Learning and Control

6.3100/2 – Week 14, Lecture B

Dec 6, 2023

Lecturer: Prof. Priya Donti

Objectives

- Summarize reinforcement learning (RL)
- Identify similarities & differences between RL and state-space control
- Assess strengths & limitations of both approaches
- Recognize the potential to combine RL and state-space approaches

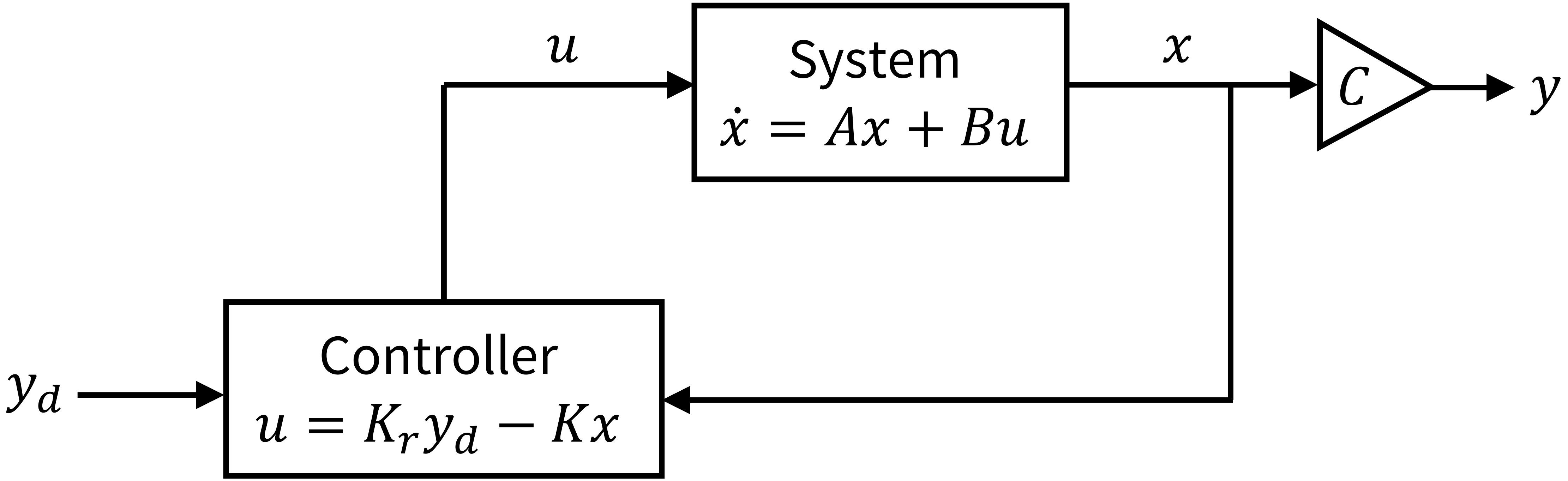
Outline

- Review of state-space control
- Overview & discussion of reinforcement learning (RL)
- Highlight of research bridging RL and state-space control

Outline

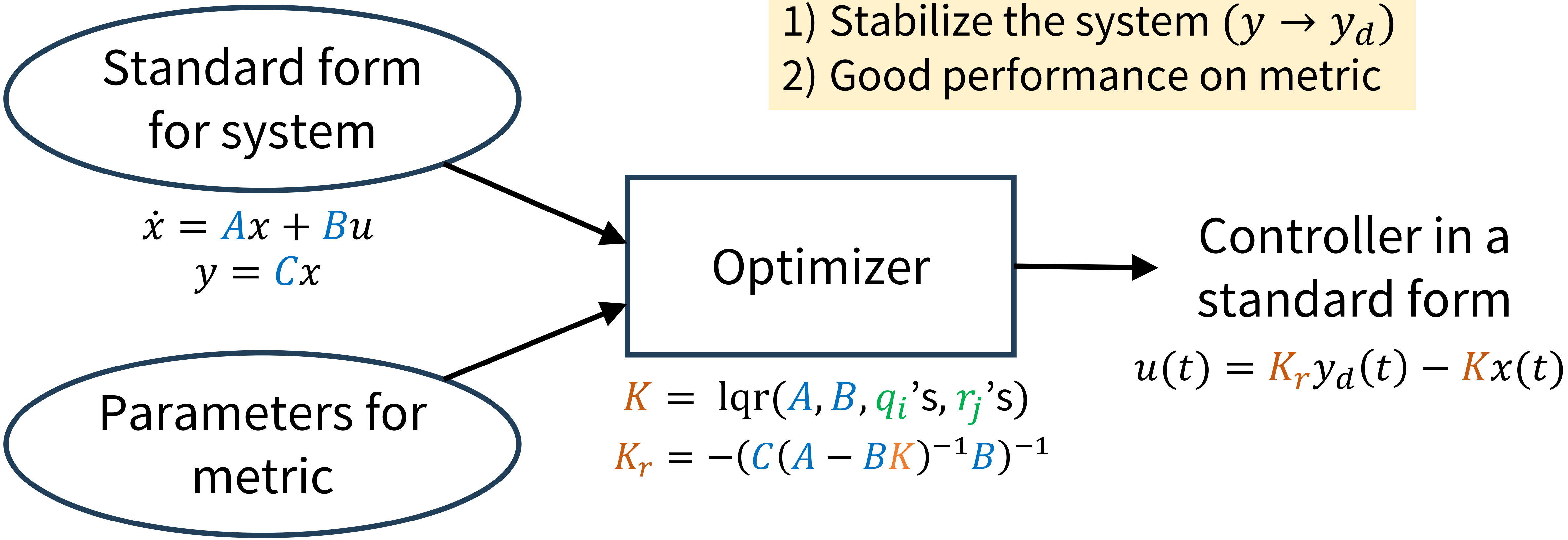
- **Review of state-space control**
- Overview & discussion of reinforcement learning (RL)
- Highlight of research bridging RL and state-space control

State-space control: Block diagram



State-space control: Design approach

Goals:
1) Stabilize the system ($y \to y_d$)
2) Good performance on metric



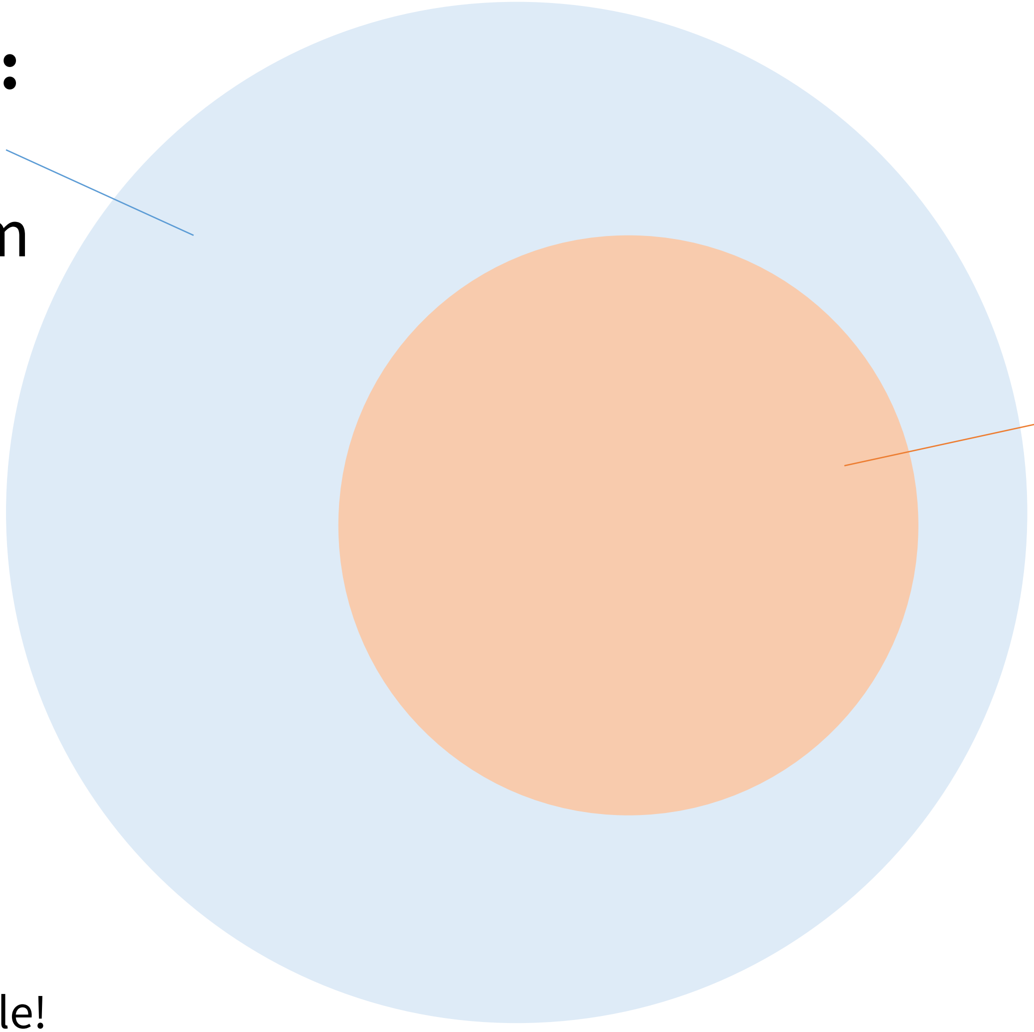
LQR: $\int_0^{\infty} \left(\sum_{i=1}^{\#\text{states}} q_i x_i(t) + \sum_{j=1}^{\#\text{inputs}} r_j u_j(t) \right) dt$

Outline

- Review of state-space control
- **Overview & discussion of reinforcement learning (RL)**
- Highlight of research bridging RL and state-space control

What is reinforcement learning?

Machine learning (ML): Techniques that automatically learn from examples (e.g., data)

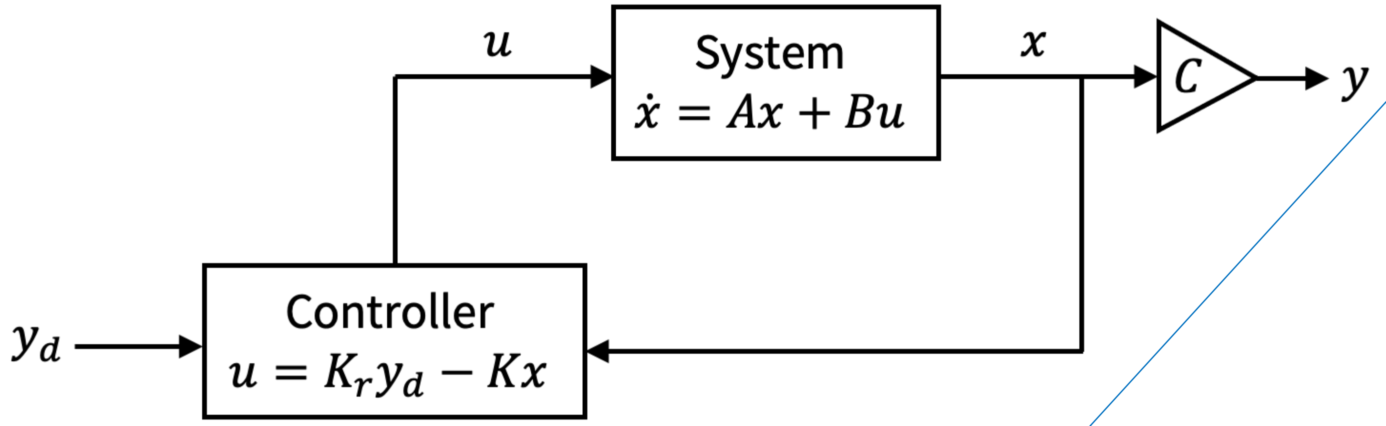


Reinforcement learning (RL): Branch of ML focused on learning control policies (via trial-and-error)

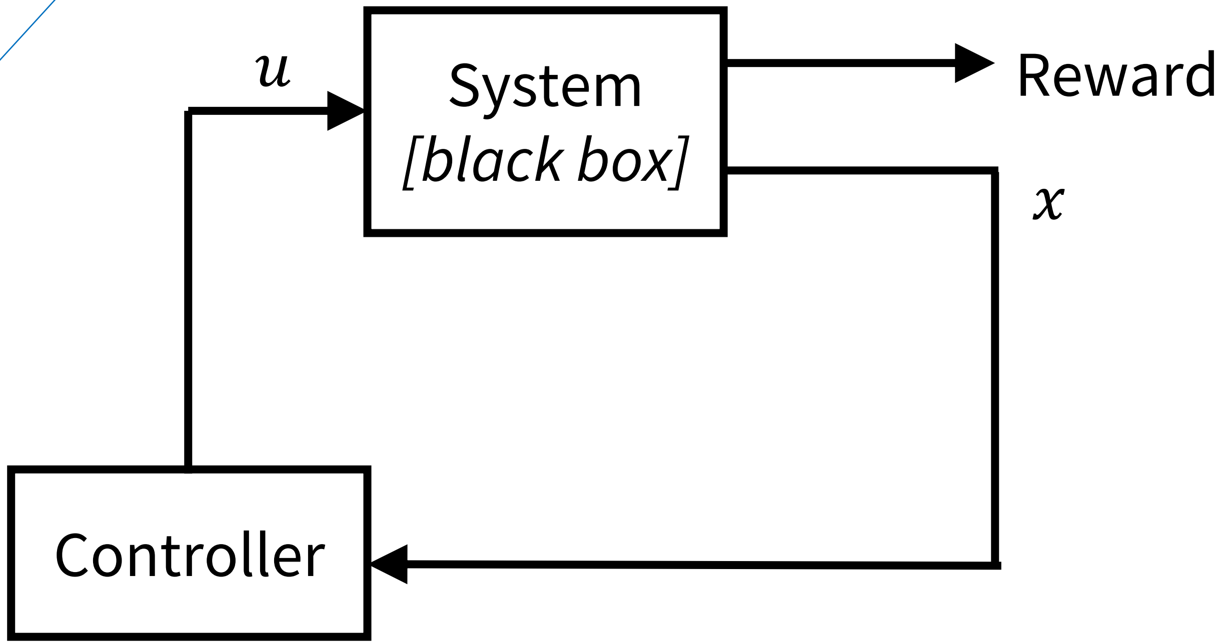
Note: Circles are not drawn to scale!

Block diagrams

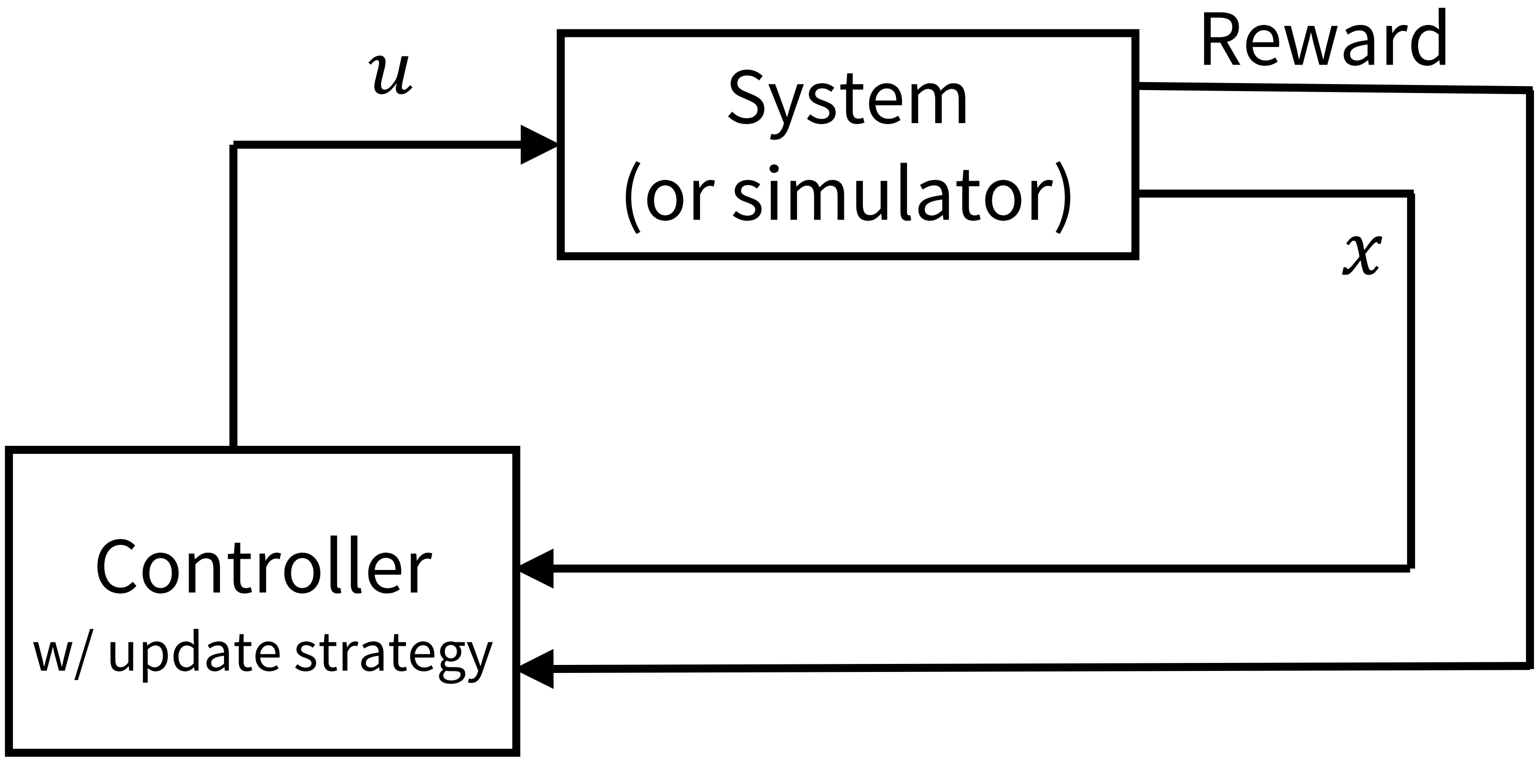
State-space control



Reinforcement learning



RL: Design approach (“trial-and-error”)



(note: must initialize with some value at t = 0)

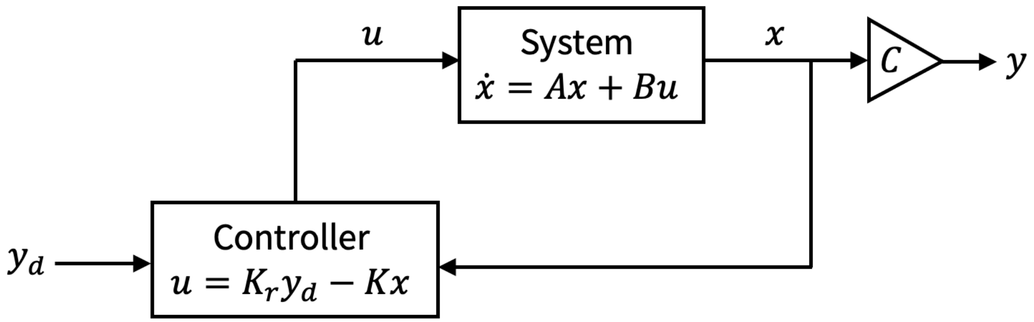
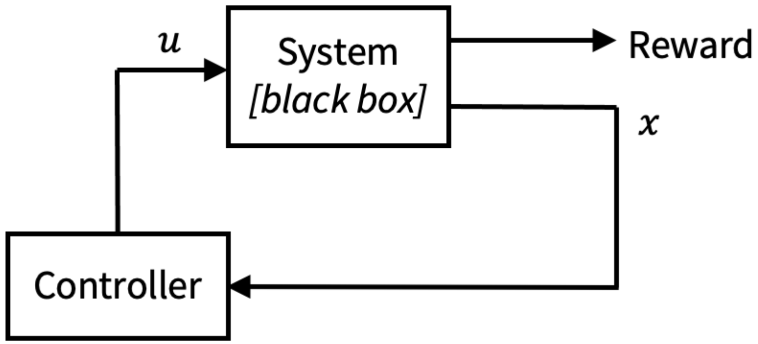
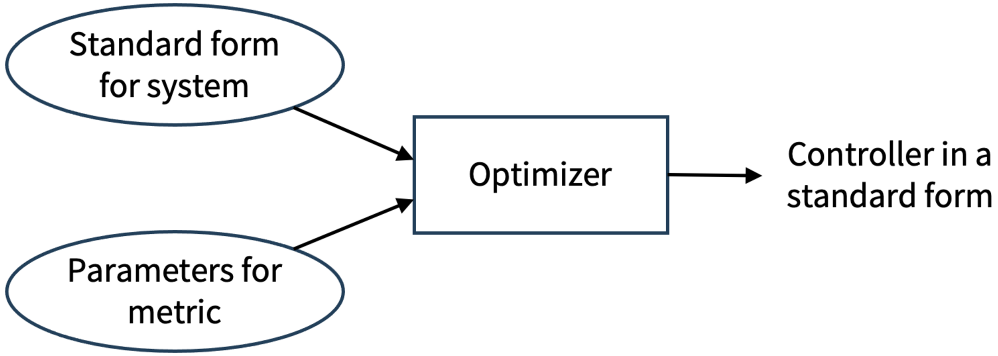
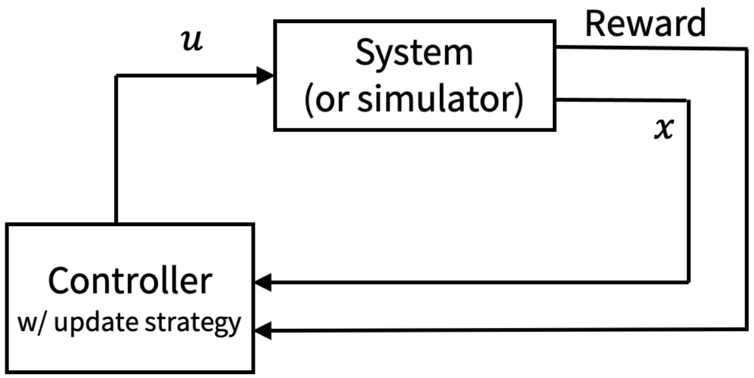
Goal: Good performance on reward

When is learning “done”?

- When some stopping criterion is met (e.g., convergence)
- Never (“lifelong learning”)

Note: In practice, controller is updated based on a *sequence* of states and actions over some time horizon (“*episodes*”), to maximize a *discounted cumulative reward* that weights present states more heavily than future states.

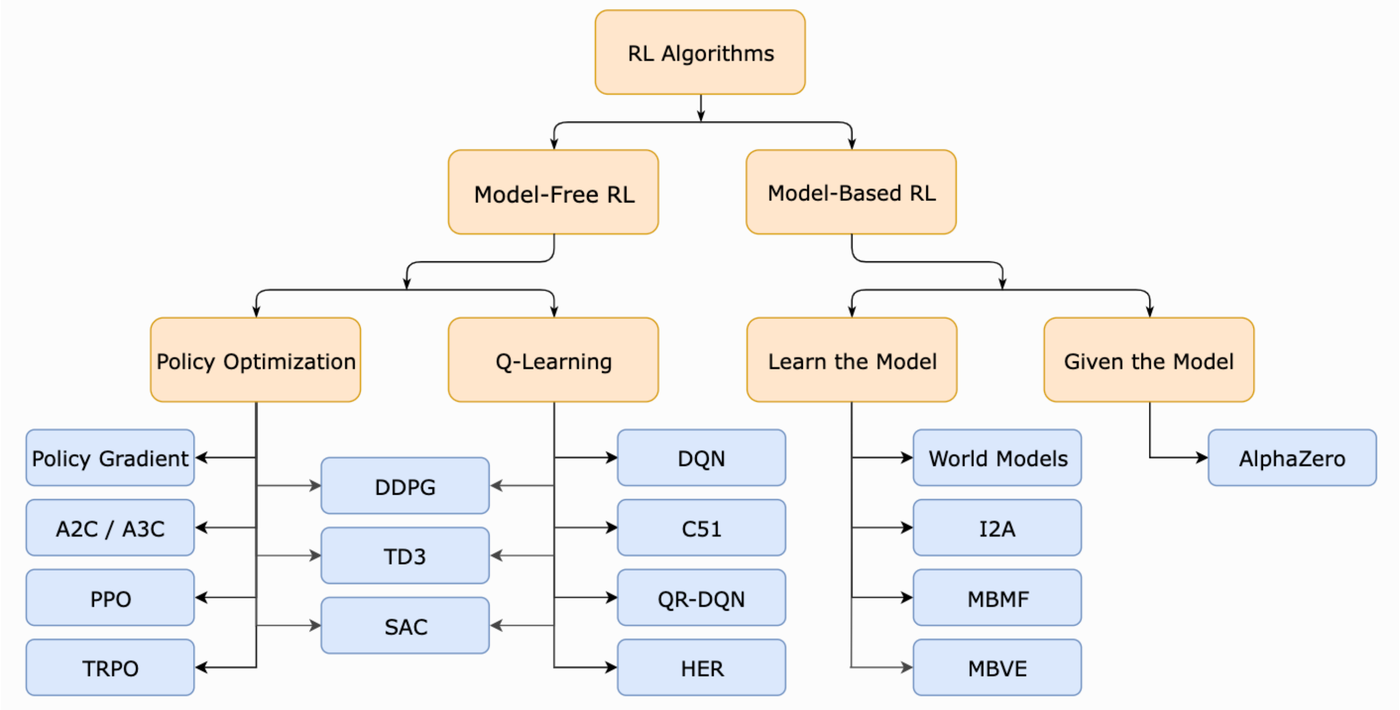
State-space control vs. RL

	State-space control	RL
Block diagram		
Design approach		
Goal	1) Stabilize the system ($y \rightarrow y_d$) 2) Good performance on metric	Good performance on reward

State-space control vs. RL

	State-space control	RL
Goals	Stability and performance	Performance (may include stability)
System	Known (function & parameters)	Unknown (but interactable)
Controller	Simple function w/ design params	Usually more complicated *[next slide]*
Perform. metric	Cost Needs to be “very nice” Hand-designed Present & future matter equally	Reward (negative cost) Does not need to be “as nice” From system (and/or hand-designed) Future matters less than present (use of “discount factor”)
Design technique	Solve an optimization problem	Learn through interaction with system

RL: What does the controller look like?



A non-exhaustive, but useful taxonomy of algorithms in modern RL
From: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Policy optimization: Function mapping from states to actions

- Learn: Parameters of function

Q-learning: Action-chooser (often greedy) based on approx. of conditional expected reward

- Learn: Approx. of conditional expected reward

Model-based methods: Some type of planning algorithm

- Learn: Approx. system model

State-space control vs. RL: Pros and cons

State-space control

- Reduced average-case performance due to simplifying assumptions

+ Can come with provable guarantees
+ Interpretable controllers

± Assumes knowledge of system (may be useful, may be incorrect)

Reinforcement learning

+ Better average-case performance due to expressive control policies & data

- Can fail catastrophically
- Usually “black box” controllers

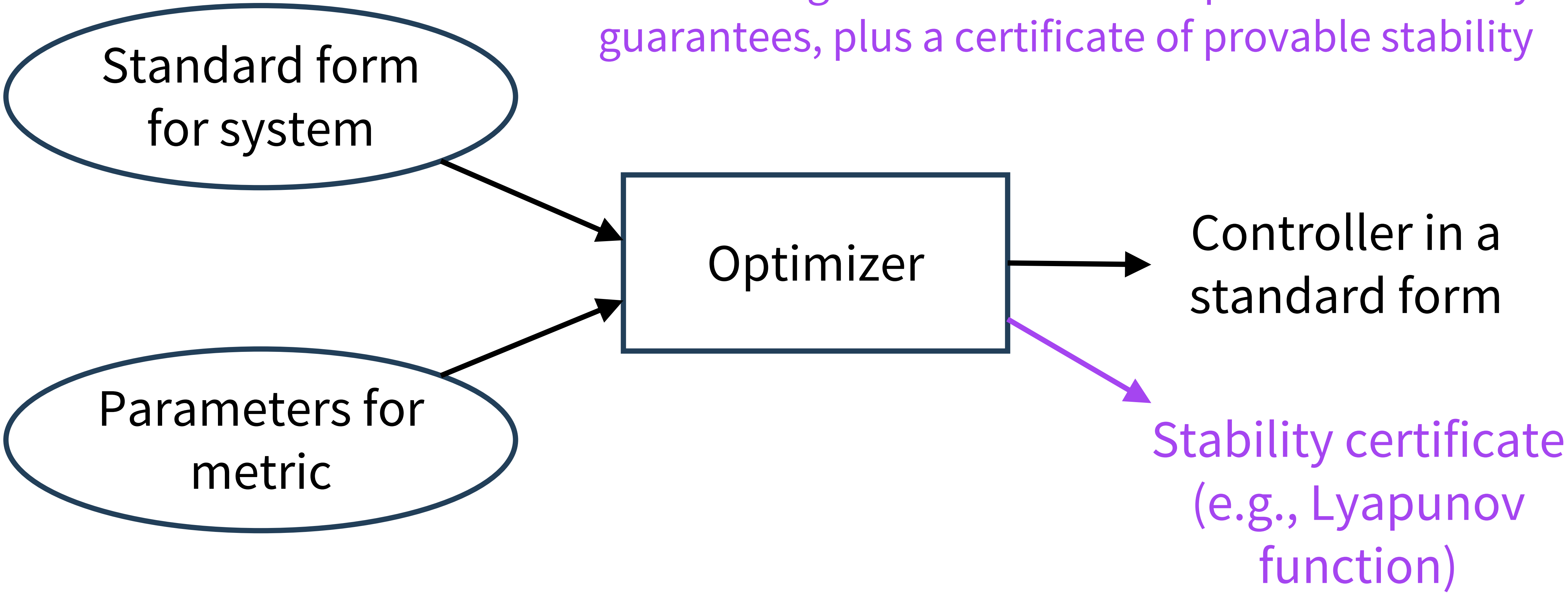
± Doesn’t assume knowledge of system (may be prudent, may be wasteful)

Outline

- Review of state-space control
- Overview & discussion of reinforcement learning (RL)
- **Highlight of research bridging RL and state-space control**

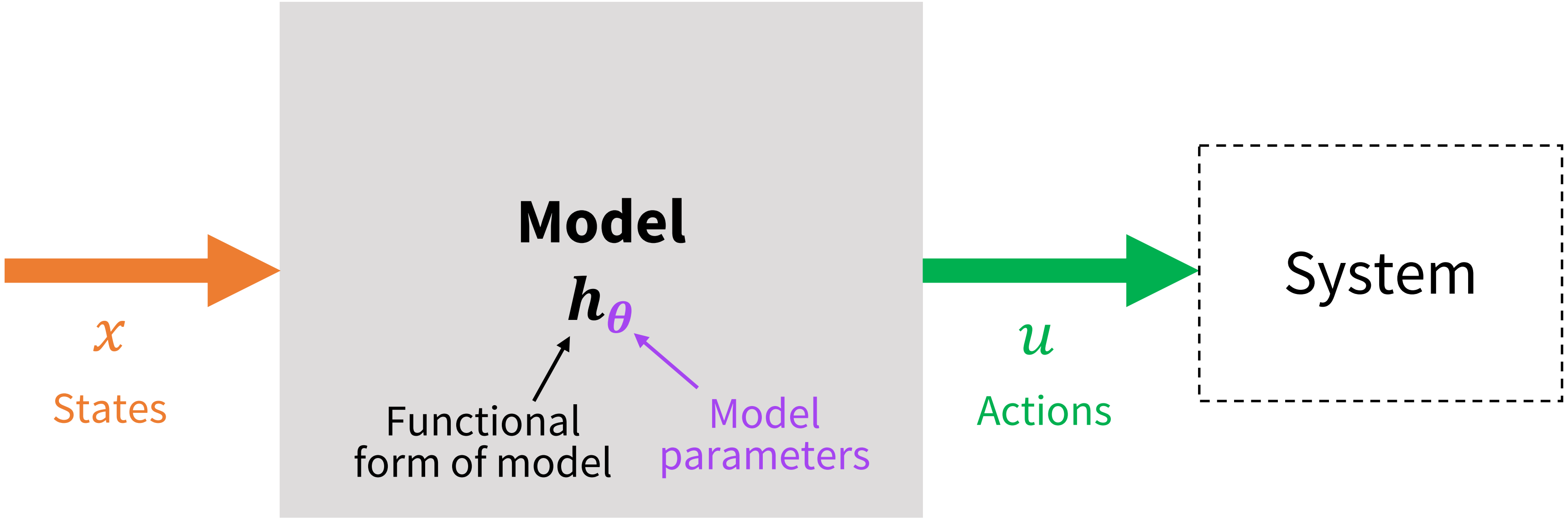
A state-space approach: Robust LQR

Solve for a good controller with provable stability guarantees, plus a certificate of provable stability



An RL approach: Policy optimization

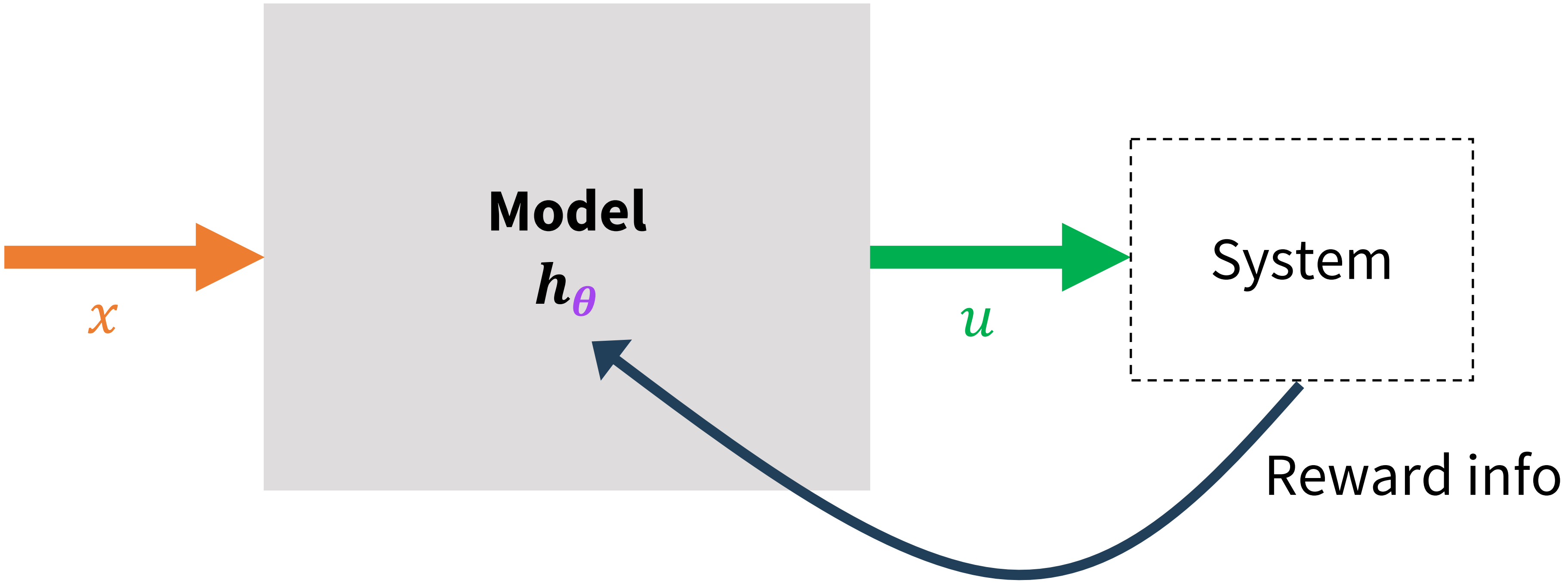
Learn the parameters of a function mapping from states to actions



An RL approach: Policy optimization

Learn the parameters of a function mapping from states to actions

Update parameters via trial-and-error using info about reward (“reward feedback”)

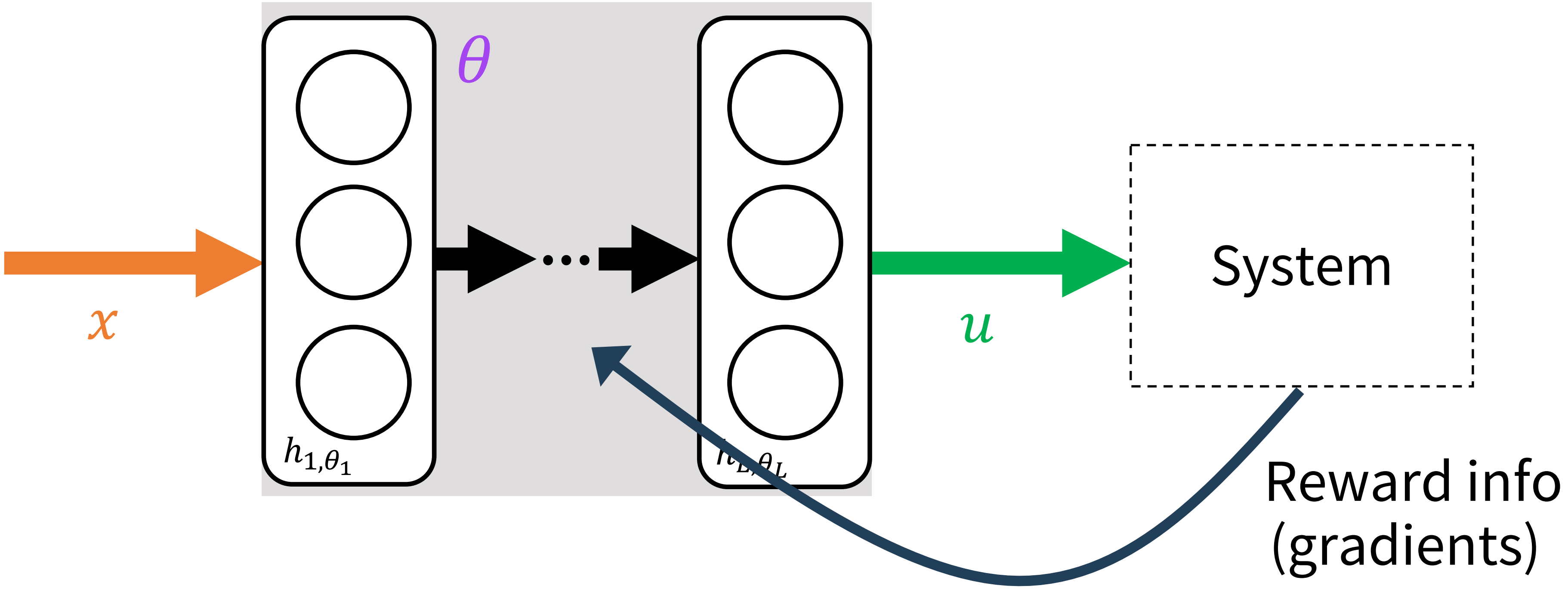


An RL approach: Policy optimization

Learn the parameters of a function mapping from states to actions

Update parameters via trial-and-error using info about reward (“reward feedback”)

Function can be a deep neural network (“deep reinforcement learning”)

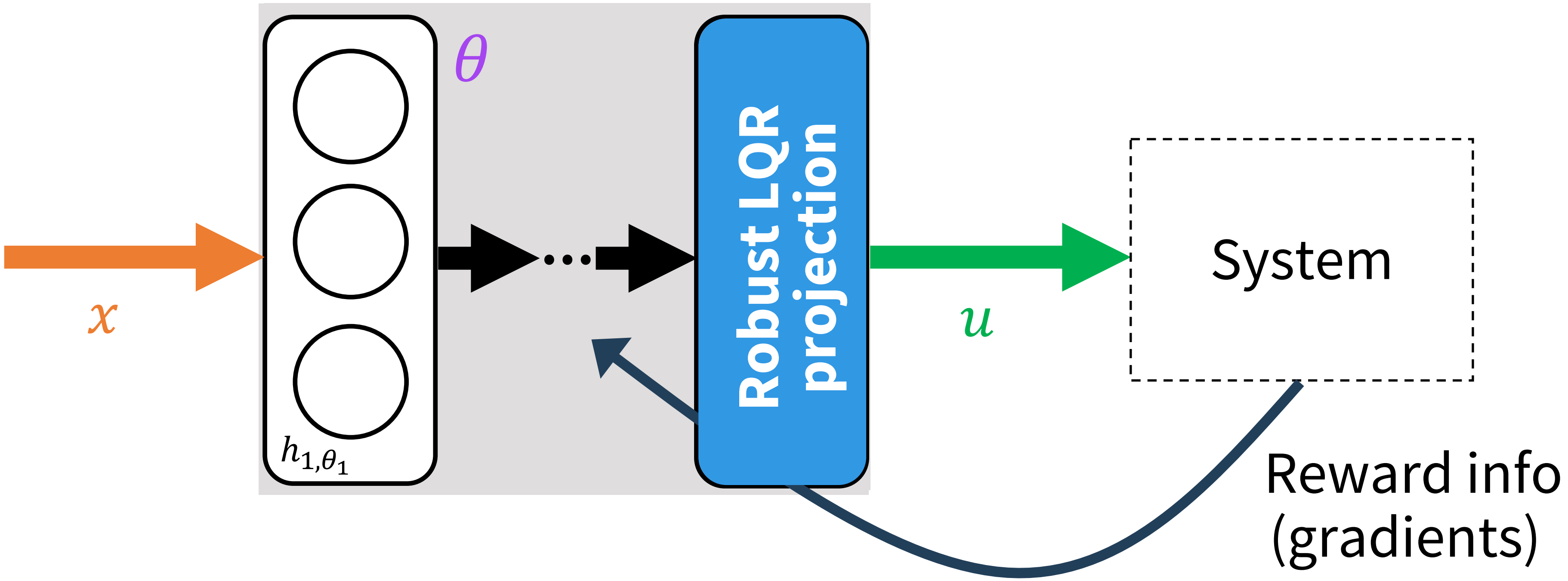


Combined: Deep RL w/ robust LQR projection

Get set of stabilizing actions using robust LQR stability certificate

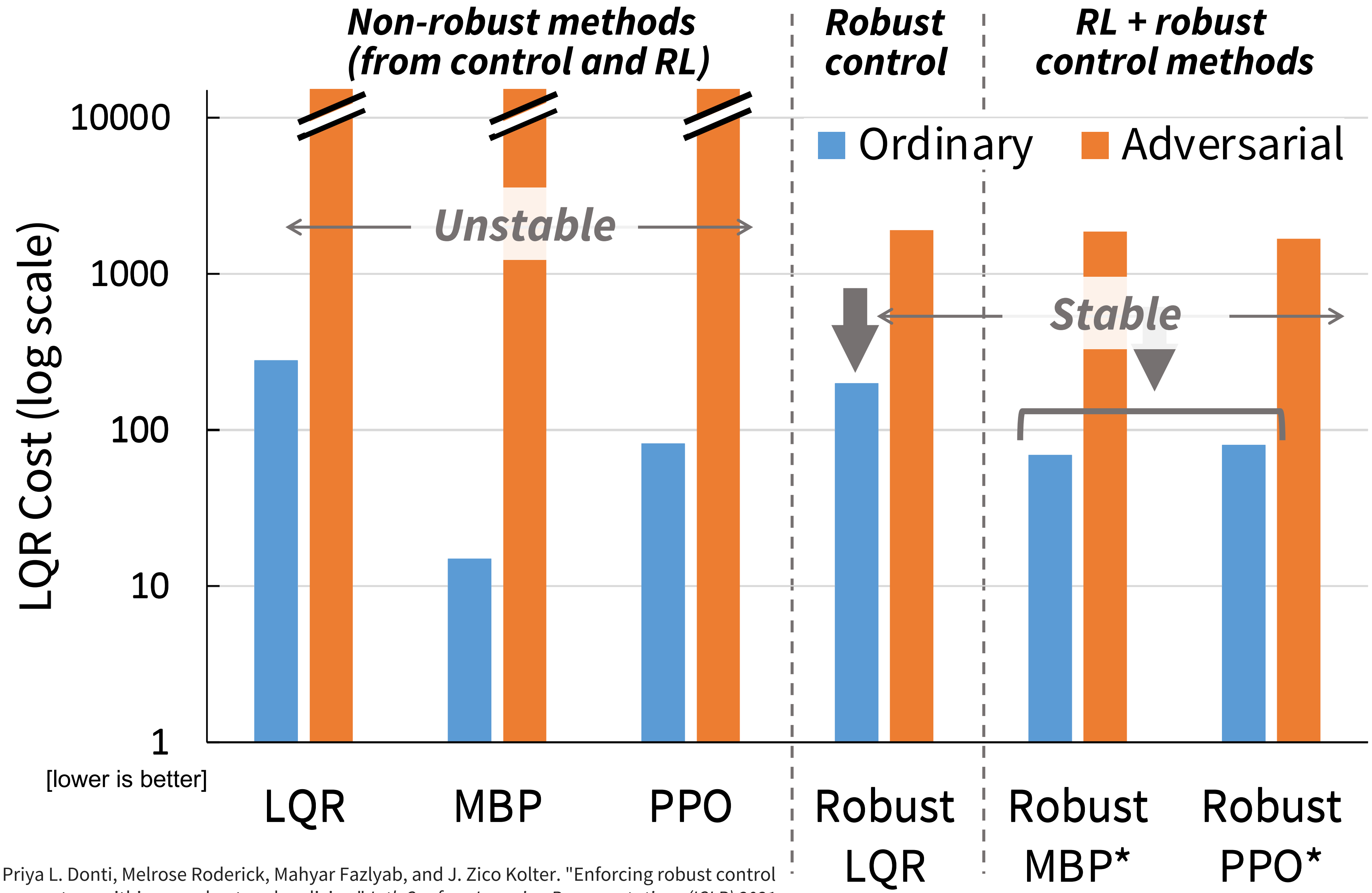
Project neural network outputs onto this set of actions

Yields deep reinforcement learning policy with provable robustness guarantees



Priya L. Donti, Melrose Roderick, Mahyar Fazlyab, and J. Zico Kolter. "Enforcing robust control guarantees within neural network policies." *Intl. Conf. on Learning Representations (ICLR) 2021.*

Illustrative results: Synthetic NLDI system



Using RL + robust control can lead to:

- **Improved "average-case" performance** over robust baselines
- **Provable stability under "worst-case" dynamics** (unlike non-robust baselines)

Priya L. Donti, Melrose Roderick, Mahyar Fazlyab, and J. Zico Kolter. "Enforcing robust control guarantees within neural network policies." *Intl. Conf. on Learning Representations (ICLR) 2021.*

Many ways to bridge machine learning & control

UNIVERSITY OF OXFORD

L4DC 2024 Conference

HOME | CALL FOR PAPERS ▾ | ORGANIZERS & PC | SPONSOR

Key dates

Submission opening	1 October 2023
Submission deadline for papers	~~1 December~~ **11 December** 2023 (23:55 GMT)

Over the next decade, the biggest generator of data is expected to be devices that sense and control the physical world.

The explosion of real-time data that is emerging from the physical world requires a rapprochement of areas such as machine learning, control theory, and optimization. While control theory has been firmly rooted in the tradition of model-based design, the availability and scale of data (both temporal and spatial) will require rethinking the foundations of our discipline. From a machine learning perspective, one of the main challenges going forward is to go beyond pattern recognition and address problems in data-driven control and optimization of dynamical processes. Our overall goal is to create a new community of people who think rigorously across the disciplines, ask new questions, and develop the foundations of this new scientific area. We are happy to welcome you to the University of Oxford for the 6th annual L4DC.

Recap

- Overview of reinforcement learning (RL)
- Some similarities & differences between RL and state-space control
- Some strengths & limitations of both approaches
- One example of combining RL and state-space approaches