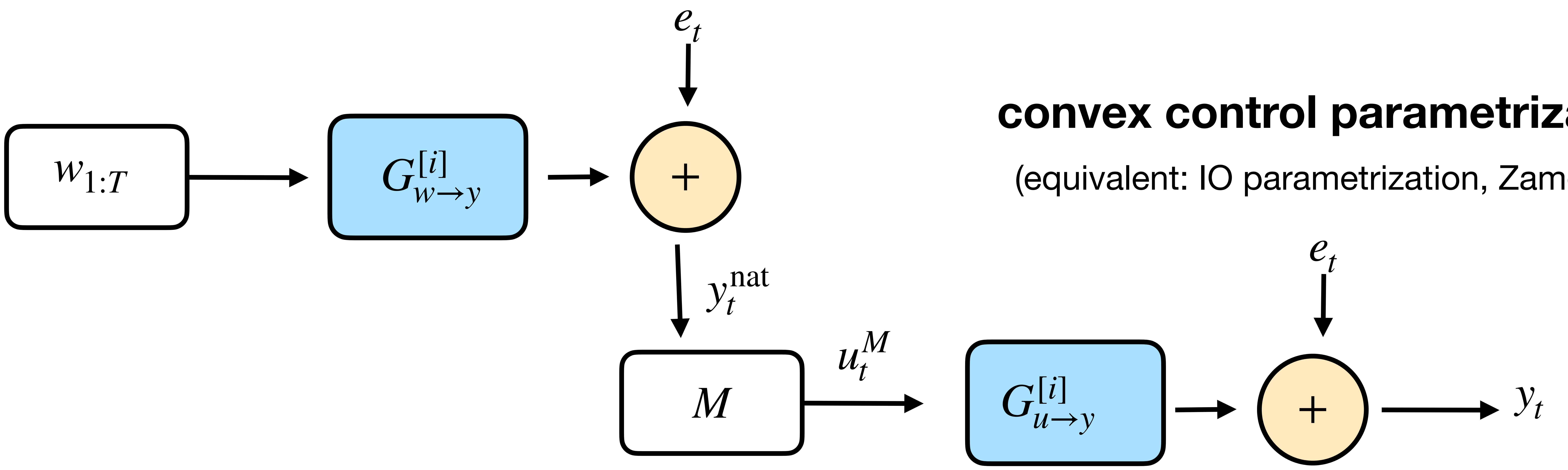
Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]} y_{t-i}^{\text{nat}}$



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]} y_{t-i}^{\text{nat}}$



convex control parametrization!

(equivalent: IO parametrization, Zames '81)

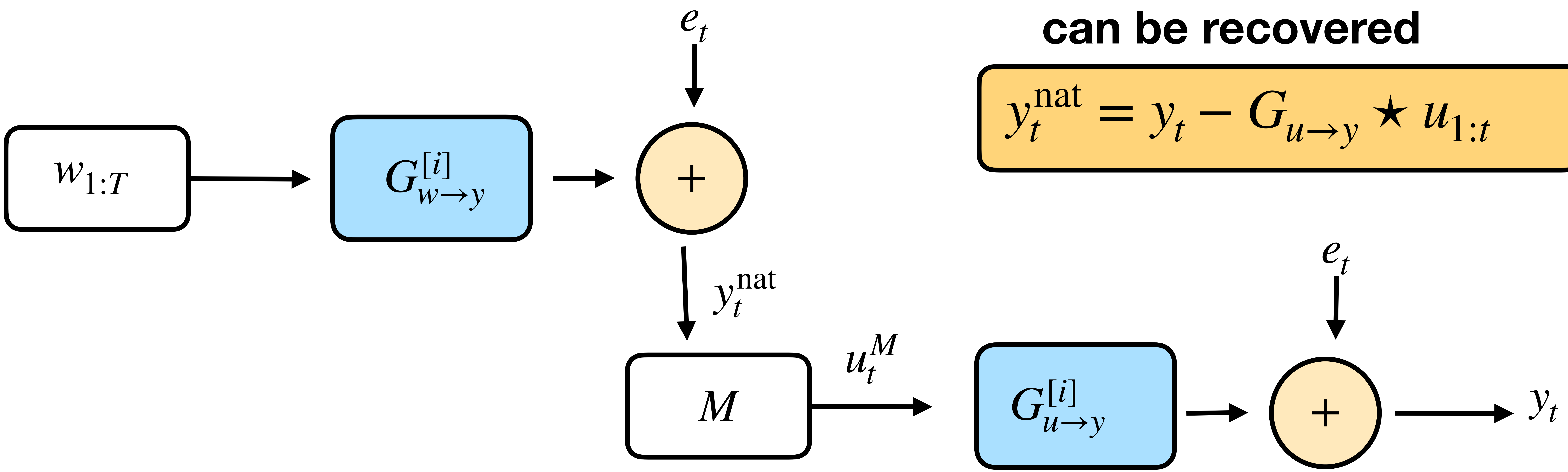
Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]} y_{t-i}^{\text{nat}}$

can be recovered

$$y_t^{\text{nat}} = y_t - G_{u\to y} \star u_{1:t}$$

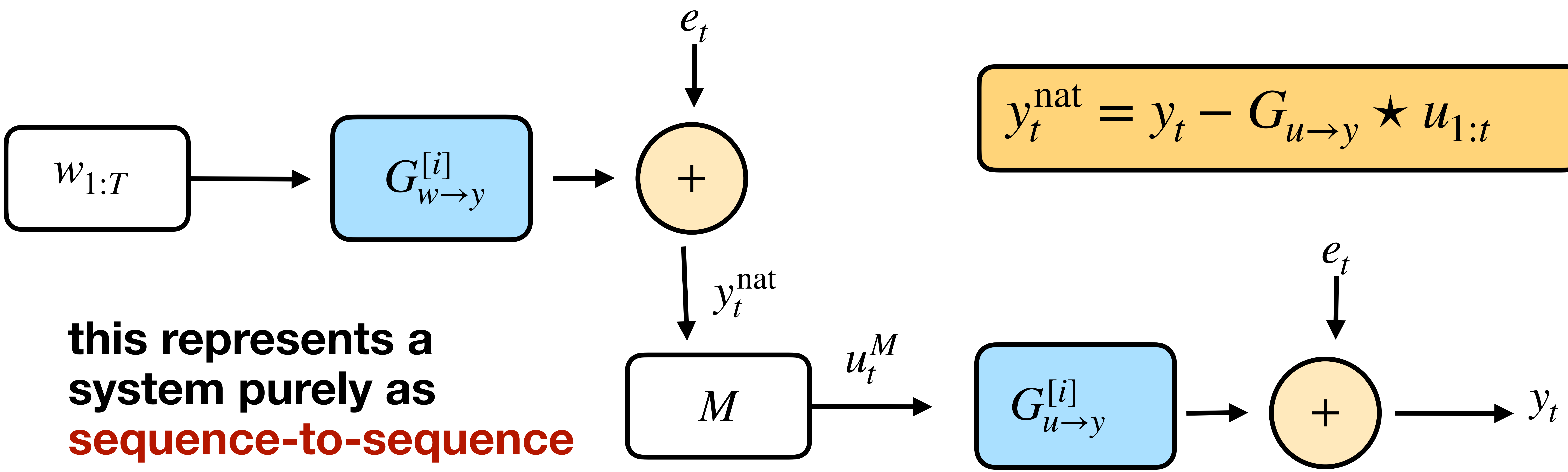


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]}\, y_{t-i}^{\text{nat}}$

$$y_t^{\text{nat}} = y_t - G_{u\to y} \star u_{1:t}$$



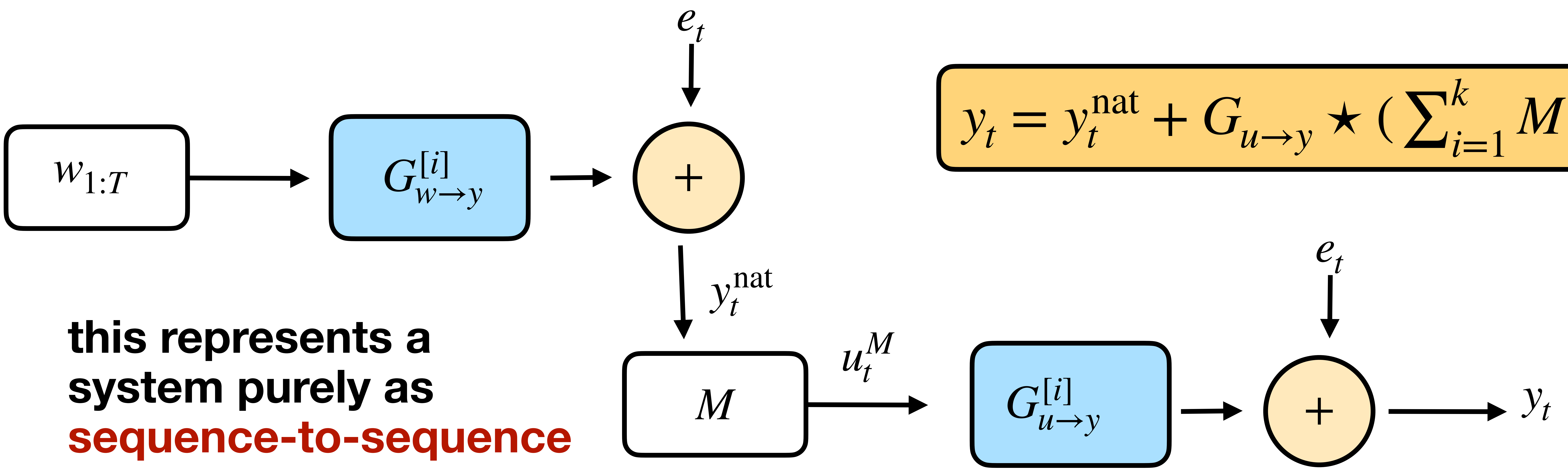
this represents a system purely as sequence-to-sequence

(e.g. Sutskever, Vinyals, Le)

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Definition: Disturbance Feedback Control (DFC) $u_t^M = \sum_{i=0}^{t} M^{[i]} y_{t-i}^{\text{nat}}$



$$y_t = y_t^{\text{nat}} + G_{u\to y} \star \left(\sum_{i=1}^{k} M^{[i]} y_i^{\text{nat}}\right)$$

this represents a system purely as sequence-to-sequence

(e.g. Sutskever, Vinyals, Le)

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C, ρ)**-stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \le C\rho^i$

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C,ρ)-**stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \le C\rho^i$

Theorem (Nature's Y's)**:** Any stabilizing, **dynamic** linear controller can be approximated by the **Disturbance Response Control** (DRC)

$$u^M_t = \sum_{i=0}^{t} M^{[i]} y^{\mathrm{nat}}_{t-i} \qquad \sum_i \|M^{[i]}\| \le O_\star(1)$$

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C, ρ)**-stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \le C\rho^i$

Theorem (Nature's Y's Regret)**:** Online gradient descent with the **Disturbance Response Control** (DRC)

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C, ρ)-**stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \le C\rho^i$

Theorem (Nature's Y's Regret)**:** Online gradient descent with the **Disturbance Response Control** (DRC)

$$u_t = \sum_{i=0}^{t} M_t^{[i]} y_{t-i}^{\text{nat}} \qquad M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t)$$

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C, ρ)**-stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \leq C\rho^i$

Theorem (Nature's Y's Regret)**:** Online gradient descent with the **Disturbance Response Control** (DRC)

$$u_t = \sum_{i=0}^{t} M_t^{[i]} y_{t-i}^{\text{nat}} \qquad M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t)$$

obtains $$\inf_M \mathrm{Reg}_T(\mathbb{A}; \Pi_{\text{drc}}) \leq \tilde{O}(\sqrt{T})$$

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C, ρ)**-stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \le C\rho^i$

Theorem (Nature's Y's Regret)**:** Online gradient descent with the **Disturbance Response Control** (DRC)

$$u_t = \sum_{i=0}^{t} M_t^{[i]} y_{t-i}^{\text{nat}} \qquad M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t)$$

Generalizes to known stabilizing controller (eg. LQG) via Youla-Kućera Par.

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Full Observation to Nature's Y's

Assume **Markov Operators*** are (C, ρ)**-stable:** $\max\{\|G^{[i]}_{w\to y}\|, \|G^{[i]}_{u\to y}\|\} \le C\rho^i$

Theorem (Nature's Y's Regret)**:** Online gradient descent with the **Disturbance Response Control** (DRC)

$$u_t = \sum_{i=0}^{t} M_t^{[i]} y_{t-i}^{\text{nat}} \qquad M_{t+1} = M_t - \eta_t \nabla \tilde{F}_t(M_t)$$

The entire algorithm can be defined using Markov operators (Improper)

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

Summary

Summary

1. We study **partial observability** ($y_t = Cx_t + e_t$)

Summary

1. We study **partial observability** $(y_t = Cx_t + e_t)$

2. We introduce and analyze the **Nature's Y's parameterization** (DFC)

Summary

1. We study **partial observability** ($y_t = Cx_t + e_t$)

2. We introduce and analyze the **Nature's Y's parameterization** (DFC)

3. We show that the same rate of regret is achievable with essentially **the same principles.**

Roadmap

3. **Unknown Dynamics**: System Identification

From Known to Unknown Dynamics

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Goal: Compete the linear controllers even if (A, B, C) **unknown**

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Goal: Compete the linear controllers even if (A, B, C) **unknown**

Challenge 1: DFC/GPC controller needed to know dynamics to recover w_t or y_t^{nat}

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Goal: Compete the linear controllers even if (A, B, C) **unknown**

Challenge 1: DFC/GPC controller needed to know dynamics to recover w_t or y_t^{nat}

Challenge 2: We need to dynamics to form to $\tilde{F}_t$ (simulated costs under M)

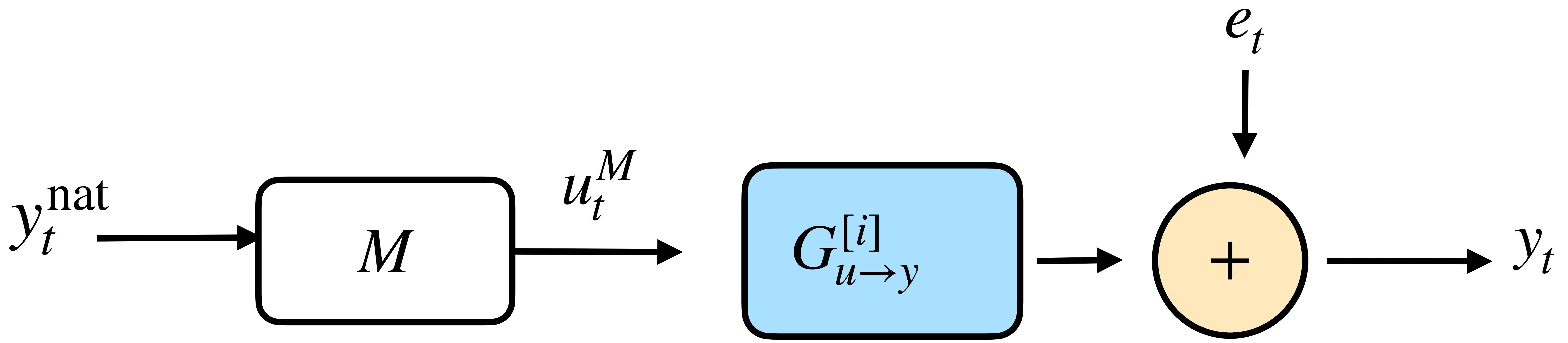
Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Goal: Compete the linear controllers even if (A, B, C) **unknown**

Challenge 1: DFC/GPC controller needed to know dynamics to recover w_t or y_t^{nat}

Challenge 2: We need to dynamics to form to $\tilde{F}_t$ (simulated costs under M)



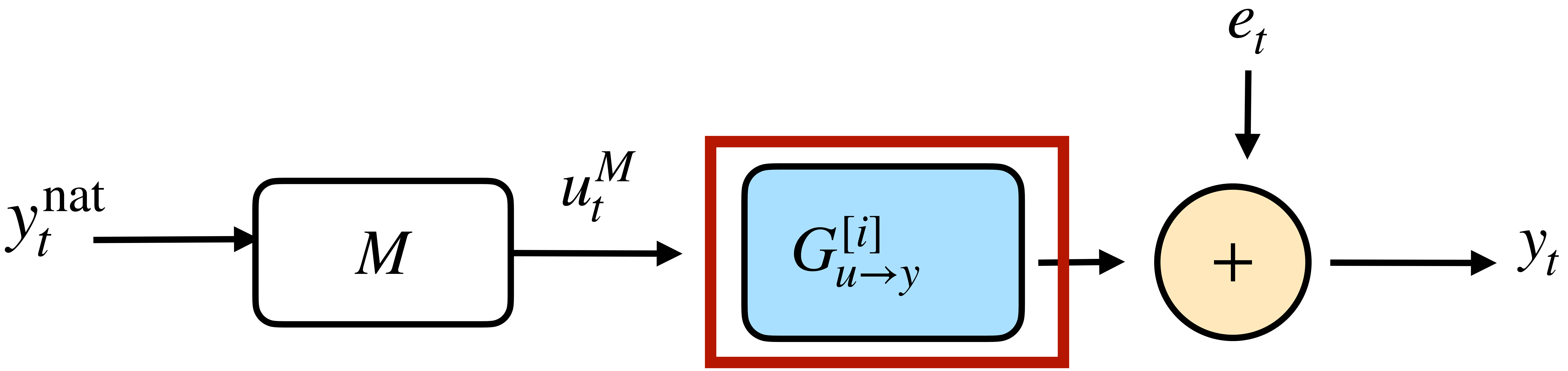
Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Goal: Compete the linear controllers even if (A, B, C) **unknown**

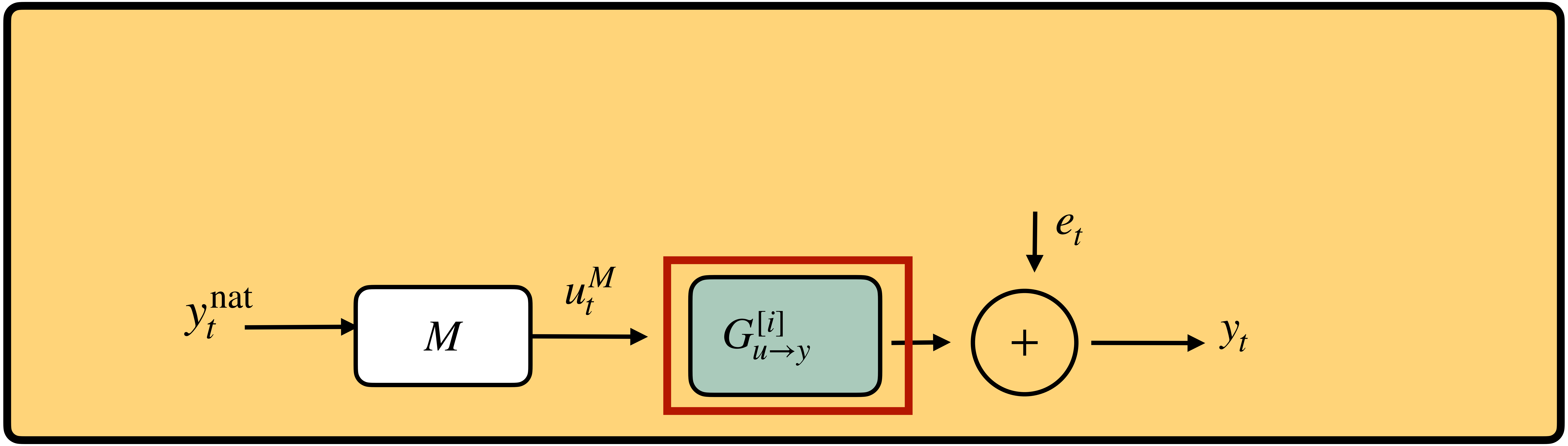
Challenge 1: DFC/GPC controller needed to know dynamics to recover w_t or y_t^{nat}

Challenge 2: We need to dynamics to form to $\tilde{F}_t$ (simulated costs under M)



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

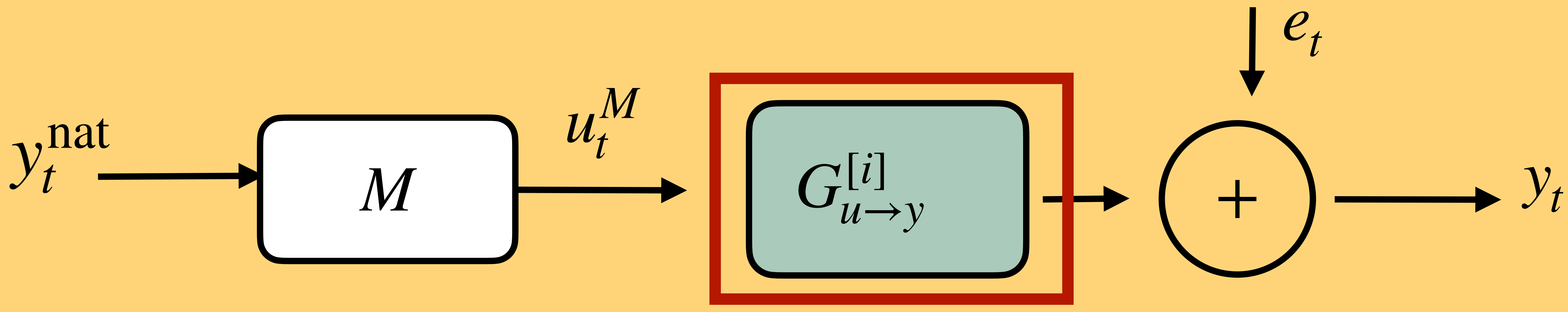
From Known to Unknown Dynamics



Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0,I)$ and estimate $G_{u \to y}$

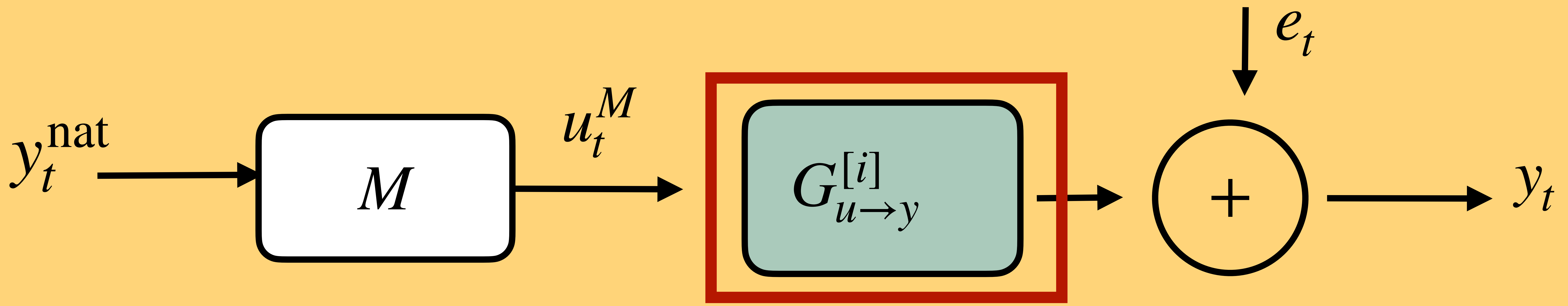


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0,I)$ and estimate $G_{u \to y}$

Step 2: Run **DFC+OGD** controller, replacing $G_{u \to y}$ least squares estimate $\hat{G}_{u \to y}$

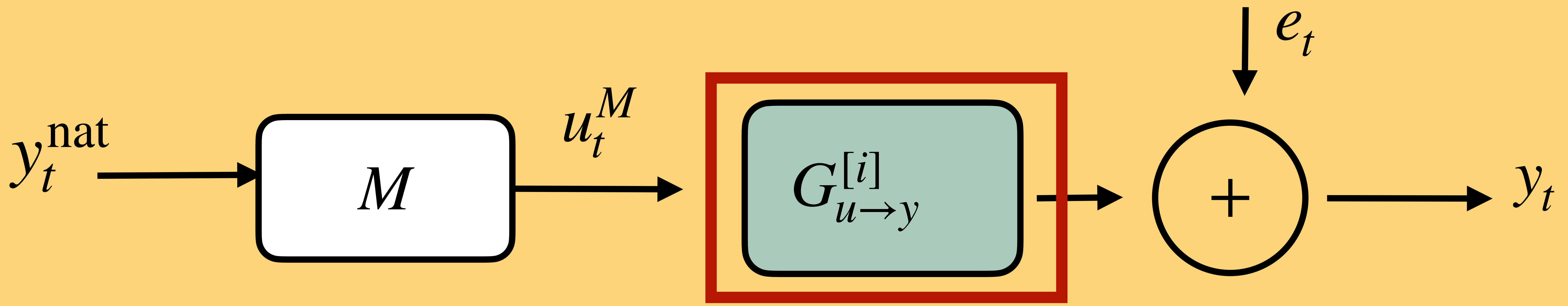


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0,I)$ and estimate $G_{u\to y}$

Step 2: Run **DFC+OGD** controller, replacing $G_{u\to y}$ least squares estimate $\hat{G}_{u\to y}$



Proposition: $\mathrm{Reg}_T \leq \tilde{O}(1)\left(\sqrt{T} + T\|\hat{G}_{\mathrm{ls}} - G\| + T_0\right)$

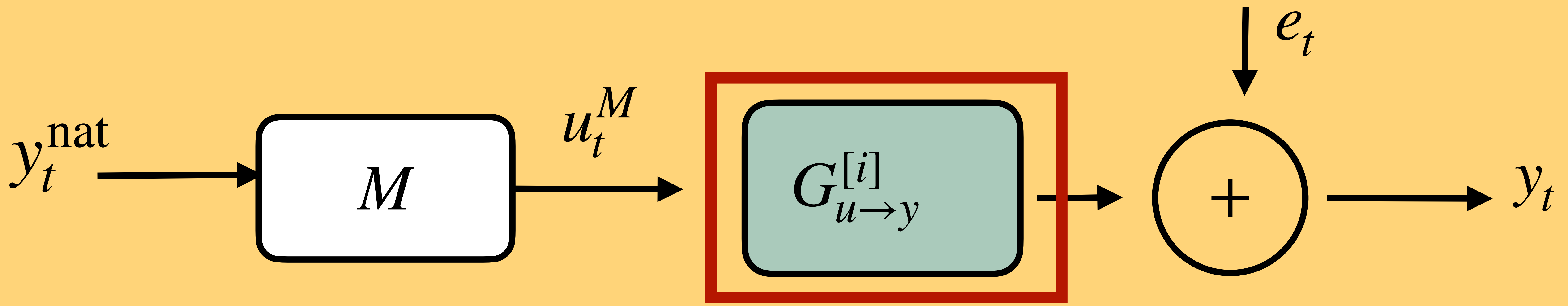
known regret **cost for error** **cost for estimation**

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0,I)$ and estimate $G_{u\to y}$

Step 2: Run **DFC+OGD** controller, replacing $G_{u\to y}$ least squares estimate $\hat{G}_{u\to y}$

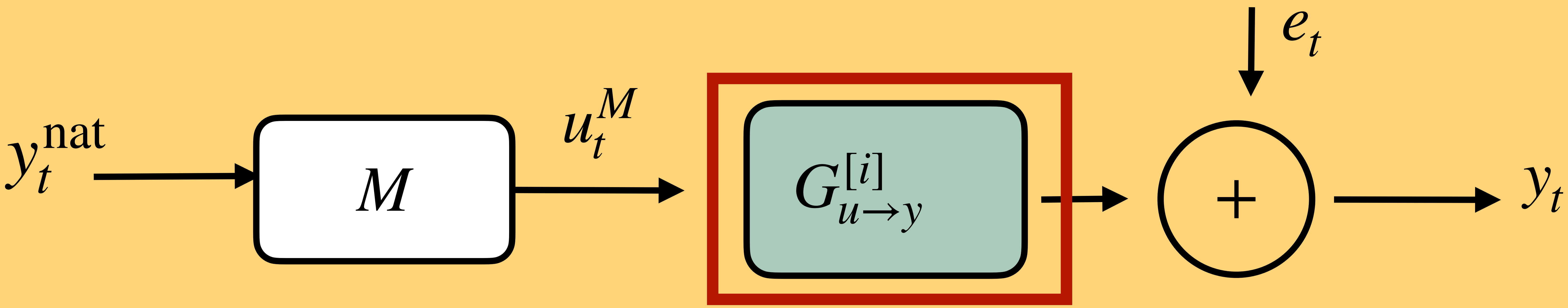


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0,I)$ and estimate $G_{u\to y}$

Step 2: Run **DFC+OGD** controller, replacing $G_{u\to y}$ least squares estimate $\hat{G}_{u\to y}$



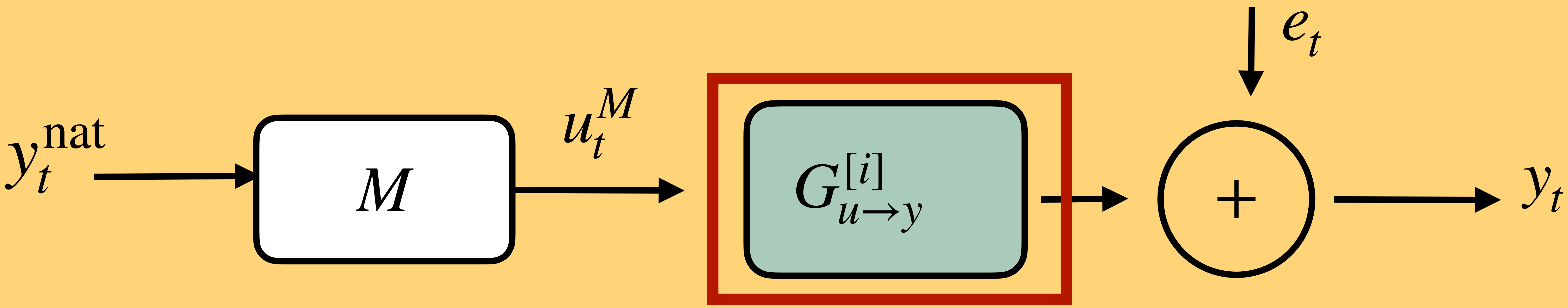
Theorem: $\mathrm{Reg}_T \le \tilde{O}(T^{2/3})$ **where** $T_0 = T^{2/3}$

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0,I)$ and estimate $G_{u\to y}$

Step 2: Run **DFC+OGD** controller, replacing $G_{u\to y}$ least squares estimate $\hat{G}_{u\to y}$

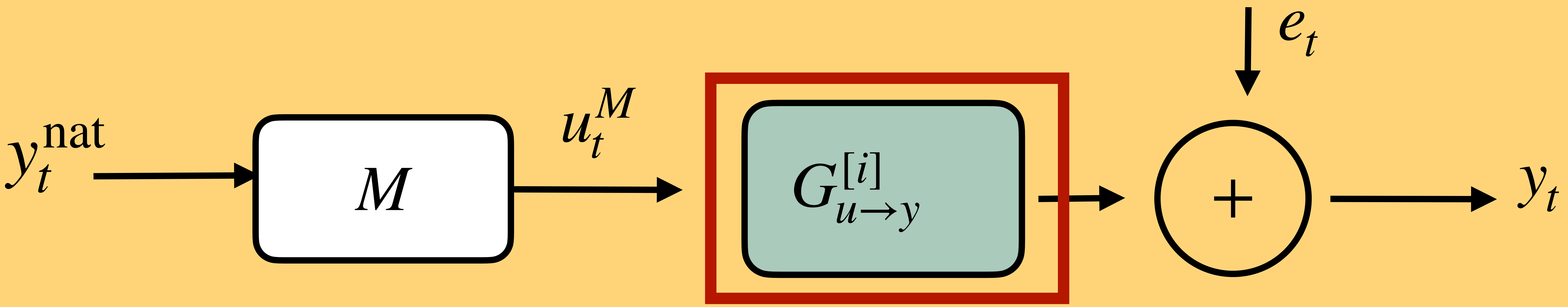


Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

From Known to Unknown Dynamics

Step 1: For first T_0 steps, use $u_t \sim \mathrm{N}(0, I)$ and estimate $G_{u \to y}$

Step 2: Run **DFC+OGD** controller, replacing $G_{u \to y}$ least squares estimate $\hat{G}_{u \to y}$



Conveniently: We only ever use and estimate the **Markov operator.**

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

Summary

Summary

1. We study **unknown dynamics**

Summary

1. We study **unknown dynamics**

2. We combine OCO with **estimating the Markov operator**

Summary

1. We study **unknown dynamics**

2. We combine OCO with **estimating the Markov operator**

3. Everything works just by working with **sequence-to-sequence** , i.e. **improper,** parameterization

Roadmap

4. Optimal Regret: Leveraging Curvature

Fast & Optimal Regret Rates

Fast & Optimal Regret Rates

Goal: How slow can we make Reg_T as a function of T?

Fast & Optimal Regret Rates

Goal: How slow can we make Reg_T as a function of T?

Also called a **fast rate** because we want $\mathrm{Reg}_T/T \to 0$ as fast as possible

Fast & Optimal Regret Rates

Goal: How slow can we make Reg_T as a function of T?

Also called a **fast rate** because we want $\mathrm{Reg}_T/T \to 0$ as fast as possible

Assume: $c_t(x, u)$ is α-**strongly convex:** $c_t(x, u) - \alpha(\|x\|^2 + \|u\|^2)/2$ **convex**

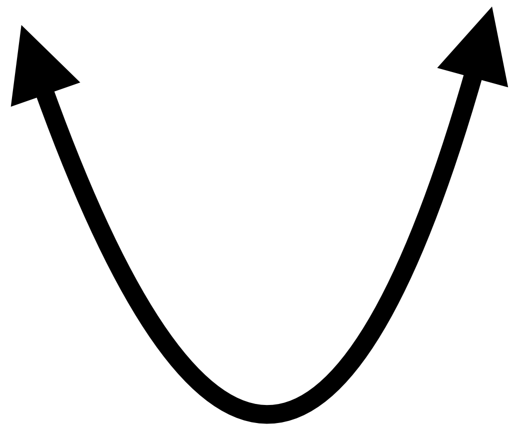
Fast & Optimal Regret Rates

Goal: How slow can we make Reg_T as a function of T?

Also called a **fast rate** because we want $\mathrm{Reg}_T/T \to 0$ as fast as possible

Assume: $c_t(x, u)$ is α-**strongly convex:** $c_t(x, u) - \alpha(\|x\|^2 + \|u\|^2)/2$ **convex**

aka **curvature:** if c_t is smooth: $\lambda_{\min}(\nabla^2 c) \geq \alpha$



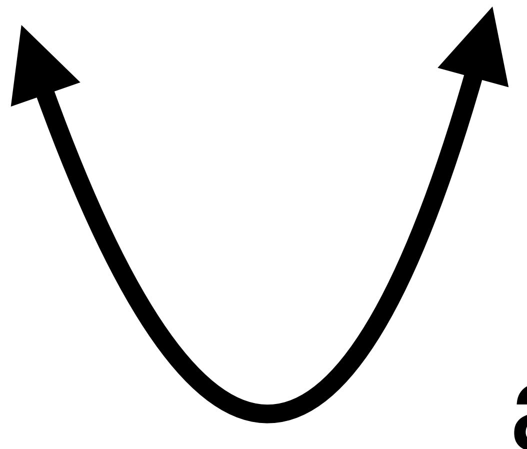
Fast & Optimal Regret Rates

Goal: How slow can we make Reg_T as a function of T?

Also called a **fast rate** because we want $\mathrm{Reg}_T / T \to 0$ as fast as possible

Assume: $c_t(x, u)$ is α**-strongly convex:** $c_t(x, u) - \alpha(\|x\|^2 + \|u\|^2)/2$ **convex**

aka **curvature:** if c_t is smooth: $\lambda_{\min}(\nabla^2 c) \geq \alpha$



accelerate learning
+ optimization

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α-**strongly convex,** there exists algorithms such that

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α**-strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α-**strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α**-strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Compare to $\sqrt{T}$ **and** $T^{2/3}$ **regret, previously**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α-**strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α**-strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**

2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Up to log factors, optimal even in online LQR (unknown A, B)

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α-**strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Up to log factors, optimal even in online LQR (unknown A, B)

fixed quadratic cost, i.i.d. Gaussian noise, full observation $y \equiv x_t$

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α-**strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α**-strongly convex,** there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Takeaway: For s.c. costs, **unknown dynamics determines regret**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Fast & Optimal Regret Rates

Theorem: If $c_t(x, u)$ is α-strongly convex, there exists algorithms such that

1. $\mathrm{Reg}_T \leq \mathrm{poly}(\log T)/\alpha$ **known dynamics**
2. $\mathrm{Reg}_T \leq \tilde{O}(\sqrt{T}/\alpha)$ **unknown dynamics**

Takeaway: For s.c. costs, **unknown dynamics determines regret**

changing costs and adversarial noise only affect rates **logarithmically.**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

Agrawal, Hazan, Singh "Logarithmic Regret for Online Control", 2019

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$

Agrawal, Hazan, Singh "Logarithmic Regret for Online Control", 2019

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$

Theorem: If noise is stochastic/persistent excitation, $\eta_t \leftarrow O(1/\alpha)$ attains fast rate

Agrawal, Hazan, Singh "Logarithmic Regret for Online Control", 2019

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \eta_t \nabla \tilde{F}_t(M_t)}$ **Proof:** $\tilde{F}_t$ **is strongly convex in expectation**

Theorem: If noise is stochastic/persistent excitation, $\eta_t \leftarrow O(1/\alpha)$ attains fast rate

Agrawal, Hazan, Singh "Logarithmic Regret for Online Control", 2019

Simchowitz, Singh, Hazan "Improper Learning for Nonstochastic Control", 2020

*

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

Theorem: For general noise, the **OnlineNewtonStep** algorithm (Hazan '07) attains fast rates.

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

*

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \mathrm{OnlineNewton}(M_t)}$

Theorem: For general noise, the **OnlineNewtonStep** algorithm (Hazan '07) attains fast rates.

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

Algorithm: Fast Rates

*

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \mathrm{OnlineNewton}(M_t)}$ **Proof:** $\tilde{F}_t$ **is exp-concave**

Theorem: For general noise, the **OnlineNewtonStep** algorithm (Hazan '07) attains fast rates.

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

*

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \mathrm{OnlineNewton}(M_t)}$ **Proof:** $\tilde{F}_t$ **is exp-concave**

Theorem: For general noise, the **OnlineNewtonStep** algorithm (Hazan '07) attains fast rates.

Intuition: Newton solves **ill-conditioned** quadratic functions

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

*

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \mathrm{OnlineNewton}(M_t)}$ **Proof:** $\tilde{F}_t$ **is exp-concave**

Theorem: For general noise, the **OnlineNewtonStep** algorithm (Hazan '07) attains fast rates.

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

*

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \mathrm{OnlineNewton}(M_t)}$ **Proof:** $\tilde{F}_t$ **is exp-concave**

Theorem: For general noise, the **OnlineNewtonStep** algorithm (Hazan '07) attains fast rates.

Fast rates for unknown dynamics relies on carefully **sensitivity to error argument** + **overparametrization.**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

*

Algorithm: Fast Rates

Optional: Estimate dynamics for first T_0 steps.

For $t = T_0, T_0 + 1, \ldots$

1. $\boxed{u_t \leftarrow u_t^{M_t}}$ **defined in terms of** $M = (M^{[0]}, \ldots, M^{[k]})$

2. $\boxed{M_t \leftarrow M_t - \mathrm{OnlineNewton}(M_t)}$ **Proof:** $\tilde{F}_t$ **is exp-concave**

Takeaway: Only thing that changes is the **optimizer + assumptions**

Simchowitz "Making Nonstochastic Control as Easy as Stochastic", 2020

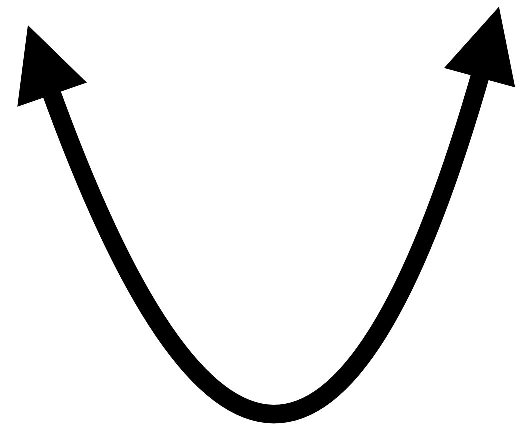
Summary

Summary

1 **Fast Rates** refer to making Reg_T grow as slow as possible.

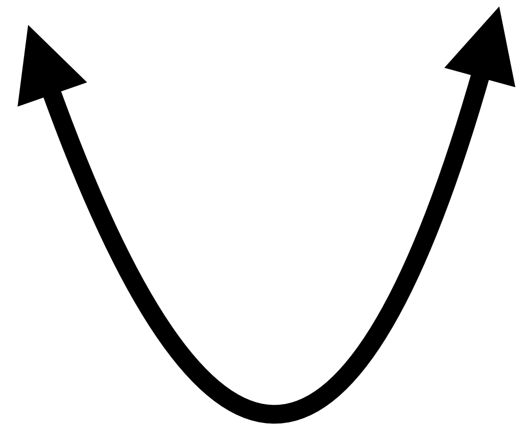
Summary

1 **Fast Rates** refer to making Reg_T grow as slow as possible.



2. With **curvature,** fast rates can be obtained only by **modification of the optimizer.**

Summary



1 **Fast Rates** refer to making Reg_T grow as slow as possible.

2. With **curvature,** fast rates can be obtained only by **modification of the optimizer.**

3. With **curvature,** the regret is determined only by **knowledge of dynamics**, and only logarithmically affected by changing costs + adversarial noise

Hardness Results and Open Questions

Roadmap

5. Open Problems / Hardness Results

The need for stabilization

- **Throughout,** we assumed a **known, stabilizing controller.**

Theorem (Chen & Hazan, ’20)**:** Without a known stabilizing controller, regret is $\Omega(\exp(\text{dimension}))$, until one stabilizes system

Open Question: What are stronger assumptions under one can stabilize the dynamics via online methods?

Beyond linear dynamics

- **Throughout,** we assumed a **fixed, linear dynamics**

Theorem (Gradu, Minyasan, Hazan, ’20)**:** If dynamics A_t, B_t, C_t change **independently** of the learner, then can obtain low **adaptive regret**

Open Question: What if dynamics change **in response to learner?**

Beyond linear dynamics

- **Throughout,** we assumed a **fixed, linear dynamics**

Theorem (Minyasan, Gradu, Simchowitz, Hazan, ’21)**:** If dynamics A_t, B_t, C_t change **independently** of the learner, then can obtain low **adaptive regret**

Open Question: How to learn for truly **nonlinear dynamics?**

Towards practical deployment

- **Thus far,** we have given mostly theoretical results

Theorems: Many of them, illustrating powerful principles in control + AI **(improperness, online learning, adaptation).**

Open Question: Using online control for the **last mile** performance.

Summary

Core Concepts:

Core Concepts:

1. From optimal/robust control to regret

Core Concepts:

1. From optimal/robust control to regret

2. From "proper controller" to convex relaxation

Core Concepts:

1. From optimal/robust control to regret

2. From “proper controller” to convex relaxation

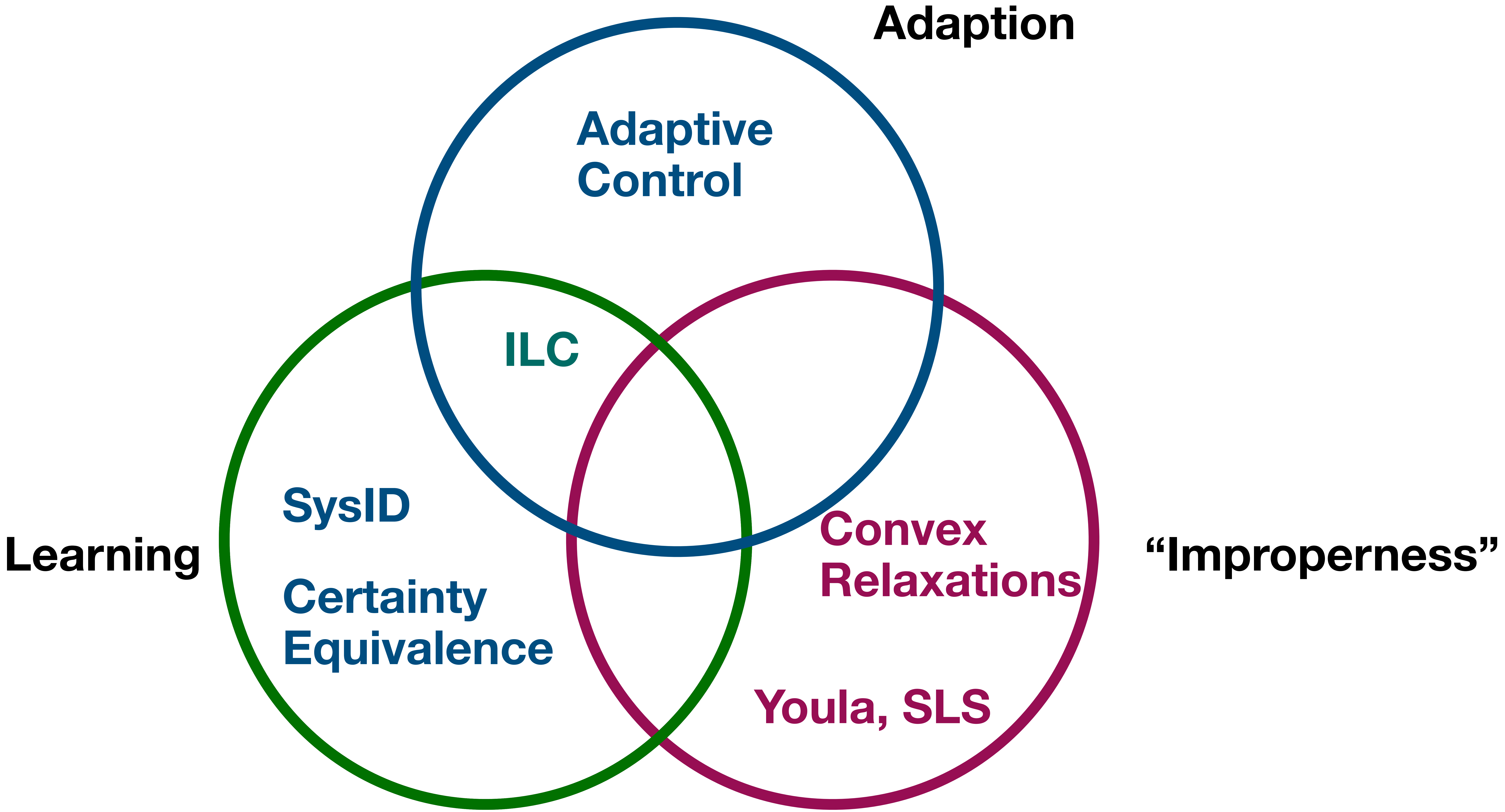
3. Combine statistical learning with online optimization

Core Concepts:

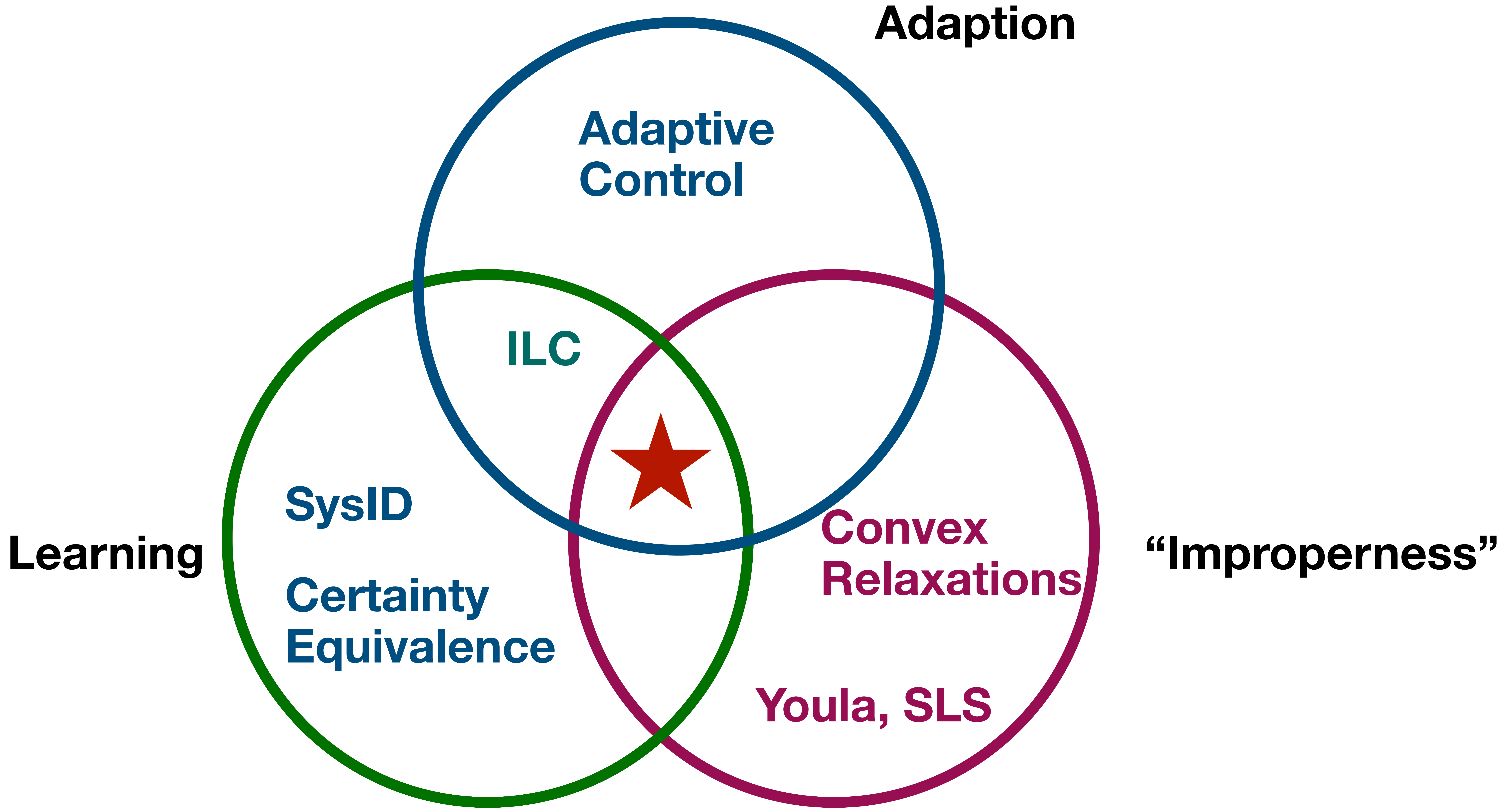
1. From optimal/robust control to regret
2. From “proper controller” to convex relaxation
3. Combine statistical learning with online optimization

Many open questions!

Non-stochastic control at the intersection



Non-stochastic control at the intersection



References

Agrawal, Bullins, Hazan, Kakade, Singh “Online Control with Adversarial Disturbances”, 2019

Agrawal, Hazan, Singh “Logarithmic Regret for Online Control”, 2019

Hazan, Kakade, Singh, “The Nonstochastic Control Policy”

Simchowitz, Singh, Hazan “Improper Learning for Nonstochastic Control”, 2020

Simchowitz “Making Nonstochastic Control as Easy as Stochastic”, 2020

Gradu, Minyasan, Hazan “Adaptive Regret for Control of Time-Varying Dynamics”, 2020

Chen, Hazan “Blackbox Control for Linear Dynamical Systems”, 2021