A modular framework for stabilizing deep reinforcement learning control

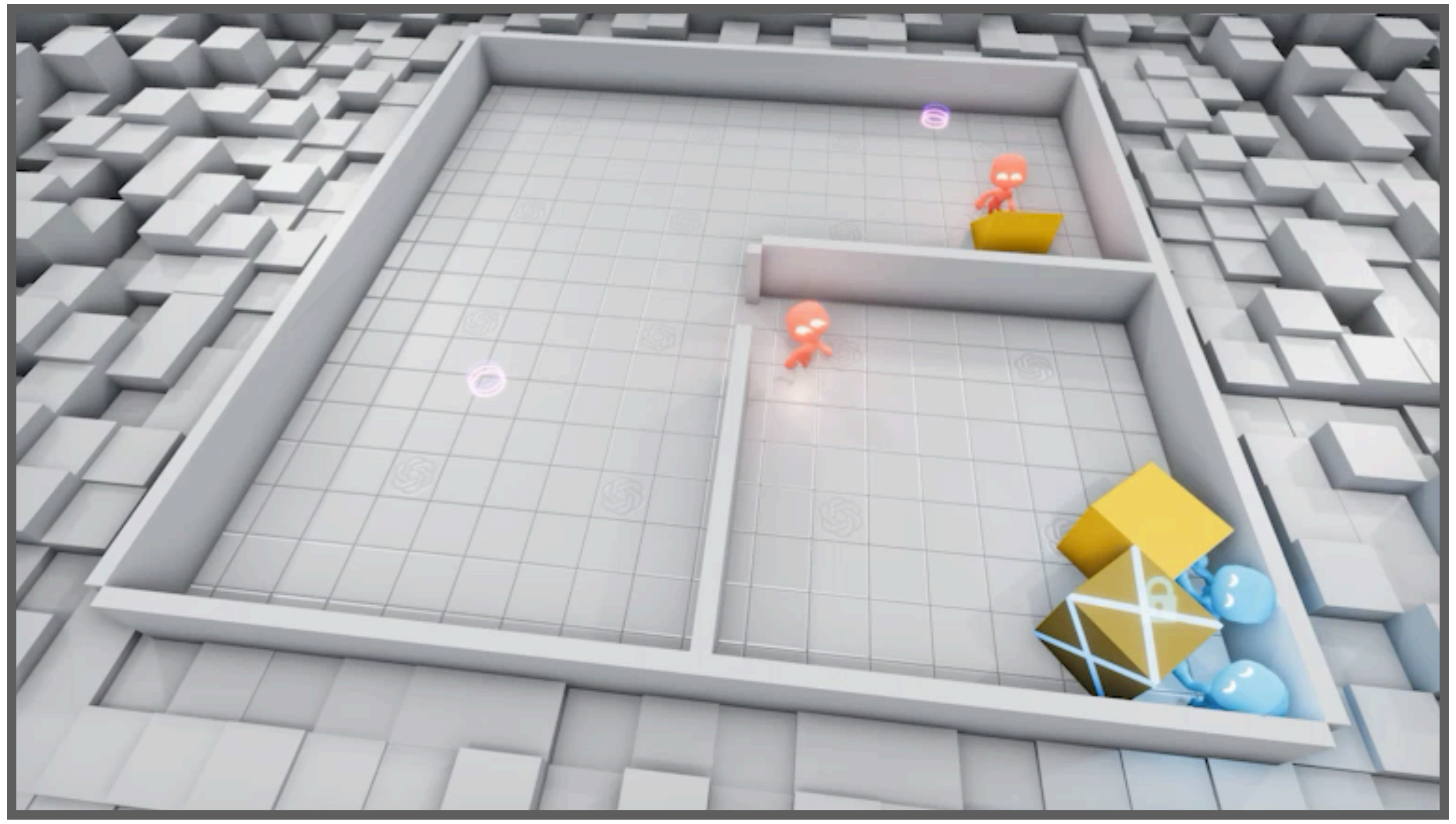
IFAC World Congress 2023

Nathan Lawrence ~ University of British Columbia ~ lawrence@math.ubc.ca

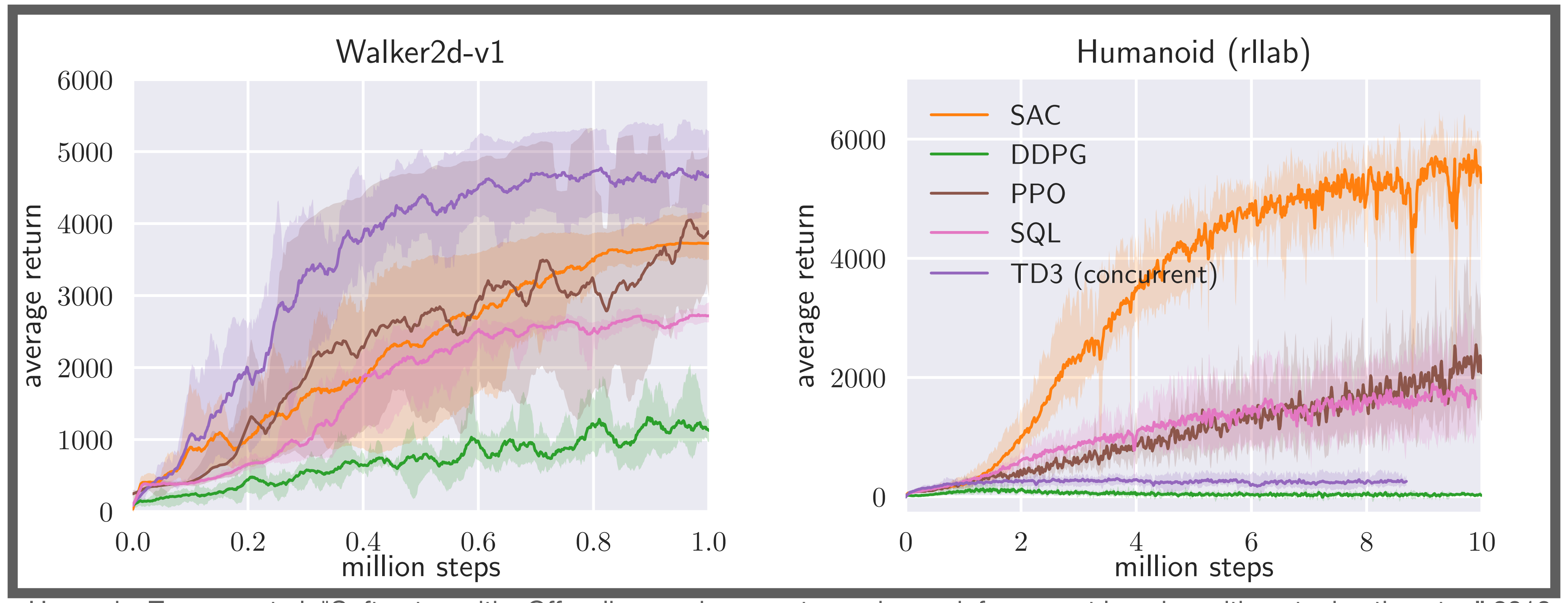
Philip Loewen, Shuyuan Wang, Michael Forbes, Bhushan Gopaluni

Reinforcement learning

Maximizing reward through experience

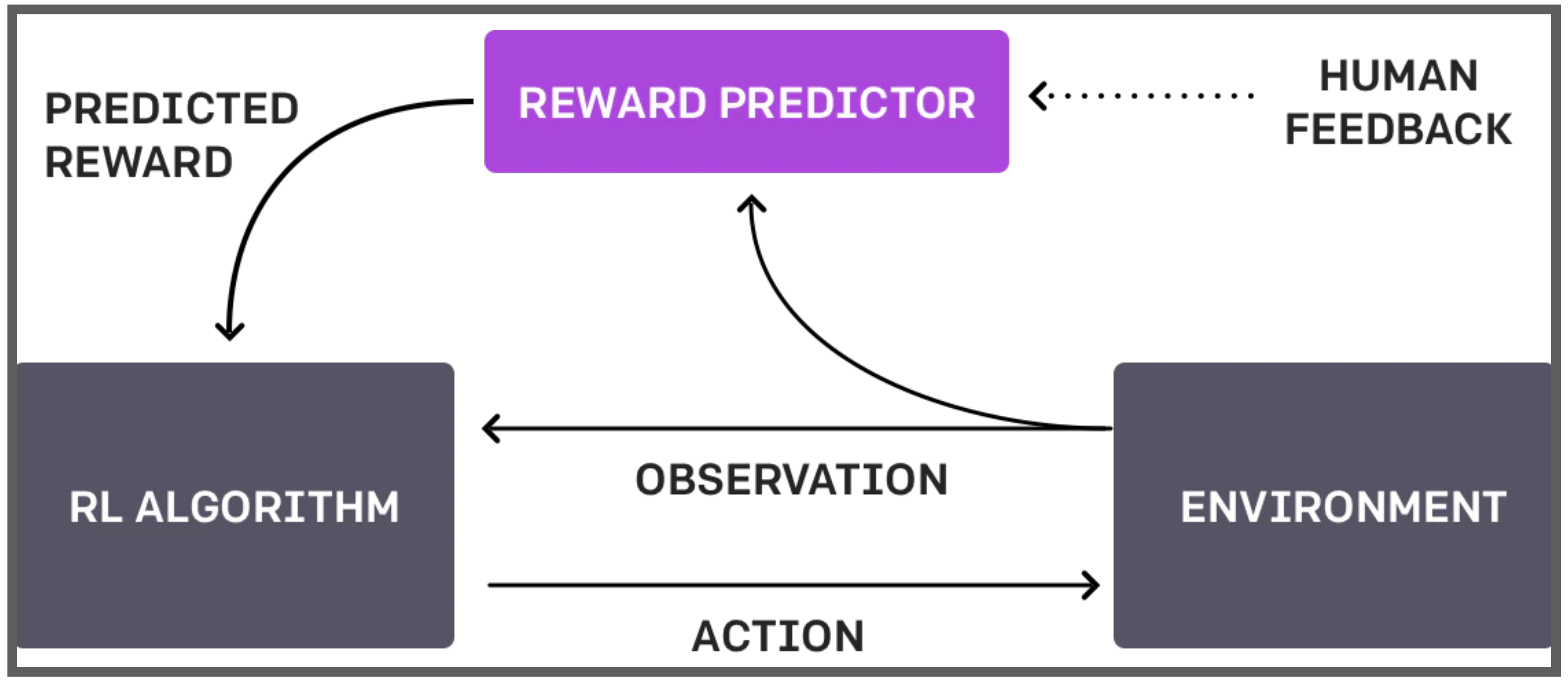


https://openai.com/research/emergent-tool-use#surprisingbehaviors

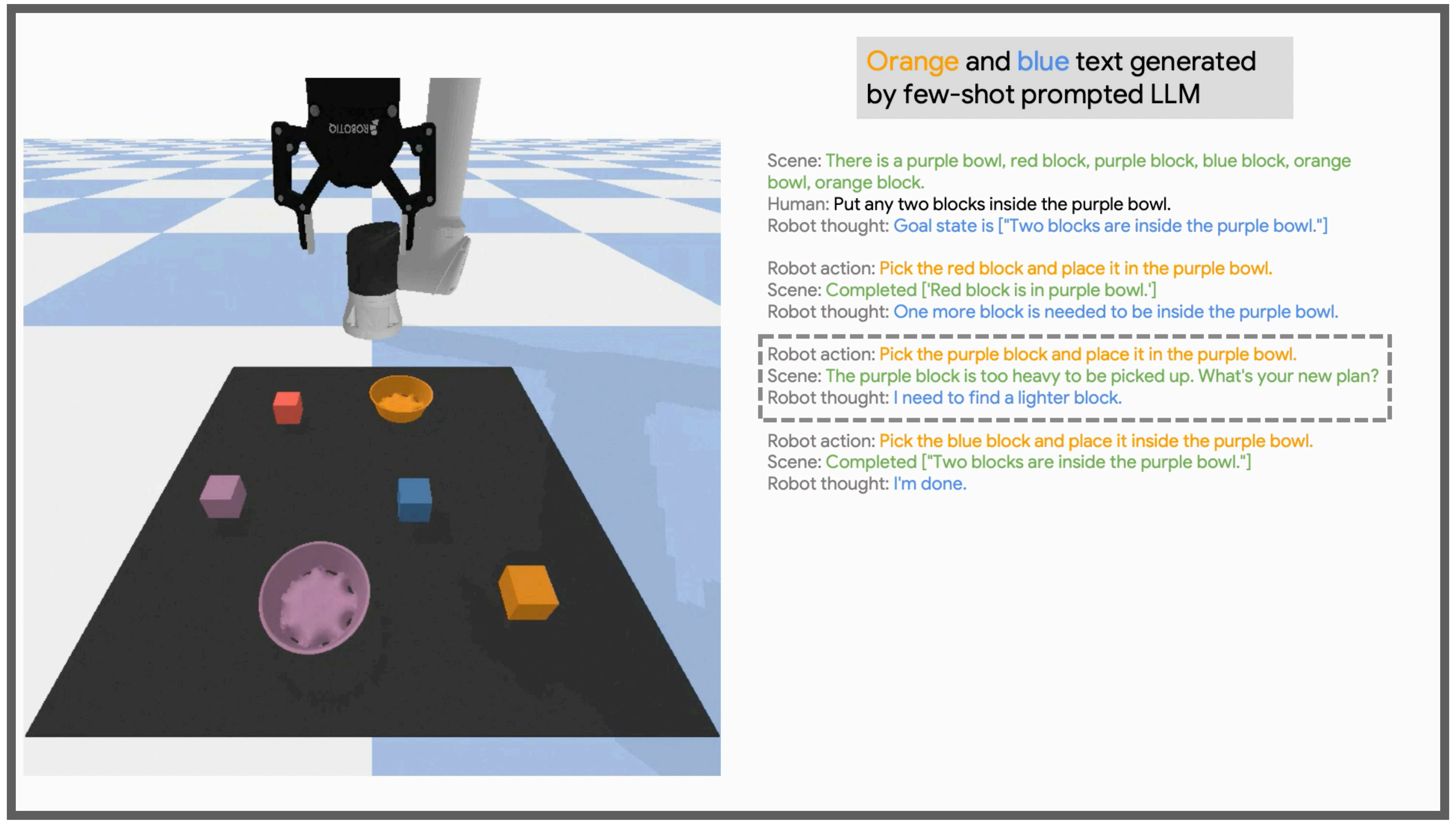


Haarnoja, Tuomas, et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." 2018.

ChatGPT

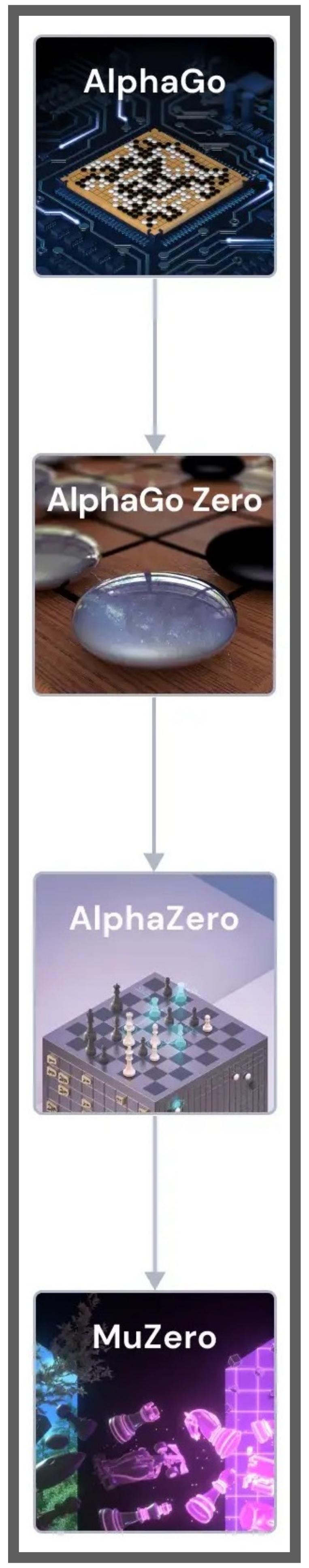


https://openai.com/research/learning-from-human-preferences



https://innermonologue.github.io/

Chess, Go, Shogi



https://www.deepmind.com/blog/muzero-mastering-go-chess-shogi-and-atari-without-rules

RL in PSE?

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A review On reinforcement learning: Introduction and applications in industrial process control

Rui Nian, Jinfeng Liu*, Biao Huang

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Reinforcement learning for batch process control: Review and perspectives

Haeun Yoo[1], Ha Eun Byun[1], Dongho Han, Jay H. Lee*

PROCESS SYSTEMS ENGINEERING

AIChE JOURNAL

Toward self-driving processes: A deep reinforcement learning approach to control

Steven Spielberg[1] | Aditya Tulsyan[1] | Nathan P. Lawrence[2] | Philip D. Loewen[2] | R. Bhushan Gopaluni[1]

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Reinforcement Learning – Overview of recent progress and implications for process control ☆

Joohyun Shin[a], Thomas A. Badgwell[b], Kuang-Hung Liu[b], Jay H. Lee[a,*]

Can reinforcement learning help maintain control loops?

It's complicated

In favor

- Leverage observed data to improve operations
- **Minimize prior domain knowledge**
- Automated maintenance on a variety of systems

Against

- Additional algorithmic complexity
- Auto-tuners exist already (but are often idle)
- **Stability during and after training**
- Sample efficiency

Our goal is to balance the automation and scalability of reinforcement learning with control-theoretic tools to create efficient and safe improvements

Reinforcement learning over all stable behaviour

Topics for today

1. Willems' lemma

   Data-based characterization of dynamics

2. Youla-Kučera parameterization

   Recipe for all stabilizing controllers

3. Learning algorithms

   A module to shape system behavior

Combining these elements gives a modular setup that decouples learning and stability

Key ingredients

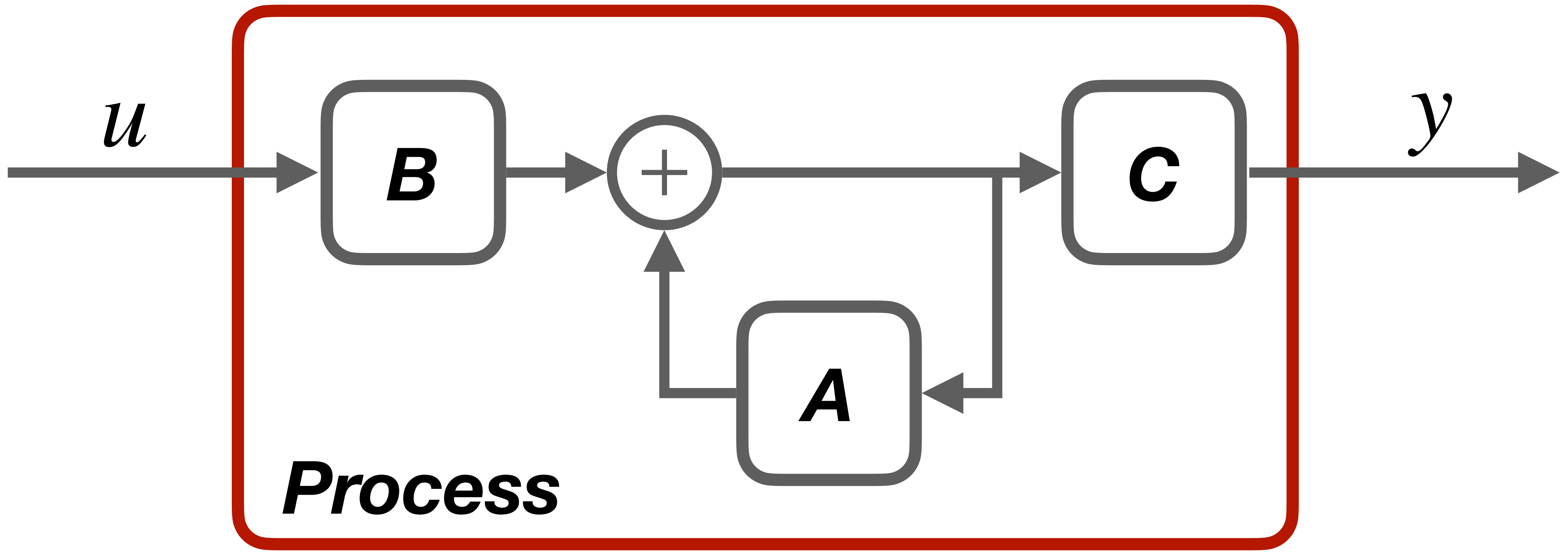
State-space model

- Define the system equations

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

where
$A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times 1}, C \in \mathbb{R}^{1 \times n}$

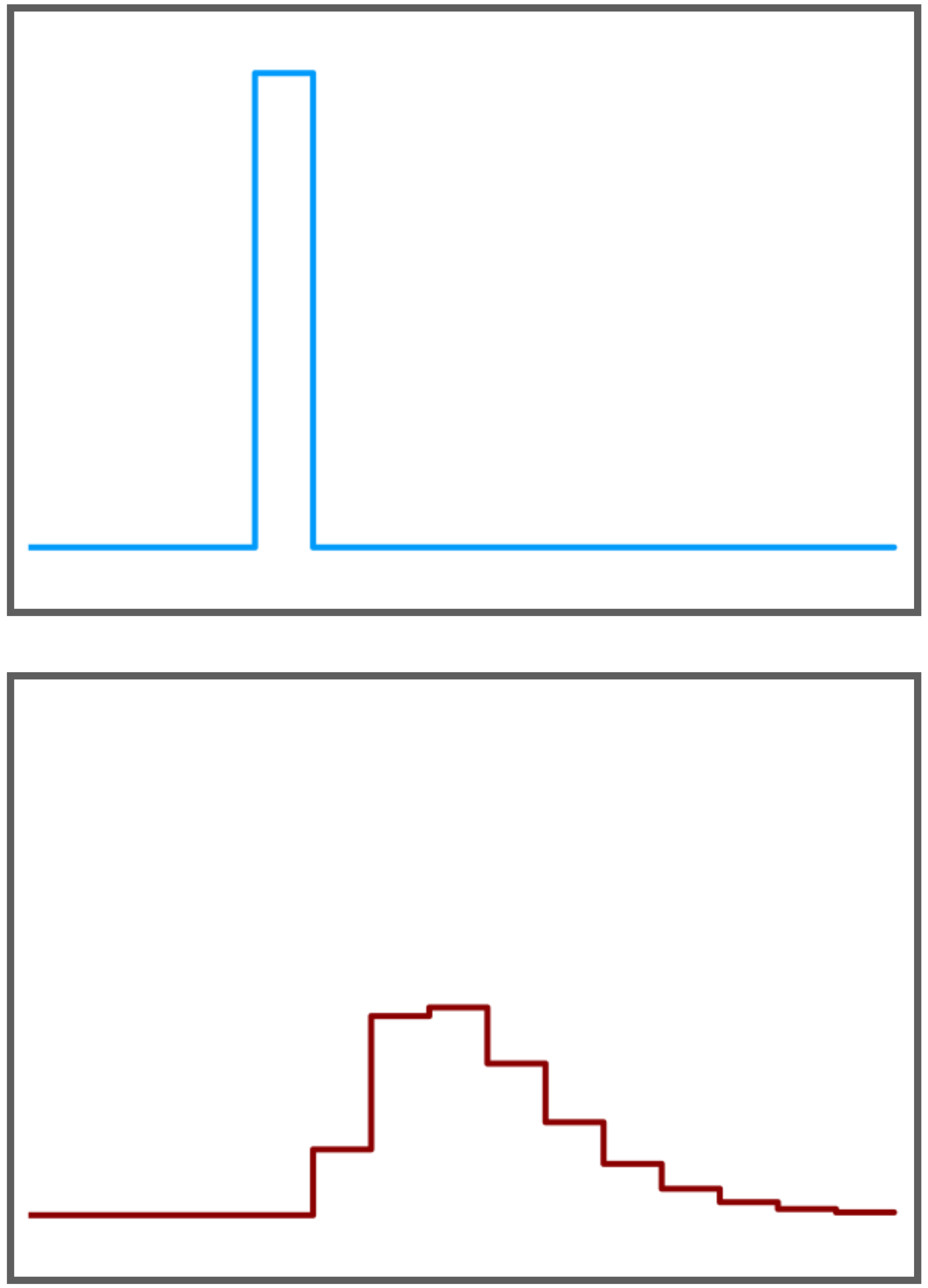
- Inputs, outputs are scalars for simplicity



Willems' fundamental lemma — a special case

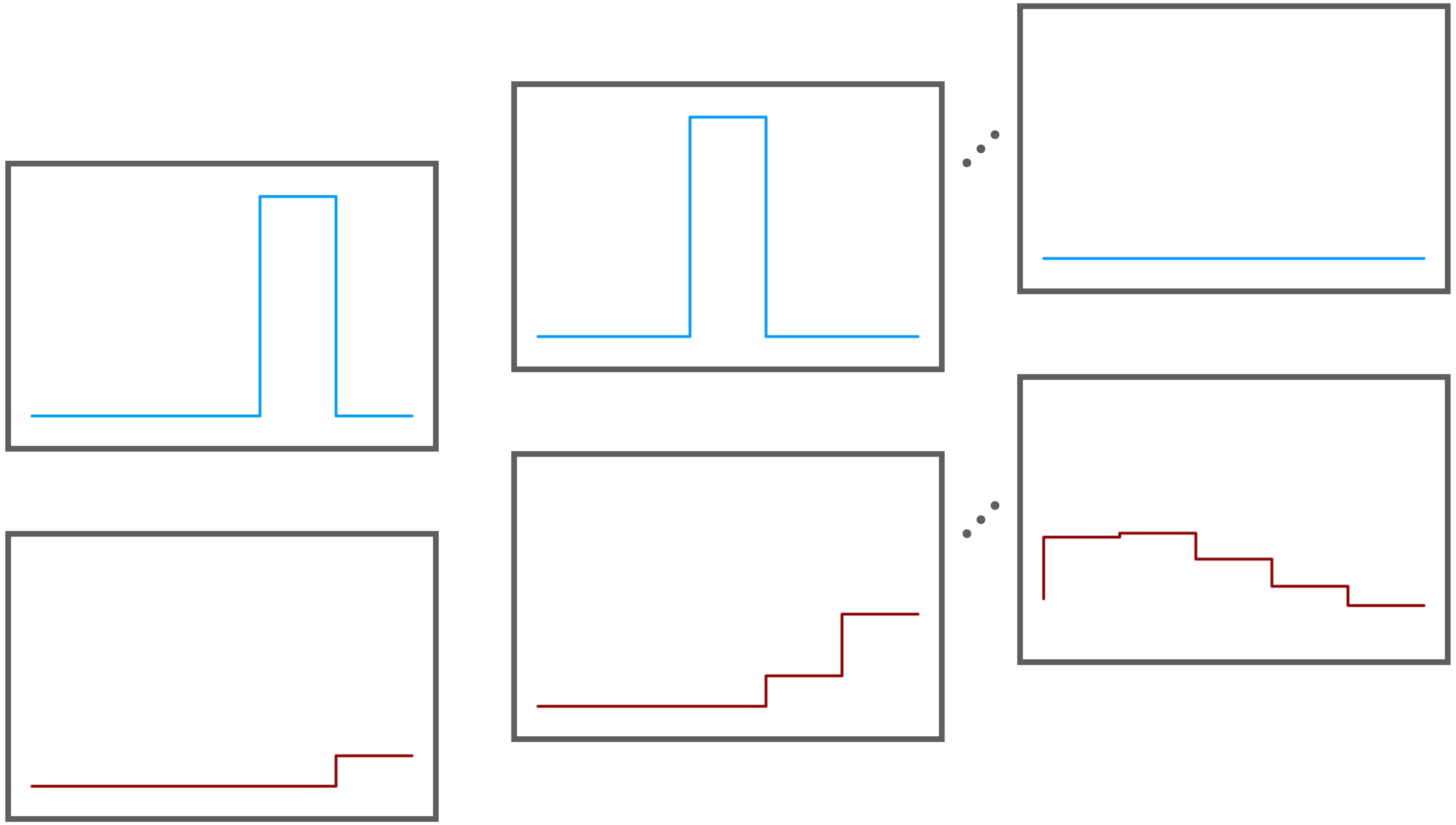
(Picture form)

Impulse



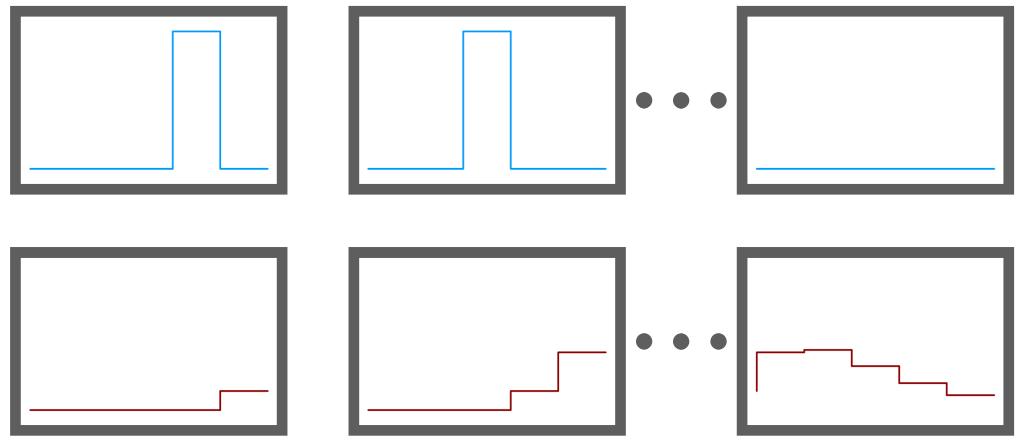
$\Longrightarrow$

What is the span of these data vectors?



1

$$\left.\left[\begin{array}{cccc|cccc} & & & 1 & & & & \\ & & \iddots & & & & & \\ & \iddots & & & & & & \\ 1 & & & & & & & \\ \hline & & & 0 & y_1 & y_2 & \cdots & y_n \\ & & \iddots & y_1 & y_2 & y_3 & \iddots & y_{n+1} \\ & \iddots & \iddots & \vdots & \vdots & \iddots & \iddots & \vdots \\ 0 & y_1 & \cdots & y_{n-1} & y_n & y_{n+1} & \cdots & y_{2n} \end{array}\right]\right\}$$



➡ Full-rank data matrix!

$$\underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}}_{\mathscr{O}} \underbrace{\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}}_{\mathscr{C}} = \begin{bmatrix} CB & CAB & \cdots & CA^{n-1}B \\ CAB & CA^2B & \cdots & CA^nB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n-1}B & CA^nB & \cdots & CA^{2n-1}B \end{bmatrix}$$

Willems' fundamental lemma — general case

Data $\Longleftrightarrow$ models

- Given a signal $z = \{z_t\}_{t=0}^{N-1}$, define its Hankel matrix of order L:

"Persistently exciting" if full rank

$$H_L(z) = \begin{bmatrix} z_0 & z_1 & \cdots & z_{N-L} \\ z_1 & z_2 & \cdots & z_{N-L+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{L-1} & z_L & \cdots & z_{N-1} \end{bmatrix}$$

- Let $\{u_t, y_t\}_{t=0}^{N-1}$ be a trajectory where u is persistently exciting of order $L+n$. Then $\{\bar{u}_t, \bar{y}_t\}_{t=0}^{L-1}$ is a trajectory if and only if there exists α such that

$$\begin{bmatrix} H_L(u) \\ H_L(y) \end{bmatrix} \alpha = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}$$

All possible data

Static, collected data

Willems, Jan C., et al. "A note on persistency of excitation." 2005.

1

A dynamic Willems' lemma

Carrying a trajectory forward

- Start with Willems' lemma:

$$\begin{bmatrix} H_L(u) \\ H_L(y) \end{bmatrix} \alpha_0 = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}$$

- How to advance to the next output? Consider nested Hankel matrices:

$$\begin{bmatrix} y_0 & y_1 & \cdots & y_{N-L} & y_{N-L+1} \\ y_1 & y_2 & \cdots & y_{N-L+1} & y_{N-L+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{L-1} & y_L & \cdots & y_{N-1} & y_N \end{bmatrix}$$

$H_L(y)$

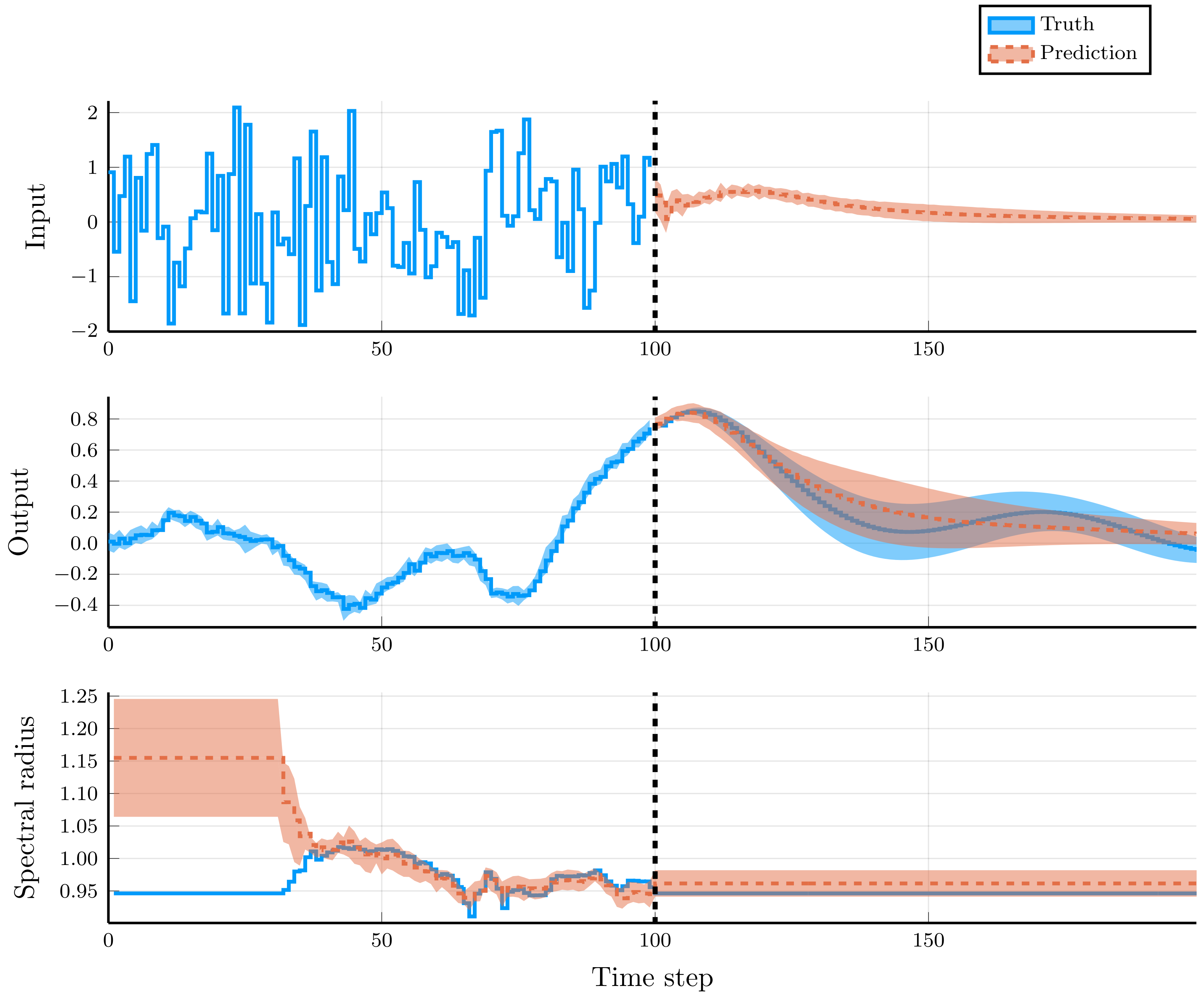
$H'_L(y)$

$$\begin{bmatrix} \bar{u}' \\ \bar{y}' \end{bmatrix} = \underbrace{\begin{bmatrix} H'_L(u) \\ H'_L(y) \end{bmatrix}}_{H'_L(y)} \alpha_0$$

Multiply previous solution by shifted Hankel matrix — Then repeat!

$$\boxed{H_L(y)\alpha' = H'_L(y)\alpha}$$

$$\Downarrow$$

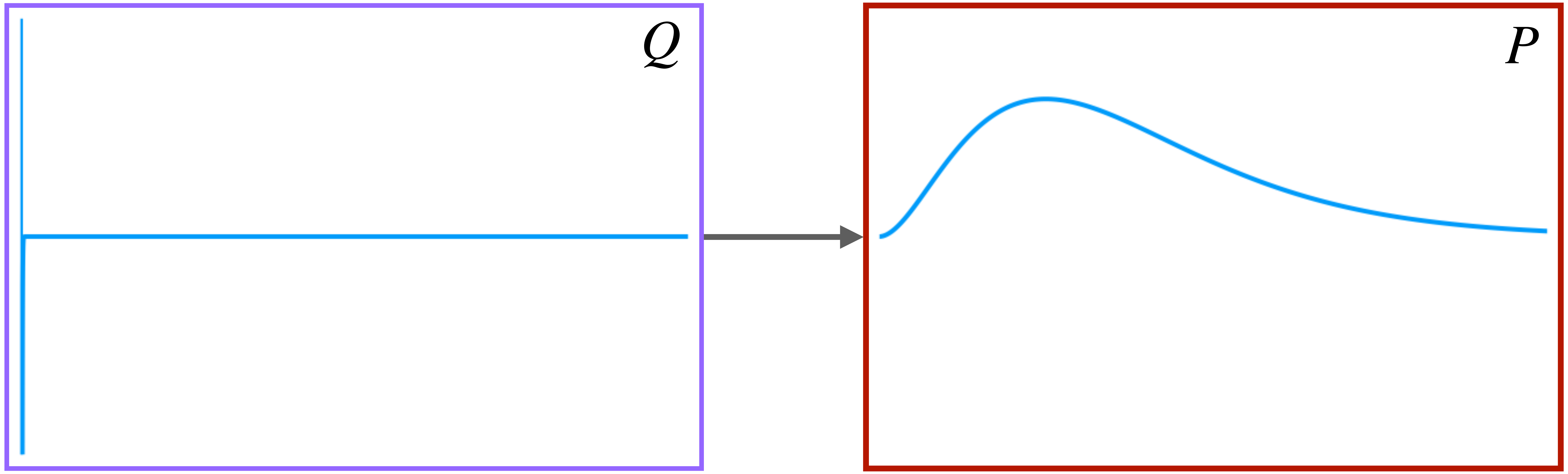


So far we have characterized a system in terms of data … how do we drive its behaviour?

Youla-Kučera parameterization

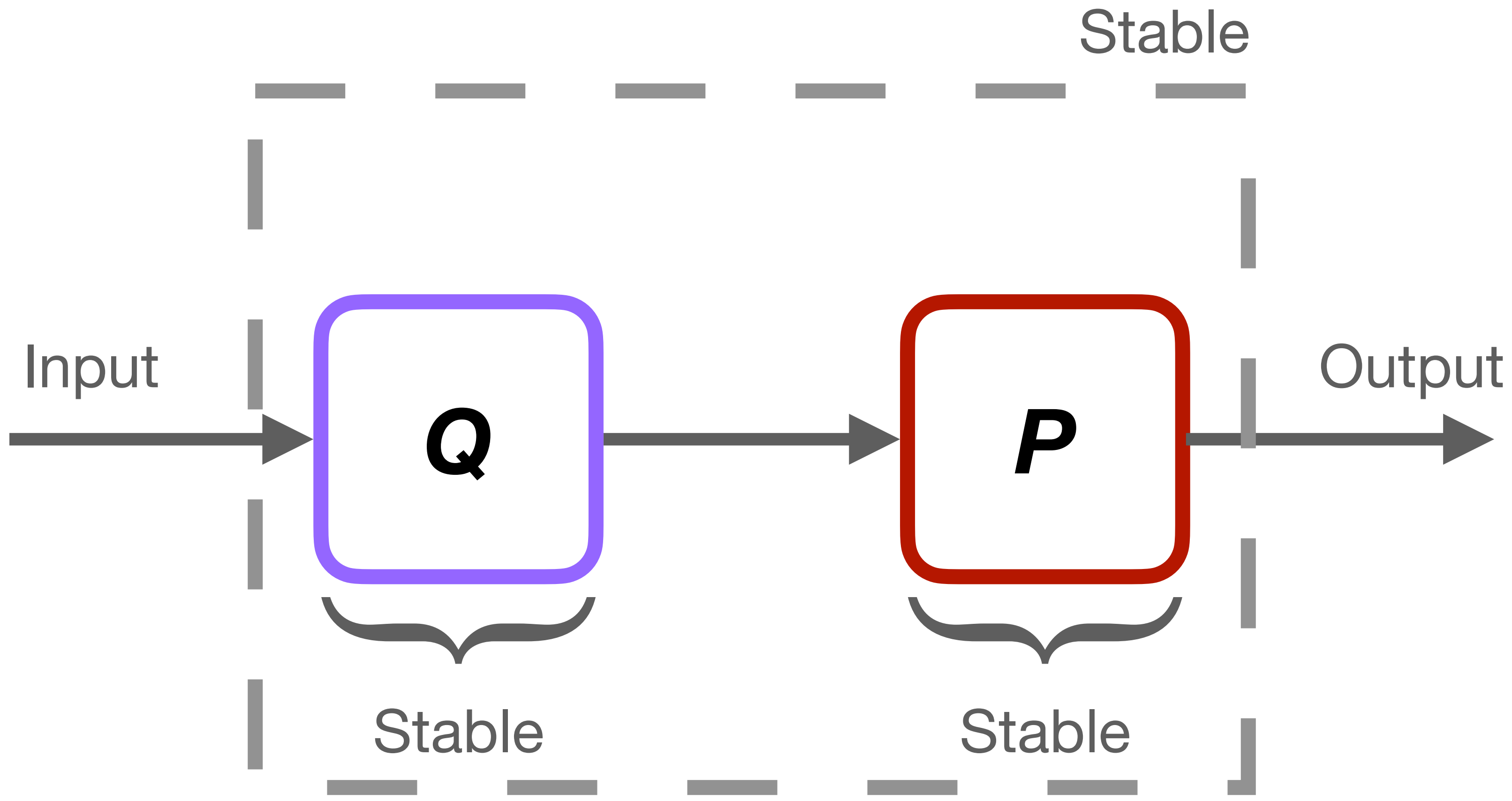
All stabilizing controllers

- Hard: Given a controller K, is it stabilizing? What is the set of all stabilizing controllers?
- Easier: What happens when you probe P with stable dynamics Q?

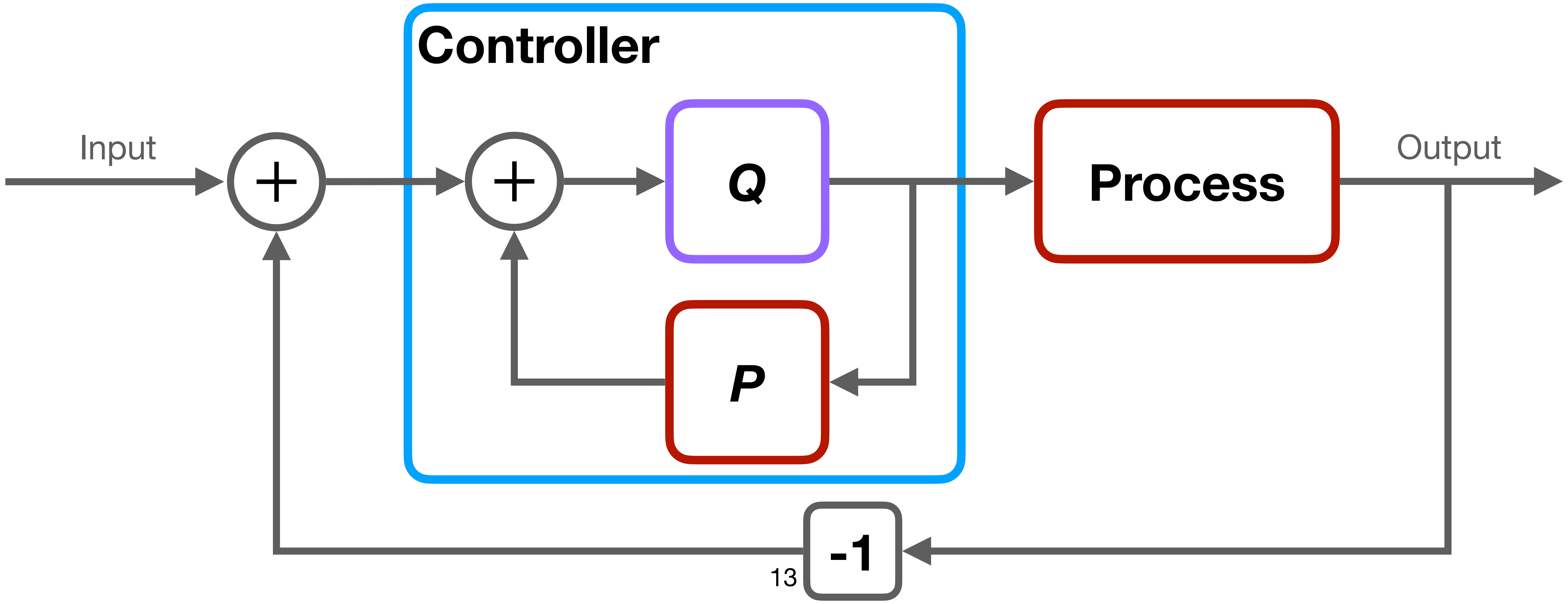


Q "parameter" characterizes stable behaviour

How do we turn it into a controller?



$\Longleftrightarrow$



Learning stable systems

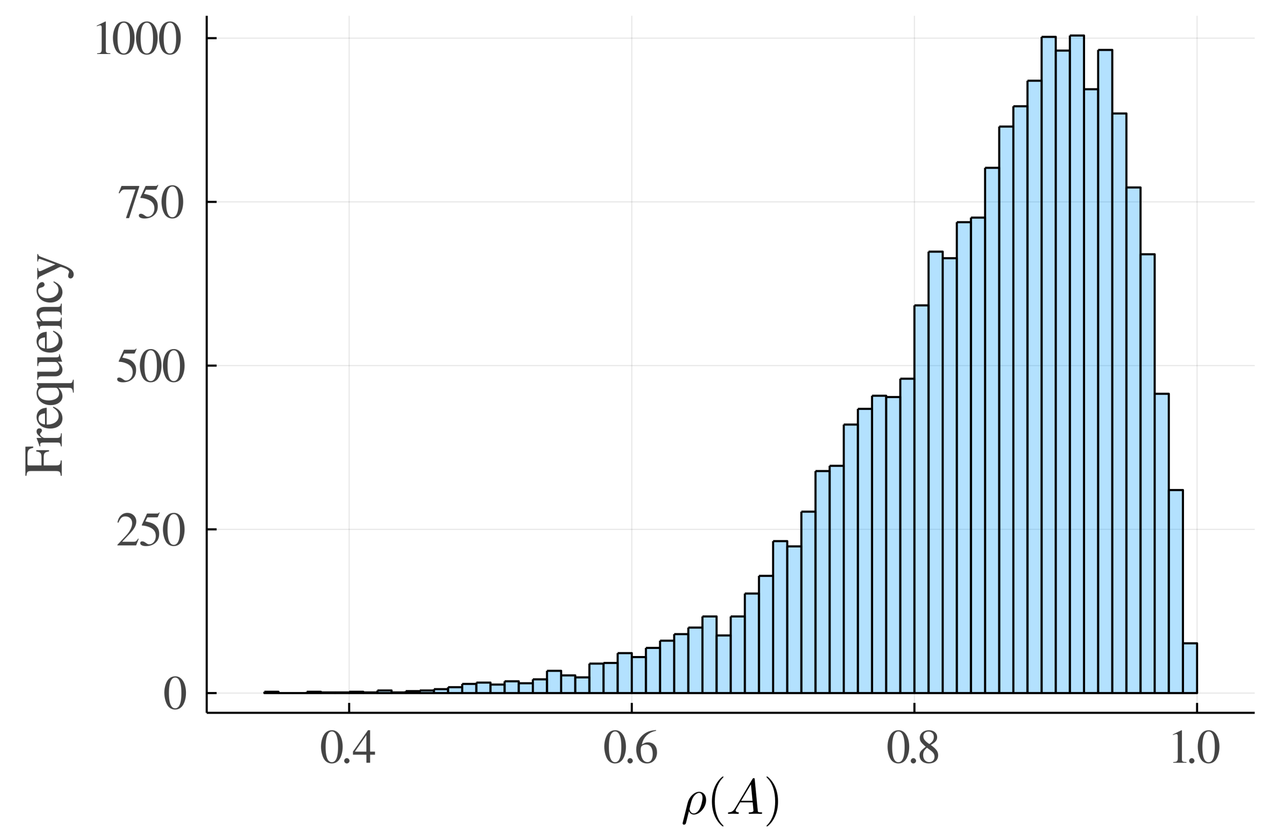
(Q in Youla-Kučera)

- Q is a global parameter, but explicitly writing it down is difficult

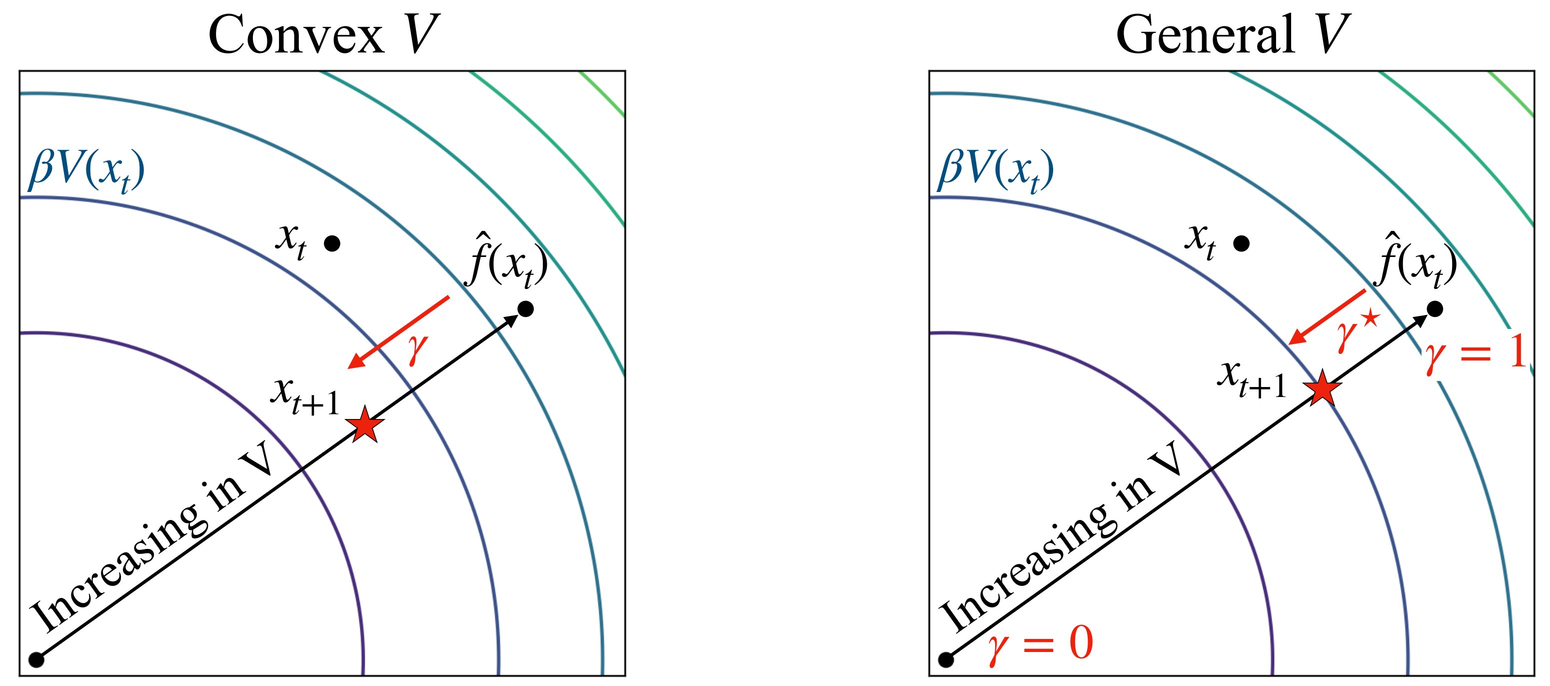
- We represent Q using an unconstrained set of trainable parameters

- Yields stable models suitable for RL or supervised learning

Linear case — matrix factorization



Nonlinear case — stable DNN



Lawrence, Nathan, et al. "Almost surely stable deep dynamics." 2020.

Final ingredient: learning algorithms

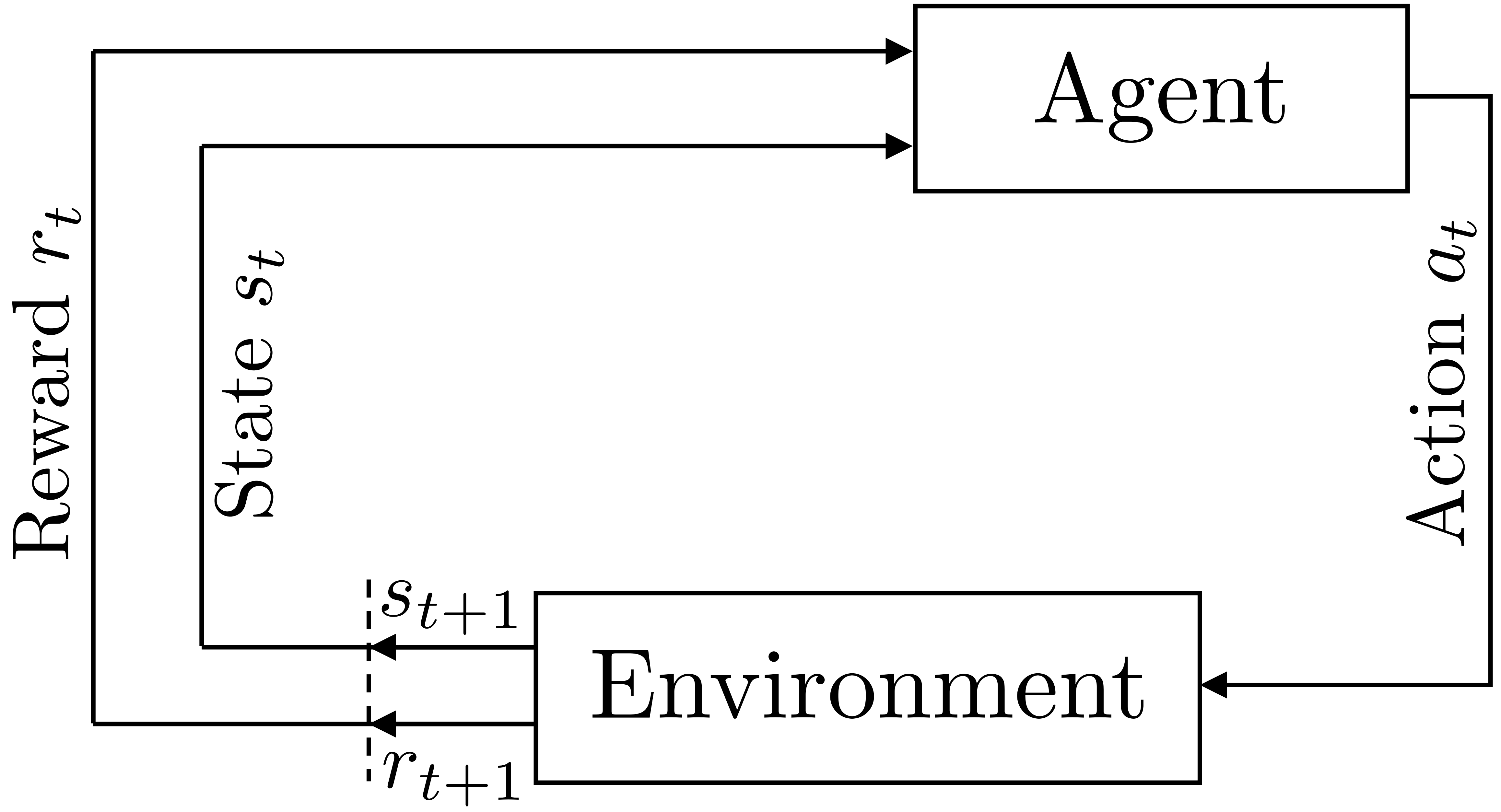
Reinforcement learning

Business as usual

- A “policy” π interacts with an “environment”, generating a trajectory $s_0, a_0, r_0, s_1, a_1, r_1, \ldots$

- A “return” is accrued and averaged: $V(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$,where $s = s_0$

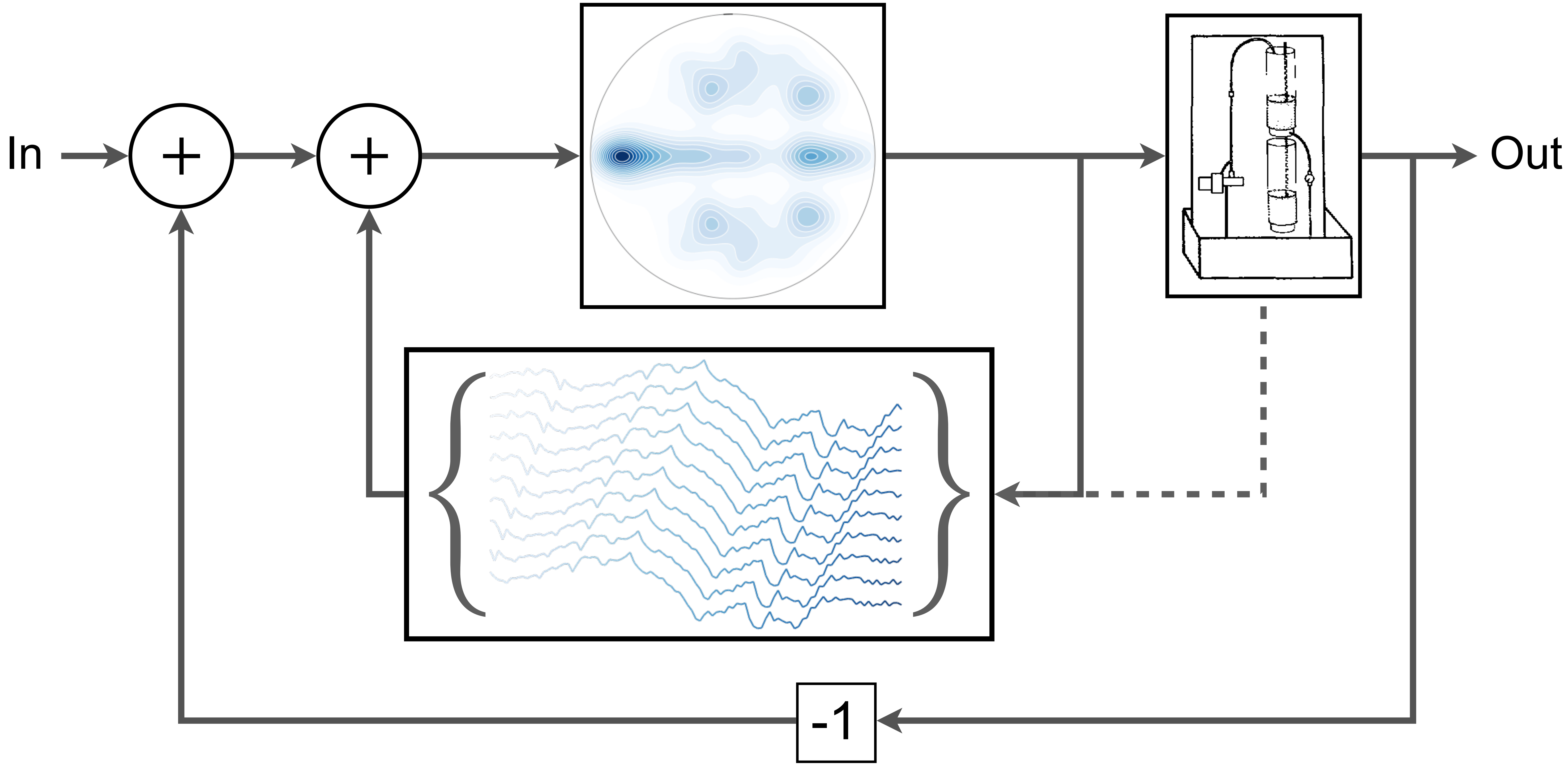
- An “agent” tries to find the “best” policy



Reinforcement learning over all stable behaviour

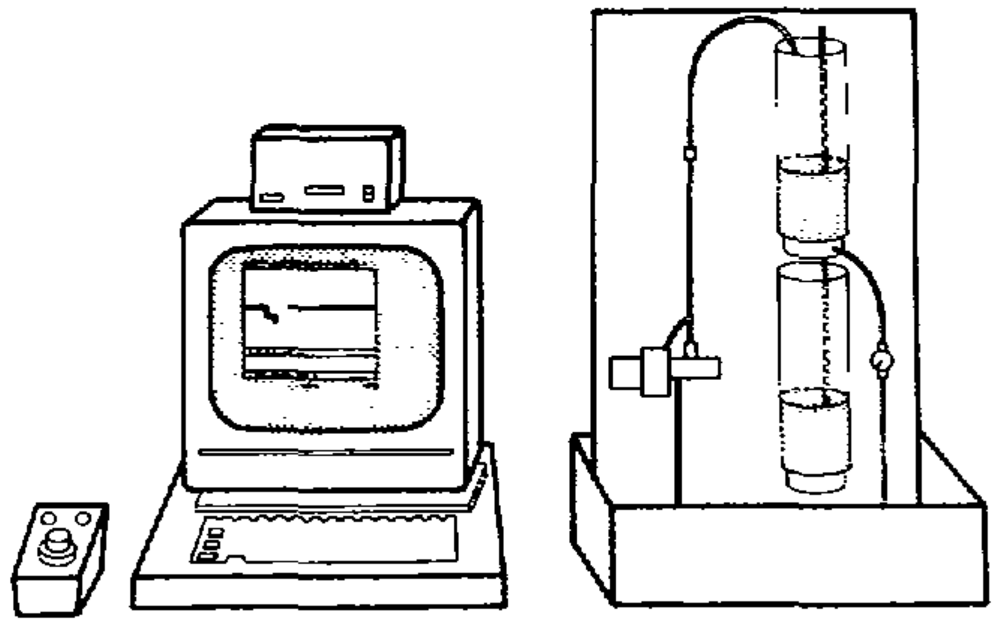
A modular setup

1. Willems' lemma
2. Youla-Kučera
3. Learning algorithm

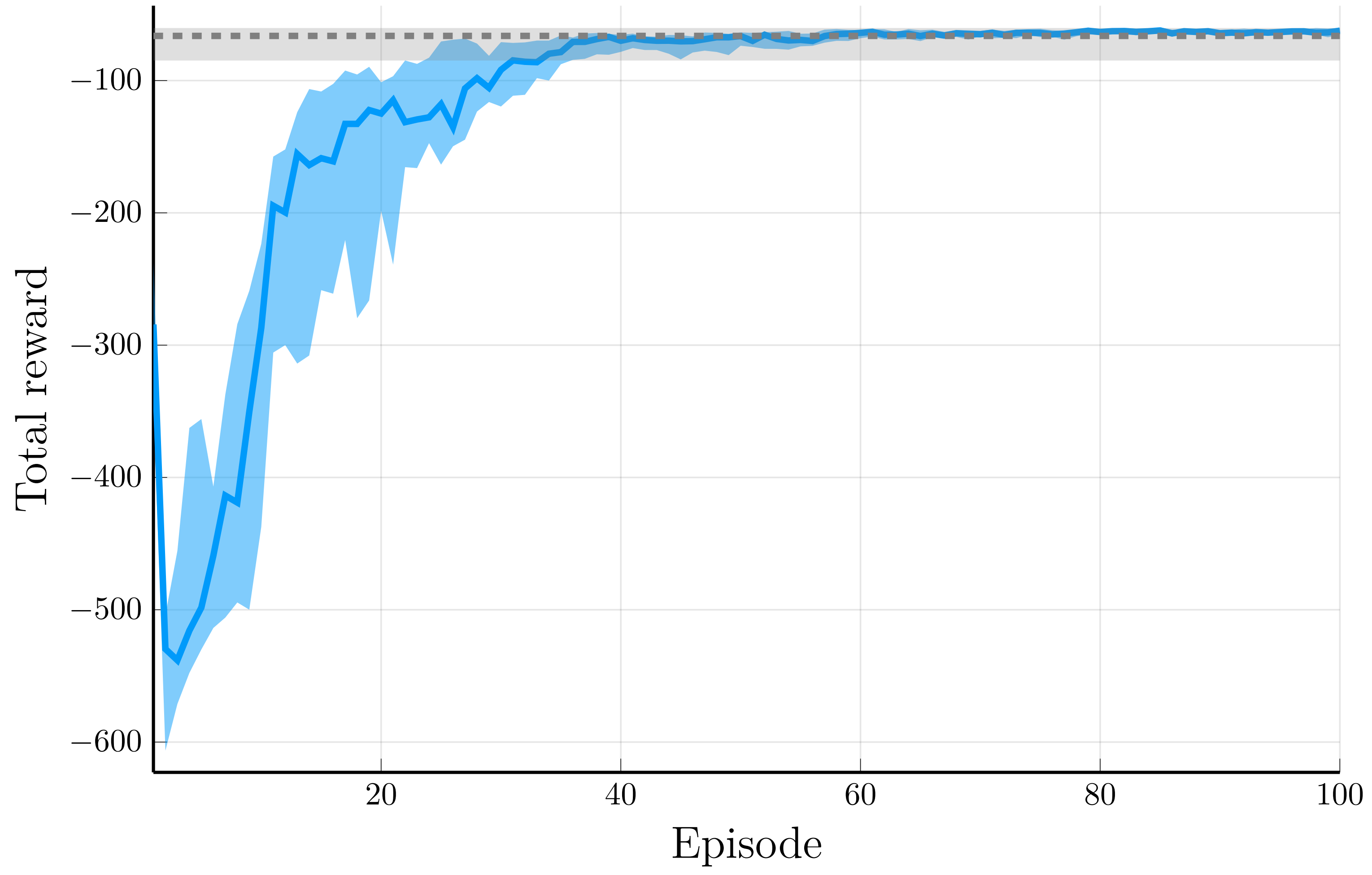


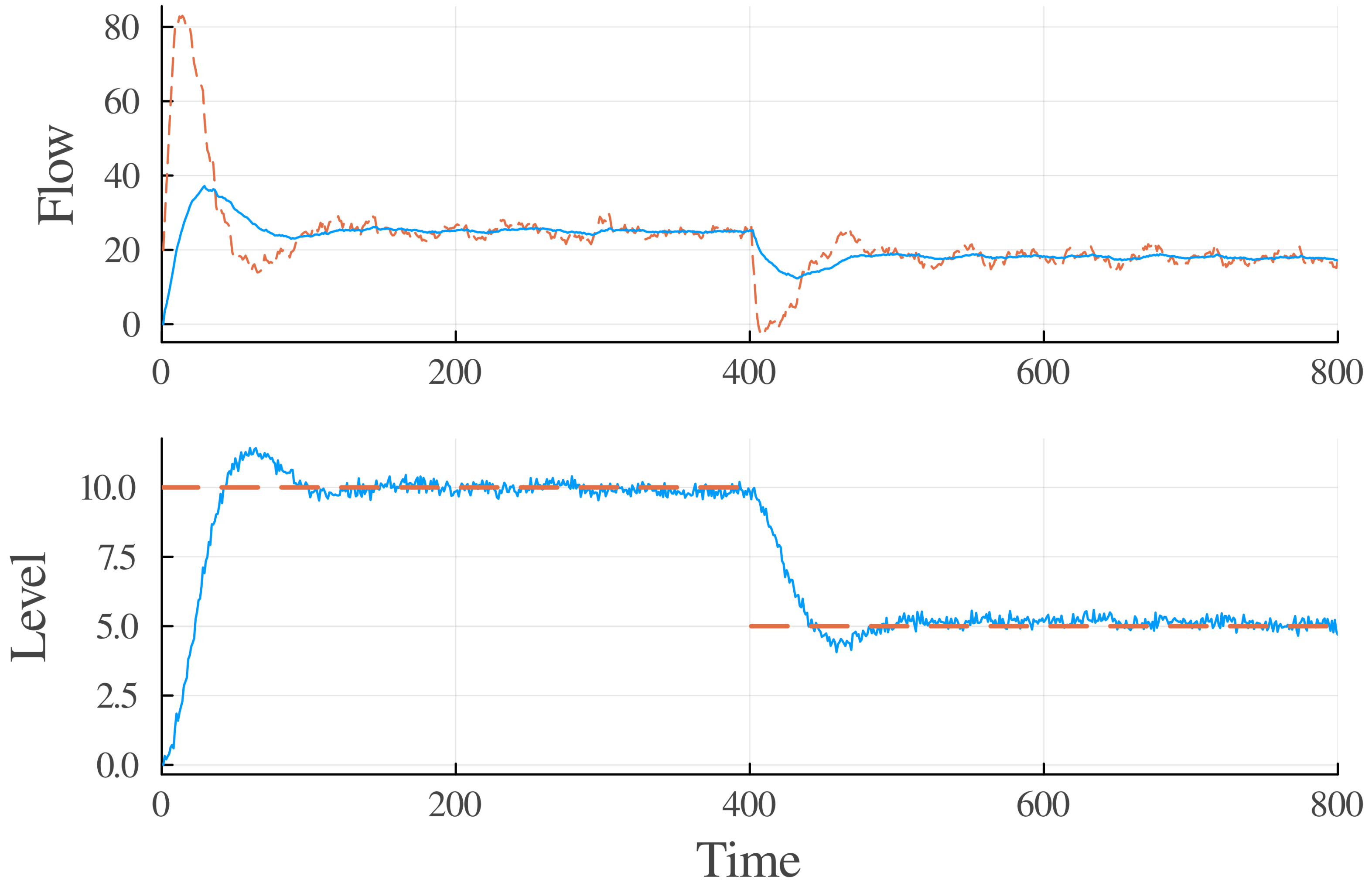
Decouples learning and stability

Industrial example



Astrom, K., and A-B. Ostberg. "A teaching laboratory for process control.", 1986





- RL agent manipulates Q parameter
- End-to-end stable learning with DNN based control
  - Stable during and after training without loss in performance

Conclusions

- Constant advances in deep RL push the boundaries of what is possible
- This success is often misaligned with industrial priorities
    - Performance is not the only metric
- We aim to preserve flexibility of general learning algorithms & maintain key system features

https://process.honeywell.com/us/en/industries/sheet-manufacturing/pulp-and-paper

References

- Willems, Jan C., et al. “A note on persistency of excitation.” 2005.
- Anderson, Brian DO. “From Youla–Kucera to identification, adaptive and nonlinear control.” 1998.
- See also: Lawrence, Nathan P. “Deep reinforcement learning agents for industrial control system design.” Electronic Theses and Dissertations, University of British Columbia. 2023.

doi:http://dx.doi.org/10.14288/1.0430547.

Samariá Gorge, Crete

Questions?

Lawrence@math.ubc.ca