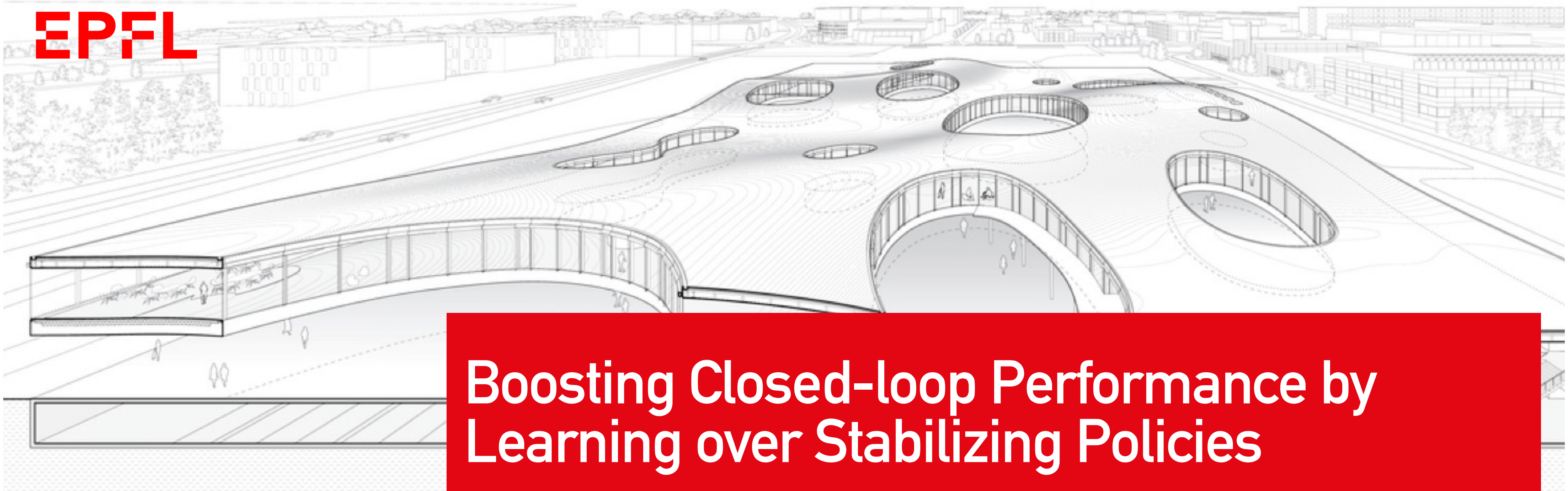


Boosting Closed-loop Performance by Learning over Stabilizing Policies

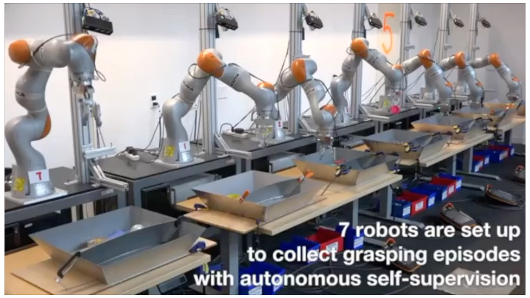
Giancarlo Ferrari Trecate & Luca Furieri
DECODE group, EPFL, Switzerland

École polytechnique fédérale de Lausanne
Swiss National Science Foundation
NCCR Automation

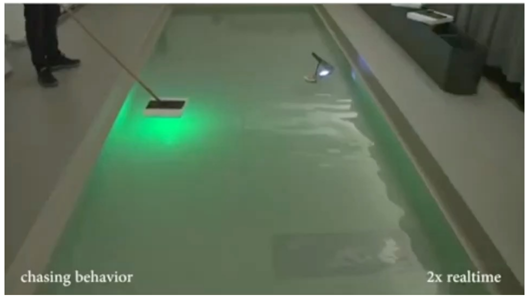
EPFL

Neural network control

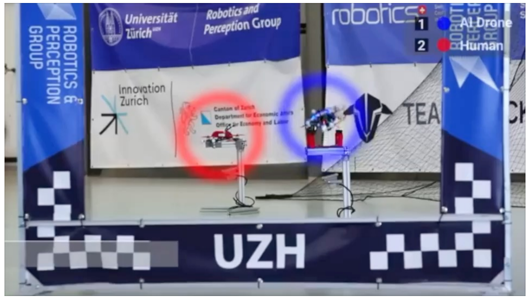
Success stories in robotics



[Kalashnikov *et al.*, ‘18]



[Youssef *et al.*, ‘20]



[Kaufmann *et al.*, ‘23]

- Flexibility of NN controllers, optimization of complex costs
- **Safety and stability guarantees for general NL systems**
  - Model-based: [Richards *et al.*, ‘18], [Chang *et al.*, ‘19], [Dawson *et al.*, ‘23], …
  - Data-driven: [Berkenkamp *et al.*, ‘17], [Recht, ‘18], [Jin & Lavaei, ‘18], …

Common scenario in engineering

Frequent availability of

- System models
- Simple stabilizing controllers around an equilibrium or a reference

G. Ferrari Trecate & L. Furieri

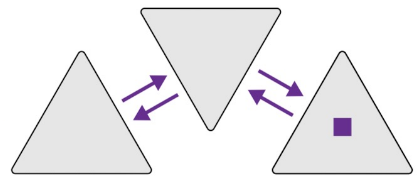
Common scenario in engineering

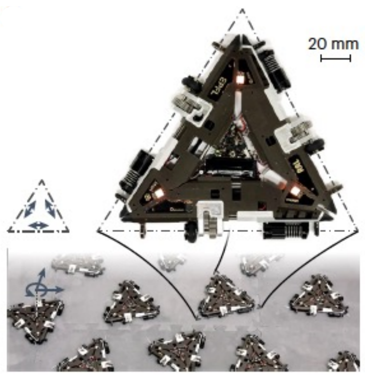
Frequent availability of

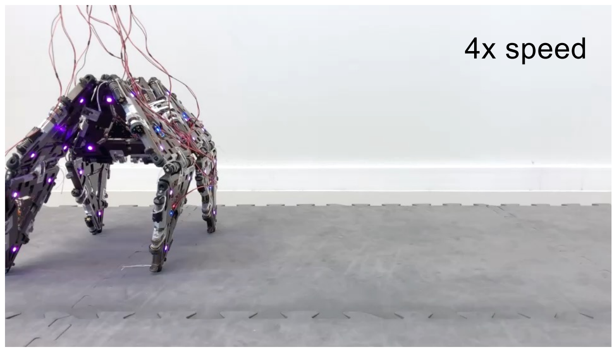
- System models
- Simple stabilizing controllers around an equilibrium or a reference

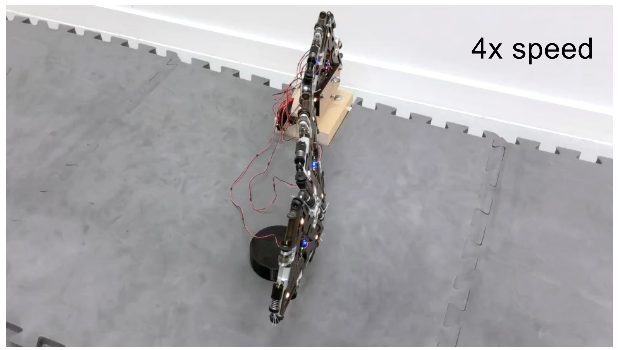
Example: Modular "origami" robot[1]

- Triangular modules that change shape and rotate around joints
- **Polygonal meshes** for several functions









[1] Belke, C.H., *et al.* "Morphological flexibility in robotic systems through physical polygon meshing." *Nature Machine Intelligence,* 2023

Common scenario in engineering

Frequent availability of

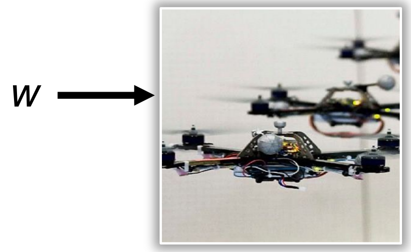
- System models
- Simple stabilizing controllers around an equilibrium or a reference

Improve performance without compromising stability?

Performance boosting

System

- Nonlinear, interconnected, stable/pre-stabilized



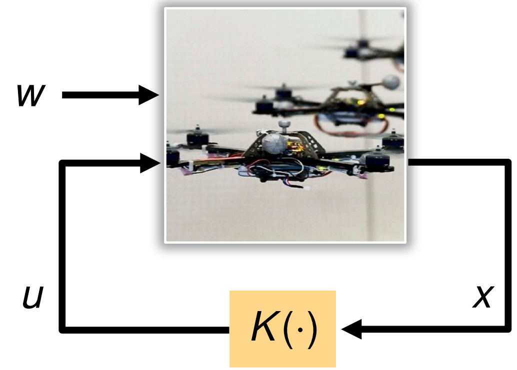
Performance boosting

System

- Nonlinear, interconnected, stable/pre-stabilized

Performance-boosting controller

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



Nonlinear Optimal Control (NOC)

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

s.t. CLOSED-LOOP STABILITY

Performance boosting

System

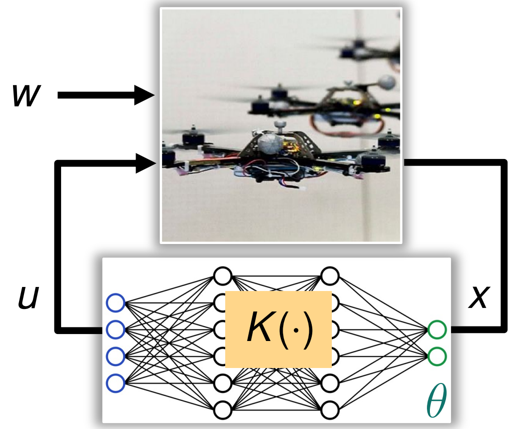
- Nonlinear, interconnected, stable/pre-stabilized

Performance-boosting controller

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

Goals

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



Nonlinear Optimal Control (NOC)

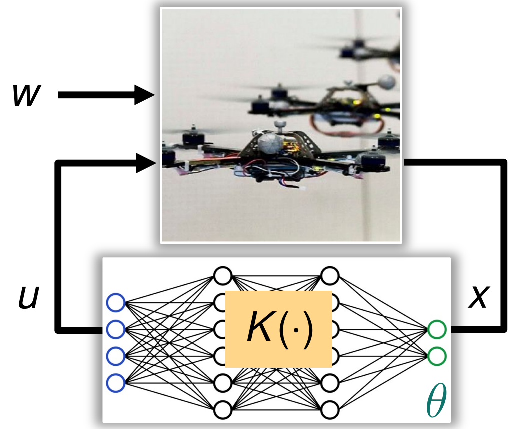
$$K(\cdot) \in \text{argmin } \frac{1}{T}\mathbb{E}_w\left[\mathcal{L}(x_{0:T}, u_{0:T})\right]$$

s.t. CLOSED-LOOP STABILITY

Outline

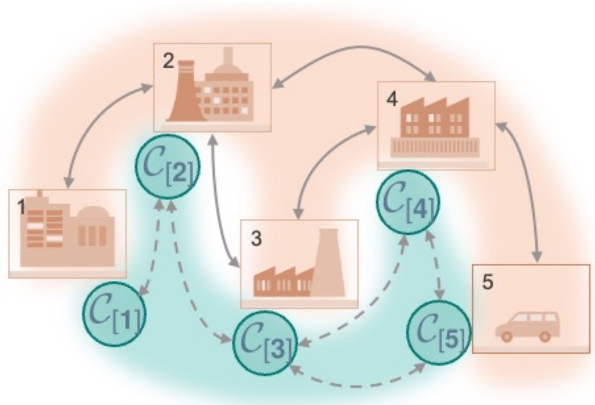
Part 1 (Gianni): Design of performance-boosting policies

- Parametrization of **all stabilizing controllers**
- NN models of stable operators
  - Solving NOC through NN training



Part 2 (Luca): Extensions for real-world deployment

- Tackling the remaining challenges
  - Uncertain models, output feedback, distributed…
- Lessons from RL: *how to shape your cost function*



EPFL

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

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

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

$\longrightarrow$

$\mathbf{x} = (x_0, x_1, \ldots)$

Operator model

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

Setup and notation

Time-varying, nonlinear, controlled system

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

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

$\longrightarrow$

$\mathbf{x} = (x_0, x_1, \dots)$

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

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

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

Operator model

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

$\mathcal{L}_2$-stability

- **A** is a **stable operator** if it is causal and $\mathbf{A}(\mathbf{x}) \in \ell_2, \forall \mathbf{x} \in \ell_2$
  - For short: $\mathbf{A} \in \mathcal{L}_2$

$\mathbf{x} \in \ell_2$ if $\sum_{t=0}^{\infty} \|x_t\|^2 < \infty$

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

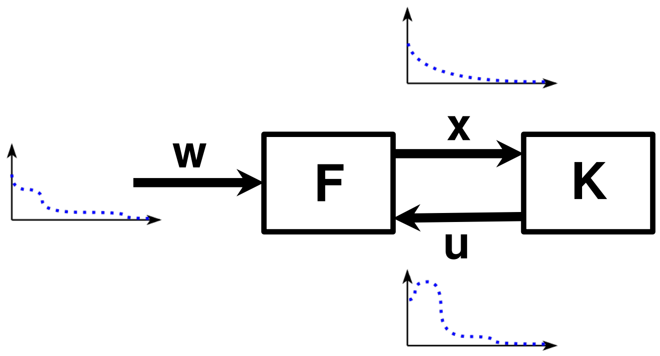
$$\begin{cases} \mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{u} = \mathbf{K}(\mathbf{x}) \end{cases}$$

$\mathcal{L}_2$-stability

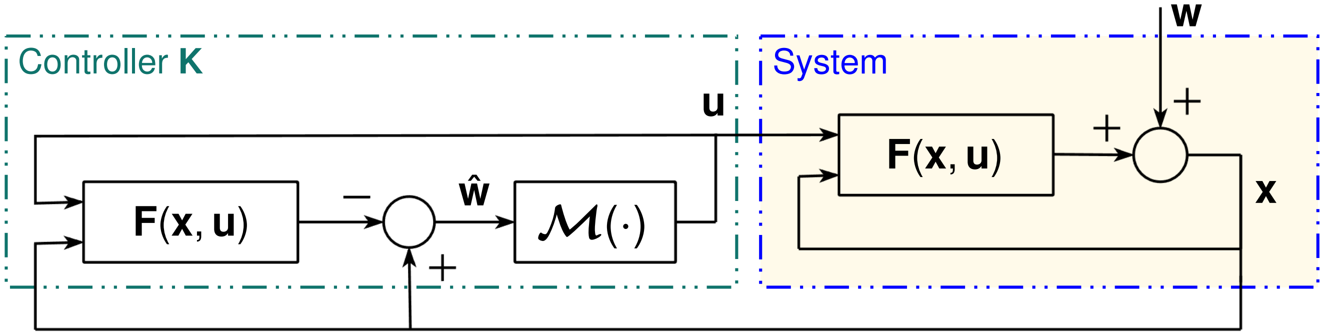
- **A** is a **stable operator** if it is causal and $\mathbf{A}(\mathbf{x}) \in \ell_2, \forall \mathbf{x} \in \ell_2$
  - For short: $\mathbf{A} \in \mathcal{L}_2$

$\mathbf{x} \in \ell_2$ if $\sum_{t=0}^{\infty} \|x_t\|^2 < \infty$

Closed-loop (CL) stability: the operators $\mathbf{w} \rightarrow \mathbf{x}$ and $\mathbf{w} \rightarrow \mathbf{u}$ are stable

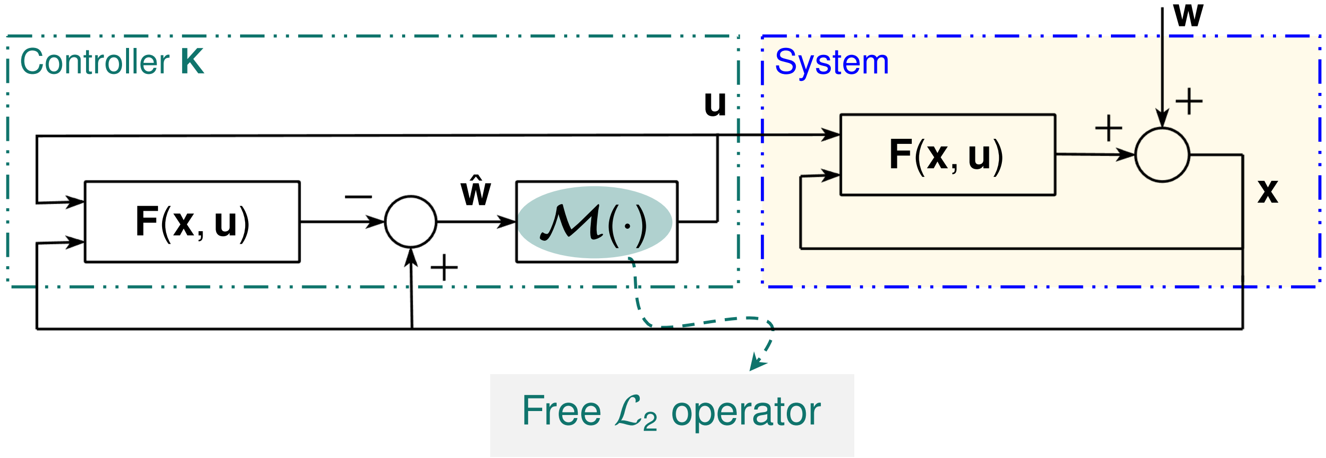


Parametrization of all stabilizing controllers[1,2]



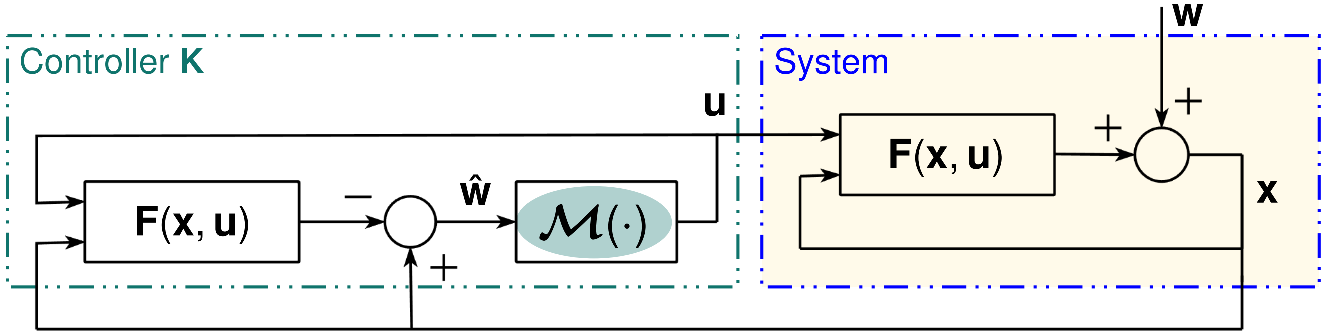
[1] L. Furieri, C. L. Galimberti, and GFT, “Neural System Level Synthesis: Learning over All Stabilizing Policies for Nonlinear Systems,” IEEE CDC 2022
[2] L. Furieri, C. L. Galimberti, and GFT, “Learning to Boost the Performance of Stable Nonlinear Systems,” ArXiv 2024

Parametrization of all stabilizing controllers[1,2]



[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,” ArXiv 2024

Parametrization of all stabilizing controllers[1,2]



Main result

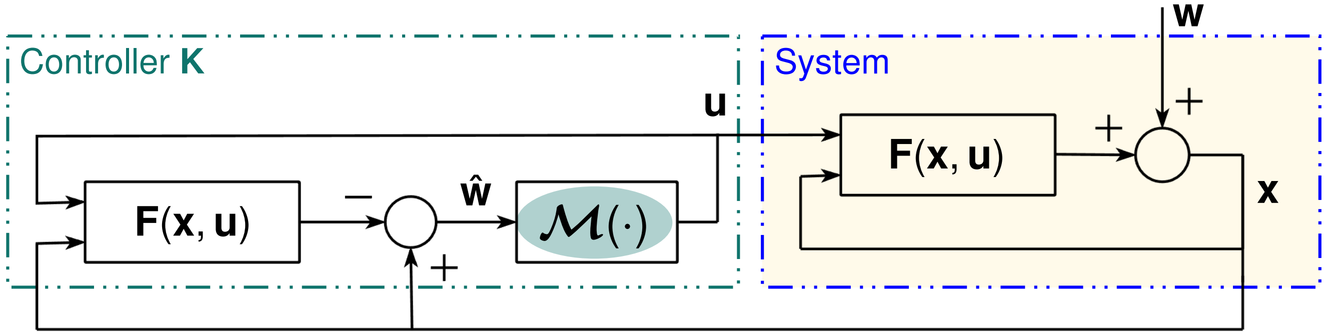
If the open-loop system is stable

(⇒) If $\mathcal{M}(\cdot) \in \mathcal{L}_2$ the CL system is stable

(⇐) If there is $\mathbf{K}'$ providing stable CL operators $\mathbf{w} \to \mathbf{x}$ and $\mathbf{w} \to \mathbf{u}$, then $\exists \mathcal{M}(\cdot) \in \mathcal{L}_2$ providing the same CL operators

[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," ArXiv 2024

Parametrization of all stabilizing controllers[1,2]



Main result

If the open-loop system is stable

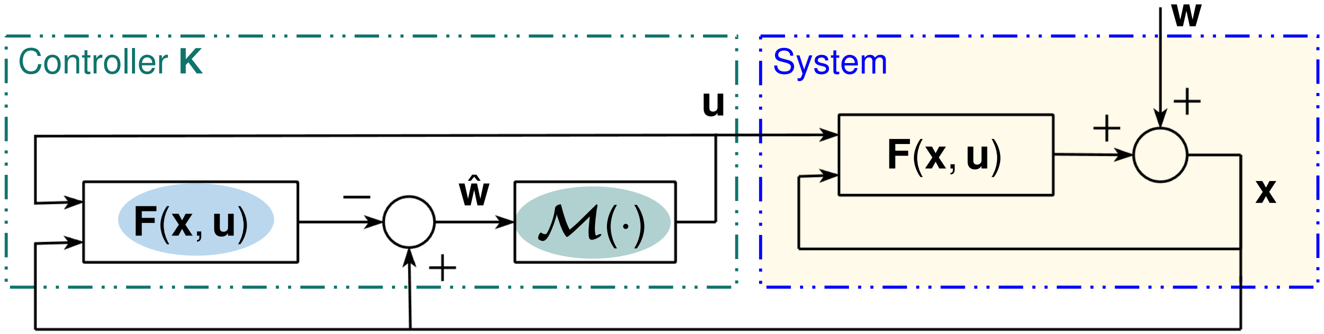
(⇒) If $\mathcal{M}(\cdot) \in \mathcal{L}_2$ the CL system is stable

(⇐) If there is $\mathbf{K}'$ providing stable CL operators $\mathbf{w} \rightarrow \mathbf{x}$ and $\mathbf{w} \rightarrow \mathbf{u}$, then $\exists \mathcal{M}(\cdot) \in \mathcal{L}_2$ providing the same CL operators

Idea behind (⇒): no model mismatch yield $\hat{\mathbf{w}} = \mathbf{w}$, opening the loop

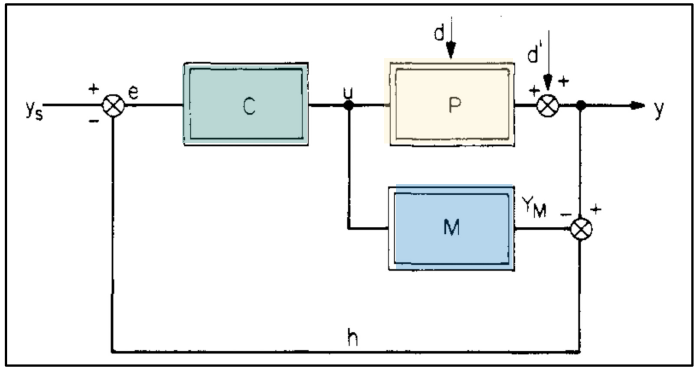
[1] L. Furieri, C. L. Galimberti, and GFT, "Neural System Level Synthesis: Learning over All Stabilizing Policies for Nonlinear Systems," IEEE CDC 2022
[2] L. Furieri, C. L. Galimberti, and GFT, "Learning to Boost the Performance of Stable Nonlinear Systems," ArXiv 2024

IMC and Youla parametrization



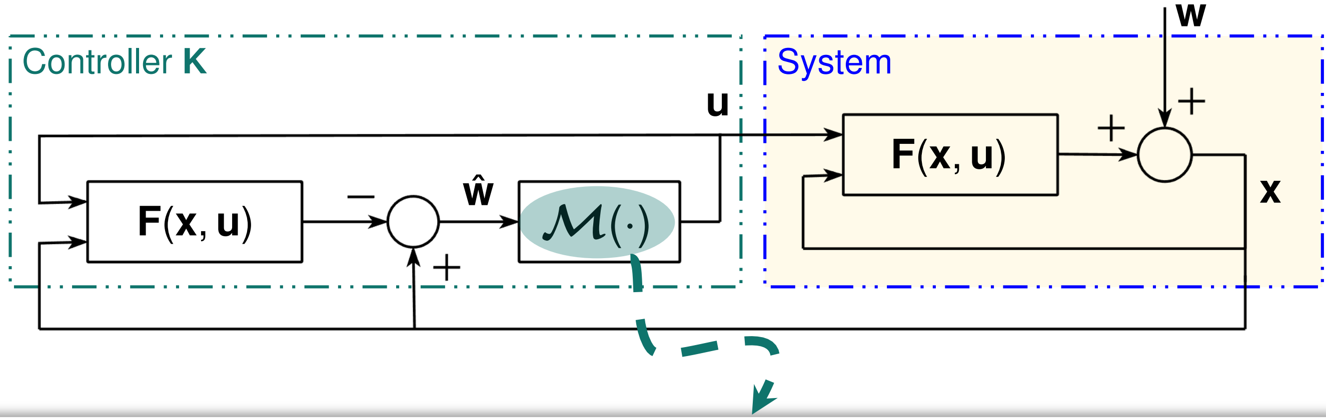
- **Internal Model Control[1,2]**
  - Sufficient for stability[1] if P=M in the I/O setting, also necessary for LTI systems[2]
  - IMC for setpoint tracking [1,2]
    - Problem: C must "invert" the plant [1]
- Nonlinear Youla parametrization[3]

Nonlinear I/O IMC structure from [1]



[1] Economou, C. G., M. Morari, and B. O. Palsson. "Internal model control: Extension to nonlinear system." Industrial & Engineering Chemistry Process Design and Development, 1986
[2] Garcia, C. E., and M. Morari. "Internal model control. A unifying review and some new results." Industrial & Engineering Chemistry Process Design and Development, 1982
[3] C.A. Desoer, R.-W. Liu. "Global parametrization of feedback systems with nonlinear plants", Systems & Control Letters, 1982

Next question



How to implement stable operators?

G. Ferrari Trecate & L. Furieri

Models of stable operators

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

- Linear operators $\mathcal{M}^{\theta} = \sum_{h=0}^{N} \frac{M_h}{z^h}$ (FIR models)

Models of stable operators

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

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

Models of stable operators

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

- Linear operators $\mathcal{M}^\theta = \sum_{h=0}^{N} \frac{M_h}{z^h}$ (FIR 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]

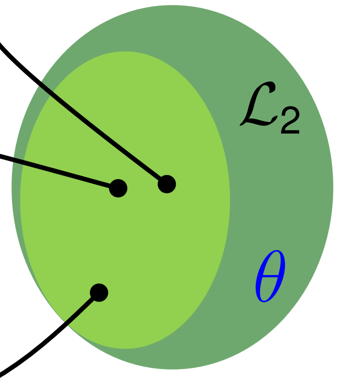
[a] *Department of Electrical Engineering, Inha University, Incheon, Republic of Korea*
[b] *Petroleum Inst of Mexico, Mexico City, Mexico*
[c] *Massachusetts Institute of Technology, Cambridge, MA, United States*

'18, '23

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

Max Revay, Ruigang Wang, Ian R. Manchester

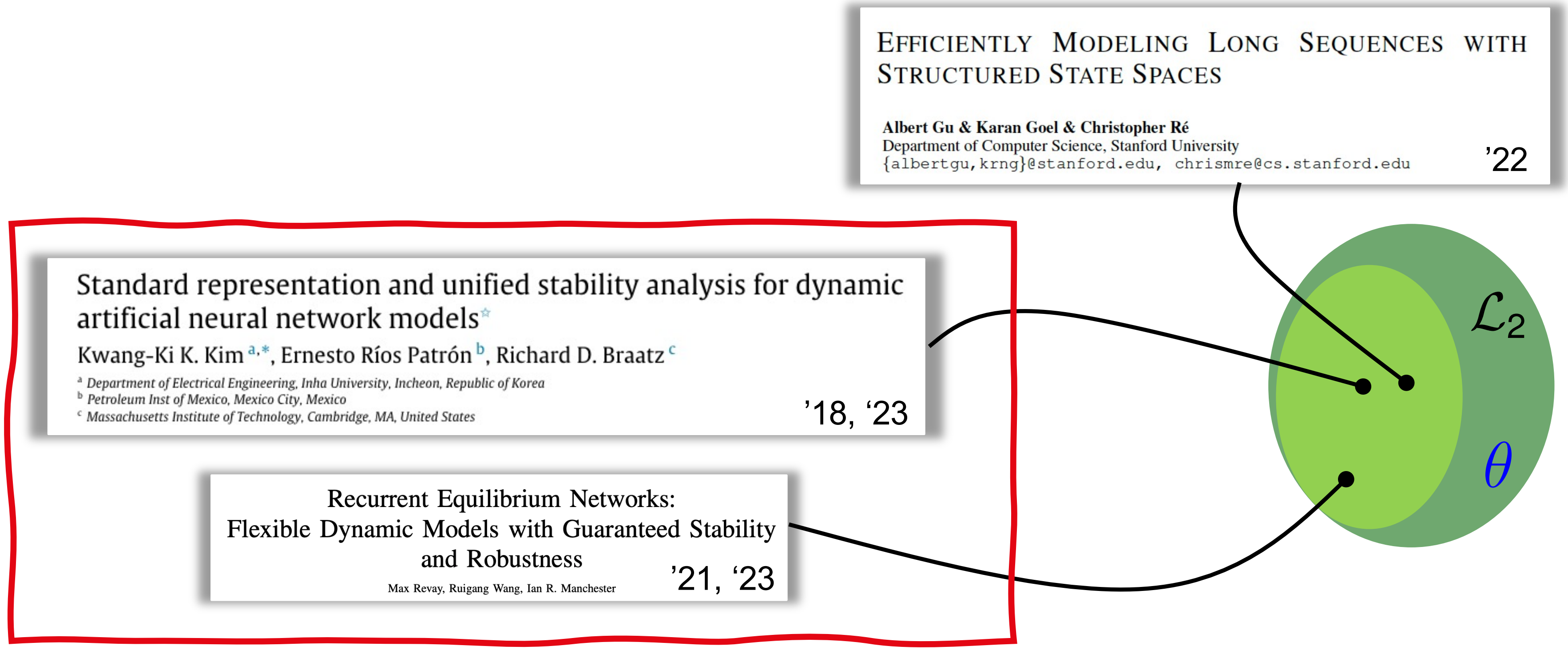
'21, '23



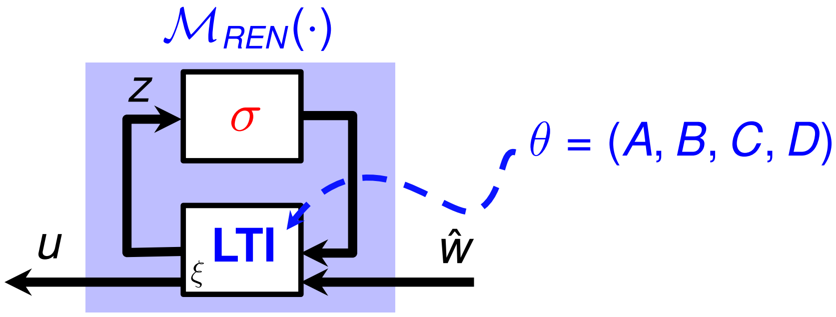
Models of stable operators

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

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



Recurrent Equilibrium Networks (RENs)[1,2]

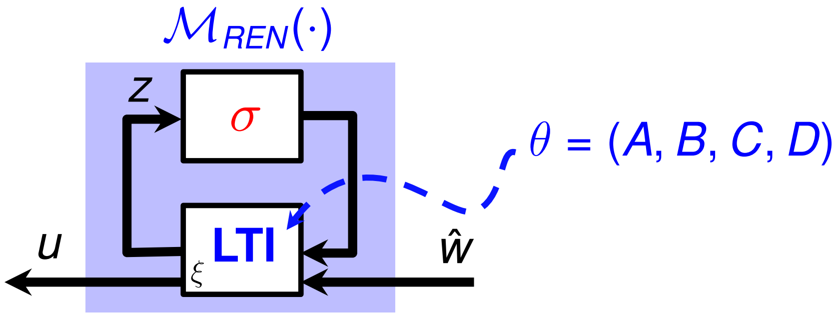


G. Ferrari Trecate & L. Furieri

ECC24 Workshop

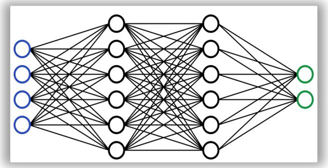
[1] Kim, K. K., E. Ríos Patrón, and R. D. Braatz. "Standard representation and unified stability analysis for dynamic artificial neural network models." *Neural Networks* 2018
[2] Revay, M, R. Wang, and I.R. Manchester. "Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness." *IEEE TAC* 2023

Recurrent Equilibrium Networks (RENs)[1,2]



- Expressive models including

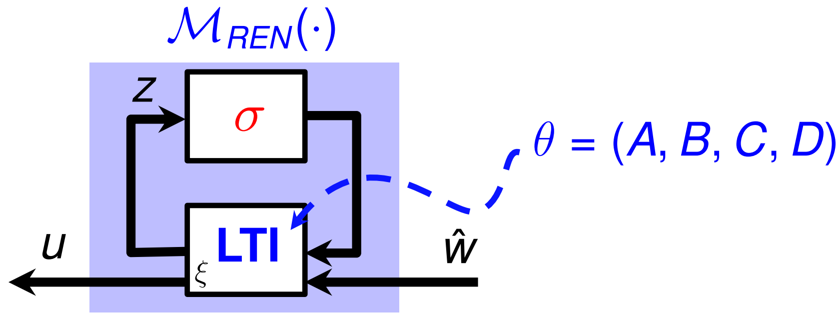
$$\xi_t = \hat{A}\xi_{t-1} + \hat{B}\,\mathrm{NN}^{\xi}(\xi_{t-1}, \hat{w}_t)$$
$$u_t = \hat{C}\xi_t + \hat{D}\,\mathrm{NN}^{u}(\xi_{t-1}, \hat{w}_t)$$



[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

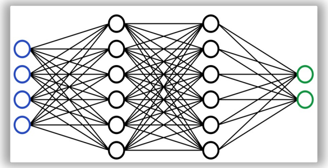
EPFL

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

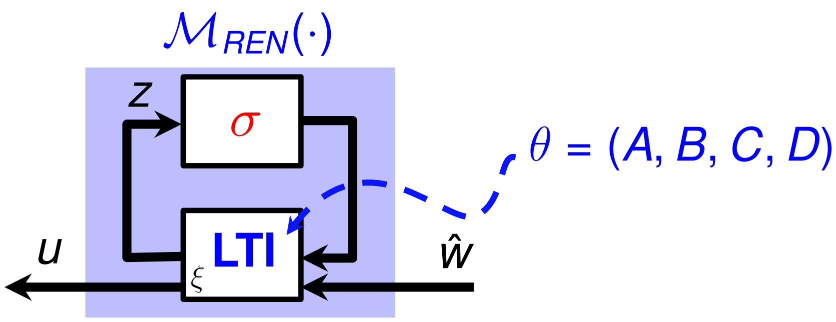
$$V(\xi_{t+1}) - V(\xi_t) \leq \gamma^2 \|\hat{w}_t\| - \|u_t\|$$

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

[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

EPFL

Recurrent Equilibrium Networks (RENs)[1,2]



- Expressive models including

$$\xi_t = \hat{A}\xi \ldots$$
$$u_t = \hat{C} \ldots$$

More details in the next talk!

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

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

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

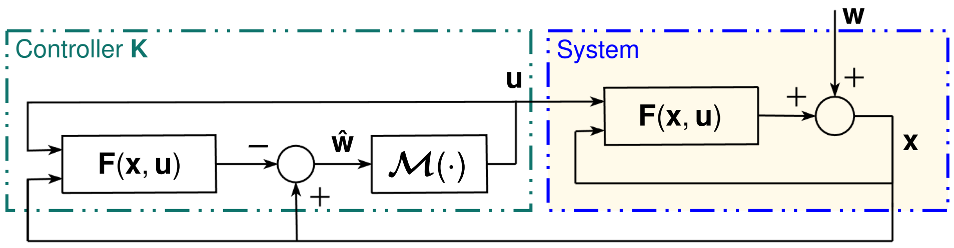
[1] Kim, K. K., E. Ríos Patrón, and R. D. Braatz. "Standard representation and unified stability analysis for dynamic artificial neural network models." *Neural Networks* 2018
[2] Revay, M, R. Wang, and I.R. Manchester. "Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness." *IEEE TAC* 2023

Deep learning for solving NOC[1]

Nonlinear Optimal Control (NOC)

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

s.t. CLOSED-LOOP STABILITY



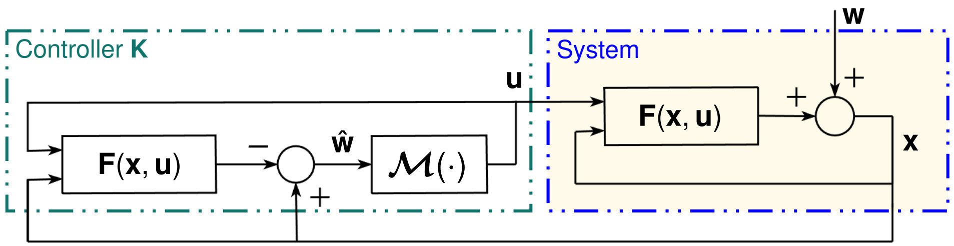
[1] L. Furieri, C. L. Galimberti, and GFT, "Neural System Level Synthesis: Learning over All Stabilizing Policies for Nonlinear Systems," IEEE CDC 2022
[2] L. Furieri, C. L. Galimberti, and GFT, "Learning to Boost the Performance of Stable Nonlinear Systems," ArXiv 2024

Deep learning for solving NOC[1]

Nonlinear Optimal Control (NOC)

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

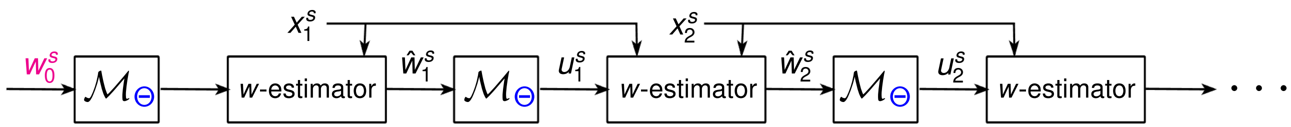
s.t. CLOSED-LOOP STABILITY



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

such that

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



[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," ArXiv 2024

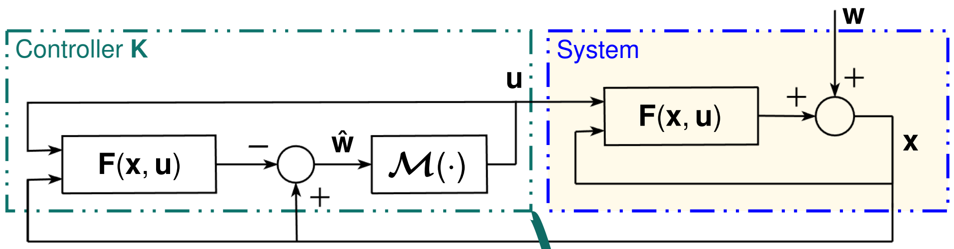
Deep learning for solving NOC[1]

G. Ferrari Trecate & L. Furieri

Nonlinear Optimal Control (NOC)

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

s.t. CLOSED-LOOP STABILITY

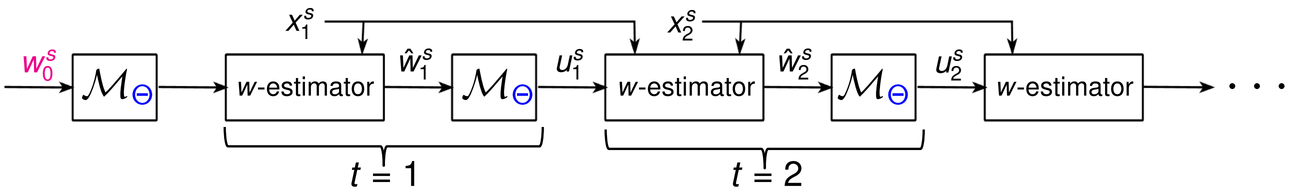


Rollout in time

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

such that

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



[1] L. Furieri, C. L. Galimberti, and GFT, “Neural System Level Synthesis: Learning over All Stabilizing Policies for Nonlinear Systems,” IEEE CDC 2022

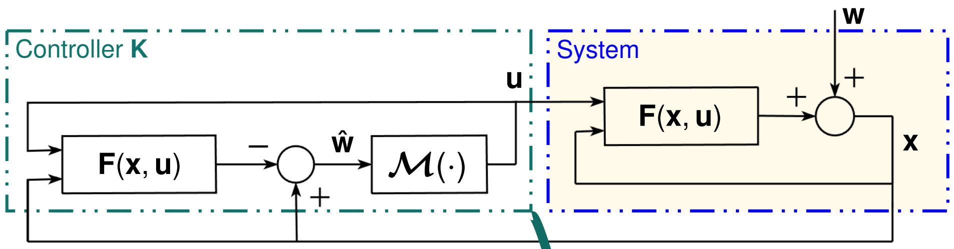
[2] L. Furieri, C. L. Galimberti, and GFT, “Learning to Boost the Performance of Stable Nonlinear Systems,” ArXiv 2024

Deep learning for solving NOC[1]

Nonlinear Optimal Control (NOC)

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

s.t. CLOSED-LOOP STABILITY

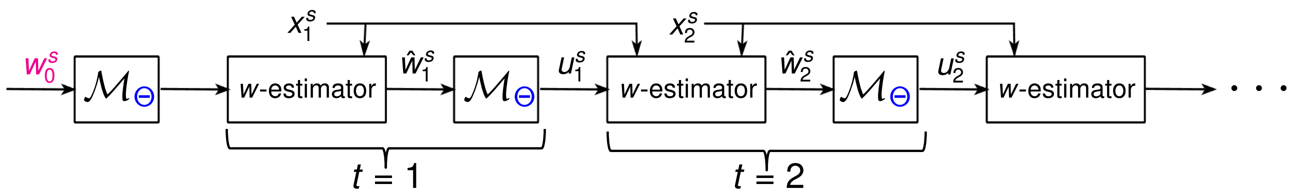


Rollout in time

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

such that

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



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

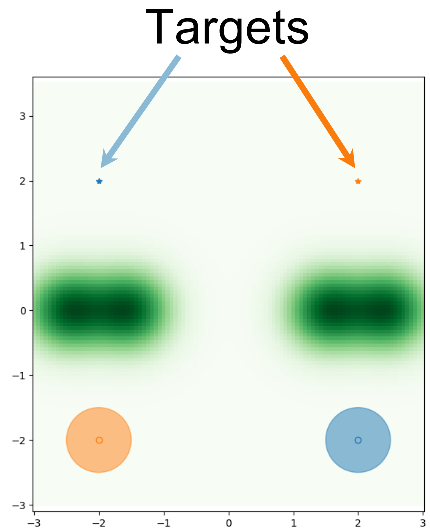
TensorFlow
PyTorch

[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," ArXiv 2024

The corridor problem

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



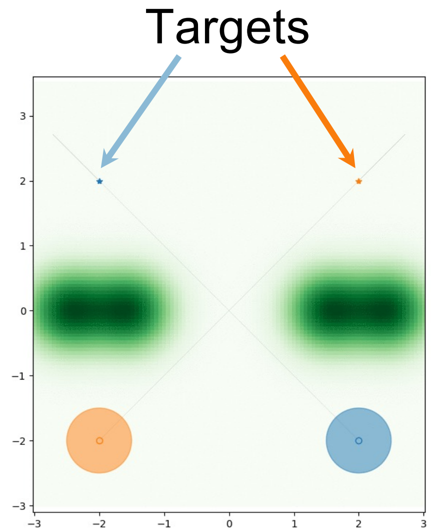
EPFL

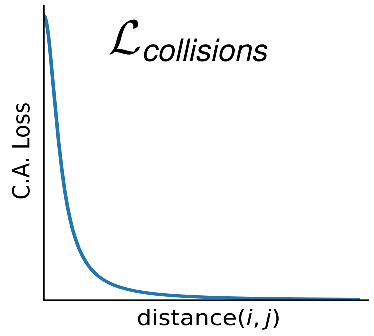
The corridor problem

G. Ferrari Trecate & L. Furieri

- 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

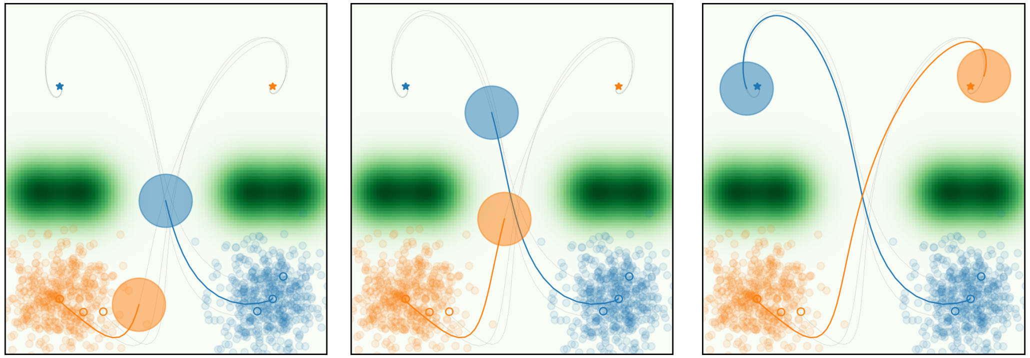
$$\mathcal{L}(\cdot) = \mathcal{L}_{target}(\cdot) + \mathcal{L}_{collisions}(\cdot) + \mathcal{L}_{obstacles}(\cdot)$$

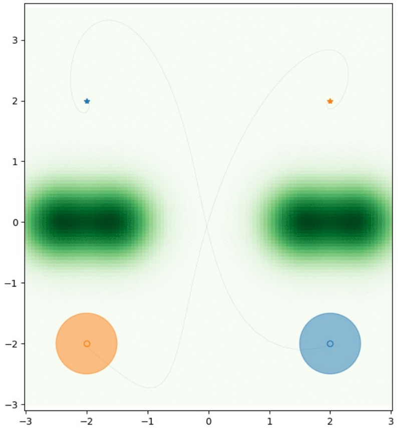




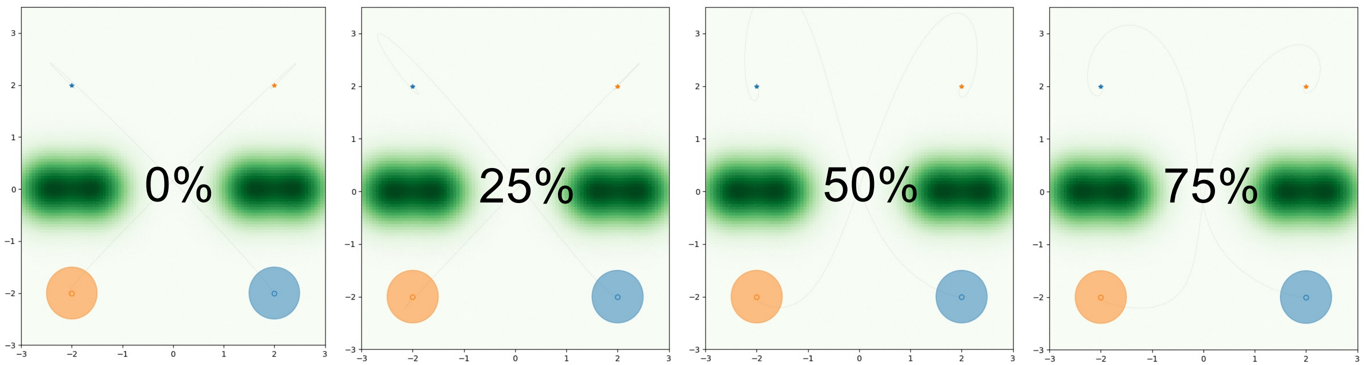
The corridor problem

- Upon training over a dataset 500 different initial conditions





- CL stability guaranteed even with early stopping of training

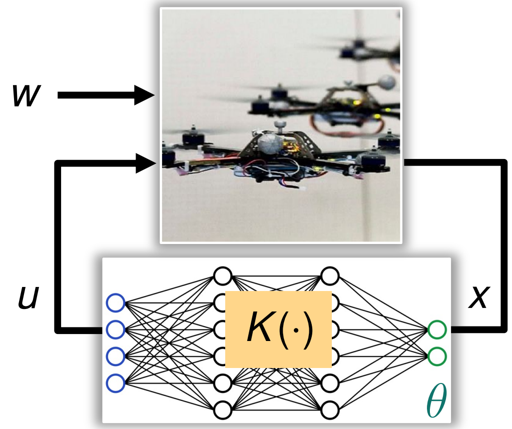


G. Ferrari Trecate & L. Furieri

Outline

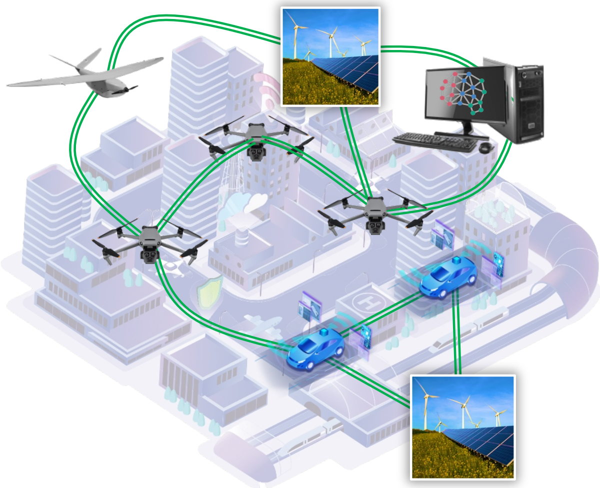
Part 1 (Gianni): Design of performance-boosting policies

- Parametrization of all stabilizing controllers
- NN models of stable operators
  - Solving NOC through NN training



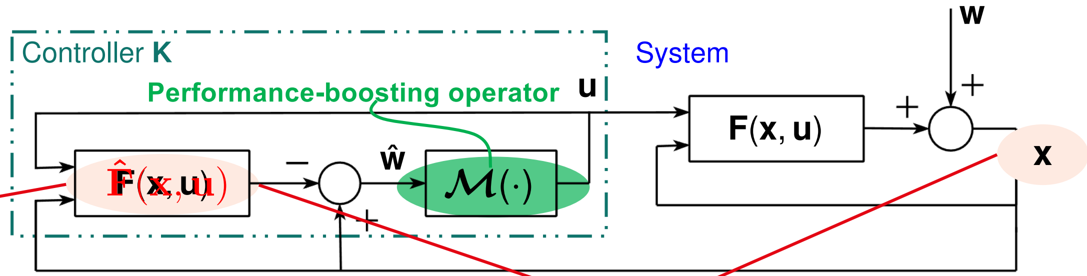
Part 2 (Luca): Extensions for real-world deployment

- Tackling the remaining challenges
  - Uncertain models, output feedback, distributed…
- Lessons from RL: how to shape your cost function



EPFL

Crucial challenges

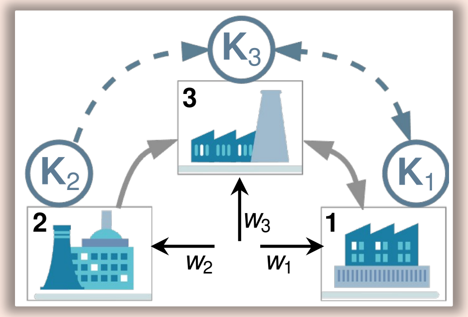


1) Inexact system models

- Only know: $\hat{\mathbf{F}}(\mathbf{x},\mathbf{u}) = (\mathbf{F} + \boldsymbol{\Delta})(\mathbf{x},\mathbf{u})$
- Stability can be compromised!
  - estimated $\hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{F}}(\mathbf{x},\mathbf{u})$ is not the real $\mathbf{w}$!
  - … and is not in ℓ_2 .

2) Noisy outputs

- Only know: $\mathbf{y} = \mathbf{H}(\mathbf{x}) + \mathbf{v}$
- Intricate closed-loop map $(\mathbf{w},\mathbf{v}) \rightarrow (\mathbf{u},\mathbf{x},\mathbf{y})$

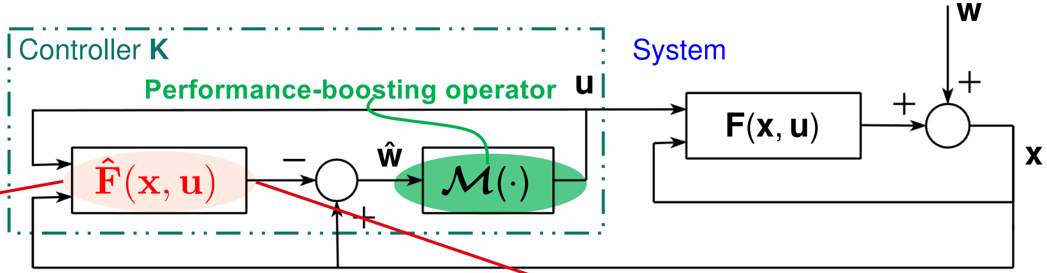


3) Local measurements

- Distributed performance-boosting?

EPFL

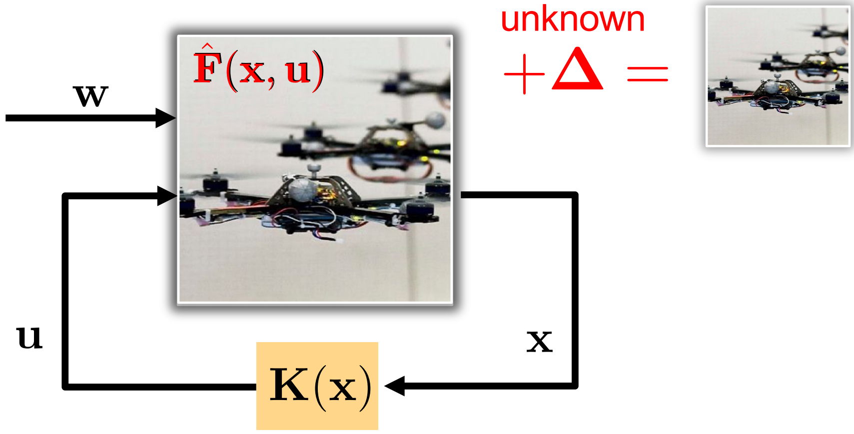
Crucial challenges



1) Inexact system models

- Only know: $\hat{\mathbf{F}}(\mathbf{x},\mathbf{u}) = (\mathbf{F}+\mathbf{\Delta})(\mathbf{x},\mathbf{u})$
- Stability is compromised!
  - estimated $\hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{F}}(\mathbf{x},\mathbf{u})$ is not the real $\mathbf{w}$!
  - … and is not in ℓ_2 .

The robust NOC problem



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

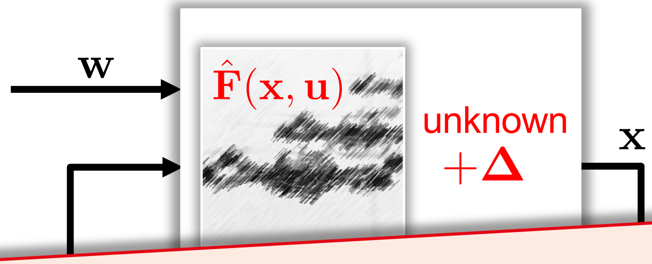
s.t. CLOSED-LOOP STABILITY

for each possible $\boldsymbol{\Delta}$

- **Assumption: !!! Incrementally??? bounded uncertainty**
  - $||\boldsymbol{\Delta}(\mathbf{a}) - \boldsymbol{\Delta}(\mathbf{b})|| \leq \gamma(\boldsymbol{\Delta})||\mathbf{a} - \mathbf{b}||$
- Uncertainty gain $\gamma(\boldsymbol{\Delta})$ estimated from data (e.g., bootstrapping techniques)
  - $\gamma(\boldsymbol{\Delta})$ as a function of #samples… open challenge![1]

[1] Tsiamis, A., Ziemann, I., Matni, N., & Pappas, G. J. (2023). Statistical learning theory for control: A finite-sample perspective. IEEE Control Systems Magazine, 43(6), 67-97.

A naïve small-gain approach



Issues with "standard" small-gain

- Conservative even if $\boldsymbol{\Delta} = 0$!
  - Our result: all and only the stabilizing controllers when $\boldsymbol{\Delta} = 0$

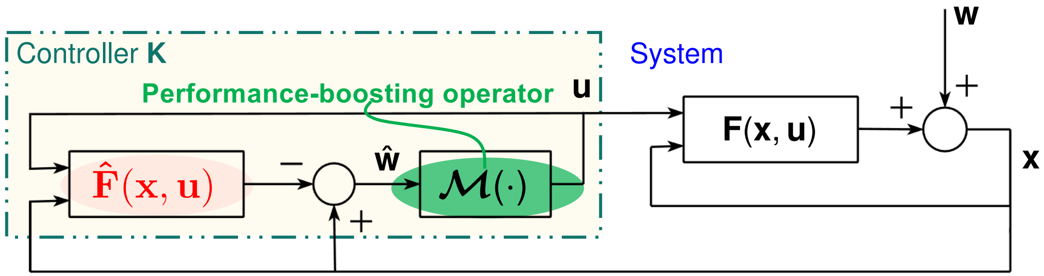
Assume nominal open-loop plant is stable, $\hat{\mathcal{F}} \in \mathcal{L}_p$. Then, if we pick $\mathbf{K}$ such that

$$\gamma(\mathbf{K}) \left(\gamma(\hat{\mathcal{F}}) + \gamma(\boldsymbol{\Delta})\right) \leq 1,$$

the real closed-loop system is stable.

Robust Performance Boosting[1]

Consider the control architecture below:



The real closed-loop system is stable if $\gamma(\mathcal{M}) < \dfrac{1}{\gamma(\boldsymbol{\Delta})\,(\gamma(\mathcal{F})+1)}$,

where $\mathcal{F}$ is the open-loop plant operator satisfying $\mathbf{x} = \mathcal{F}(\mathbf{u}, \mathbf{w})$.

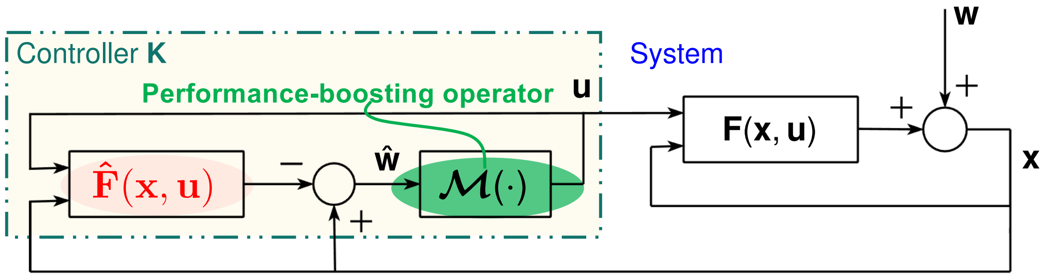
Remarks

- Unconstrained learning over robustly stabilizing controllers
  - e.g., can specify maximal gain $\gamma(\mathcal{M})$ using REN models (Part 1)
- Conservatism vanishes as $\boldsymbol{\Delta} \to 0$!
  - right-hand-side becomes infinity → all and only stabilizing policies (Part 1)

[1] L. Furieri, C. L. Galimberti, and GFT, “Learning to Boost the Performance of Stable Nonlinear Systems,” ArXiv 2024

Robust Performance Boosting[1]

Consider the control architecture below:



The real closed-loop system is stable if $\gamma(\boldsymbol{\mathcal{M}}) < \dfrac{1}{\gamma(\boldsymbol{\Delta})\left(\gamma(\boldsymbol{\mathcal{F}})+1\right)}$,

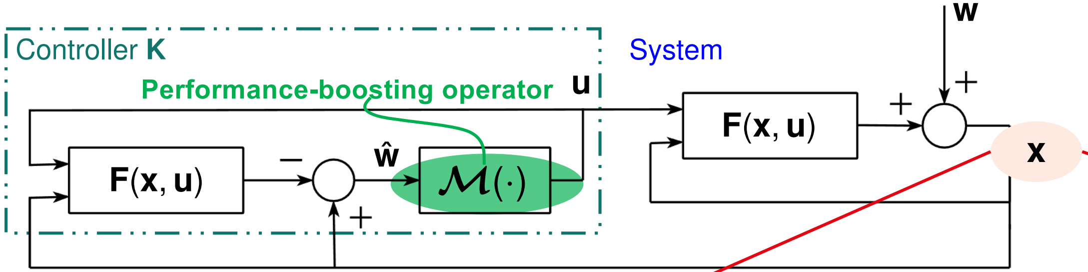
where $\boldsymbol{\mathcal{F}}$ is the open-loop plant operator satisfying $\mathbf{x} = \boldsymbol{\mathcal{F}}(\mathbf{u}, \mathbf{w})$.

Proof sketch

- Notice that $\mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{F}(\boldsymbol{\mathcal{F}}(\mathbf{u}, \mathbf{w}), \mathbf{u})$, and $\hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{F}}(\mathbf{x}, \mathbf{u})$
  - by substitution we reveal $\hat{\mathbf{w}} = \boldsymbol{\Delta}(\boldsymbol{\mathcal{F}}(\mathbf{u}, \mathbf{w}), \mathbf{u}) + \mathbf{w}$
- Upperbounding through the operator gains, the above implies

$$|\hat{\mathbf{w}}| \leq \left(\frac{\gamma(\boldsymbol{\Delta})\gamma(\boldsymbol{\mathcal{F}})+1}{1-\gamma(\boldsymbol{\Delta})\gamma(\boldsymbol{\mathcal{M}})\left(\gamma(\boldsymbol{\mathcal{F}})+1\right)}\right)|\mathbf{w}|\,.$$

Crucial challenges



2) Noisy outputs

- Only know: $\mathbf{y} = \mathbf{H}(\mathbf{x}) + \mathbf{v}$
- Intricate closed-loop map
  $(\mathbf{w}, \mathbf{v}) \rightarrow (\mathbf{u}, \mathbf{x}, \mathbf{y})$

EPFL

Towards the output-feedback case

- Classical results based on
  - Youla-like formulations[1]
  - Kernel-based representations[2]
- Recent results using REN parametrizations
  - Contractive closed-loops for linear systems[3]
  - Extension to contractive and Lipschitz nonlinear systems[4]

Lack of a general theory

- Different modeling setups (e.g., state-space, input/output…)
- Different guarantees ($\mathcal{L}_p$-stability, contractivity…)

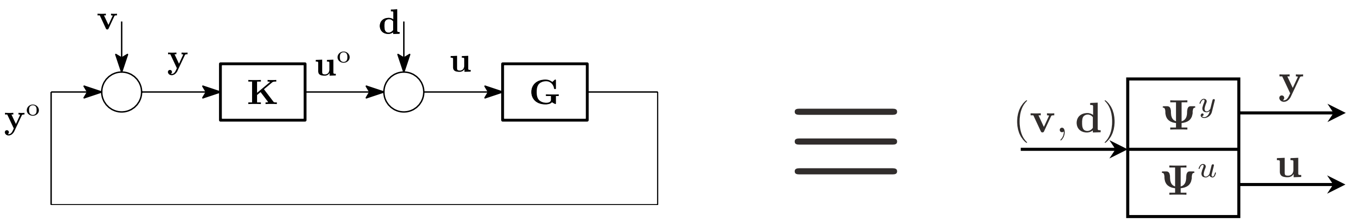
[1] V. Anantharam and C. A. Desoer. "On the stabilization of nonlinear systems," *IEEE Transactions on Automatic Control,* 1984.
[2] K. Fujimoto and T. Sugie. "Youla-Kucera Parameterization for Nonlinear Systems via Observer Based Kernel Representations," *Trans. of the Soc. of Inst. and Control Engineers,* 1998.
[3] Wang, R., & Manchester, I. R. "Youla-ren: Learning nonlinear feedback policies with robust stability guarantees". *2022 American Control Conference (ACC)*. IEEE.
[4] N.H. Barbara, R. Wang and I.R. Manchester, "Learning Over Contracting and Lipschitz Closed-Loops for Partially-Observed Nonlinear Systems," *IEEE Conf. Decision & Control*, 2023.

A closed-loop operator perspective

- We focus on nonlinear systems in input-output form

$$\mathbf{y} = \mathbf{G}(\mathbf{u} + \mathbf{d}) + \mathbf{v} \qquad \mathbf{G} \in \mathcal{L}_p$$

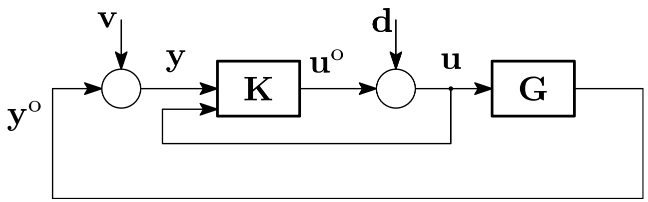
- Control loop and induced closed-loop operators $(\mathbf{\Psi}^y, \mathbf{\Psi}^u)$



- Novel characterization of all achievable closed-loop operators[1]

$$\mathbf{\Psi}^y = \mathbf{G} \circ \mathbf{\Psi}^u, \quad \cancel{\mathbf{\Psi}^u = \mathbf{\Psi}^u \circ \left(\mathbf{\Psi}^{(y,u)}\right)^{-1} \mathbf{\Psi}^y}$$

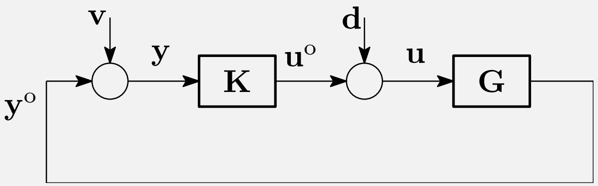
- Drop the second constraint… new architecture



$$\mathbf{\Psi}^y = \mathbf{G} \circ \mathbf{\Psi}^u$$

[1] Galimberti, C., Furieri, L., GFT, "Performance-boosting output-feedback controllers for nonlinear systems", in preparation, 2024

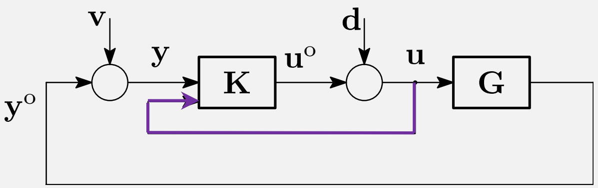
Performance-boosting in output-feedback



Equivalent to Youla

$$\min_{\mathcal{Q}\in\mathcal{L}_p} \quad \frac{1}{T}\left[\mathscr{L}(y_{T:0}, u_{T:0})\right]$$
$$\text{s.t.} \quad \mathbf{y} = \mathbf{G}(\mathbf{u}+\mathbf{d}) + \mathbf{v},$$
$$\mathbf{u} = \mathcal{Q}(\mathbf{y} - \mathbf{G}(\mathbf{u}))$$

- Learn over all and only $\mathcal{L}_p$-stabilizing controllers



New architecture

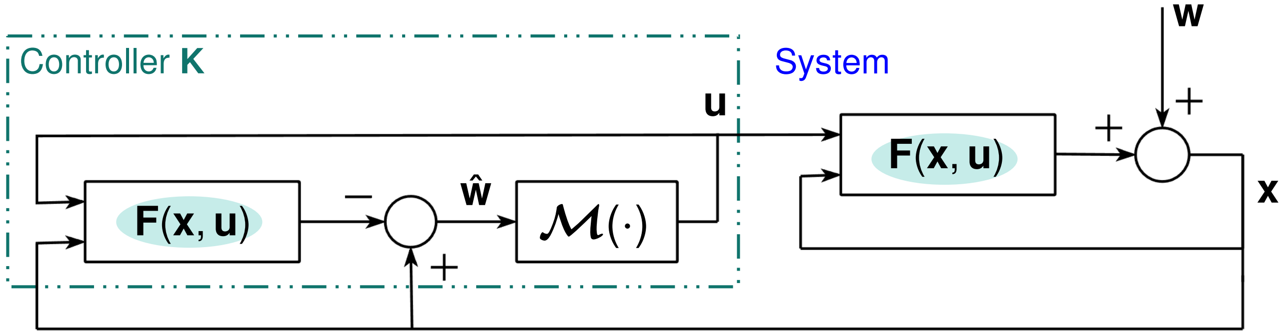
$$\min_{\mathbf{\Psi}^u\in\mathcal{L}_p} \quad \frac{1}{T}\left[\mathscr{L}(y_{T:0}, u_{T:0})\right]$$
$$\text{s.t.} \quad \delta_{t-1} = u_{t-1} - \mathbf{\Psi}^u_{t-1}(\beta_{t-1}, \delta_{t-2})$$
$$\beta_t = y_t - (\mathbf{G}_t \circ \mathbf{\Psi}^u_t)(\beta_{t-1}, \delta_{t-1})$$
$$u_t = \mathbf{\Psi}^u_t(\beta_t, \delta_{t-1})$$

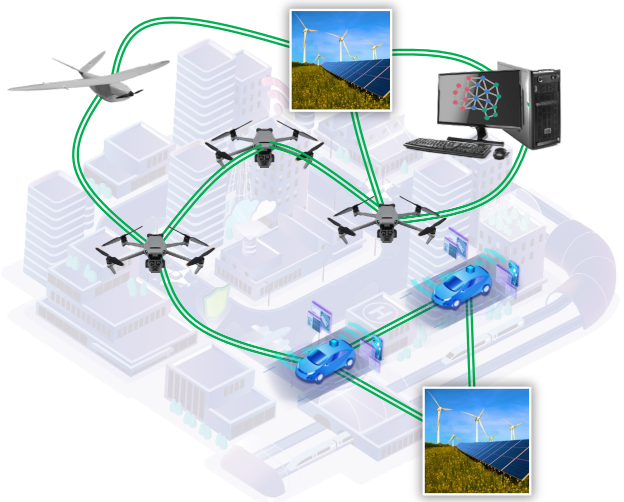
- Additional insight $\mathbf{\Psi}^y = \mathbf{G} \circ \mathbf{\Psi}^u$
  - Learn over closed-loop maps with stronger properties, e.g. [2]
  - E.g., $\mathbf{\Psi}^u$ is a REN, $\mathbf{G}$ is contracting $\implies$ $\mathbf{\Psi}^y$ inherits contractivity

[1] Galimberti, C., Furieri, L., GFT, "Performance-boosting output-feedback controllers for nonlinear systems", in preparation, 2024

[2] N.H. Barbara, R. Wang and I.R. Manchester, "Learning Over Contracting and Lipschitz Closed-Loops for Partially-Observed Nonlinear Systems," *IEEE Conf. Decision & Control*, 2023.

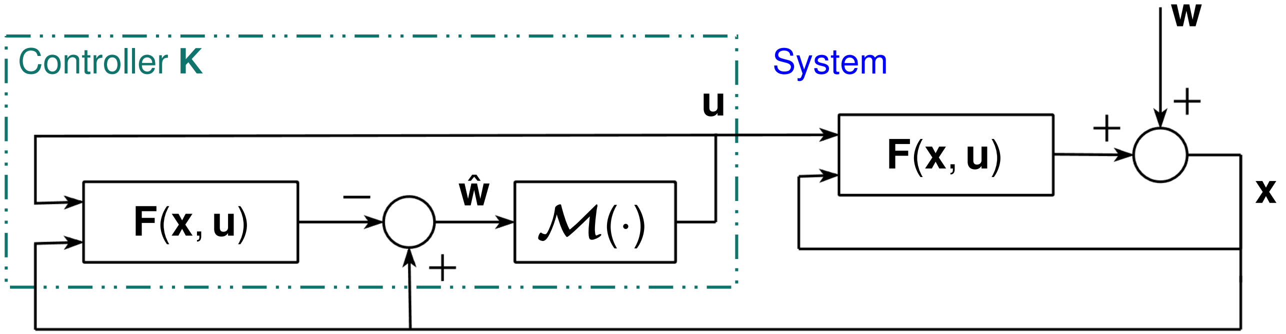
Distributed control





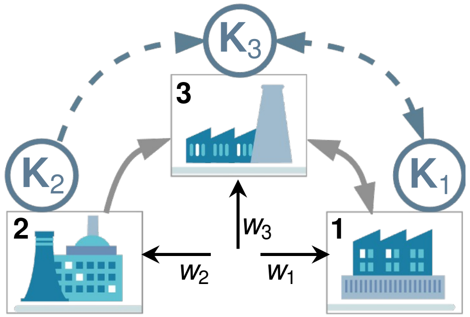
G. Ferrari Trecate & L. Furieri

Distributed control



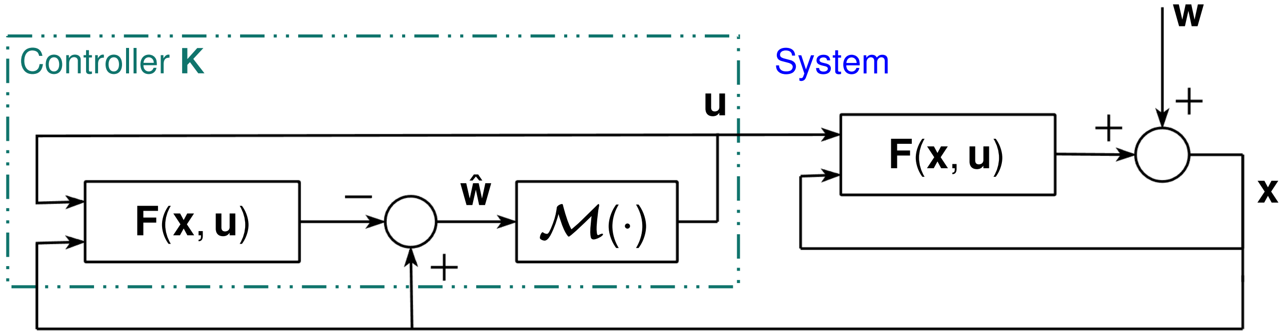
The sparsity of F is replicated in the controller

State-coupled subsystems



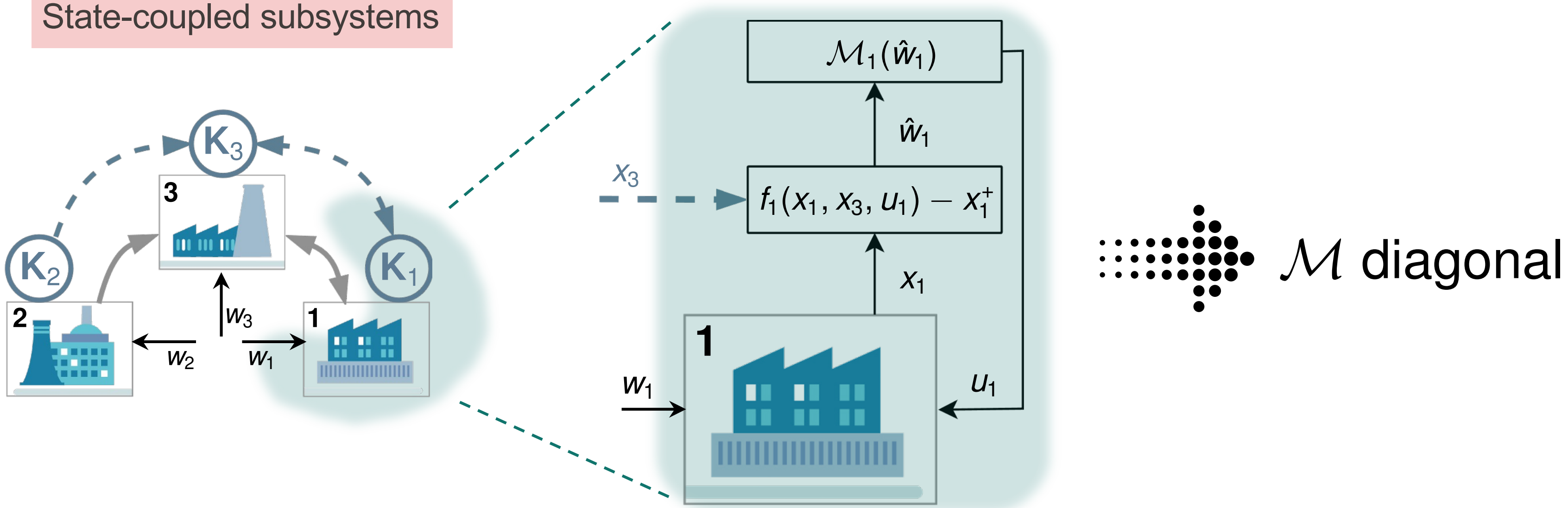
G. Ferrari Trecate & L. Furieri

Distributed control



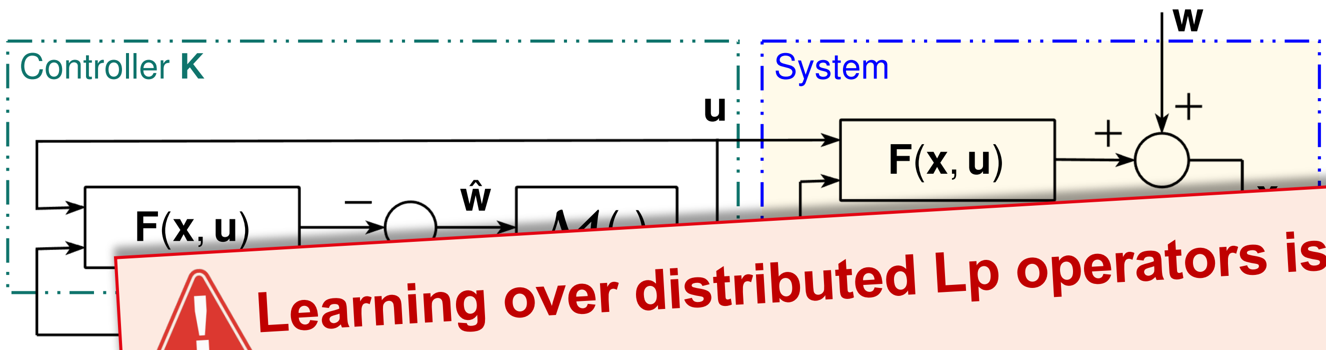
The sparsity of F is replicated in the controller

State-coupled subsystems



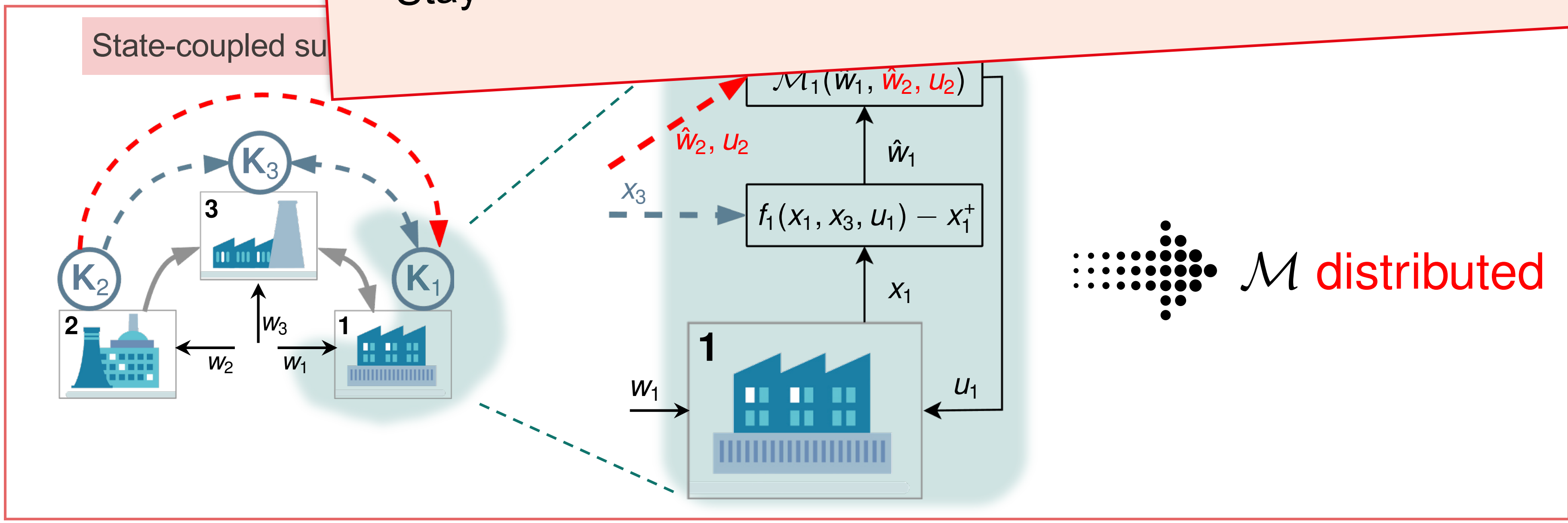
Distributed control

G. Ferrari Trecate & L. Furieri



Learning over distributed Lp operators is challenging!

- Stay tuned for the next talk…



The magic of the cost

Nonlinear Optimal Control (NOC)

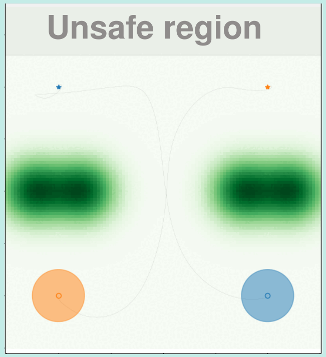
$$K(\cdot) \in \operatorname{argmin} \frac{1}{T}\mathbb{E}_w\left[\mathcal{L}(x_{0:T}, u_{0:T})\right]$$

s.t. CLOSED-LOOP STABILITY

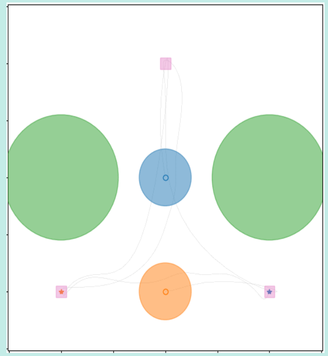
The magic of the cost

G. Ferrari Trecate & L. Furieri

Safety via invariance



Waypoint tracking



Nonlinear Optimal Control (NOC)

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

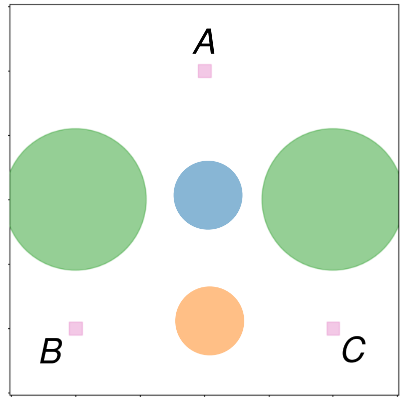
s.t. CLOSED-LOOP STABILITY

Boosting open-loop performance

- $\mathcal{L}_2$-gain, settling time, overshoot, ...

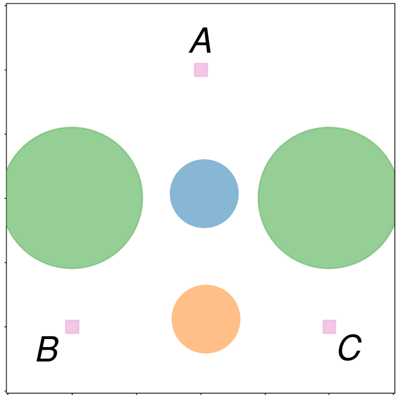
Waypoints tracking

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



Waypoints tracking

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

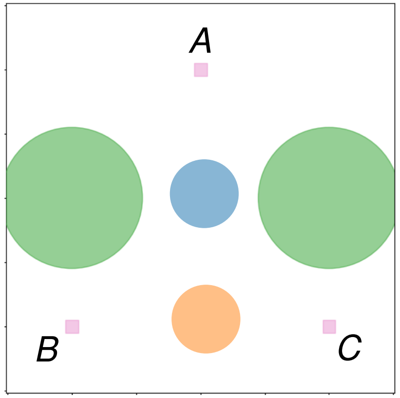


- Waypoints $\to$ **Linear Temporal Logic** formulae[1] $\to$ cost $\mathcal{L}_{way}$

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

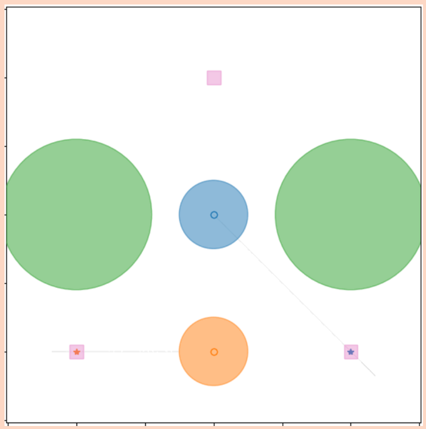
Waypoints tracking

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

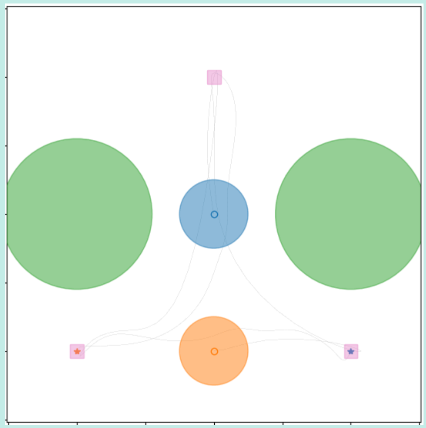


- Waypoints $\rightarrow$ **Linear Temporal Logic** formulae[1] $\rightarrow$ cost $\mathcal{L}_{way}$

Base controller



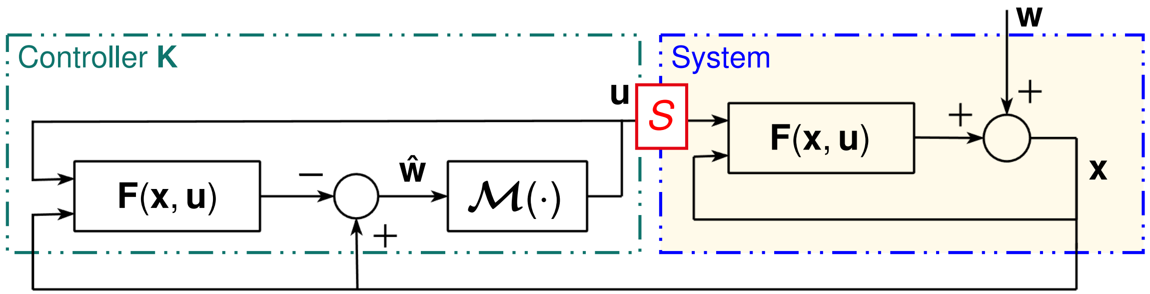
Performance boosting



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

G. Ferrari Trecate & L. Furieri

Safety

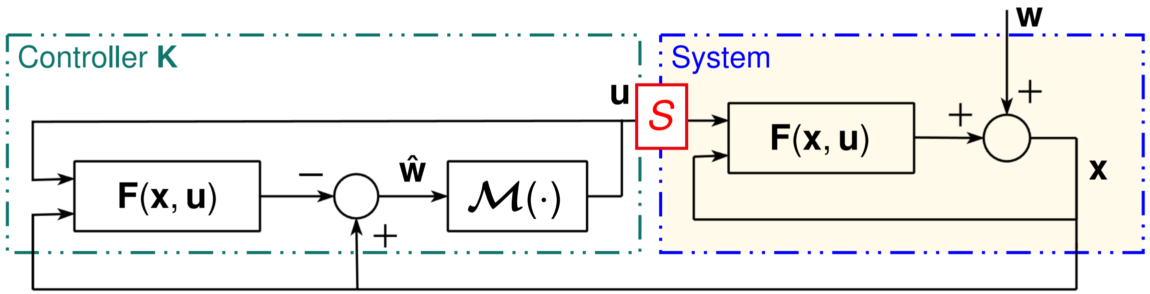


- Add a safety filter[1] guaranteeing $(x_t, u_t) \in \mathcal{C}, \forall t > 0$
  - Requires online optimization
  - Tweaks **u** only if needed

[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

Safety



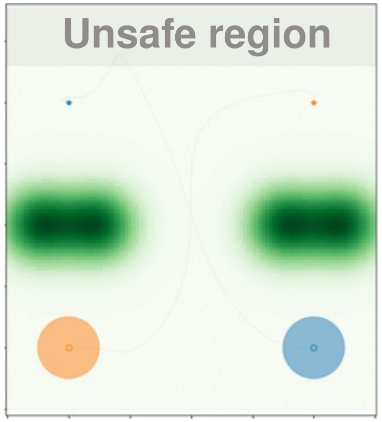
- Add a safety filter[1] guaranteeing $(x_t, u_t) \in \mathcal{C}, \forall t > 0$
  - Requires online optimization
  - Tweaks **u** only if needed
- **Reduce filter activation** embedding **soft safety specs** in the cost
  - Promote constraint fulfillment $\rightarrow \mathcal{L}_{safe} = \max_{t<T}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

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

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

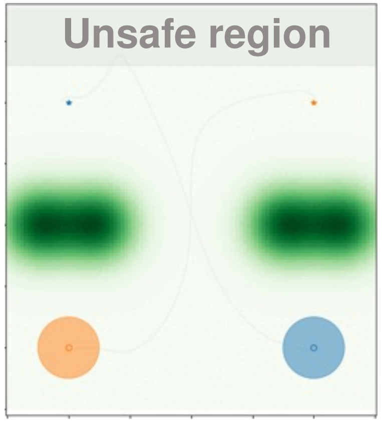
[2] Agrawal, A., and K. Sreenath. "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." *Robotics: Science and Systems.* 2017

The safe corridor problem

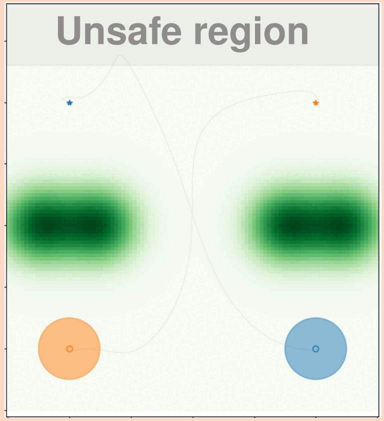


G. Ferrari Trecate & L. Furieri

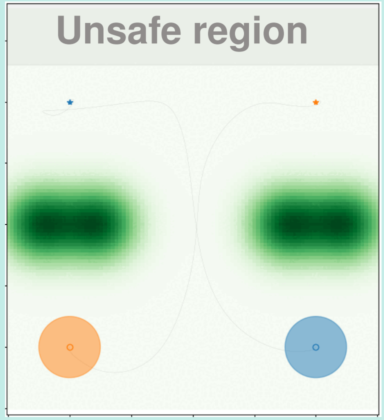
The safe corridor problem



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



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

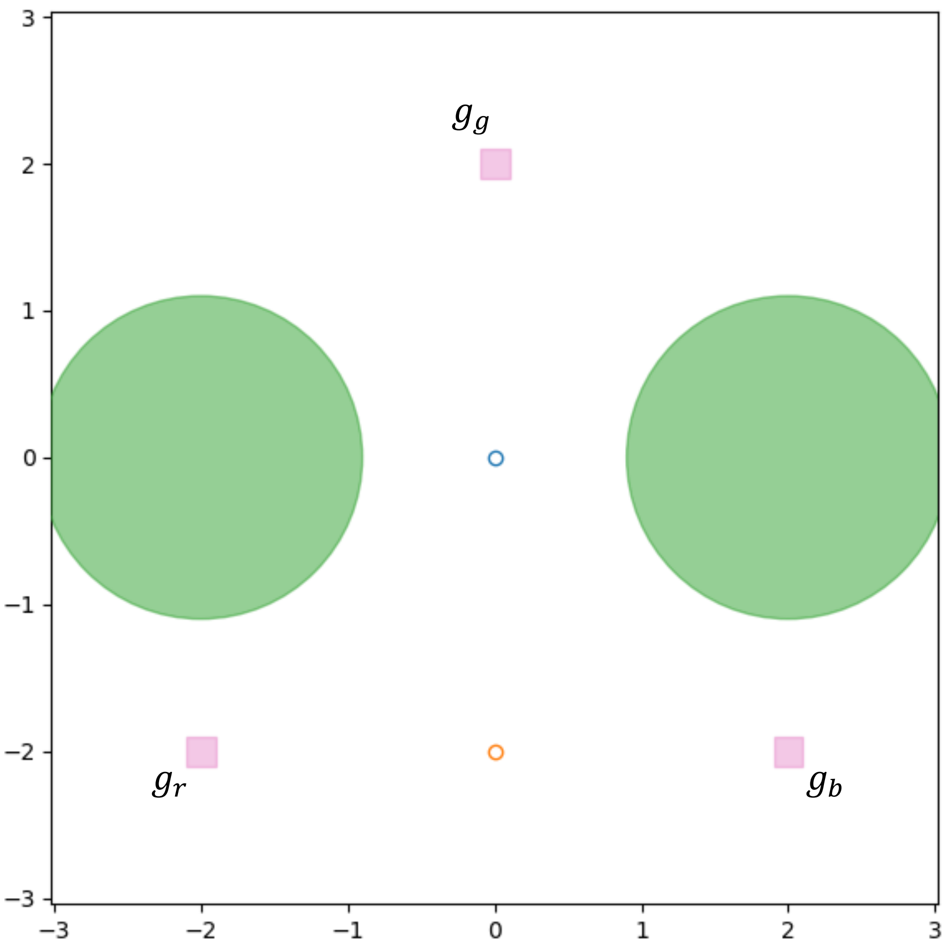


EPFL

THANKS !

G. Ferrari Trecate & L. Furieri

Waypoints



- The objective is to have two vehicles that visits the pink squares, starting from the small circles, in the following order:
  - Blue vehicle: $g_r \to g_g \to g_b$
  - Orange vehicle: $g_b \to g_g \to g_r$

- We use the TLTL specification. It reads, for vehicle 1:

$$\phi_1 = (\psi_{g_r} \mathcal{T} \psi_{g_g} \mathcal{T} \psi_{g_b}) \wedge (\neg (\psi_{g_g} \vee \psi_{g_b}) \mathcal{U} \psi_{g_r}) \wedge (\neg \psi_{g_b} \mathcal{U} \psi_{g_g}) \wedge \left(\bigwedge_{i=r,g,b} \Box (\psi_{g_i} \Rightarrow \bigcirc \Box \neg \psi_{g_i}) \right) \wedge \left(\bigwedge_{i=1,2} \Box \psi_{o_i} \right) \wedge \Diamond \Box \psi_{g_b}$$

  where: $\psi_{g_i} = d_{g_i} < 0.05$ and $\psi_{o_i} = d_{o_i} > r_{o_i}$, being r_{o_i} the radius of the obstacle.

  The specification ϕ_2 for vehicle 2 can be constructed similarly.

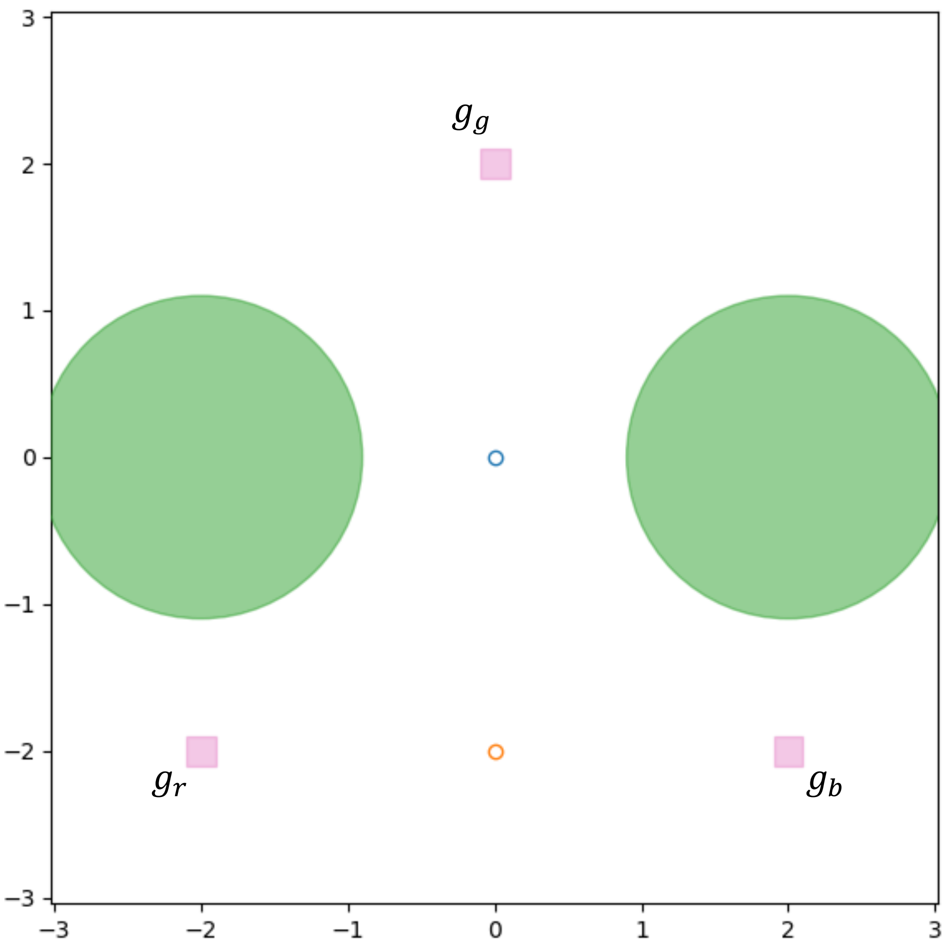
  The collision avoidance specification reads: $\phi_{ca} = d_{1,2} > 2 * r_{vehicle}$

- The final specification is then: $\phi_1 \wedge \phi_2 \wedge \phi_{ca}$

- Cost: we translate the TLTL specification to a cost function $\mathcal{L}_{TL}$. Moreover, we also consider a regularization term of the form $\mathcal{L}_x = (x - \bar{x})^\top Q(x - \bar{x})$

  for promoting the vehicles to do the minimal possible path.
  The final cost is $\mathcal{L}_{TL} + \alpha_x \mathcal{L}_x$

Waypoints



$$\phi_1 = \left(\psi_{g_r} \,\mathcal{T}\, \psi_{g_g} \,\mathcal{T}\, \psi_{g_b}\right) \wedge \left(\neg\left(\psi_{g_g} \vee \psi_{g_b}\right) \mathcal{U}\, \psi_{g_r}\right) \wedge \left(\neg\psi_{g_b} \,\mathcal{U}\, \psi_{g_g}\right) \wedge \left(\bigwedge_{i=r,g,b} \Box\left(\psi_{g_i} \Rightarrow \bigcirc\Box\neg\psi_{g_i}\right)\right) \wedge \left(\bigwedge_{i=1,2} \Box\psi_{o_i}\right) \wedge \Diamond\Box\psi_{g_b}$$

The TLTL specification for vehicle 1 reads:

$\left(\psi_{g_r} \,\mathcal{T}\, \psi_{g_g} \,\mathcal{T}\, \psi_{g_b}\right)$: visit g_r then g_g then g_b,

$\wedge$: and

$\left(\neg\left(\psi_{g_g} \vee \psi_{g_b}\right) \mathcal{U}\, \psi_{g_r}\right)$: don't visit g_g or g_b until visiting g_r,

$\wedge$: and

$\left(\neg\psi_{g_b} \,\mathcal{U}\, \psi_{g_g}\right)$: don't visit g_b until visiting g_g,

$\wedge$: and

$\left(\bigwedge_{i=r,g,b} \Box\left(\psi_{g_i} \Rightarrow \bigcirc\Box\neg\psi_{g_i}\right)\right)$: always if visited g_i implies next always don't visit g_i,

$\wedge$: and

$\left(\bigwedge_{i=1,2} \Box\psi_{o_i}\right)$: always avoid obstacles,

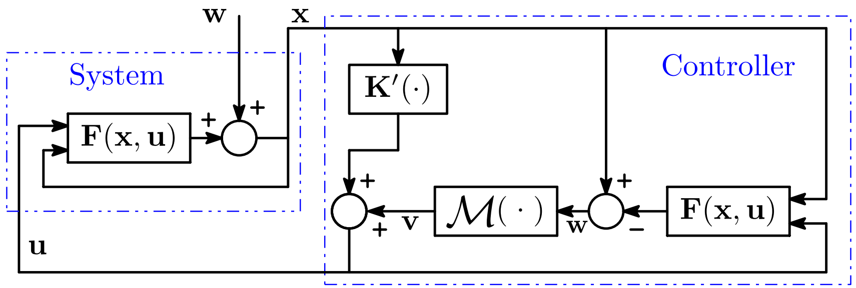
$\wedge$: and

$\Diamond\Box\psi_{g_b}$: eventually always state at the final goal (g_b).

Main result

II. Parametrize **v(x)** as follows

$$\mathbf{v} = \mathcal{M}(\mathbf{w}) = \mathcal{M}(\mathbf{x} - \mathbf{F}(\mathbf{x}, \mathbf{u}))$$



Result part 1 (sufficiency)

The CL maps $(\mathbf{\Phi}^x, \mathbf{\Phi}^u)$ achieved by the control scheme above are stable for any $\mathcal{M}(\cdot) \in \mathcal{L}_p$

Proof

- By hypothesis, disturbance sequence $\mathbf{w} \in \ell_p$
- Since $\mathcal{M}(\cdot) \in \mathcal{L}_p$, then $\mathbf{v} = \mathcal{M}(\mathbf{w}) \in \ell_p$
- By hypothesis, base controller $\mathbf{K}'(\cdot)$ such that $(\mathbf{w}, \mathbf{v}) \in \ell_p \implies (\mathbf{x}, \mathbf{u}) \in \ell_p$

Main result

Result part 2 (necessity)

If $\mathbf{K}' \in \mathcal{L}_p$, we can obtain any achievable CL maps $(\mathbf{\Psi}^{\mathbf{x}}, \mathbf{\Psi}^{\mathbf{u}}) \in \mathcal{L}_p$ by searching over the space of stable operators $\mathcal{M} \in \mathcal{L}_p$.

$\Longrightarrow$ **Globally optimal** CL maps by searching over $\mathcal{M} \in \mathcal{L}_p$

Proof

- Select $\mathcal{M} = -\mathbf{K}'(\mathbf{\Psi}^{\mathbf{x}}) + \mathbf{\Psi}^{\mathbf{u}}$. Then, $(\mathbf{K}', \mathbf{\Psi}^{\mathbf{x}}, \mathbf{\Psi}^{\mathbf{u}}) \in \mathcal{L}_p \implies \mathcal{M} \in \mathcal{L}_p$
- So, the corresponding policy $\mathbf{u} = \mathbf{K}'(\mathbf{x}) + \mathcal{M}(\mathbf{x} - \mathbf{F}(\mathbf{x}, \mathbf{u}))$ is within our search space ☺

What closed-loop maps do we achieve?

- We prove by induction that $(\mathbf{\Phi}^{\mathbf{x}}, \mathbf{\Phi}^{\mathbf{u}}) = (\mathbf{\Psi}^{\mathbf{x}}, \mathbf{\Psi}^{\mathbf{u}})$, i.e., we achieve the desired CL maps.
- Inductive Step: assume $(\Phi^x_{j:0}, \Phi^u_{j:0}) = (\Psi^x_{j:0}, \Psi^u_{j:0})$. Then

$$\Phi^u_{j+1} = K'_{j+1}(\underbrace{F_{j+1:0}(\Phi^x_{j:0}, \Phi^u_{j:0}) + I}_{=\Phi^x_{j+1}}) - K'_{j+1}(\underbrace{F_{j+1:0}(\Psi^x_{j:0}, \Psi^u_{j:0}) + I}_{=\Psi^x_{j+1}}) + \Psi^u_{j+1} = \Psi^u_{j+1}$$ ☺

- Base Step: $\Phi^x_0 = \Psi^x_0 = I \ldots$ (*the initial state is the initial state*)