Some fundamental limitations of learning for dynamics and control

Necmiye Ozay

Electrical Engineering and Computer Science

Robotics

University of Michigan

American Control Conference

New Orleans, LA

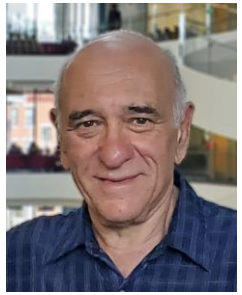
May 28, 2026

Research funded by ONR MURI

Outline

- **The role of learning in dynamics and control**

- **On some fundamental limitations of learning immersions (Koopman embeddings)**



Joint work with
Zexiang Liu (Michigan) and Eduardo Sontag (Northeastern)
arXiv:2312.17045 **(IFAC WC'23, Automatica'25)**

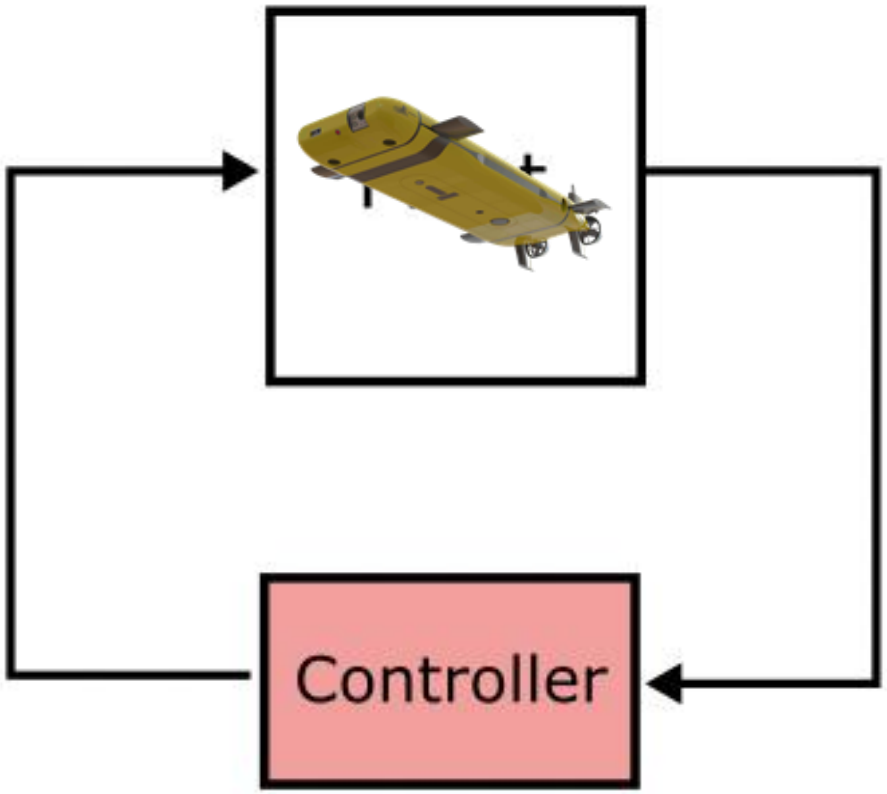
- **On the hardness of learning to stabilize linear systems**

Joint work with
Xiong Zeng (Michigan), Zexiang Liu (Michigan), Zhe Du (Michigan), and Mario Sznaier (Northeastern)
arXiv:2311.11151 **(CDC'23)**

Plant (+ operating environment)

Task or mission specification

φ



Many components in a feedback loop can be learned:

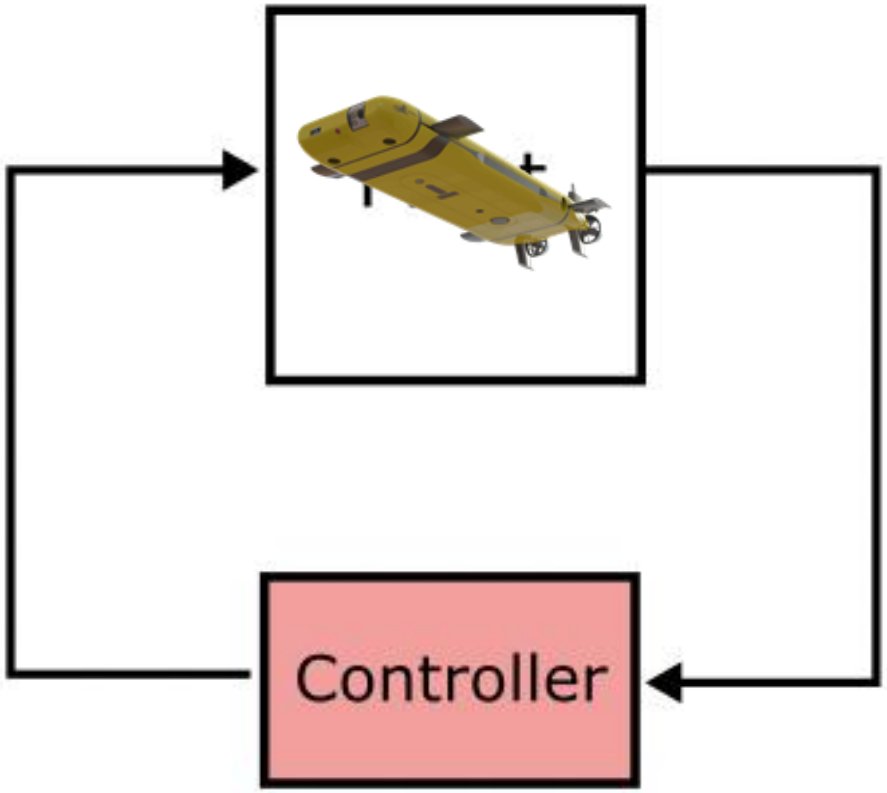
- **Plant model:** model-based learning, system identification
- **Task specification:** inverse optimal control, inverse reinforcement learning
- **Controller:** model-free learning
- **Perception/estimation modules**

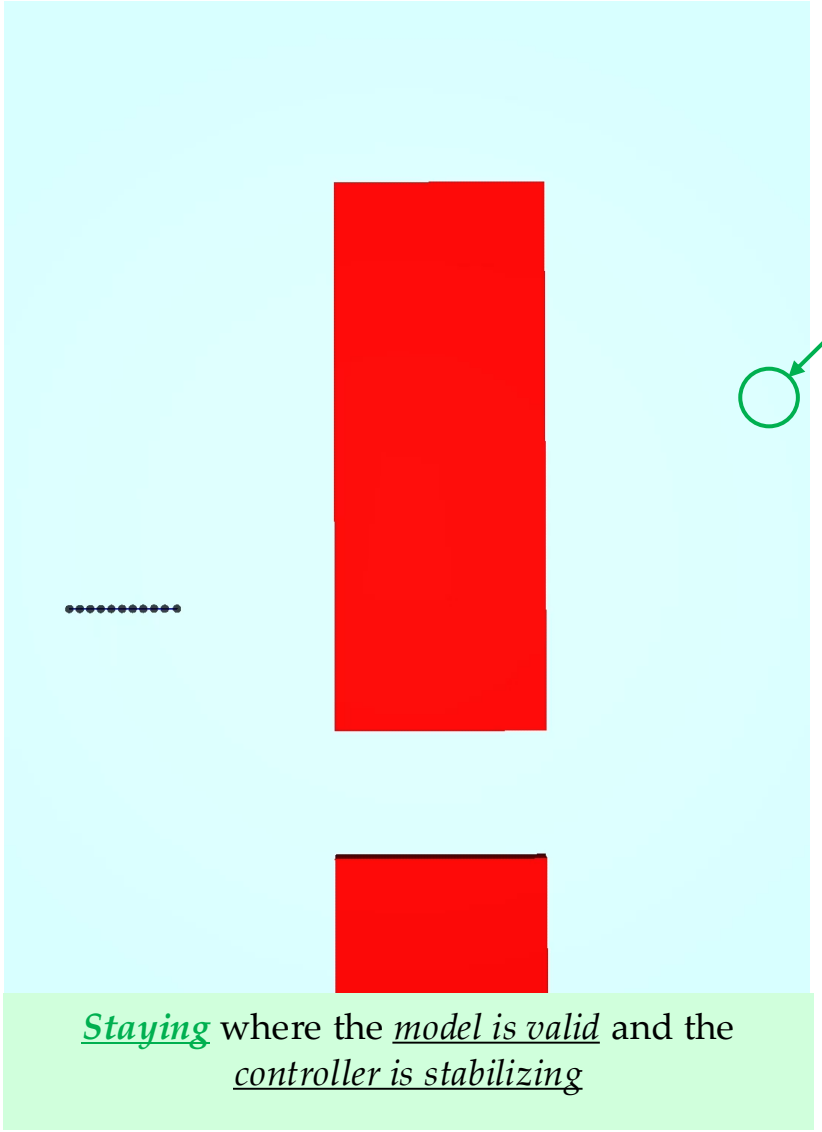
GOAL: **Ensure** system-level safety and performance when some of these pieces are learned!

Plant (+ operating environment)

Task or mission specification

φ



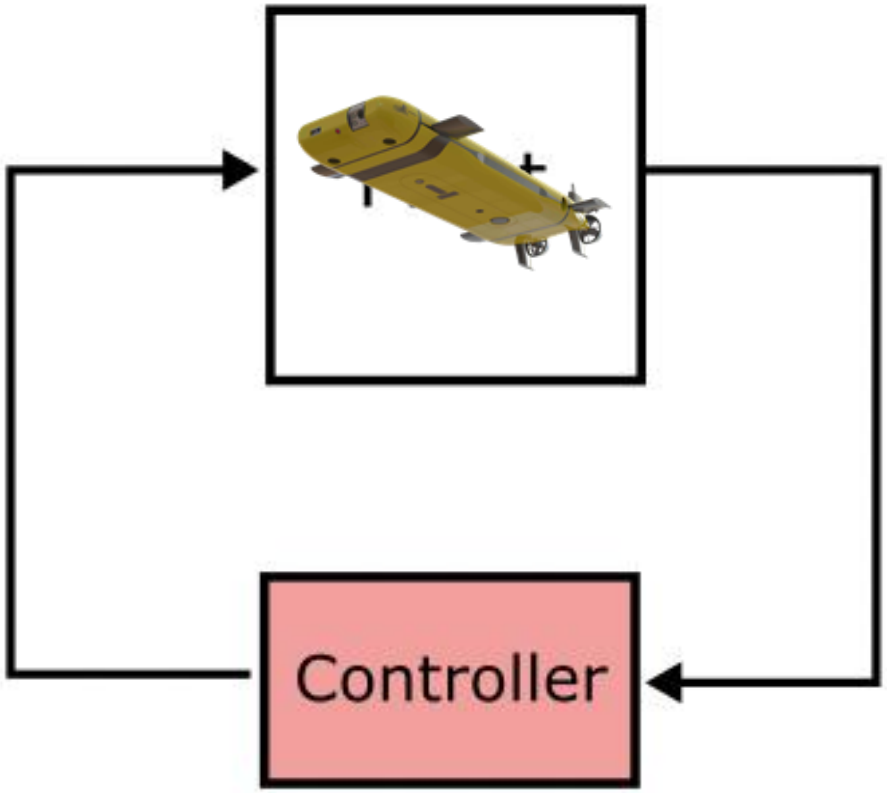


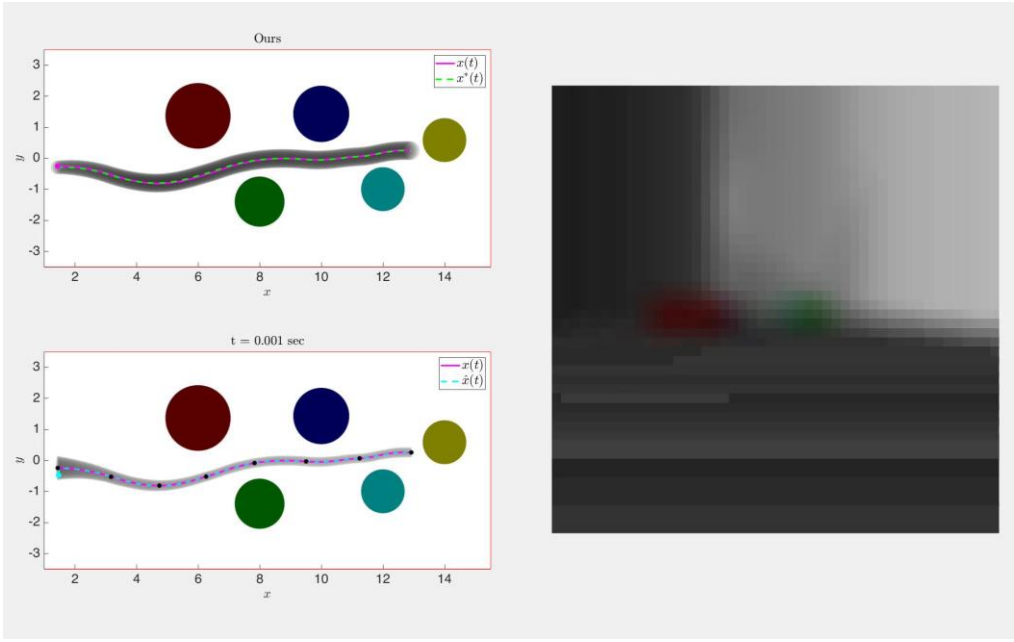
Learned plant model
(Chou, Ozay, Berenson CDC'21)

Task or mission specification

φ

Plant (+ operating environment)





Legend

Tubes executed trajectory planned trajectory estimated trajectory

Nominal dynamics:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ u_1 \\ u_2 \end{bmatrix}$$

Goal: safely reach goal using RGB-D images at runtime

Learned perception module

(Chou, Ozay, Berenson WAFR'22)

Plant (+ operating environment)

Task or mission specification

Demonstration (in VR)

Execution in new scene

Task:

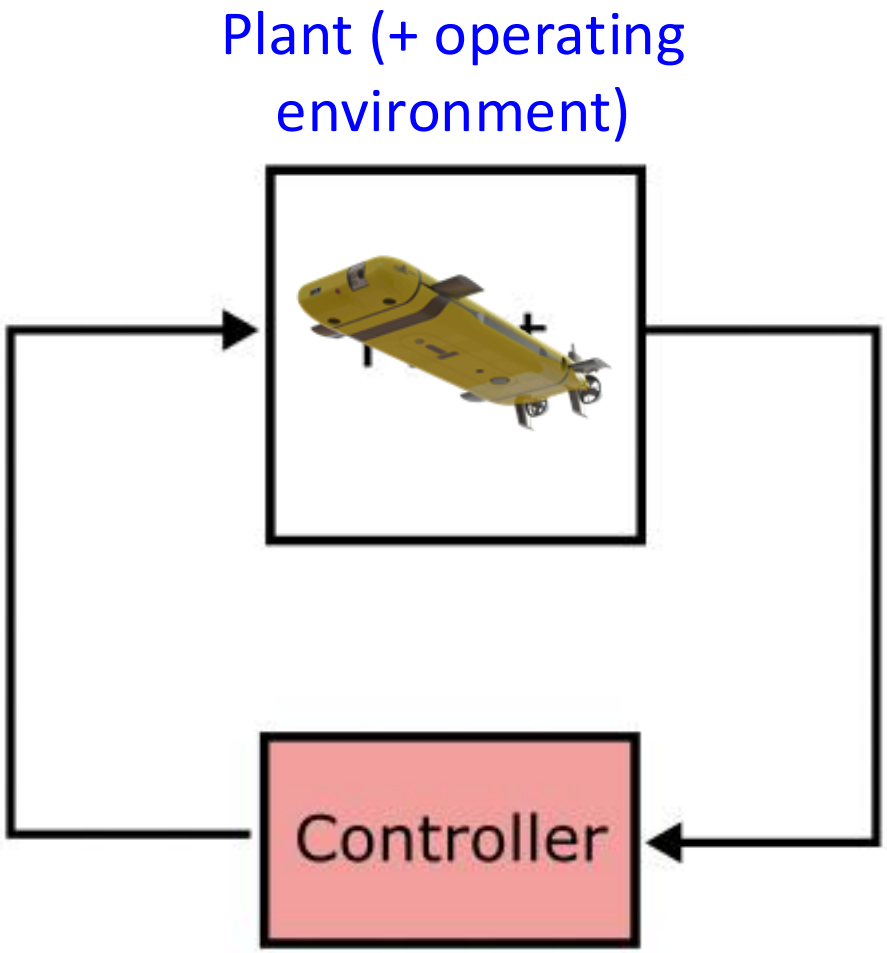
grasp soup → place in small box → deliver to blue region → grasp Cheez-Its → place in large box

Learned specification

(Chou, Ozay, Berenson RSS'20, AURO'22)

Task or mission specification

φ



Many components in a feedback loop can be learned:

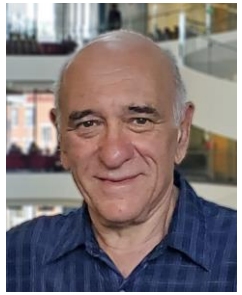
- **Plant model:** model-based learning, system identification
- **Task specification:** inverse optimal control, inverse reinforcement learning
- **Controller:** model-free learning
- **Perception/estimation modules**

What are the fundamental limits in learning-based control?

Outline

- The role of learning in dynamics and control

- **On some fundamental limitations of learning immersions (Koopman embeddings)**



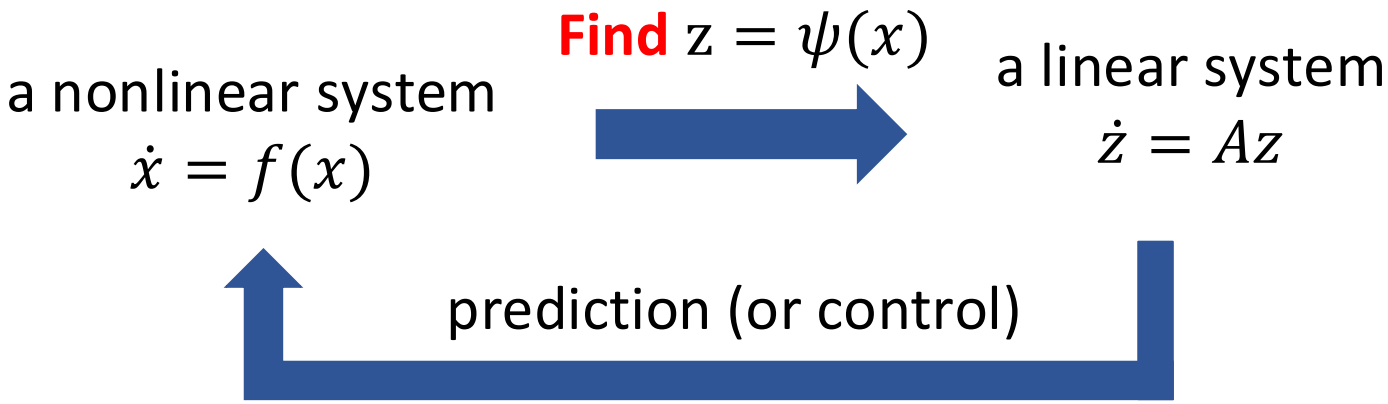
Joint work with
Zexiang Liu (Michigan) and Eduardo Sontag (Northeastern)
arXiv:2312.17045 **(IFAC WC'23, Automatica'25)**

- On the hardness of learning to stabilize linear systems

Joint work with
Xiong Zeng (Michigan), Zexiang Liu (Michigan), Zhe Du (Michigan), and Mario Sznaier (Northeastern)
arXiv:2311.11151 **(CDC'23)**

Linear immersions

The idea of Koopman-inspired methods:

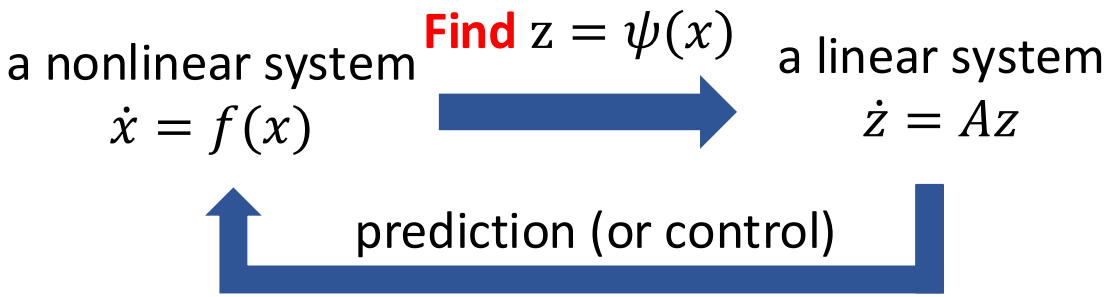


In Practice: Represent $\psi(x)$ as a neural network and learn the network weights and *A* from data

Question: Does such a $\psi(x)$ **exist or not**?

Existence of linear immersion?

The idea of Koopman-inspired methods:



Question: Does such a $\psi(x)$ **exist or not**?

Answer: Sure! Take $\psi(x) = c \ \ \forall x$, and $A = 0$
(constant function)

Question: Does such a one-to-one $\psi(x)$ **exist or not**?

Existence of linear immersion?

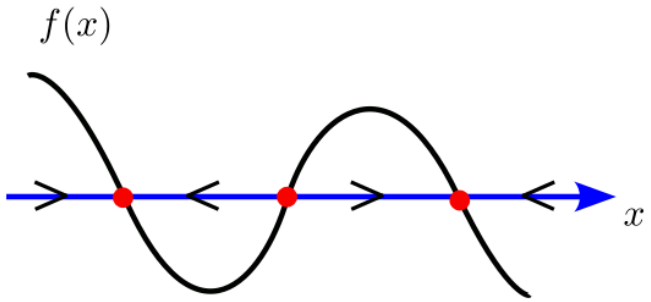
The idea of Koopman-inspired methods:

a nonlinear system $\dot{x} = f(x)$ → **Find** $z = \psi(x)$ → a linear system $\dot{z} = Az$

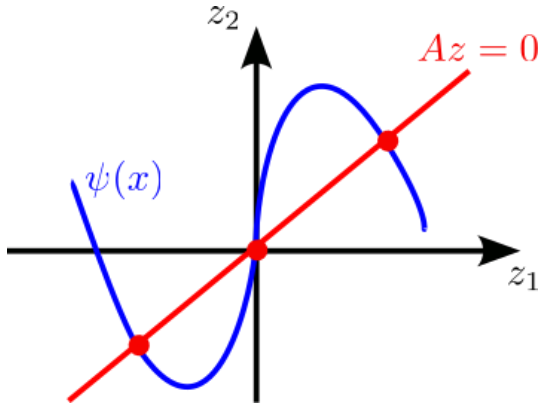
prediction (or control)

Question: Does a one-to-one $\psi(x)$ **exist or not**?

1D example:



$\psi(x)$?



The only possibility is that A is rank-deficient!

The power of discontinuity

1D example:

Consider the 1-d system with **three isolated equilibria**

$$\dot{x} = \frac{x(1-x^2)}{1+x^2}.$$

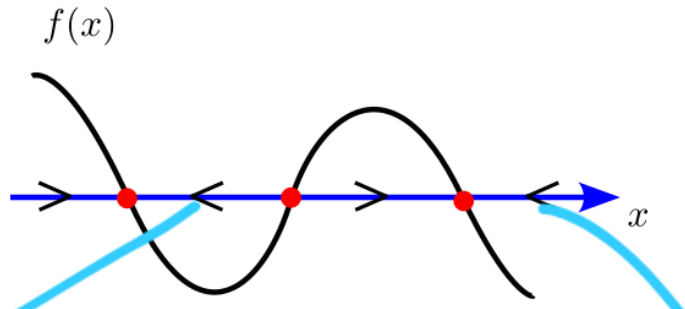
Equilibrium points are $\{-1, 0, 1\}$.

Then,

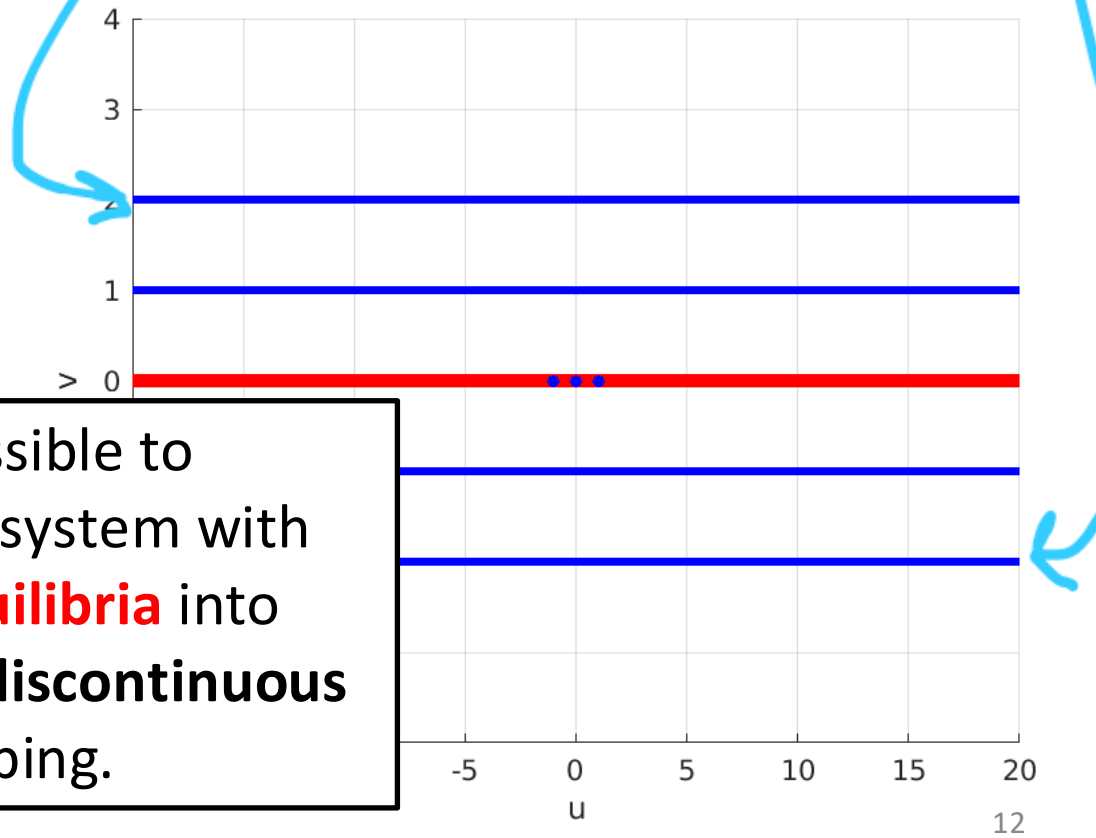
$$\dot{x} = \frac{x(1-x^2)}{1+x^2} \xrightarrow{\psi(x)} \begin{cases} \dot{u} = v \\ \dot{v} = 0 \end{cases}$$

where

$$\psi(x) = \begin{cases} (x, 0) & f(x) = 0 \\ \left(\ln\left|\frac{3x}{-2+2x^2}\right|, 1\right) & \\ \left(2\ln\left|\frac{3x}{-2+2x^2}\right|, 2\right) & \\ \left(-\ln\left|\frac{3x}{-2+2x^2}\right|, -1\right) & \\ \left(-2\ln\left|\frac{3x}{-2+2x^2}\right|, -2\right) & x > 1. \end{cases}$$



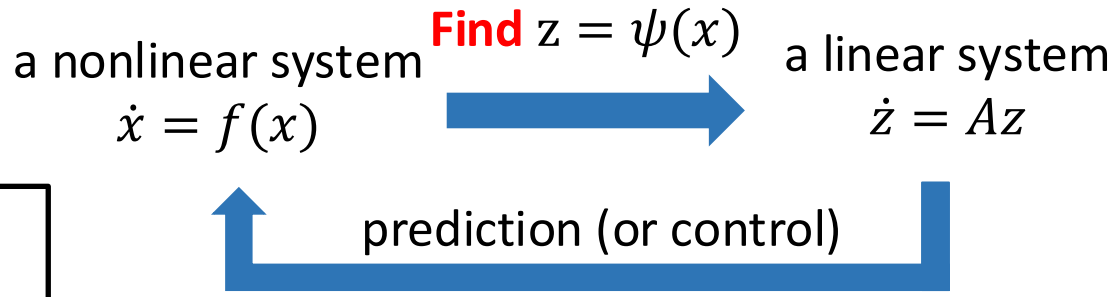
Question: Does a one-to-one $\psi(x)$ **exist or not**?



Observation: It is possible to immerse a nonlinear system with **multiple isolated equilibria** into **linear systems** via a **discontinuous** and **one-to-one** mapping.

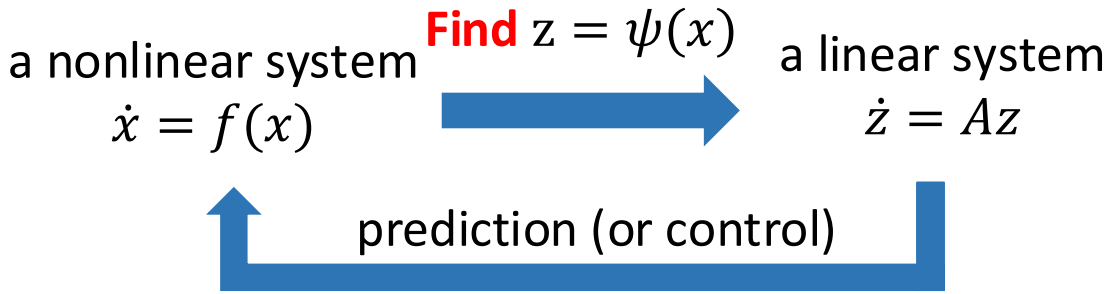
Discontinuity?

When ψ is learned via a neural network NN (with continuous activation functions), ψ is a **continuous** function.



Question: Does a one-to-one $\psi(x)$ **exist or not**?

Some definitions



Given an initial state $\xi \in \mathcal{X}$, the **ω-limit set** $\omega^+(\xi)$ is the set of all $x \in \mathcal{X}$ where there exists a sequence $t_n \to +\infty$ such that $\varphi(t_n, \xi) \to x$.

Given an initial state $\xi \in \mathcal{X}$, the trajectory $\varphi(t, \xi)$ is **precompact** *in* $\mathcal{X}$ if the closure of the set $\{\varphi(t, \xi) | t \geq 0\}$ is compact with respect to the subspace topology on $\mathcal{X}$.

For each ω-limit set Ω, its **domain of attraction** is defined by $D^+(\Omega) := \{\xi \in \mathcal{X} \mid \omega^+(\xi) = \Omega\}$.

Question: A **linear system** does not possess **more than one isolated ω-limit sets**. Can nonlinear systems with **multiple isolated ω-limit sets** be immersed into **linear systems** via **continuous** and **one-to-one** mappings?

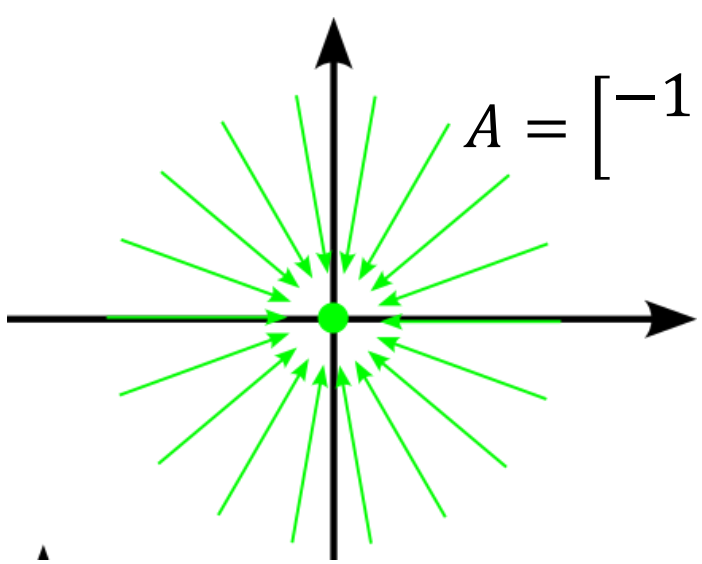
Systems with closed basins

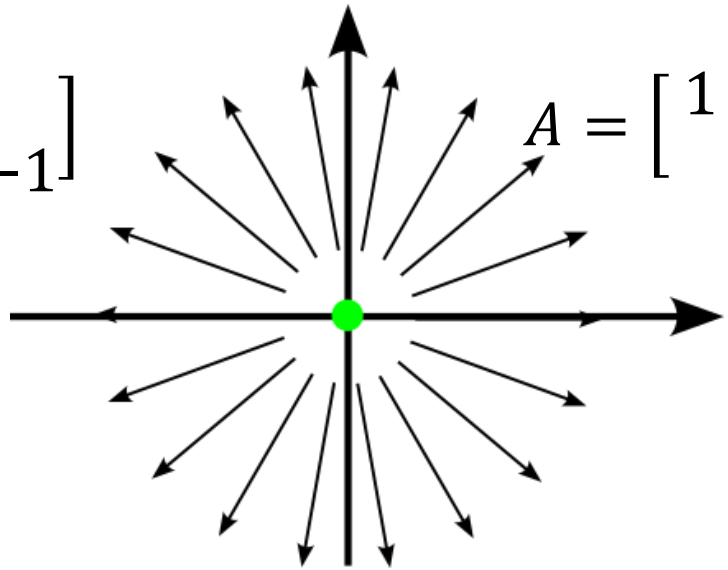
a nonlinear system $\dot{x} = f(x)$ — **Find** $z = \psi(x)$ → a linear system $\dot{z} = Az$

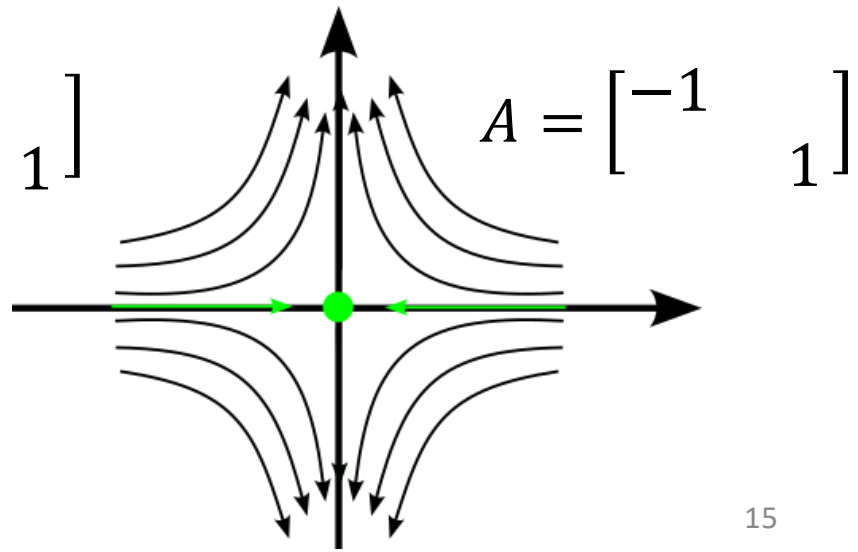
prediction (or control)

Defn. A system $\dot{x} = f(x)$ on $\mathcal{X}$ has ***closed basins*** if the domain of attraction $D^+(\Omega)$ is closed for all ω-limit set Ω in $\mathcal{X}$.

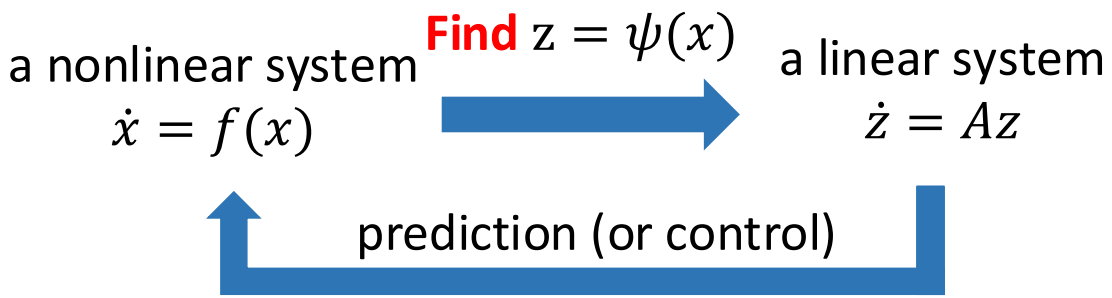
Lemma: Every linear system with $\mathcal{X} = \mathbb{R}^n$ has closed basins.







On the non-existence of linear immersion



Main result:

Theorem: Under the assumptions on $\mathcal{X}$ (connected, forward invariant), suppose that:

(T1) $\dot{x} = f(x)$ on $\mathcal{X}$ can be immersed in a system with closed basins by a one-to-one continuous mapping ψ;

(T2) trajectories of $\dot{x} = f(x)$ on $\mathcal{X}$ are precompact in $\mathcal{X}$;

(T3) the set $\mathcal{W}$ of all the ω-limit sets in $\mathcal{X}$ is finite or countable.

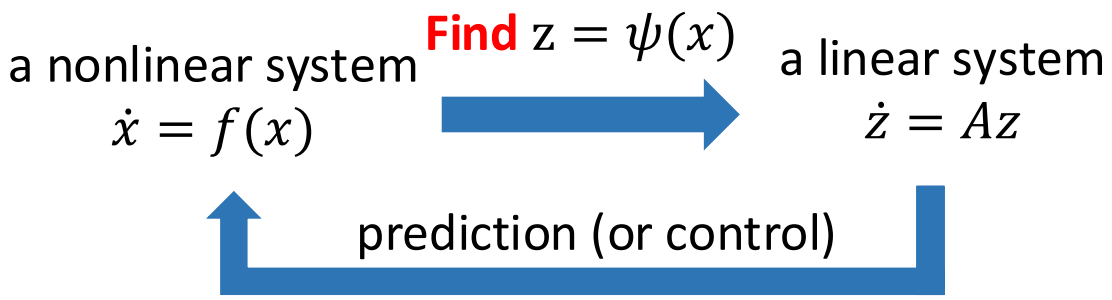
Then $\mathcal{W}$ has exactly one element.

In other words,

- if ψ is continuous and one-to-one, and
- if trajectories of $\dot{x} = f(x)$ on $\mathcal{X}$ are precompact in $\mathcal{X}$
- if f has countable but more than one ω-limit sets in $\mathcal{X}$

Then, f cannot be immersed in a linear system!

On the non-existence of linear immersion



Main result "less formal":

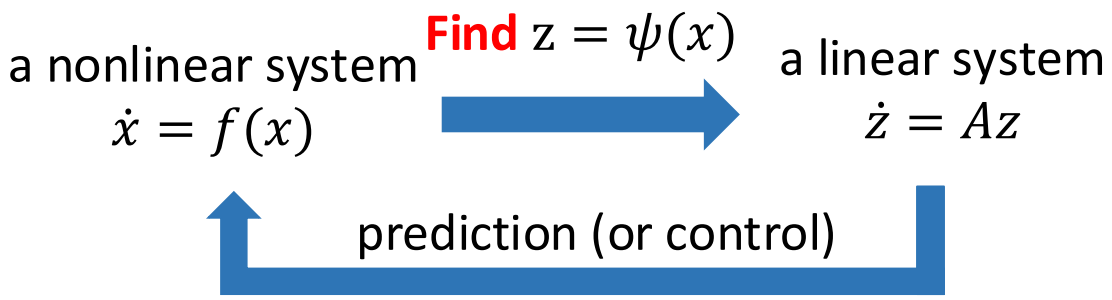
- if ψ is continuous and one-to-one, and
- if trajectories of $\dot{x} = f(x)$ on $\mathcal{X}$ are precompact (or backward precompact) in $\mathcal{X}$
- if f has countable but more than one ω-limit sets in $\mathcal{X}$

Then, f cannot be immersed in a linear system with ψ!

These conditions are not only sufficient for non-existence but also necessary in the sense that

- if ψ is discontinuous but other conditions hold, we saw an example of a 1-1 linear immersion
- if there are trajectories that are neither precompact nor backward precompact, there exists a system with countable ω-limit sets that admits a continuous 1-1 linear immersion (example due to Kvalheim and Arathoon)
- if f has uncountably many ω-limit sets, there exist examples with continuous 1-1 linear immersions

On the non-existence of linear immersion



Main result "less formal":

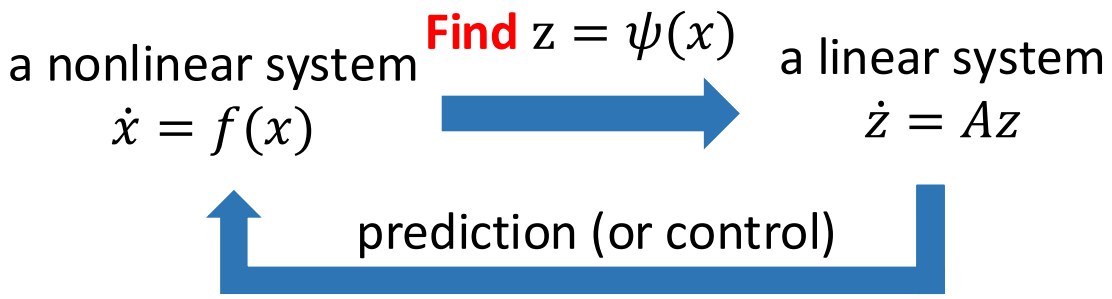
- if ψ is continuous and one-to-one, and
- if trajectories of $\dot{x} = f(x)$ on $\mathcal{X}$ are precompact (or backward precompact) in $\mathcal{X}$
- if f has countable but more than one ω-limit sets in $\mathcal{X}$

Then, f cannot be immersed in a linear system with ψ!

Implications for finding/learning $\boldsymbol{\psi(x)}$: If using a NN with continuous activation functions:

- All omega limit sets will collapse to one (cannot be inverted to predict x)
- If we embed x in ψ (i.e., $\psi(x) = [x; \phi(x)]$), need discontinuities (can lead to high Lipschitz constants for the NN, degrading robustness)

More examples

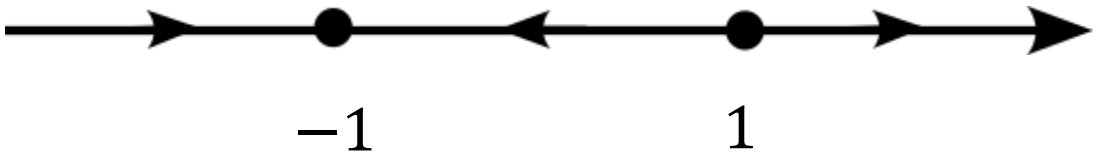


Consider the 1-d system

$$\dot{x} = x^2 - 1.$$

The ω-limit sets are $x = 1$ and $x = -1$.

Let $\mathcal{X} = (-\infty, 1)$. Then,



$$\dot{x} = x^2 - 1 \quad \longrightarrow \quad \dot{z} = -2z$$

$$\psi(x) = \frac{x+1}{x-1}$$

If we extend $\mathcal{X}$ by a point to $\mathcal{X}' = (-\infty, 1]$, ψ is not an immersion anymore as $\psi(1)$ is undefined.

More examples

a nonlinear system $\dot{x} = f(x)$ **Find** $z = \psi(x)$ a linear system $\dot{z} = Az$

prediction (or control)

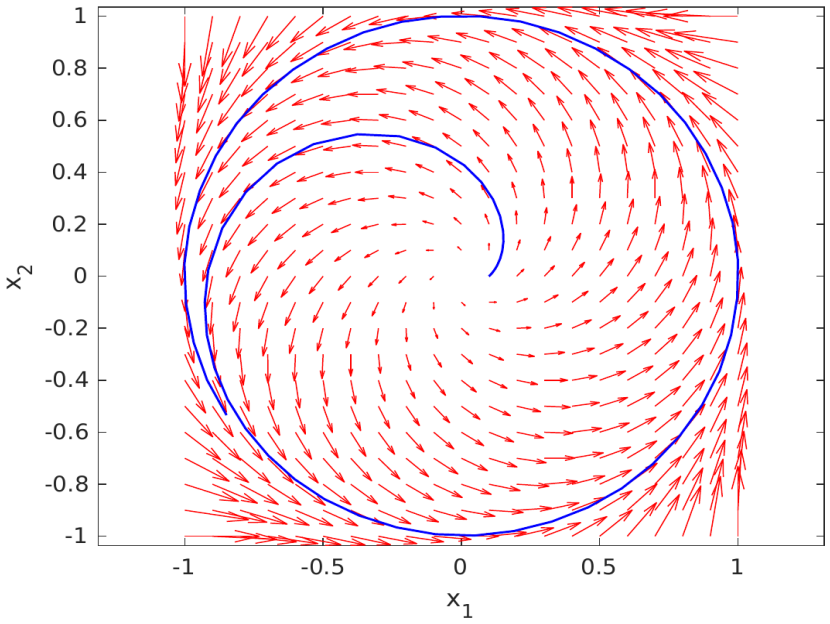
Consider the 2-d system

$$\begin{cases} \dot{x}_1 = x_1 - x_2 - x_1(x_1^2 + x_2^2) \\ \dot{x}_2 = x_1 + x_2 - x_2(x_1^2 + x_2^2) \end{cases}$$

The ω-limit sets are:

- the origin $\{0\}$ and the unit circle $\{x \mid \|x\|_2 = 1\}$.

Let $\mathcal{X} = \mathbb{R}^2/\{0\}$. Then,



$$\begin{cases} \dot{x}_1 = x_1 - x_2 - x_1(x_1^2 + x_2^2) \\ \dot{x}_2 = x_1 + x_2 - x_2(x_1^2 + x_2^2) \end{cases} \longrightarrow \begin{cases} \dot{u} = -v \\ \dot{v} = u \\ \dot{w} = -2w \end{cases}$$

$$\psi(x) = \left(\frac{x_1}{\|x\|_2}, \frac{x_2}{\|x\|_2}, \|x\|_2^{-2} - 1 \right)$$

If we extend $\mathcal{X}$ by a point to $\mathbb{R}^2$,ψ is not an immersion anymore as $\psi(0)$ is undefined.

Implications on learning

Formally, given data $\{x_l, x_l^+\}_{l=1}^N$

$$(\psi^*, A^*) \in \underset{\substack{\text{contin.}\psi \\ A\in\mathbb{R}^{m\times m}}}{\operatorname{argmin}} \sum_{l=1}^{N} \|\psi(x_l^+) - e^{A\tau}\psi(x_l)\|$$

$$\text{s.t. } x_l^+ = \varphi(\tau, x_l)\ \forall l \in [1, N]$$

τ: sampling time.

N: number of sampled states

φ: flow of the nonlinear system

Implications on learning

Formally, given data $\{x_l, x_l^+\}_{l=1}^N$

$$(\psi^*, A^*) \in \underset{\substack{\text{contin.}\psi \\ A\in\mathbb{R}^{m\times m}}}{\operatorname{argmin}} \sum_{l=1}^{N} \|\psi(x_l^+) - e^{A\tau}\psi(x_l)\|$$

$$\text{s.t.}\, x_l^+ = \varphi(\tau, x_l)\ \forall l \in [1, N]$$

τ: sampling time.

N: number of sampled states

φ: flow of the nonlinear system

The set of optima:

$$\mathcal{F}^*(\tau, N) = \{the\ set\ of\ cont.interpolating\ func.for\ the\ data\ set\ \{x_l, x_l^+\}_{l=1}^N\}$$

Implications on learning

Formally, given data $\{x_l, x_l^+\}_{l=1}^N$

$$(\psi^*, A^*) \in \underset{\substack{\text{contin.}\psi \\ A\in\mathbb{R}^{m\times m}}}{\operatorname{argmin}} \sum_{l=1}^{N} \|\psi(x_l^+) - e^{A\tau}\psi(x_l)\|$$

$$\text{s.t. } x_l^+ = \varphi(\tau, x_l)\ \forall l \in [1, N]$$

τ: sampling time.
N: number of sampled states
φ: flow of the nonlinear system

The set of optima:

$$\mathcal{F}^*(\tau, N) = \{the\ set\ of\ cont.interpolating\ func.for\ the\ data\ set\ \{x_l, x_l^+\}_{l=1}^N\}$$

$\mathcal{F}^*(\tau, N)$ is non-empty because there is a trivial solution: ψ with $\psi(x) = 0,\ \forall x.$
This ψ is always contained by $\mathcal{F}^*(\tau, N)$ for any τ and N

If you want to eliminate trivial solutions, you can add regularization or additional constraints to make $\mathcal{F}^*$ smaller.

Implications on learning

Implication on learning formalized:

$$\mathcal{F}^*(\tau, N) = \{the\ set\ of\ cont.interpolating\ func.for\ the\ data\ set\ \{x_l, x_l^+\}_{l=1}^N\}$$

Theorem: Let $\{x_l\}_{l=1}^{\infty}$ be a dense subset of $\mathcal{X}$. Let ψ be a continuous function such that $\psi(\Omega_1) \neq \psi(\Omega_2)$ for some ω-limit sets Ω_1 and Ω_2 of the system. Then, there exists a small enough $t^* > 0$ such that for all sampling times $\tau \in (0, t^*]$, ψ is not in $\mathcal{F}^*(\tau, N)$ *for some* $N > 0$.

Intuitively, any continuous function that ***can*** distinguish different ω-limit sets cannot be an optimizer of the learning problem as more data is collected.

Conclusions

Key contribution: Under mild conditions, for nonlinear systems **with multiple isolated ω-limit sets**, there are **NO** linear immersions that are **continuous** and **one-to-one** (even if A is low rank).

Implications for finding/learning $\psi(x)$: When **multiple ω-limit sets** are present, a valid one-to-one $\psi(x)$ needs to be **discontinuous** at certain points.

$\Rightarrow$ Continuous parameterization will not work. **Hybrid** structure is required!

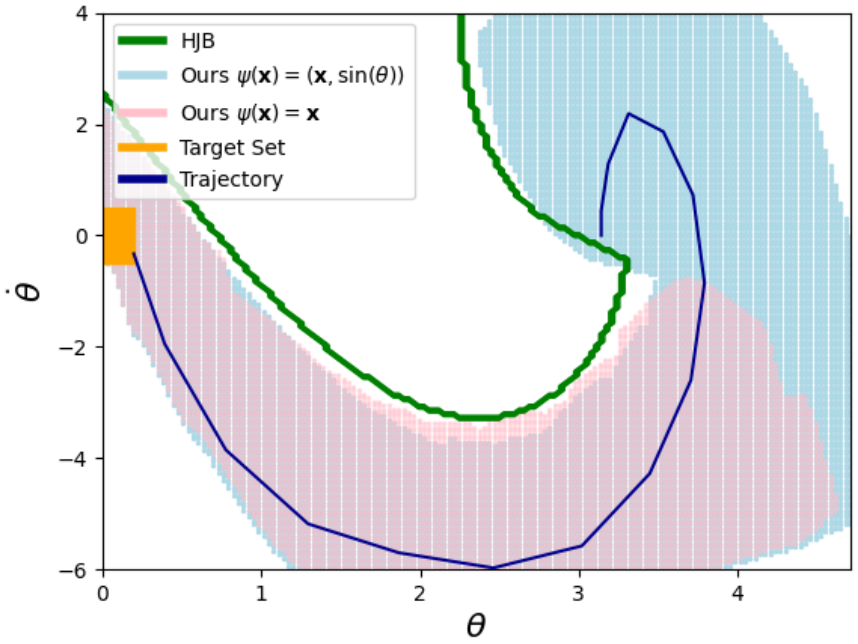
Current work (for control systems):

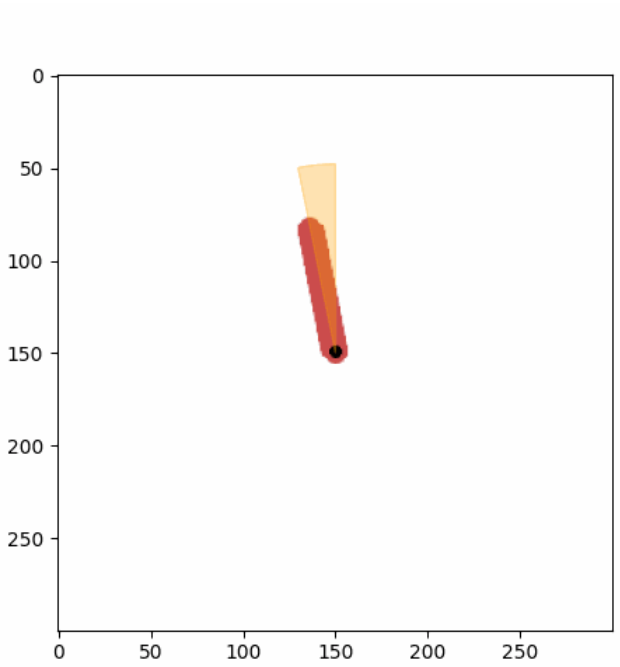
- Learn **local** and approximate Koopman liftings
  - Applications in safety constrained reachability (Balim, Aspeel, Liu, Ozay, L-CSS'23)
- Given multiple approximate Koopman liftings, which one is better?
  - A preorder relation between approximate Koopman liftings (Aspeel, Ozay, ADHS'24)
- Extension to discrete-time systems (Ristich, Sontag, Ozay, IFAC WC'26)

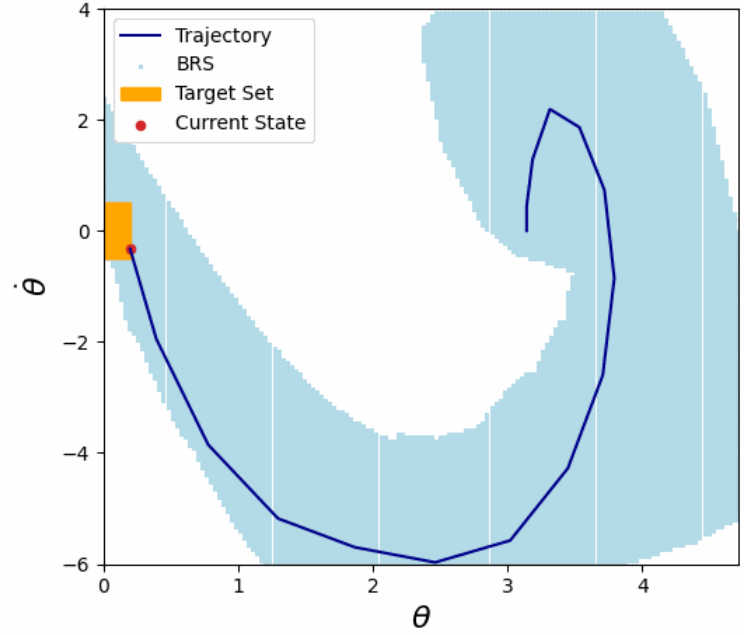
Conclusions

Current work (for control systems):

- Learn **local** and approximate Koopman liftings
  - Applications in safety constrained reachability (Balim, Aspeel, Liu, Ozay, L-CSS'23)
- Given multiple approximate Koopman liftings, which one is better?
  - A preorder relation between approximate Koopman liftings (Aspeel, Ozay, ADHS'24)

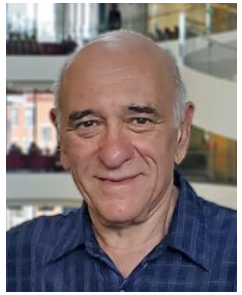






Outline

- The role of learning in dynamics and control
- On some fundamental limitations of learning immersions (Koopman embeddings)



Joint work with
Zexiang Liu (Michigan) and Eduardo Sontag (Northeastern)
arXiv:2312.17045 **(IFAC WC'23, Automatica'25)**

- **On the hardness of learning to stabilize linear systems**

Joint work with
Xiong Zeng (Michigan), Zexiang Liu (Michigan), Zhe Du (Michigan), and Mario Sznaier (Northeastern)
arXiv:2311.11151 **(CDC'23)**

How to measure hardness?

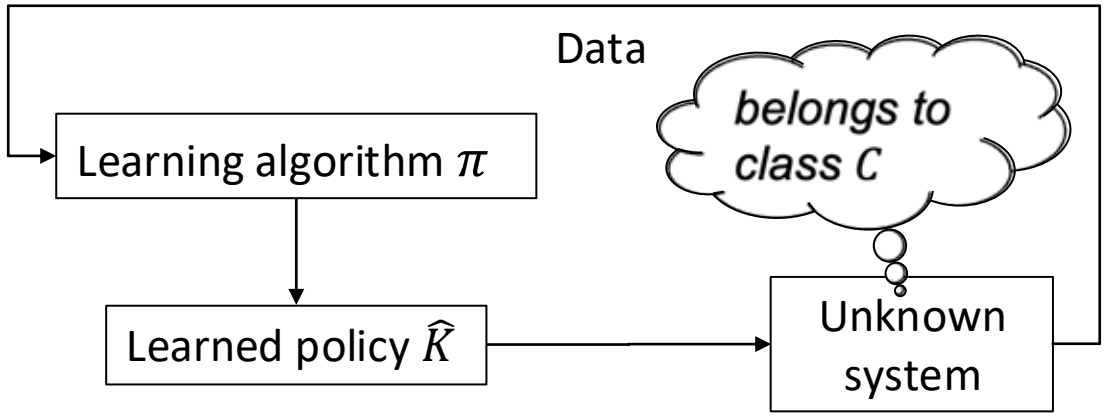
- Computational complexity: how the **computational cost** of an algorithm depends on the **input size**?

  Comp. Cost = $\mathcal{O}$(**input size**)

- Sample complexity: how a **performance metric** of a *learning* algorithm depends on the **sample size** (# of data points)?

  Performance = $\mathcal{O}$(**sample size**)

  Examples of performance metric:
  approximation error,
  cumulative cost or regret,
  probability of stability (this work)



complexity of a problem ≈ complexity of the best algorithm

Setup

Given

- A discrete-time linear time-invariant (LTI) system S from **a class $\mathcal{C}_n$**:

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

where $x_t \in \mathbb{R}^n$ and $w_t \sim \mathcal{N}(0, \sigma_w^2 \boldsymbol{I})$

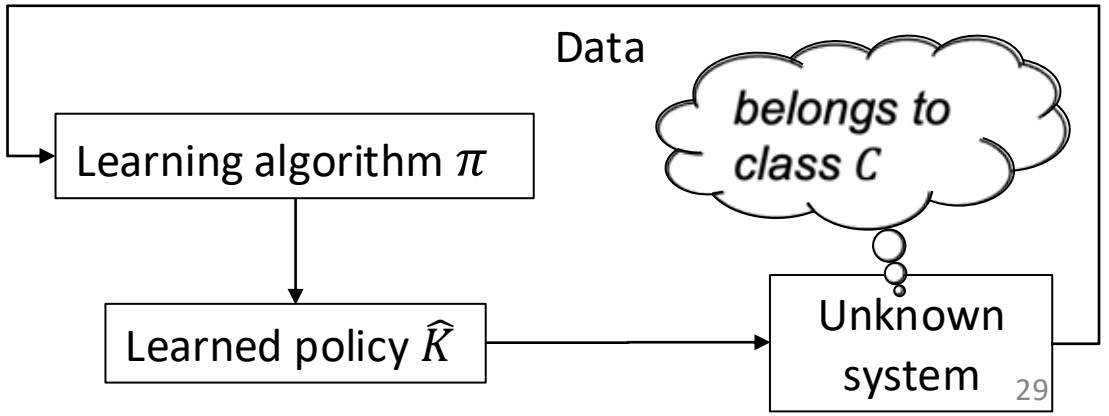
- A time budget N and an input budget $\mathbb{E}[\|u_t\|^2] \leq \sigma_u^2$

Consider a learning algorithm π that

(i) interacts with the system for N units of time, and

(ii) outputs a linear static state-feedback controller $\widehat{K}$ at time N

We want $\widehat{K}$ to stabilize the (unknown) system S.



Setup

Given

- A discrete-time linear time-invariant

$$x_{t+1} =$$

where $x_t \in \mathbb{R}^n$ and $w_t \sim \mathcal{N}(0, \sigma_w^2 \boldsymbol{I})$

- A time budget N and an input budge

Use your favorite algorithm π:

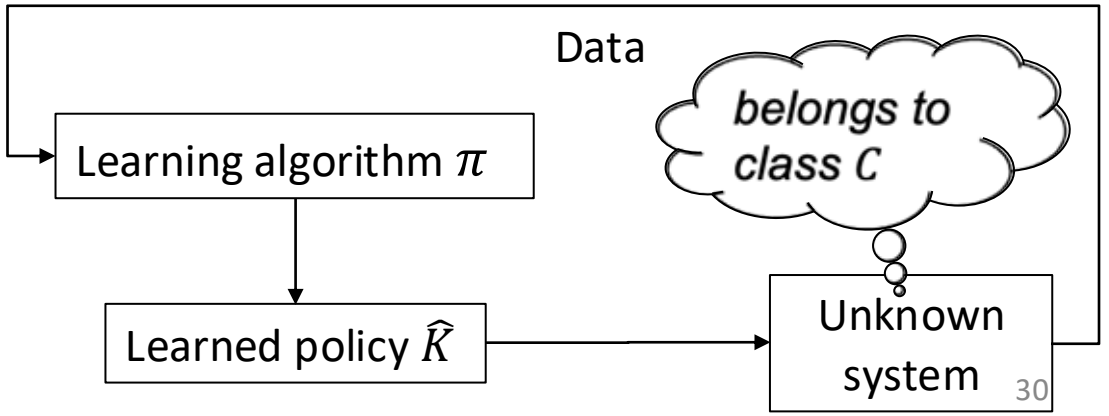
- Excite the system for N steps and do (direct) data-driven control
- Excite the system for N steps and do system id and certainty equivalent control
- Use any other online/active learning method

Consider a learning algorithm π that

(i) interacts with the system for N units of time, and

(ii) outputs a linear static state-feedback controller $\widehat{K}$ at time N

We want $\widehat{K}$ to stabilize the (unknown) system S.



Problem statement

Denote the probability that $\widehat{K}$ stabilizes an unknown system $S = (A, B) \in \mathcal{C}_n$ by

$$P_{S,\pi}^{N}\left(\left(A + B\,\widehat{K}\right)\text{is stable}\right)$$

A learning algorithm π is evaluated by the **worst-case** stabilization probability over $\mathcal{C}_n$:

$$\inf_{S \in \mathcal{C}_n} P_{S,\pi}^{N}((A + B\,\widehat{K}) \text{ is stable})$$

Sample complexity of π: Given a $\delta \in [0, 1]$, the **minimum time/data budget N** such that

$$\inf_{S \in \mathcal{C}_n} P_{S,\pi}^{N}((A + B\,\widehat{K}) \text{ is stable}) \geq 1 - \delta.$$

Question: Given a class $\mathcal{C}$ of linear systems, how does the **sample complexity** for learning stabilizing static state-feedback controllers depend on the **system dimension n**?

Roadmap to establish hardness

Find a class $\mathcal{C}_n$ of systems such that for any algorithm π and for all $\delta \in [0,0.5]$, we have

$$\inf_{S \in \mathcal{C}_n} P_{S,\pi}^{N}((A + B\,\widehat{K}) \text{ is stable}) \geq 1 - \delta.$$

only if

$$N \geq O(exp(n)).$$

Key insight: if the class $\mathcal{C}_n$ contains a pair of systems (A_1, B_1) and (A_2, B_2) that are

- hard to distinguish, and
- not co-stabilizable

then learning to stabilize is hard.

Roadmap to establish hardness

Find a class $\mathcal{C}_n$ of systems such that for any algorithm π and for all $\delta \in [0,0.5]$, we have

$$\inf_{S \in \mathcal{C}_n} P_{S,\pi}^{N}((A + B\,\widehat{K}) \text{ is stable}) \geq 1 - \delta.$$

only if

$$N \geq O(exp(n)).$$

Key insight: if the class $\mathcal{C}_n$ contains a pair of systems (A_1, B_1) and (A_2, B_2) that are

- hard to distinguish, and

  when driven by the same algorithm π, the resulting state trajectories are similar

- not co-stabilizable

  there does not exist a single controller that simultaneously stabilizes the two systems

then learning to stabilize is hard. $\nexists\, K$ s.t. both $A_1 + B_1 K$ *and* $A_2 + B_2 K$ are stable

A pair of systems that are hard to learn to stabilize

$$\mathbf{A} = \begin{bmatrix} r & v & 0 & \cdots & 0 \\ 0 & 0 & v & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \cdots & v \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}, \ \mathbf{B} = \begin{bmatrix} b^{(1)} \\ 0 \\ \vdots \\ 0 \\ v \end{bmatrix} \in \mathbb{R}^n$$

$x_{t+1} = Ax_t + Bu_t + w_t, \quad r > 1, 0 < v < \frac{r-1}{2}$, and $b^{(1)} \geq 0$.

First order system with actuation delay of length n-1

Take $\mathrm{B}_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ v \end{bmatrix}, \mathrm{B}_2 = \begin{bmatrix} m \\ 0 \\ \vdots \\ v \end{bmatrix}$

A pair of systems that are hard to learn to stabilize

$$\mathbf{A} = \begin{bmatrix} r & v & 0 & \cdots & 0 \\ 0 & 0 & v & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \cdots & v \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}, \ \mathbf{B} = \begin{bmatrix} b^{(1)} \\ 0 \\ \vdots \\ 0 \\ v \end{bmatrix} \in \mathbb{R}^n$$

$x_{t+1} = Ax_t + Bu_t + w_t, \quad r > 1, 0 < v < \frac{r-1}{2}$, and $b^{(1)} \geq 0$.

First order system with actuation delay of length n-1

Take $\mathrm{B}_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ v \end{bmatrix}, \mathrm{B}_2 = \begin{bmatrix} m \\ 0 \\ \vdots \\ v \end{bmatrix}$

Proposition: There exists a state-feedback controller K that stabilizes both (A, B_1) and (A, B_2) only if

$$m \leq \mathcal{O}\left(\frac{1}{\exp(n)}\right)$$

A pair of systems that are hard to learn to stabilize

Proposition: There exists a state-feedback controller K that stabilizes both (A, B_1) and (A, B_2) only if

$$m \leq \mathcal{O}\left(\frac{1}{\exp(n)}\right)$$

Proof sketch: Use Ackerman's formula to analytically construct K to place the poles of system 1 arbitrarily inside the unit circle and check the stability of system 2 when controlled by K using Jury stability test.

A pair of systems that are hard to learn to stabilize

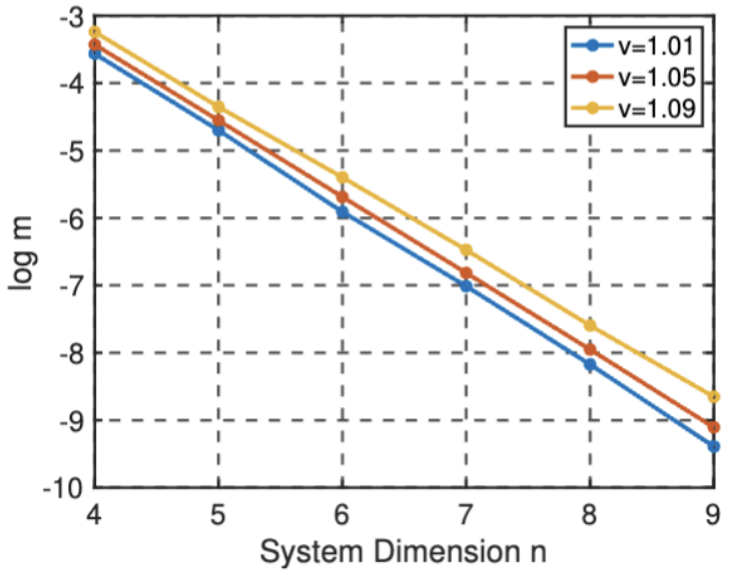
Proposition: There exists a state-feedback controller K that stabilizes both (A, B_1) and (A, B_2) only if

$$m \leq \mathcal{O}\left(\frac{1}{\exp(n)}\right)$$

Proof sketch: Use Ackerman's formula to analytically construct K to place the poles of system 1 arbitrarily inside the unit circle and check the stability of system 2 when controlled by K using Jury stability test.

We can also do a sanity check numerically using LMI-based sufficient condition for co-stabilizability

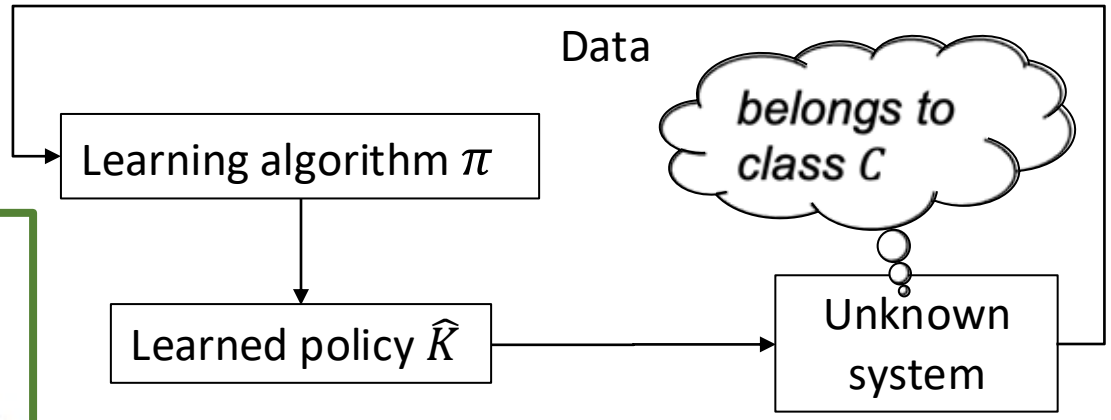
$$\begin{aligned} find \quad & K, P \\ s.t. \quad & (A + B_1 K)^T P (A + B_1 K) \prec P \\ & (A + B_2(m) K)^T P (A + B_2(m) K) \prec P \\ & P \succ 0 \end{aligned}$$



Main result

$$\mathbf{A} = \begin{bmatrix} r & v & 0 & \cdots & 0 \\ 0 & 0 & v & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \cdots & v \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \ \mathbf{B} = \begin{bmatrix} b^{(1)} \\ 0 \\ \vdots \\ 0 \\ v \end{bmatrix} \in \mathbb{R}^{n}$$

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad r > 1, 0 < v < \tfrac{r-1}{2}, \text{ and } b^{(1)} \geq 0.$$



Take $\mathrm{B}_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ v \end{bmatrix}, \mathrm{B}_2 = \begin{bmatrix} m_n \\ 0 \\ \vdots \\ v \end{bmatrix}$

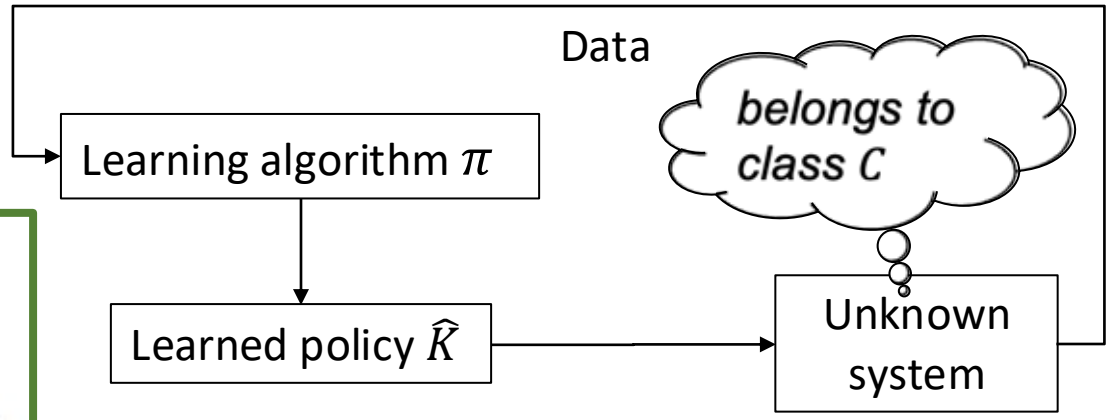
Theorem: For any class C of systems containing such a pair with $m_n = 2\left(\frac{2v}{r-1}\right)^n$ and any learning algorithm, the least amount of data for learning to stabilize arbitrary systems in C grows exponentially with the state dimension!

$$\inf_{S\in C_n} P^N_{S,\pi}((A + B\,\widehat{K}) \text{ is stable}) \geq 1 - \delta \quad \text{only if} \quad N \geq \frac{\sigma_w^2}{2\sigma_u^2}\left(\frac{r-1}{2v}\right)^{2n} \log\left(\frac{1}{3\delta}\right)$$

Main result

$$\mathbf{A} = \begin{bmatrix} r & v & 0 & \cdots & 0 \\ 0 & 0 & v & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \cdots & v \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \ \mathbf{B} = \begin{bmatrix} b^{(1)} \\ 0 \\ \vdots \\ 0 \\ v \end{bmatrix} \in \mathbb{R}^{n}$$

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad r > 1, 0 < v < \tfrac{r-1}{2}, \text{ and } b^{(1)} \geq 0.$$



Take $\mathrm{B}_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ v \end{bmatrix}, \mathrm{B}_2 = \begin{bmatrix} m_n \\ 0 \\ \vdots \\ v \end{bmatrix}$

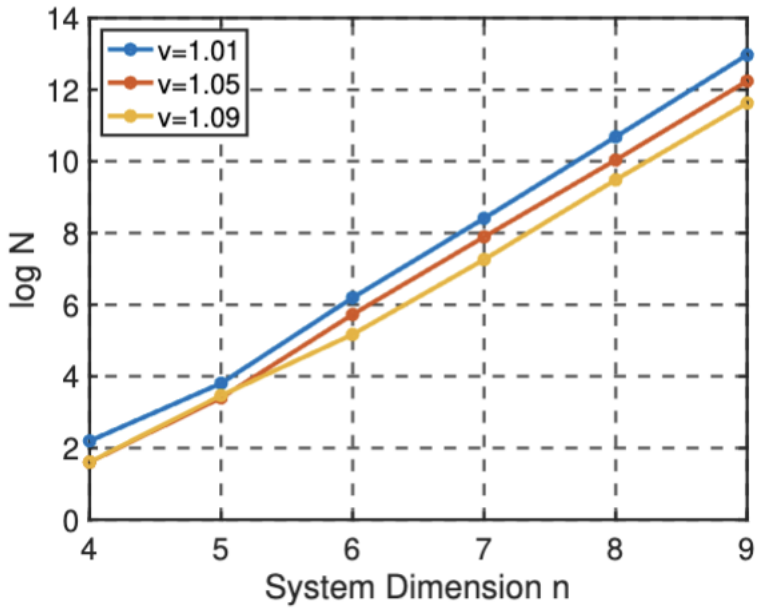
Theorem: For any class C of systems containing such a pair with $m_n = 2\left(\frac{2v}{r-1}\right)^n$ (< 1) and any learning algorithm, the least amount of data for learning to stabilize arbitrary systems in C grows exponentially with the state dimension!

$$\inf_{S\in C_n} P^N_{S,\pi}((A + B\,\widehat{K}) \text{ is stable}) \geq 1 - \delta \quad \text{only if} \quad N \geq \underbrace{\frac{\sigma_w^2}{2\sigma_u^2}}_{1/\mathrm{SNR}} \left(\frac{r-1}{2v}\right)^{2n} \log\left(\frac{1}{3\delta}\right)$$

exp in dimension n

Numerical illustration

Number of samples required for 90% probability of finding a stabilizing controller using certainty equivalent LQR*:



y-axis: observed least sample size N for CE LQR being stabilizing with probability 0.9

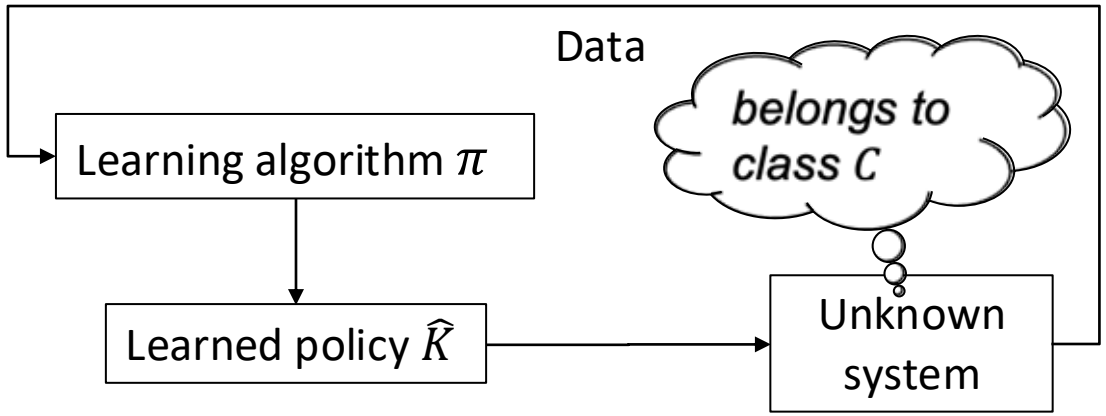
Certainty Equivalent LQR

- Excite the system with iid Gaussian input for N steps
- Learn system matrices
- Do LQR with learned model

$$\min_{u_0,u_{1,\dots}} \lim_{T\to\infty} E[\frac{1}{T}\sum_{t=0}^{T}(x_t^T Q x_t + u_t^T R u_t)]$$
$$s.t.\, x_{t+1} = Ax_t + Bu_t + w_t$$

* By our result, same trend holds for any algorithm

Conclusions



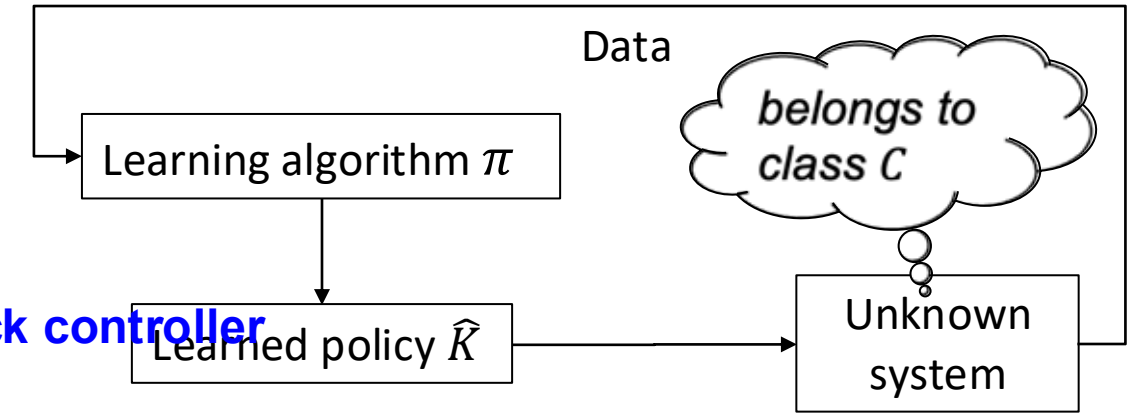
Key contributions:

- Learning to stabilize a linear system is hard if we have systems that are

  hard to distinguish AND hard to co-stabilize

- Compared with earlier work by Tsiamis et al., our analysis
  - reveals a new obstruction for the hardness of learning to stabilize;
  - covers a larger class of systems (including open-loop unstable ones) – encompasses many examples from literature where some hardness was observed

Conclusions



Key contributions: with a static linear state-feedback controller

- Learning to stabilize a linear system is hard if we have systems that are

  hard to distinguish AND hard to co-stabilize

- Compared with earlier work by Tsiamis et al., our analysis
  - reveals a new obstruction for the hardness of learning to stabilize;
  - covers a larger class of systems (including open-loop unstable ones) – encompasses many examples from literature where some hardness was observed

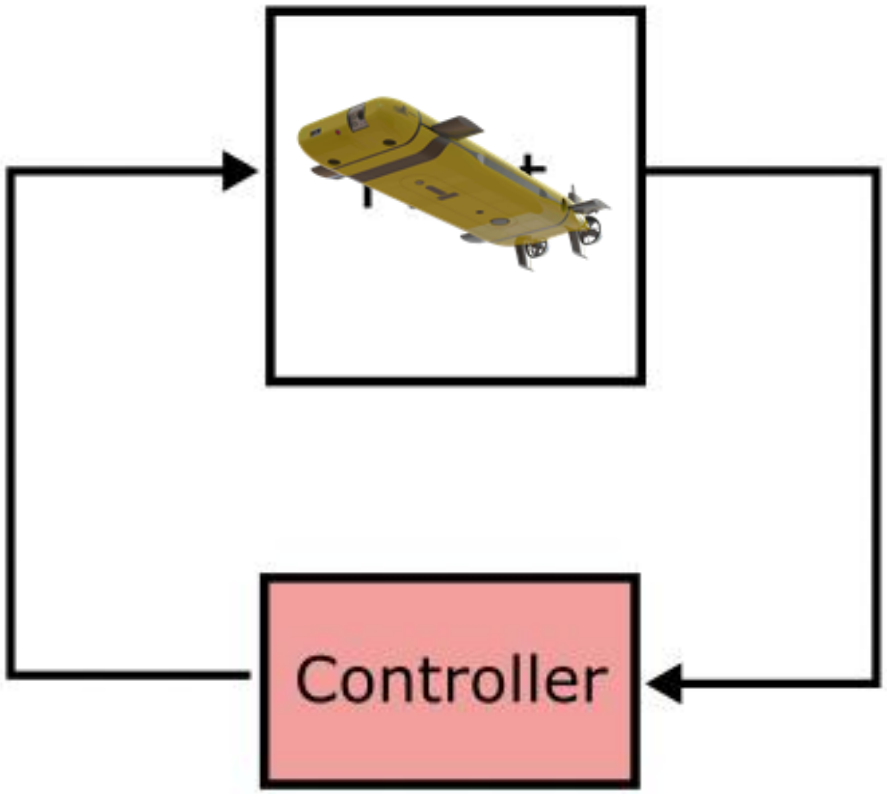
Current work:

- Observation: Static linear state feedback controllers are not a rich enough class for co-stabilizing linear systems (memory helps!)
- Can we reduce sample complexity by using dynamic time-varying controllers?

Plant (+ operating environment)

Task or mission specification

φ



Many components in a feedback loop can be learned:

- **Plant model:** model-based learning, system identification
- **Task specification:** inverse optimal control, inverse reinforcement learning
- **Controller:** model-free learning
- **Perception/estimation modules**

Learn but be aware of the fundamental limits (e.g., hardness and impossibility results) ➔ Learn better

More info and preprints @ https://web.eecs.umich.edu/~necmiye/