Closing the loop between optimal nonlinear control and learning-based optimization

Luca Furieri

Swiss National Science Foundation

EPFL

UNIVERSITY OF OXFORD

Neural network control

Success stories in robotics

[Kalashnikov *et al.*, ‘18]

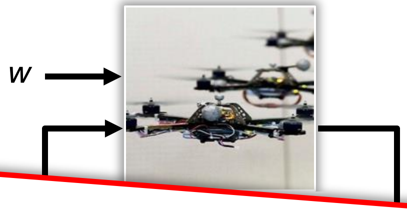
[Youssef *et al.*, ‘20]

[Kaufmann *et al.*, ‘23]

- Flexibility of NN controllers, optimization of complex objective functions

Two challenges

1. Certify closed-loop stability <u>during</u> the learning?

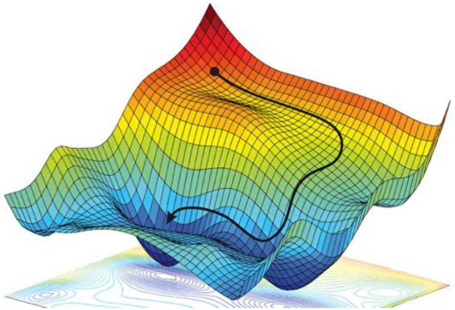


Goal: stability for all choices of θ

Unified methodology

From designing **stabilizing** controllers… to designing **stable** closed-loop maps

2. Navigate the highly nonconvex optimization landscape



Goal: converge, fast, to good (local) solution

Part 1

Stable NN closed-loop maps for nonlinear optimal control

[1] Furieri L., Galimberti C., Ferrari-Trecate G, "*Neural system level synthesis: learning over all stabilizing policies for nonlinear systems*", [CDC 2022]
[2] Furieri L., Galimberti C., Ferrari-Trecate G., "*Learning to boost the performance of stable nonlinear systems*", [OJ-CSYS, 2024]

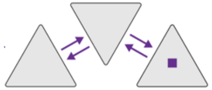
Common scenario in engineering

- … frequent availability of stabilizing controllers around equilibrium or a reference

… however, stability is not enough

Modular “origami” robot[1]

Distributed PID control



8x speed

All-terrain legged-robots[2]

>4x speed

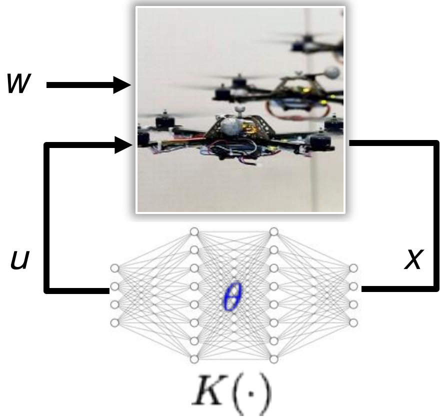
Boost performance without compromising stability?

[1] Wisth, Camurri, Fallon, “*VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots*." IEEE Transactions on Robotics, 2022
[2] Belke, Holdcroft, Sigrist, Paik, *"Morphological flexibility in robotic systems through physical polygon meshing."* Nature Machine Intelligence, 2023

Performance boosting

System

- Nonlinear, stable/pre-stabilized



Performance-boosting controller

- Stability-preserving, optimizing complex costs
- Performance = task execution, safety, robustness, …

Nonlinear Optimal Control

$$K(\cdot) \in \operatorname{argmin} \frac{1}{T}\mathbb{E}_w\left[\mathcal{L}(x_{0:T}, u_{0:T})\right]$$

s.t. CLOSED-LOOP STABILITY

Goals

- Leverage NNs flexibility
- Harness open-loop stability for control design

Setup and notation

Time-varying, nonlinear, controlled system

$$\begin{cases} x_t = f_t(x_{t-1}, u_{t-1}) + w_t \\ u_t = K_t(x_{t:0}) \end{cases}$$

Setup and notation

Time-varying, nonlinear, controlled system

$$\begin{cases} x_t = f_t(x_{t-1}, u_{t-1}) + w_t \\ u_t = K_t(x_{t:0}) \end{cases}$$

Process noise

Dynamic controller

Setup and notation

Time-varying, nonlinear, controlled system

$$\begin{cases} x_t = f_t(x_{t-1}, u_{t-1}) + w_t \\ u_t = K_t(x_{t:0}) \end{cases}$$

$$\xrightarrow[\mathbf{x} = (x_0, x_1, \dots)]{\mathbf{K}(\mathbf{x}) = (K_0(x_0), K_1(x_{1:0}), \dots)}$$

Operator model

$$\begin{cases} \mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{u} = \mathbf{K}(\mathbf{x}) \end{cases}$$

Setup and notation

Time-varying, nonlinear, controlled system

$$\begin{cases} x_t = f_t(x_{t-1}, u_{t-1}) + w_t \\ u_t = K_t(x_{t:0}) \end{cases}$$

$$\mathbf{K}(\mathbf{x}) = (K_0(x_0), K_1(x_{1:0}), \ldots) \quad \longrightarrow \quad \mathbf{x} = (x_0, x_1, \ldots)$$

Operator model

$$\begin{cases} \mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{u} = \mathbf{K}(\mathbf{x}) \end{cases}$$

LTI system: $x_t = Ax_{t-1} + Bu_{t-1} + w_t$

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots \\ A & 0 & 0 & \cdots \\ 0 & A & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \cdots \\ B & 0 & 0 & \cdots \\ 0 & B & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} x_0 \\ w_1 \\ w_2 \\ \vdots \end{bmatrix}$$

Setup and notation

Time-varying, nonlinear, controlled system

$$\begin{cases} x_t = f_t(x_{t-1}, u_{t-1}) + w_t \\ u_t = K_t(x_{t:0}) \end{cases} \xrightarrow[\mathbf{x} = (x_0, x_1, \ldots)]{\mathbf{K}(\mathbf{x}) = (K_0(x_0), K_1(x_{1:0}), \ldots)} $$

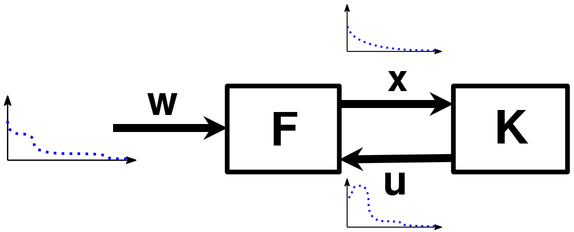
Operator model

$$\begin{cases} \mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{u} = \mathbf{K}(\mathbf{x}) \end{cases}$$

Stability in the $\mathcal{L}_2$ sense

- **A** is a **stable operator** if it is causal and $\mathbf{A}(\mathbf{x}) \in \ell_2, \forall \mathbf{x} \in \ell_2$ ← - - $\mathbf{x} \in \ell_2$ if $\sum_{t=0}^{\infty} \|x_t\|^2 < \infty$
  - For short: $\mathbf{A} \in \mathcal{L}_2$

- **Stabilizing controller**: the closed-loop maps $\mathbf{w} \to \mathbf{x}$ and $\mathbf{w} \to \mathbf{u}$ are stable operators



Setup and notation

Time-varying, nonlinear, controlled system

$$\begin{cases} x_t = f_t(x_{t-1}, u_{t-1}) + w_t \\ u_t = K_t(x_{t:0}) \end{cases} \xrightarrow[\mathbf{x} = (x_0, x_1, \ldots)]{\mathbf{K}(\mathbf{x}) = (K_0(x_0), K_1(x_{1:0}), \ldots)}$$

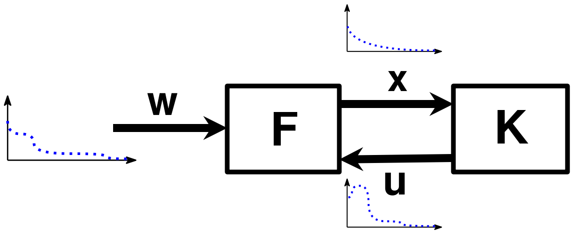
Operator model

$$\begin{cases} \mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{u} = \mathbf{K}(\mathbf{x}) \end{cases}$$

Stability in the $\mathcal{L}_2$ sense

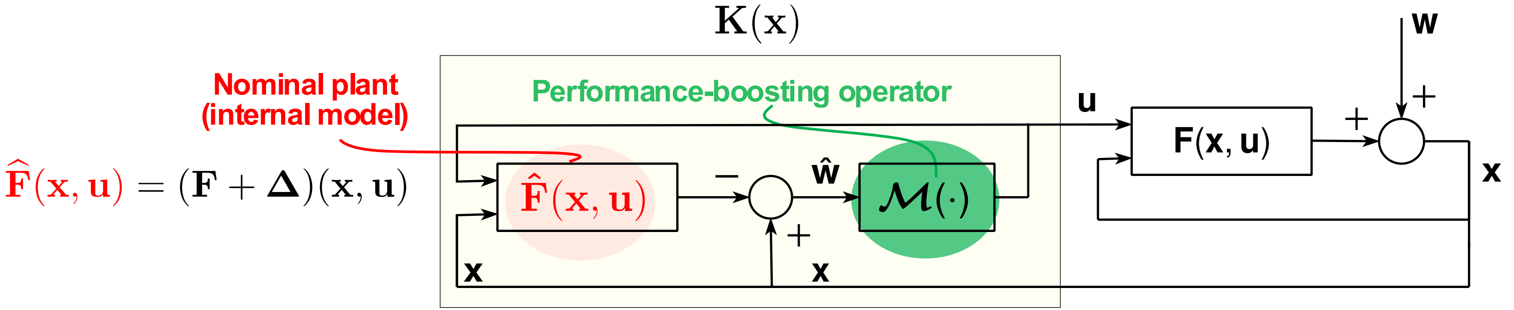
- **A** is a **stable operator** if it is causal and $\mathbf{A}(\mathbf{x}) \in \ell_2, \forall \mathbf{x} \in \ell_2$ ← - - $\mathbf{x} \in \ell_2$ if $\sum_{t=0}^{\infty} \|x_t\|^2 < \infty$
  - For short: $\mathbf{A} \in \mathcal{L}_2$

- **Stabilizing controller**: the closed-loop maps $\mathbf{w} \rightarrow \mathbf{x}$ and $\mathbf{w} \rightarrow \mathbf{u}$ are stable operators



Assumption: the open-loop plant $\mathbf{x} = \mathcal{F}(\mathbf{u}, \mathbf{w})$ is a stable operator

Parametrization of all nonlinear stabilizing controllers



1. **Sufficiency:** the controller $\mathbf{K}(\mathbf{x})$ as above stabilizes the <u>real</u> system $\mathbf{F}(\mathbf{x},\mathbf{u})$ if

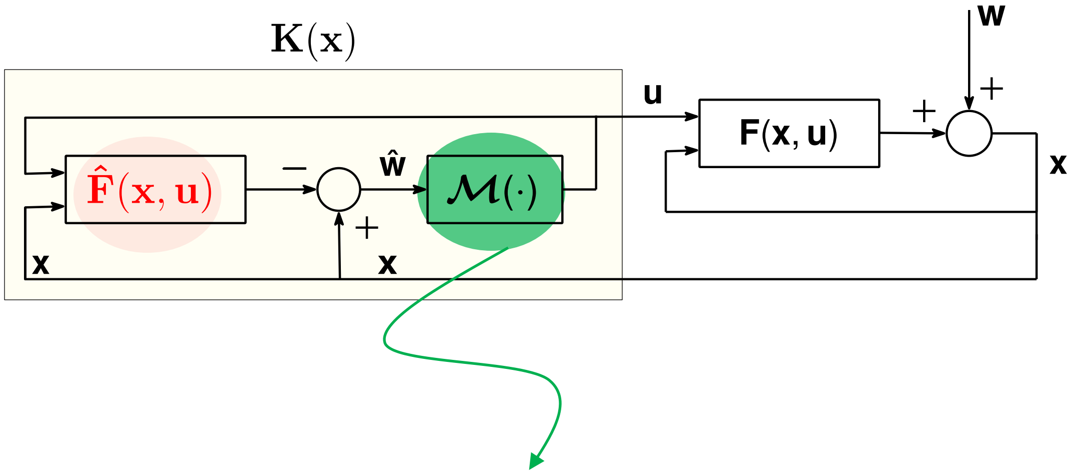
$$\mathcal{M} \in \mathcal{L}_2 \quad \text{and} \quad \text{gain}(\mathcal{M}) < \frac{1}{\text{gain}(\boldsymbol{\Delta})(\text{gain}(\mathcal{F})+1)}$$

2. **Necessity:** if the real model is known $(\boldsymbol{\Delta}=0)$, then $\mathcal{M}$ <u>is</u> the closed-loop map $\mathbf{w} \to \mathbf{u}$

➔ Therefore, <u>*any stable closed-loop behavior can be obtained*</u> by selecting $\mathcal{M} \in \mathcal{L}_2$

[1] L. Furieri, C. L. Galimberti, and GFT, "Neural System Level Synthesis: Learning over All Stabilizing Policies for Nonlinear Systems," IEEE CDC 2022
[2] L. Furieri, C. L. Galimberti, and GFT, "Learning to Boost the Performance of Stable Nonlinear Systems," OJ-CSYS 2024

Next question...



How to implement *neural-network* stable operators?

Models of stable operators

Finite-dimensional parametrization of $\mathcal{M}^\theta \in \mathcal{L}_2$

- Linear operators $\mathcal{M}^\theta = \sum_{h=0}^{N} \frac{M_h}{z^h}$ (Finite Impulse Response models)
- *Nonlinear operators*?

Models of stable operators

Finite-dimensional parametrization of $\mathcal{M}^\theta \in \mathcal{L}_2$

- Linear operators $\mathcal{M}^\theta = \sum_{h=0}^{N} \frac{M_h}{z^h}$ (Finite Impulse Response models)
- *Nonlinear operators*?

EFFICIENTLY MODELING LONG SEQUENCES WITH STRUCTURED STATE SPACES

Albert Gu & Karan Goel & Christopher Ré
Department of Computer Science, Stanford University
{albertgu,krng}@stanford.edu, chrismre@cs.stanford.edu

’22

Standard representation and unified stability analysis for dynamic artificial neural network models

Kwang-Ki K. Kim [a,*], Ernesto Ríos Patrón [b], Richard D. Braatz [c]

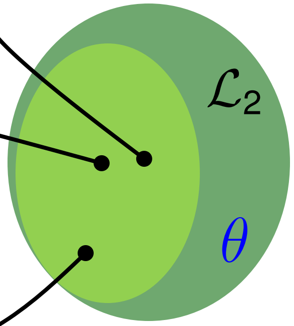
[a] *Department of Electrical Engineering, Inha University, Incheon, Republic of Korea*
[b] *Petroleum Inst of Mexico, Mexico City, Mexico*
[c] *Massachusetts Institute of Technology, Cambridge, MA, United States*

’18, ‘23

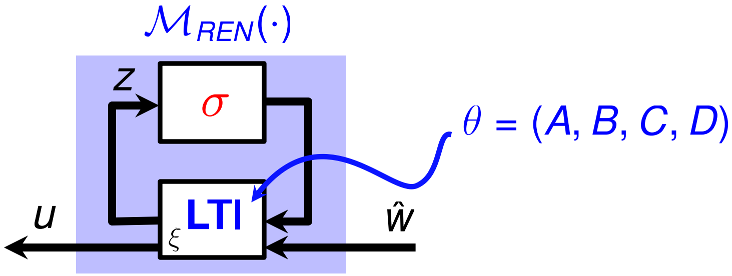
Recurrent Equilibrium Networks:
Flexible Dynamic Models with Guaranteed Stability and Robustness

Max Revay, Ruigang Wang, Ian R. Manchester

’21, ‘23

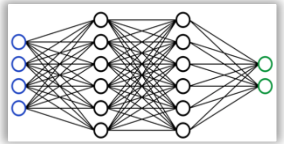


Recurrent Equilibrium Networks (RENs)[1,2]



- Expressive models including

$$\xi_t = \hat{A}\xi_{t-1} + \hat{B}\,\mathrm{NN}^{\xi}(\xi_{t-1}, \hat{w}_t)$$
$$u_t = \hat{C}\xi_t + \hat{D}\,\mathrm{NN}^{u}(\xi_{t-1}, \hat{w}_t)$$



- $\mathcal{M}_{REN} \in \mathcal{L}_2$ if there is a storage function $V(\xi) = \xi^T P \xi$ verifying

$$V(\xi_{t+1}) - V(\xi_t) \leq \gamma^2 \|\hat{w}_t\| - \|u_t\|$$

- **Free parametrization**[2]: explicit map $\Theta \mapsto (\theta, P)$ such that $\mathcal{M}_{REN} \in \mathcal{L}_2$ **for any** $\Theta \in \mathbb{R}^d$
  - **Limitation**: contractive models

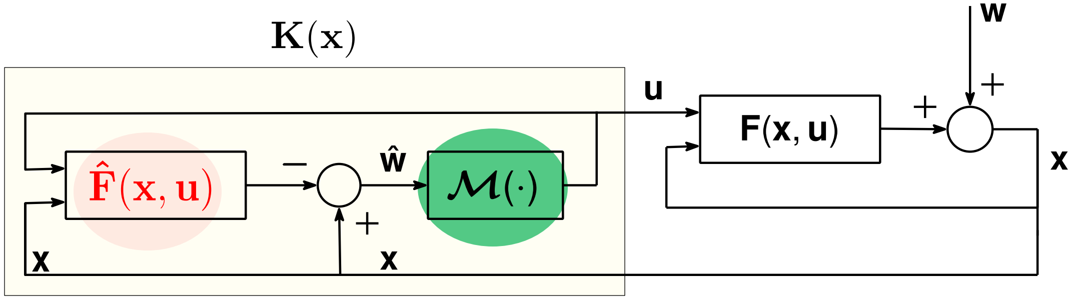
[1] Kim, K. K., E. Ríos Patrón, and R. D. Braatz. "Standard representation and unified stability analysis for dynamic artificial neural network models." *Neural Networks* 2018

[2] Revay, M, R. Wang, and I.R. Manchester. "Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness." *IEEE TAC* 2023

Deep learning formulation

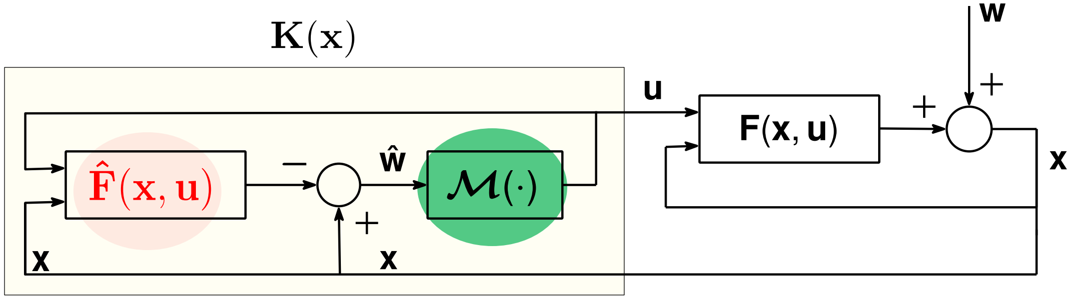
$$K(\cdot) \in \operatorname{argmin} \frac{1}{T}\mathbb{E}_w\left[\mathcal{L}(x_{0:T}, u_{0:T})\right]$$

s. t. CLOSED-LOOP STABILITY



Deep learning formulation

$$K(\cdot) \in \operatorname{argmin} \frac{1}{T}\mathbb{E}_w\left[\mathcal{L}(x_{0:T}, u_{0:T})\right]$$
$$\text{s. t.} \quad \text{CLOSED-LOOP STABILITY}$$

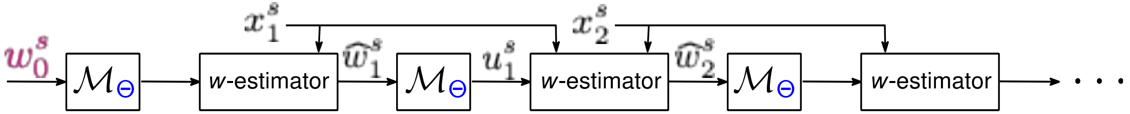


Rollout in time

$$\min_{\Theta \in \mathbb{R}^d} \frac{1}{S}\sum_{s=1}^{S} \mathcal{L}(x^s_{0:T}, u^s_{0:T})$$

subject to

$$x^s_t = \widehat{f}_t(x^s_{t-1}, u^s_{t-1}) + w^s_t, \quad x^s_0 = w^s_0$$



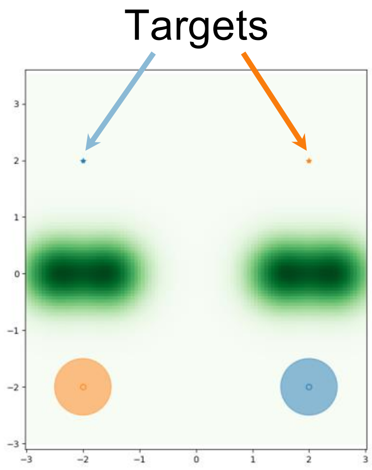
- Free parametrization of $\mathcal{M}$ $\rightarrow$ unconstrained optimization $\rightarrow$ backprop
- CL stability guaranteed even if optimization stops early

TensorFlow

PyTorch

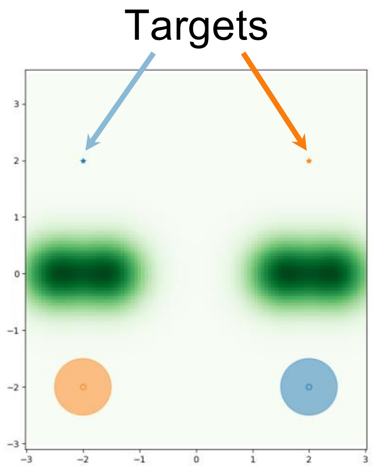
Numerical example: the corridor problem

- 2 robots: point-mass dynamics, nonlinear drag
- **Goal**: CL stability on targets, avoid collisions & obstacles

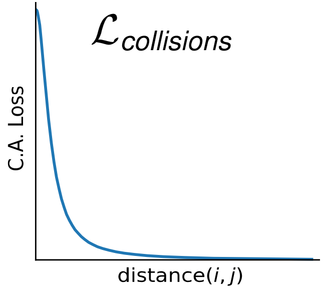


Numerical example: the corridor problem

- 2 robots: point-mass dynamics, nonlinear drag
- **Goal**: CL stability on targets, avoid collisions & obstacles
- **Separation of concerns:**
  1. Design a simple stabilizing base controller
     - Linear spring at rest on target (overshoot, collisions….)
  2. Performance-boosting controller minimizing

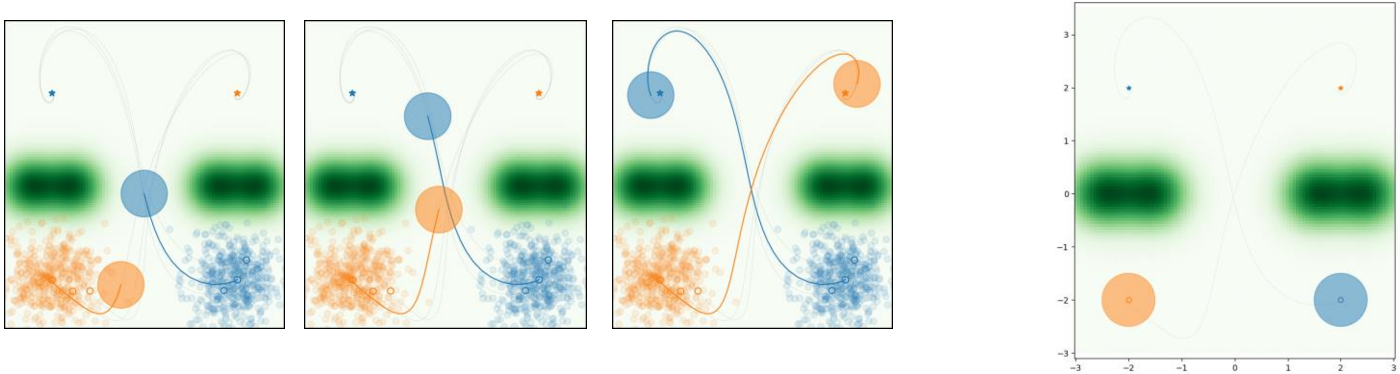


$$\mathcal{L}(\cdot) = \mathcal{L}_{target}(\cdot) + \mathcal{L}_{collisions}(\cdot) + \mathcal{L}_{obstacles}(\cdot)$$

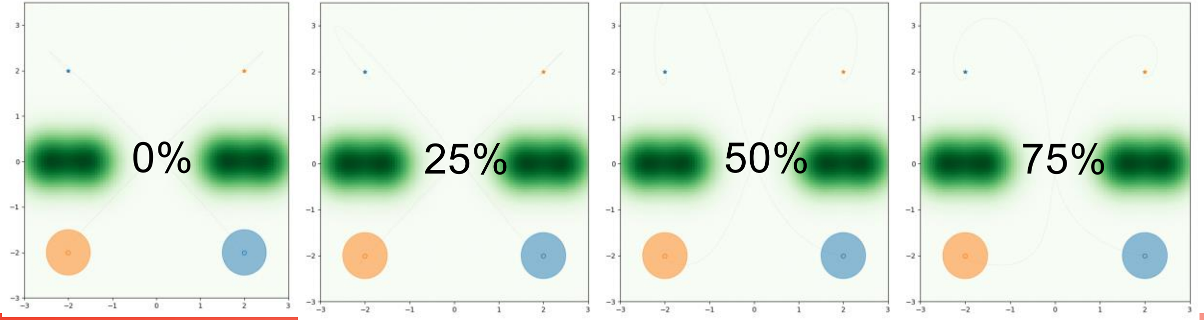


Numerical example: the corridor problem

- Upon training over a dataset 500 different initial conditions



- CL stability guaranteed even with early stopping of training



The power of the cost: lessons from RL

- «Reward» shaping does the magic in RL
- Our result: *decoupling* reward from stability

Performance-boosting problem

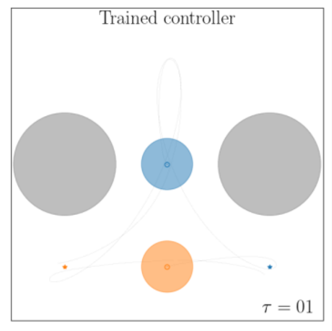
$$\min_{\Theta \in \mathbb{R}^d} \quad \frac{1}{S}\sum_{s=1}^{S}\left[\mathscr{L}(x^s_{T:0}, u^s_{T:0})\right]$$

$$\text{s.t.} \quad x^s_t = f_t(x^s_{t-1}, u^s_{t-1}) + w^s_t, \qquad x^s_0 = w^s_0$$

$$u^s_t = \mathcal{M}^{\Theta}_t\left(x^s_t - f_t(x^s_{t-1}, u^s_{t-1})\right)$$

The power of the cost: lessons from RL

Waypoint tracking



- «Reward» shaping does the magic in RL
- Our result: *decoupling* reward from stability

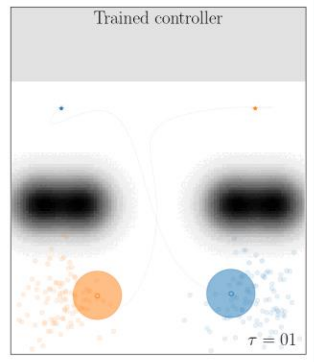
Performance-boosting problem

$$\min_{\Theta \in \mathbb{R}^d} \quad \frac{1}{S} \sum_{s=1}^{S} [\mathscr{L}(x^s_{T:0}, u^s_{T:0})]$$

$$\text{s.t.} \quad x_t^s = f_t(x^s_{t-1}, u^s_{t-1}) + w_t^s, \qquad x_0^s = w_0^s$$

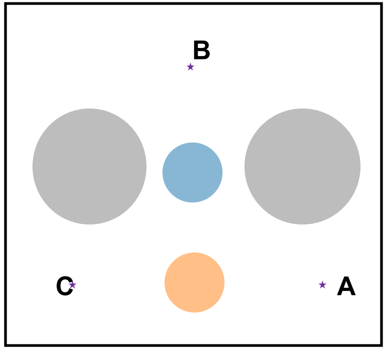
$$u_t^s = \mathcal{M}_t^{\Theta}\left(x_t^s - f_t(x^s_{t-1}, u^s_{t-1})\right)$$

Safety via invariance



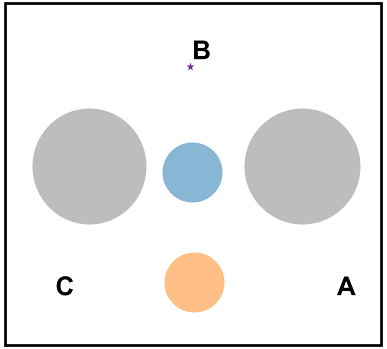
Waypoints tracking

- **Task specs**:
  - No collisions
  - Blue robot: $A \rightarrow B \rightarrow C$, stabilizing around C
  - Orange robot: $C \rightarrow A \rightarrow B$, stabilizing around B



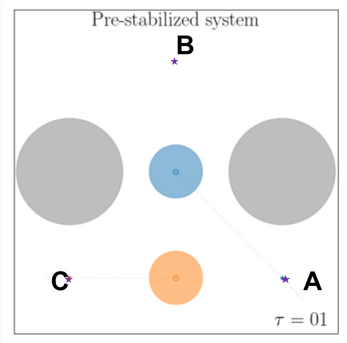
Numerical example: waypoints tracking

- **Task specs**:
  - No collisions
  - Blue robot: $B \to C \to A$, stabilizing around A
  - Orange robot: $A \to B \to C$, stabilizing around C

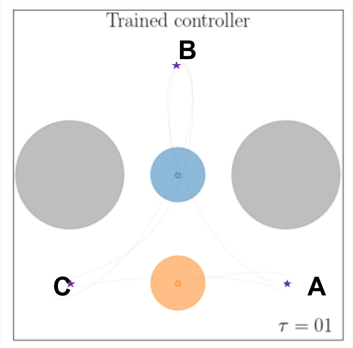


- Waypoints $\to$ **Linear Temporal Logic** formulae[1] $\to$ cost $\mathcal{L}_{way}$

Base controller

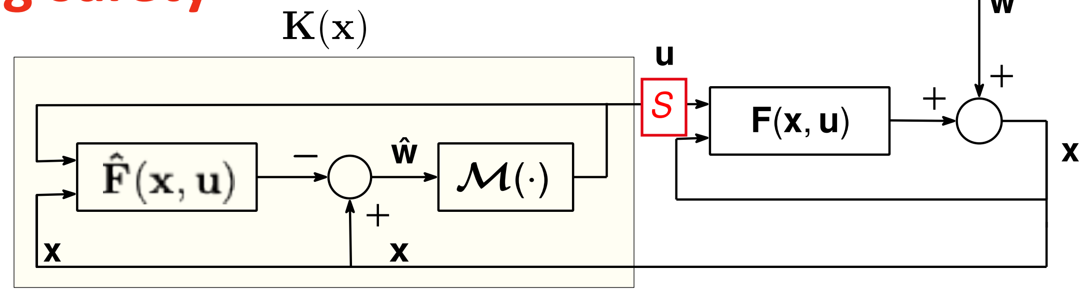


Performance boosting



[1] Li, X., C.-I. Vasile, and C. Belta. "Reinforcement learning with temporal logic rewards.", IEEE IROS, 2017

Embedding safety

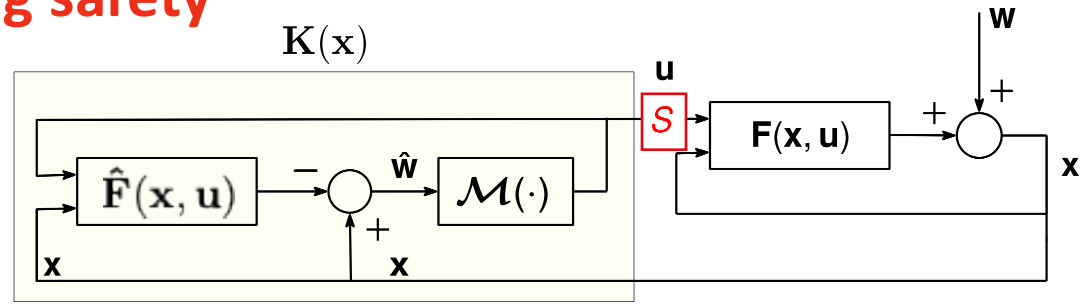


- Add a safety filter[1] guaranteeing $(x_t, u_t) \in \mathcal{C}, \forall t > 0$
  - Requires online optimization
  - Tweaks **u** only if needed

[1] Hewing, L., *et al.* "Learning-based model predictive control: Toward safe learning in control." Annual Review of Control, Robotics, and Autonomous Systems, 2020

[2] Agrawal, A., and K. Sreenath. "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." *Robotics: Science and Systems*. 2017

Embedding safety



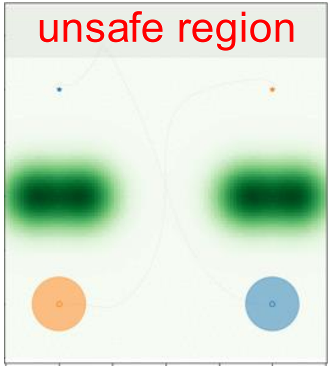
- Add a safety filter[1] guaranteeing $(x_t, u_t) \in \mathcal{C}, \forall t > 0$
  - Requires online optimization
  - Tweaks **u** only if needed
- **Reduce filter activation** embedding **soft safety specs** in the cost
  - Promote constraint fulfillment $\rightarrow \mathcal{L}_{safe} = \max_{t<T} Barrier_{\mathcal{C}}(x_t, u_t)$
  - Promote invariance[2] of $\mathcal{X} = \{x : h(x) \leq 0\}$

$$\mathcal{L}_{inv} = \max_{t<T} \mathrm{ReLU}\left(h(x_t) - h(x_{t+1}) - \gamma h(x_t)\right)$$

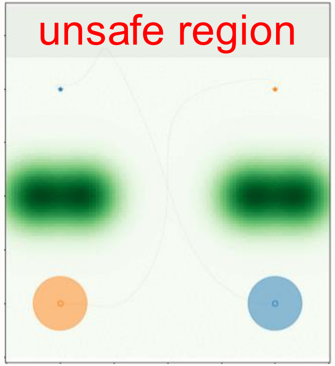
[1] Hewing, L., *et al.* "Learning-based model predictive control: Toward safe learning in control." Annual Review of Control, Robotics, and Autonomous Systems, 2020

[2] Agrawal, A., and K. Sreenath. "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." *Robotics: Science and Systems*. 2017

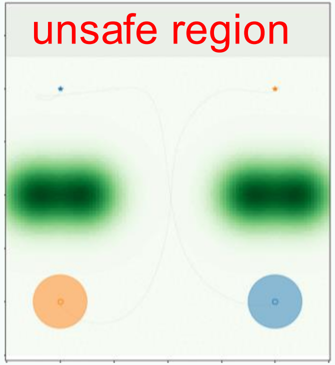
Numerical example: the safe corridor problem



$\mathcal{L}$ **without** safety-promoting terms
Average violation: 43%



$\mathcal{L}$ **including** $\mathcal{L}_{inv}$
Average violation: 1.4%



Not in this talk: extensions

- Interpretation as nonlinear System Level Synthesis and Youla parametrizations[1,2]

- The output-feedback case[2]
    - Equivalent to *internal model control*, first proof of necessity

- Applications to distributed system identification and control
    - Interconnected stable operators + dissipativity arguments[3,4]

[1] Furieri L., Galimberti C., Ferrari-Trecate G, "*Neural system level synthesis: learning over all stabilizing policies for nonlinear systems*", [CDC 2022]

[2] Galimberti C., Furieri L., Ferrari-Trecate G., "*Parametrizations of All Stable Closed-loop Responses: From Theory to Neural Network Control Design*", [Arxiv, 2025]

[3] Massai L., Saccani D., Furieri L., Ferrari-Trecate G., «Unconstrained learning of networked nonlinear systems via free parametrization of stable interconnected operators», [ECC 2024]

[4] Massai L., Saccani D., Furieri L., Ferrari-Trecate G., «Optimal distributed control with stability guarantees by training a network of neural closed-loop maps», [CDC 2024]

From designing ***stabilizing*** policies... to designing ***stable*** closed-loop operators

Part 2: … Is this shift in perspective useful for nonconvex optimization?