Safety Assurance in Learning-enabled Autonomous Systems

Saber Jafarpour

University of Colorado Boulder

March 5, 2024

Energy/power systems

Air mobility

Autonomous driving

Manufacturing

Transportation systems

Agriculture

Safety-critical Autonomous Systems

Introduction

Energy/power systems

Air mobility

Autonomous driving

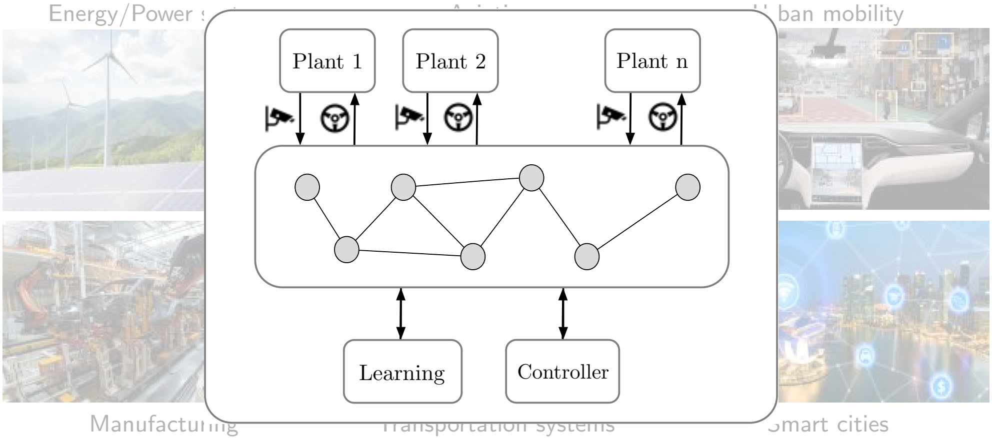
Manufacturing

Transportation systems

Agriculture

An important goal (Safe Autonomy)

Perform their tasks while ensuring **safety** and **robustness** of the system.

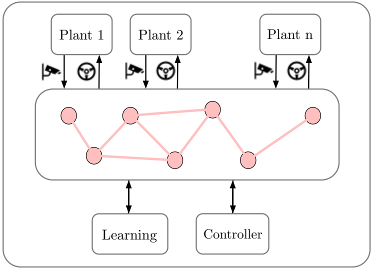


An important goal (Safe Autonomy)

Perform their tasks while ensuring **safety** and **robustness** of the system.

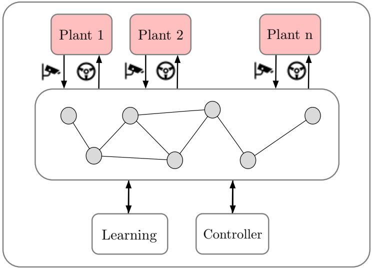
Challenges for ensuring **safety** in autonomous systems:

1. large number of agents
2. complex and highly nonlinear components
3. uncertain environment with unmodeled dynamics



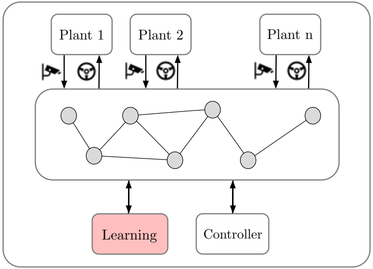
Challenges for ensuring **safety** in autonomous systems:

1. large number of agents
2. complex and highly nonlinear components
3. uncertain environment with unmodeled dynamics



Challenges for ensuring **safety** in autonomous systems:

1. large number of agents
2. complex and highly nonlinear components
3. uncertain environment with unmodeled dynamics



Challenges for ensuring **safety** in autonomous systems:

1. large number of agents
2. complex and highly nonlinear components
3. uncertain environment with unmodeled dynamics

My Research

Different aspect of autonomy with safety and robustness considerations

Tools: Systems and Control (dynamical systems, optimization theory)

Research summary

My past and current research

Large-scale systems

- threshold of frequency synchronization (TAC 2020, SICON 2019)
- multi-stability via partitioning the state-space (SIAM Review 2021, Nature Com 2022)
- dynamic stability of low-inertia power grids (TCNS 2019)

Optimization-based systems

- time-varying optimization (TAC 2021)
- non-Euclidean monotone operator theory (CDC 2022)

Nonlinear systems

- weak and semi-contraction theory (TAC 2021)
- non-Euclidean contraction theory (TAC 2022, TAC 2023)
- small time local controllability (SICON 2020)

Learning-enabled systems

- contraction-based reachability of neural networks (NeurIPS 2021, L4DC 2022)
- interval-based reachability of neural networks (L4DC 2023, ADHS 2024)
- safety verification of neural feedback loops (submitted 2023)

In this talk: Autonomous Systems with Learning-enabled components

In this talk: Autonomous Systems with Learning-enabled components

Machine learning was one of the deriving forces for developments

In this talk: Autonomous Systems with Learning-enabled components

Machine learning was one of the deriving forces for developments

- availability of data and computation tools
- performance and efficiency

In this talk: Autonomous Systems with Learning-enabled components

Machine learning was one of the deriving forces for developments

- availability of data and computation tools
- performance and efficiency

Success stories and potential applications

NVIDIA self driving car

Amazon fulfillment centers

Manufacturing

But can we ensure their safety?

Tesla Slams Right Into Overturned Truck While on Autopilot

Waymo driverless car strikes bicyclist in San Francisco, causes minor injuries

But can we ensure their safety?

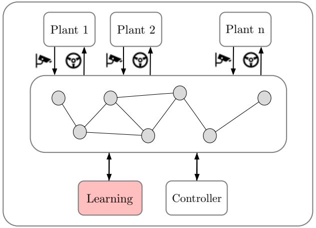
Perception-based Obstacle Avoidance

Video courtesy of Dr. Taylor Johnson at CS department of the Vanderbilt University

But can we ensure their safety?

Tesla Slams Right Into Overturned Truck While on Autopilot

What is different with Learning-based components?



But can we ensure their safety?

Tesla Slams Right Into Overturned Truck While on Autopilot

- limited guarantee in their design

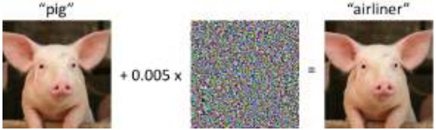


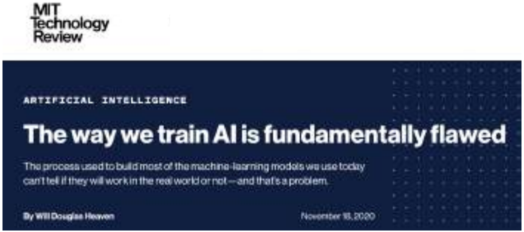
Image credit: MIT CSAIL

But can we ensure their safety?

Tesla Slams Right Into Overturned Truck While on Autopilot

Waymo driverless car strikes bicyclist in San Francisco, causes minor injuries

- limited guarantee in their design

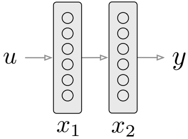


But can we ensure their safety?

Tesla Slams Right Into Overturned Truck While on Autopilot

Waymo driverless car strikes bicyclist in San Francisco, causes minor injuries

- limited guarantee in their design
- large # of parameters with nonlinearity



$478 \times 100 \times 100 \times 10$

$\#$ of parameters ~ 90000
$\#$ of activation patterns $\sim 10^{60}$

But can we ensure their safety?

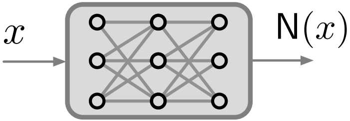
Tesla Slams Right Into Overturned Truck While on Autopilot

Waymo driverless car strikes bicyclist in San Francisco, causes minor injuries

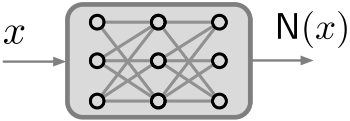
- limited guarantee in their design
- large # of parameters with nonlinearity

Rigorous and **computationally efficient** methods for safety assurance

ML focus on safety and robustness of **stand-alone** learning algorithms



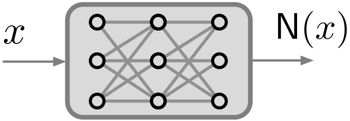
ML focus on safety and robustness of **stand-alone** learning algorithms



Different approaches:

- analysis (Goodfellow et al., 2015, Zhang et al., 2019, Fazlyab et al., 2023)
- design (Papernot et al., 2016, Carlini and Wagner, 2017, Madry et al., 2018)

ML focus on safety and robustness of **stand-alone** learning algorithms

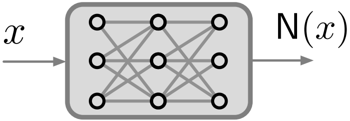


Different approaches:

- analysis (Goodfellow et al., 2015, Zhang et al., 2019, Fazlyab et al., 2023)
- design (Papernot et al., 2016, Carlini and Wagner, 2017, Madry et al., 2018)

In autonomous systems, learning algorithms are **a part of the system** (controller, motion planner, obstacle detection)

ML focus on safety and robustness of **stand-alone** learning algorithms



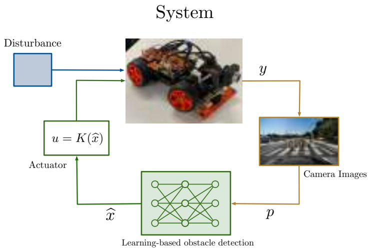
Different approaches:

- analysis (Goodfellow et al., 2015, Zhang et al., 2019, Fazlyab et al., 2023)
- design (Papernot et al., 2016, Carlini and Wagner, 2017, Madry et al., 2018)

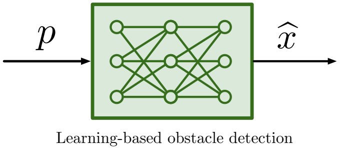
In autonomous systems, learning algorithms are **a part of the system** (controller, motion planner, obstacle detection)

New challenges arises when learning algorithms are used **in-the-loop**

Perception-based Obstacle Avoidance



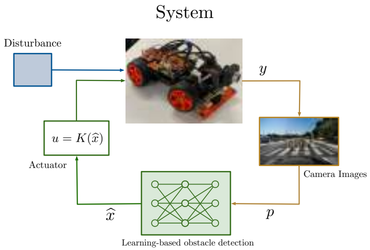
In-the-loop



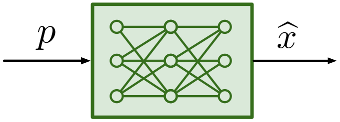
trained offline using images

Stand-alone

Perception-based Obstacle Avoidance



In-the-loop



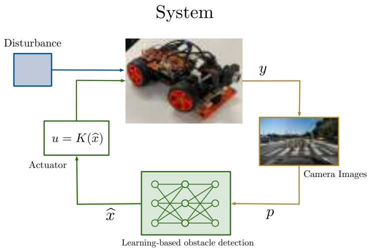
Learning-based obstacle detection

trained offline using images

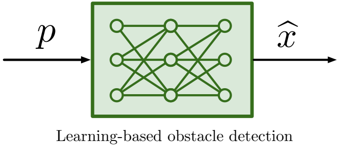
Stand-alone

- **stand-alone**: estimation of states using learning algorithm
- **in-the-loop**: closed-loop system avoid the obstacle

Perception-based Obstacle Avoidance



In-the-loop



trained offline using images

Stand-alone

- **stand-alone**: estimation of states using learning algorithm
- **in-the-loop**: closed-loop system avoid the obstacle

> **In-the-loop**: how the autonomous system perform with the learning algorithm as a part of it.

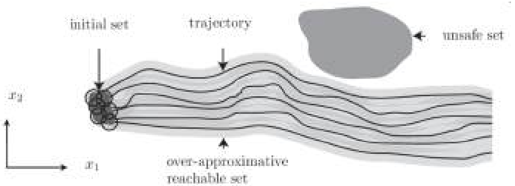
Ensure safety of the autonomous system with learning algorithms **in-the-loop**

Ensure safety of the autonomous system with learning algorithms **in-the-loop**

Safety of autonomous system using **reachability analysis**

Ensure safety of the autonomous system with learning algorithms **in-the-loop**

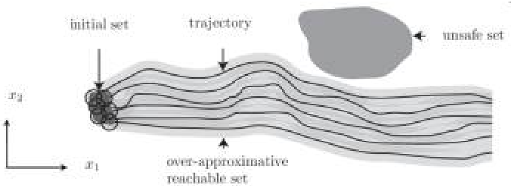
Safety of autonomous system using **reachability analysis**



Reachability analysis estimates the evolution of the autonomous system

Ensure safety of the autonomous system with learning algorithms **in-the-loop**

Safety of autonomous system using **reachability analysis**



Reachability analysis estimates the evolution of the autonomous system

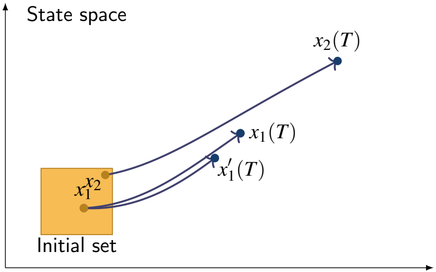
In this talk:

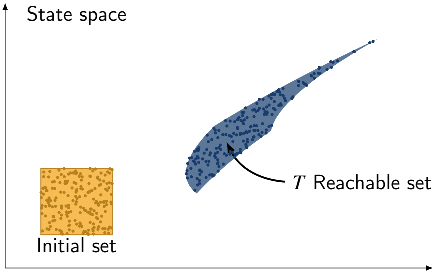
1. control-theoretic tools for efficient and scalable reachability
2. applications to safety assurance of learning-enabled systems

Outline of this talk

- **Reachability Analysis**
- Neural Network Controlled Systems
- Future Research Directions

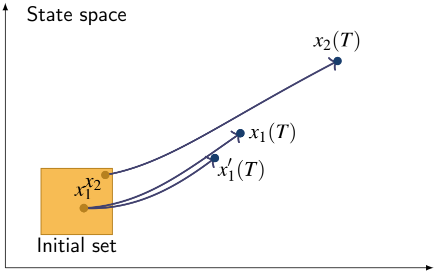
System : $\dot{x} = f(x, w)$ **State** : $x \in \mathbb{R}^n$ **Uncertainty** : $w \in \mathcal{W} \subseteq \mathbb{R}^m$

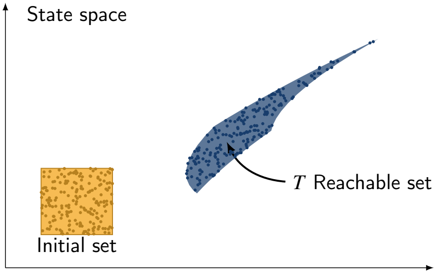




What are the possible states of the system at time T?

System : $\dot{x} = f(x, w)$ **State** : $x \in \mathbb{R}^n$ **Uncertainty** : $w \in \mathcal{W} \subseteq \mathbb{R}^m$





What are the possible states of the system at time T?

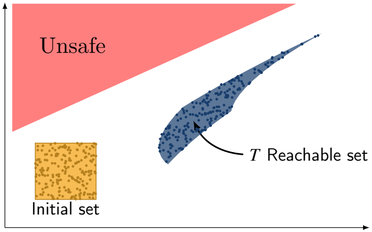
- **T-reachable sets** characterize evolution of the system

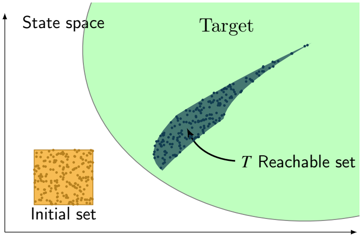
$$\mathcal{R}_f(T, \mathcal{X}_0, \mathcal{W}) = \{x_w(T) \mid x_w(\cdot) \text{ is a traj for some } w(\cdot) \in \mathcal{W} \text{ with } x_0 \in \mathcal{X}_0\}$$

A large number of **safety specifications** can be represented using T-reachable sets

A large number of **safety specifications** can be represented using T-reachable sets

- Example: Reach-avoid problem



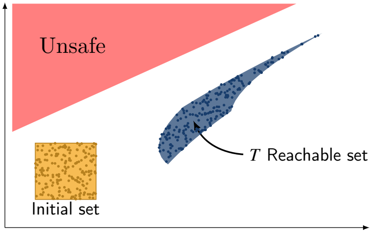


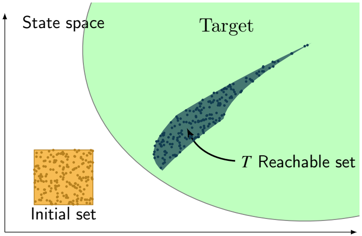
$$\mathcal{R}_f(T, \mathcal{X}_0, \mathcal{W}) \cap \text{ Unsafe set } = \emptyset$$

$$\mathcal{R}_f(T_{\text{final}}, \mathcal{X}_0, \mathcal{W}) \subseteq \text{Target set}$$

A large number of **safety specifications** can be represented using T-reachable sets

- Example: Reach-avoid problem





$$\mathcal{R}_f(T, \mathcal{X}_0, \mathcal{W}) \cap \text{ Unsafe set } = \emptyset$$

$$\mathcal{R}_f(T_{\text{final}}, \mathcal{X}_0, \mathcal{W}) \subseteq \text{Target set}$$

Combining different instantiation of Reach-avoid problem $\Longrightarrow$
diverse range of specifications
(complex planning using logics, invariance, stability)

Computing the T-reachable sets are computationally challenging

Computing the T-reachable sets are computationally challenging

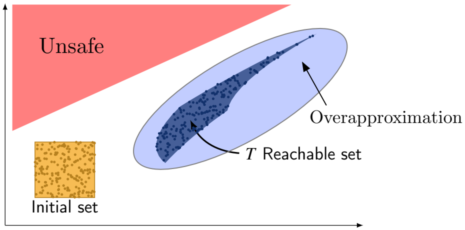
Solution: over-approximations of reachable sets

Over-approximation: $\mathcal{R}_f(T, \mathcal{X}_0, \mathcal{W}) \subseteq \overline{\mathcal{R}}_f(T, \mathcal{X}_0, \mathcal{W})$

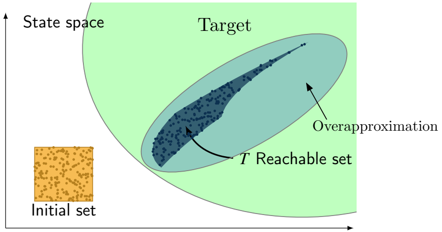
Computing the T-reachable sets are computationally challenging

Solution: over-approximations of reachable sets

Over-approximation: $\mathcal{R}_f(T, \mathcal{X}_0, \mathcal{W}) \subseteq \overline{\mathcal{R}}_f(T, \mathcal{X}_0, \mathcal{W})$

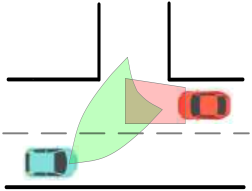


$\overline{\mathcal{R}}_f(T, \mathcal{X}_0, \mathcal{W}) \cap \text{Unsafe set} = \emptyset$



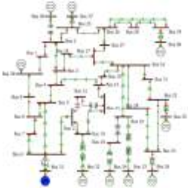
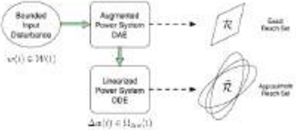
$\overline{\mathcal{R}}_f(T_{\text{final}}, \mathcal{X}_0, \mathcal{W}) \subseteq \text{Target set}$

Autonomous Driving:

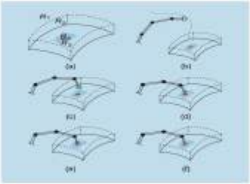
Althoff, 2014

Power grids:

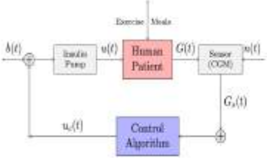



Chen and Domınguez-Garcıa, 2016

Robot-assisted Surgery:




Drug Delivery:

Chen, Dutta, and Sankaranarayanan, 2017

Reachability of dynamical system is an old problem: $\sim$ 1980

Reachability of dynamical system is an old problem: ~ 1980

Different approaches for approximating reachable sets

- Linear, and piecewise linear systems (Ellipsoidal methods) (Kurzhanski and Varaiya, 2000)
- Optimization-based approaches (Hamilton-Jacobi, Level-set method) (Bansal et al., 2017, Mitchell et al., 2002, Herbert et al., 2021)
- Matrix measure-based (Fan et al., 2018, Maidens and Arcak, 2015)

Reachability of dynamical system is an old problem: $\sim$ 1980

Different approaches for approximating reachable sets

- Linear, and piecewise linear systems (Ellipsoidal methods) (Kurzhanski and Varaiya, 2000)
- Optimization-based approaches (Hamilton-Jacobi, Level-set method) (Bansal et al., 2017, Mitchell et al., 2002, Herbert et al., 2021)
- Matrix measure-based (Fan et al., 2018, Maidens and Arcak, 2015)

Most of the classical reachability approaches are computationally heavy and not scalable to large-size systems

Reachability of dynamical system is an old problem: $\sim$ 1980

Different approaches for approximating reachable sets

- Linear, and piecewise linear systems (Ellipsoidal methods) (Kurzhanski and Varaiya, 2000)
- Optimization-based approaches (Hamilton-Jacobi, Level-set method) (Bansal et al., 2017, Mitchell et al., 2002, Herbert et al., 2021)
- Matrix measure-based (Fan et al., 2018, Maidens and Arcak, 2015)

Most of the classical reachability approaches are computationally heavy and not scalable to large-size systems

In this talk: use control-theoretic tools to develop scalable and computationally efficient approaches for reachability

Approach #1: Contraction Theory

A framework for stability analysis

$\dot{x} = f(x, w)$ is contracting wrt $\|\cdot\|$ with rate c if
the dist between every two traj is decreasing/increasing with exp rate c wrt $\|\cdot\|$

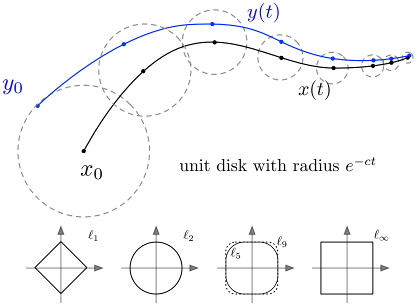
Approach #1: Contraction Theory

A framework for stability analysis

$\dot{x} = f(x, w)$ is contracting wrt $\|\cdot\|$ with rate c if
the dist between every two traj is decreasing/increasing with exp rate c wrt $\|\cdot\|$

Applications

- convergence to reference trajectories
- efficient equilibrium point computation
- input-output robustness
- entrainment to periodic orbits



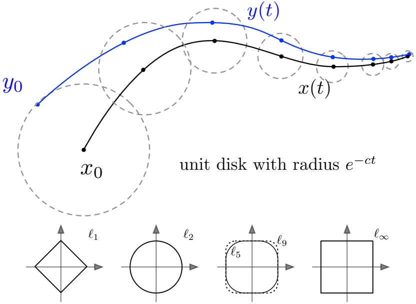
Approach #1: Contraction Theory

A framework for stability analysis

$\dot{x} = f(x, w)$ is contracting wrt $\|\cdot\|$ with rate c if
the dist between every two traj is decreasing/increasing with exp rate c wrt $\|\cdot\|$

Applications

- convergence to reference trajectories
- efficient equilibrium point computation
- input-output robustness
- entrainment to periodic orbits



In this talk: contraction theory for reachability analysis

How to characterize contractivity using vector fields?

How to characterize contractivity using vector fields?

Matrix measure

Given a matrix $A \in \mathbb{R}^{n\times n}$ and a norm $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,
- **In the literature**: one-sided Lipschitz constant, logarithmic norm

How to characterize contractivity using vector fields?

Matrix measure

Given a matrix $A \in \mathbb{R}^{n\times n}$ and a norm $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}$$

Closed-form expressions:

$$\mu_2(A) = \tfrac{1}{2}\lambda_{\max}(A + A^\top)$$
$$\mu_1(A) = \max_j \left(a_{jj} + \sum_{i\neq j} |a_{ij}|\right)$$
$$\mu_\infty(A) = \max_i \left(a_{ii} + \sum_{j\neq i} |a_{ij}|\right)$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,
- **In the literature**: one-sided Lipschitz constant, logarithmic norm

How to characterize contractivity using vector fields?

Matrix measure

Given a matrix $A \in \mathbb{R}^{n\times n}$ and a norm $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}$$

Closed-form expressions:

$$\mu_2(A) = \tfrac{1}{2}\lambda_{\max}(A + A^\top)$$
$$\mu_1(A) = \max_j \left(a_{jj} + \sum\nolimits_{i\neq j} |a_{ij}|\right)$$
$$\mu_\infty(A) = \max_i \left(a_{ii} + \sum\nolimits_{j\neq i} |a_{ij}|\right)$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,
- **In the literature**: one-sided Lipschitz constant, logarithmic norm

Classical result

$\dot{x} = f(x, w)$ is contracting wrt $\|\cdot\|$ with rate c iff

$$\mu_{\|\cdot\|}\left(\tfrac{\partial f}{\partial x}(x, w)\right) \leq c, \qquad \text{for all } x, w$$

How to characterize contractivity using vector fields?

Matrix measure

Given a matrix $A \in \mathbb{R}^{n \times n}$ and a norm $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h \to 0^+} \frac{\|I_n + hA\| - 1}{h}$$

Closed-form expressions:

$$\mu_2(A) = \tfrac{1}{2}\lambda_{\mathsf{max}}(A + A^\top)$$
$$\mu_1(A) = \max_j \Big(a_{jj} + \sum_{i \neq j} |a_{ij}|\Big)$$
$$\mu_\infty(A) = \max_i \Big(a_{ii} + \sum_{j \neq i} |a_{ij}|\Big)$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,
- **In the literature**: one-sided Lipschitz constant, logarithmic norm

Classical result

$\dot{x} = f(x, w)$ is contracting wrt $\|\cdot\|$ with rate c iff

$$\mu_{\|\cdot\|}\Big(\frac{\partial f}{\partial x}(x, w)\Big) \leq c, \qquad \text{for all } x, w$$

- Efficient methods to find minimum c (Aylward et al., 2006, Giesl et al. 2023)

Assume $\mu_{\|\cdot\|}\left(\frac{\partial f}{\partial x}(x,w)\right) \leq c$ and $\left\|\frac{\partial f}{\partial w}(x,w)\right\| \leq \ell$ for almost every x, u.

[1] A. Davydov and **SJ** and F.Bullo, IEEE TAC, 2022.

Assume $\mu_{\|\cdot\|}\left(\frac{\partial f}{\partial x}(x,w)\right) \leq c$ and $\left\|\frac{\partial f}{\partial w}(x,w)\right\| \leq \ell$ for almost every x, u.

Input-to-state stability

$$\|x(t) - x^*(t)\| \leq e^{ct}\|x(0) - x^*(0)\| + \frac{\ell}{c}(e^{ct} - 1)\sup_{\tau\in[0,t]} \|w(\tau) - w^*\|$$

[1] A. Davydov and **SJ** and F.Bullo, IEEE TAC, 2022.

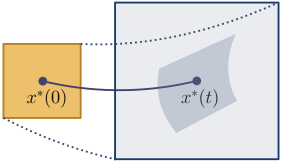
Assume $\mu_{\|\cdot\|}\left(\frac{\partial f}{\partial x}(x,w)\right) \leq c$ and $\left\|\frac{\partial f}{\partial w}(x,w)\right\| \leq \ell$ for almost every x, u.

Theorem[1]

If $\mathcal{X}_0 = B_{\|\cdot\|}(r_1, x_0^*)$ and $\mathcal{W} = B_{\|\cdot\|}(r_2, w^*)$, then

$$\mathcal{R}_f(t, \mathcal{X}_0, \mathcal{W}) \subseteq B_{\|\cdot\|}(e^{ct} r_1 + \tfrac{\ell}{c}(e^{ct} - 1) r_2, x^*(t))$$

where $x^*(\cdot)$ is the solution of $\dot{x} = f(x, w^*)$ with $x(0) = x_0^*$.



[1] A. Davydov and **SJ** and F.Bullo, IEEE TAC, 2022.

Approach #1: Contraction-based Reachability

Input-to-state stability

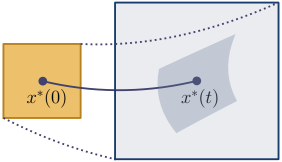
Assume $\mu_{\|\cdot\|}\left(\frac{\partial f}{\partial x}(x,w)\right) \leq c$ and $\left\|\frac{\partial f}{\partial w}(x,w)\right\| \leq \ell$ for almost every x, u.

Theorem

If $\mathcal{X}_0 = B_{\|\cdot\|}(r_1, x_0^*)$ and $\mathcal{W} = B_{\|\cdot\|}(r_2, w^*)$, then

$$\mathcal{R}_f(t, \mathcal{X}_0, \mathcal{W}) \subseteq B_{\|\cdot\|}(e^{ct} r_1 + \tfrac{\ell}{c}(e^{ct} - 1) r_2, x^*(t))$$

where $x^*(\cdot)$ is the solution of $\dot{x} = f(x, w^*)$ with $x(0) = x_0^*$.



(Computationally efficient): only need estimates of c and ℓ

(Scalable): efficient methods for computing c and ℓ for large-scale systems

[1] A. Davydov and **SJ** and F.Bullo, IEEE TAC, 2022.

- **Key idea:** embed the dynamical system on $\mathbb{R}^n$ into a dynamical system on $\mathbb{R}^{2n}$
- Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$

Original system

$$\dot{x} = f(x, w)$$

Embedding system

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w})$$

- **Key idea:** embed the dynamical system on $\mathbb{R}^n$ into a dynamical system on $\mathbb{R}^{2n}$
- Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$

Original system

$$\dot{x} = f(x, w)$$

Embedding system

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w})$$

$\underline{d}, \overline{d}$ are **decomposition functions** s.t.

1. $f(x, w) = \underline{d}(x, x, w, w)$ for every x, w
2. cooperative: $(\underline{x}, \underline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
3. competitive: $(\overline{x}, \overline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
4. the same properties for $\overline{d}$

- **Key idea:** embed the dynamical system on $\mathbb{R}^n$ into a dynamical system on $\mathbb{R}^{2n}$
- Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$

Original system

$$\dot{x} = f(x, w)$$

Embedding system

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w})$$

$\underline{d}, \overline{d}$ are **decomposition functions** s.t.

1. $f(x, w) = \underline{d}(x, x, w, w)$ for every x, w
2. cooperative: $(\underline{x}, \underline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
3. competitive: $(\overline{x}, \overline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
4. the same properties for $\overline{d}$

Embedding system is monotone (order preserving):

$$\overline{x}_i \uparrow \quad \Longrightarrow \quad \overline{x}_j \downarrow \text{ and } \underline{x}_j \uparrow \quad \text{for all j}$$
$$\underline{x}_i \downarrow \quad \Longrightarrow \quad \overline{x}_j \uparrow \text{ and } \underline{x}_j \downarrow \quad \text{for all j}$$

- **Key idea:** embed the dynamical system on $\mathbb{R}^n$ into a dynamical system on $\mathbb{R}^{2n}$
- Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$

Original system

$$\dot{x} = f(x, w)$$

Embedding system

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w})$$

$\underline{d}, \overline{d}$ are **decomposition functions** s.t.

1. $f(x, w) = \underline{d}(x, x, w, w)$ for every x, w
2. cooperative: $(\underline{x}, \underline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
3. competitive: $(\overline{x}, \overline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
4. the same properties for $\overline{d}$

Every system has at least one decomposition function

- **Key idea:** embed the dynamical system on $\mathbb{R}^n$ into a dynamical system on $\mathbb{R}^{2n}$
- Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$

Original system

$$\dot{x} = f(x, w)$$

Embedding system

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w})$$

$\underline{d}, \overline{d}$ are **decomposition functions** s.t.

1. $f(x, w) = \underline{d}(x, x, w, w)$ for every x, w
2. cooperative: $(\underline{x}, \underline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
3. competitive: $(\overline{x}, \overline{w}) \mapsto \underline{d}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w})$
4. the same properties for $\overline{d}$

Every system has at least one decomposition function

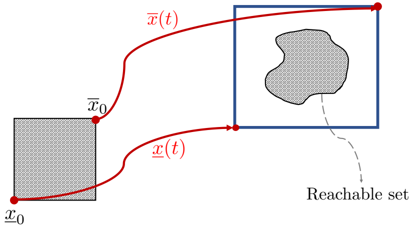
In this talk: we use mixed monotone theory for reachability analysis

Theorem[2]

Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ and

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}), \qquad \underline{x}(0) = \underline{x}_0$$
$$\dot{\overline{x}} = \overline{d}(\overline{x}, \underline{x}, \overline{w}, \underline{w}), \qquad \overline{x}(0) = \overline{x}_0$$

Then $\mathcal{R}_f(t, \mathcal{X}_0) \subseteq [\underline{x}(t), \overline{x}(t)]$



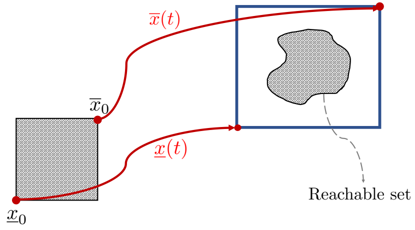
[2] H. Smith, Journal of Difference Equations and Applications, 2008

Theorem[2]

Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ and

$$\begin{aligned} \dot{\underline{x}} &= \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}), & \underline{x}(0) &= \underline{x}_0 \\ \dot{\overline{x}} &= \overline{d}(\overline{x}, \underline{x}, \overline{w}, \underline{w}), & \overline{x}(0) &= \overline{x}_0 \end{aligned}$$

Then $\mathcal{R}_f(t, \mathcal{X}_0) \subseteq [\underline{x}(t), \overline{x}(t)]$



a single trajectory of embedding system provides **lower bound** ($\underline{x}$) and **upper bound** ($\overline{x}$) for the trajectories of the original system.

[2]H. Smith, Journal of Difference Equations and Applications, 2008

Approach #2: Interval-based Reachability

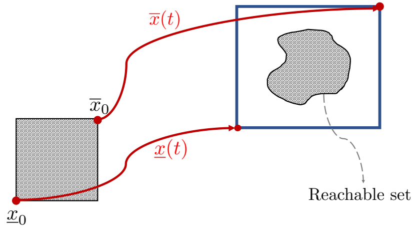
Embedding Systems

Theorem[2]

Assume $\mathcal{W} = [\underline{w}, \overline{w}]$ and $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ and

$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{w}, \overline{w}), \qquad \underline{x}(0) = \underline{x}_0$$
$$\dot{\overline{x}} = \overline{d}(\overline{x}, \underline{x}, \overline{w}, \underline{w}), \qquad \overline{x}(0) = \overline{x}_0$$

Then $\mathcal{R}_f(t, \mathcal{X}_0) \subseteq [\underline{x}(t), \overline{x}(t)]$



a single trajectory of embedding system provides **lower bound** ($\underline{x}$) and **upper bound** ($\overline{x}$) for the trajectories of the original system.

(Computational efficient): solve for one trajectory of embedding system

(Scalable): embedding system is $2n$-dimensional

[2] H. Smith, Journal of Difference Equations and Applications, 2008

How to compute a decomposition function for a system?

[3] **SJ** and A. Harapanahalli and S. Coogan, L4DC, 2023

How to compute a decomposition function for a system?

- Assume $f : \mathbb{R} \to \mathbb{R}$ is scalar:

Mean-value Inequality

$$\underbrace{f(\underline{x}) + \left[\min_{z\in[\underline{x},\overline{x}]} \frac{\partial f}{\partial x}\right]^{-} (\overline{x} - \underline{x})}_{\underline{d}(\underline{x},\overline{x})} \leq f(x) \leq \underbrace{f(\underline{x}) + \left[\max_{z\in[\underline{x},\overline{x}]} \frac{\partial f}{\partial x}\right]^{+} (\overline{x} - \underline{x})}_{\overline{d}(\underline{x},\overline{x})}$$

where $[A]^{+} = \max\{A, 0\}$ and $[A]^{-} = \min\{A, 0\}$.

[3] **SJ** and A. Harapanahalli and S. Coogan, L4DC, 2023

Approach #2: Interval-based Reachability

A Jacobian-based decomposition function

How to compute a decomposition function for a system?

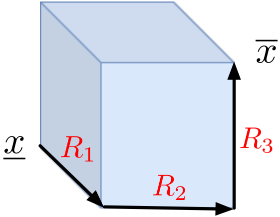
Theorem[3]

Jacobian-based: $\dot{x} = f(x,u)$ such that $\frac{\partial f}{\partial x} \in [\underline{J}_{[\underline{x},\overline{x}]}, \overline{J}_{[\underline{x},\overline{x}]}]$ and $\frac{\partial f}{\partial u} \in [\underline{J}_{[\underline{u},\overline{u}]}, \overline{J}_{[\underline{u},\overline{u}]}]$, then

$$\begin{bmatrix} \underline{d}(\underline{x},\overline{x},\underline{u},\overline{u}) \\ \overline{d}(\underline{x},\overline{x},\underline{u},\overline{u}) \end{bmatrix} = \begin{bmatrix} -[\underline{M}]^- & [\underline{M}]^- \\ -[\overline{M}]^+ & [\overline{M}]^+ \end{bmatrix} \begin{bmatrix} \underline{x} \\ \overline{x} \end{bmatrix} + \begin{bmatrix} -[\underline{N}]^- & [\underline{N}]^- \\ -[\overline{N}]^+ & [\overline{N}]^+ \end{bmatrix} \begin{bmatrix} \underline{u} \\ \overline{u} \end{bmatrix} + \begin{bmatrix} f(\underline{x},\underline{u}) \\ f(\underline{x},\underline{u}) \end{bmatrix}$$

$\underline{x} \mapsto R_1 \mapsto R_2 \mapsto \ldots \mapsto R_n \mapsto \overline{x}$, then the i-th column of $\underline{M}$ is $\min_{z\in R_i, w\in[\underline{u},\overline{u}]} \frac{\partial f_i}{\partial x}(z,w)$

- Interval analysis for computing Jacobian bounds.
- `immrax`: Toolbox that implements interval analysis in `JAX`.



[3] **SJ** and A. Harapanahalli and S. Coogan, L4DC, 2023

How to compute a decomposition function for a system?

Theorem[3]

Jacobian-based: $\dot{x} = f(x,u)$ such that $\frac{\partial f}{\partial x} \in [\underline{J}_{[\underline{x},\overline{x}]}, \overline{J}_{[\underline{x},\overline{x}]}]$ and $\frac{\partial f}{\partial u} \in [\underline{J}_{[\underline{u},\overline{u}]}, \overline{J}_{[\underline{u},\overline{u}]}]$, then

$$\begin{bmatrix} \underline{d}(\underline{x},\overline{x},\underline{u},\overline{u}) \\ \overline{d}(\underline{x},\overline{x},\underline{u},\overline{u}) \end{bmatrix} = \begin{bmatrix} -[\underline{M}]^- & [\underline{M}]^- \\ -[\overline{M}]^+ & [\overline{M}]^+ \end{bmatrix} \begin{bmatrix} \underline{x} \\ \overline{x} \end{bmatrix} + \begin{bmatrix} -[\underline{N}]^- & [\underline{N}]^- \\ -[\overline{N}]^+ & [\overline{N}]^+ \end{bmatrix} \begin{bmatrix} \underline{u} \\ \overline{u} \end{bmatrix} + \begin{bmatrix} f(\underline{x},\underline{u}) \\ f(\underline{x},\underline{u}) \end{bmatrix}$$

$\underline{x} \mapsto R_1 \mapsto R_2 \mapsto \dots \mapsto R_n \mapsto \overline{x}$, then the i-th column of $\underline{M}$ is $\min_{z\in R_i, w\in[\underline{u},\overline{u}]} \frac{\partial f_i}{\partial x}(z,w)$

- Interval analysis for computing Jacobian bounds.
- `immrax`: Toolbox that implements interval analysis in `JAX`.

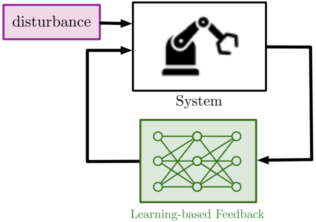
[3] **SJ** and A. Harapanahalli and S. Coogan, L4DC, 2023

Outline of this talk

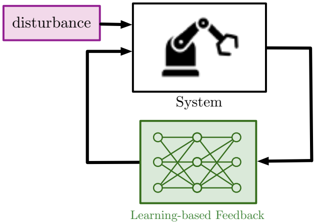
- Reachability Analysis
- **Neural Network Controlled Systems**
- Future Research Directions

- **In this part:** Learning-based component as a controller

- **In this part:** Learning-based component as a controller



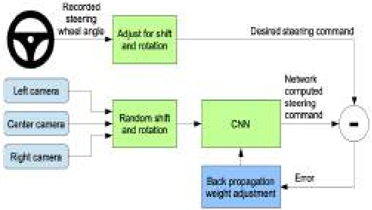
- **In this part:** Learning-based component as a controller



Issues with traditional controllers:

1. computationally burdensome
2. interaction with human
3. complicated representation

Self driving vehicles:

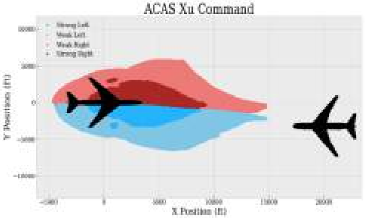


M. Bojarski, et al., NeurIPS, 2016.

Robotic motion planning:

M. Everett, et. al., IROS, 2018.

Collision avoidance:

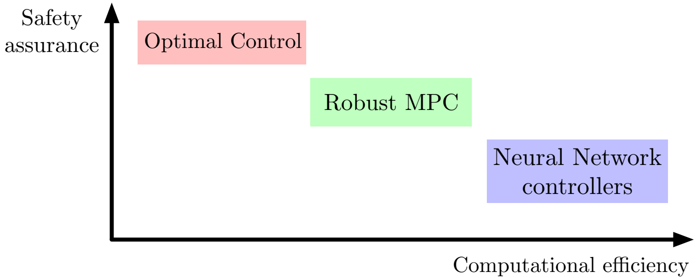


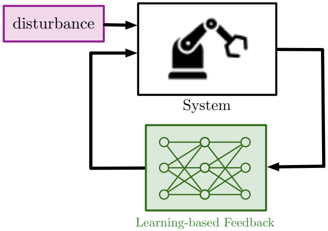
K. Julian, et. al., DASC, 2016.

Safety of learning-enabled autonomous systems **cannot be completely ensured** at the design level[4]

[4] Institute for Defense Analysis, The Status of Test, Evaluation, Verification, and Validation of Autonomous Systems, 2018

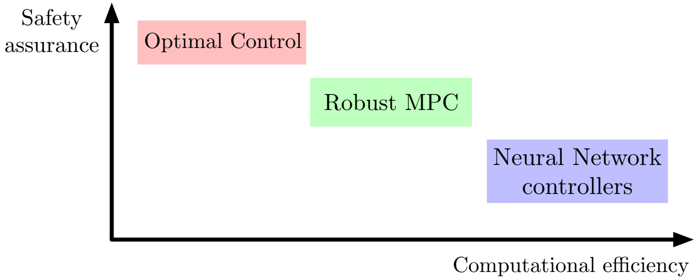
Safety of learning-enabled autonomous systems **cannot be completely ensured** at the design level[4]

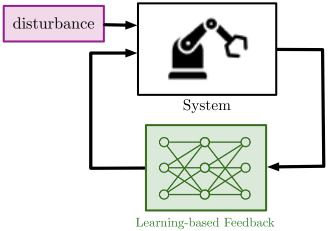




[4] Institute for Defense Analysis, The Status of Test, Evaluation, Verification, and Validation of Autonomous Systems, 2018

Safety of learning-enabled autonomous systems **cannot be completely ensured** at the design level[4]

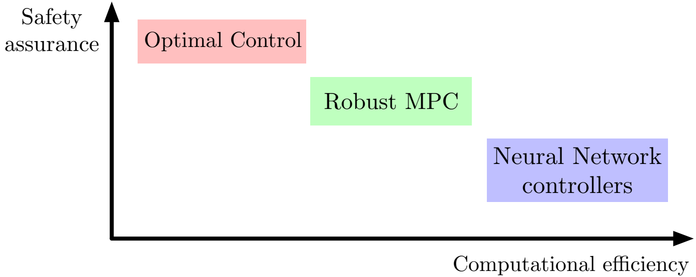


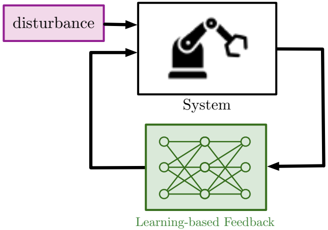


Design a mechanism that can do **run-time** safety verification

[4] Institute for Defense Analysis, The Status of Test, Evaluation, Verification, and Validation of Autonomous Systems, 2018

Safety of learning-enabled autonomous systems **cannot be completely ensured** at the design level[4]





Design a mechanism that can do **run-time** safety verification

Our approach: reachable set over-approximations for some time in future.

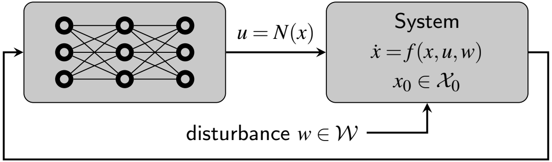
[4] Institute for Defense Analysis, The Status of Test, Evaluation, Verification, and Validation of Autonomous Systems, 2018

An open-loop nonlinear system with a neural network controller

$$\dot{x} = f(x, u, w),$$
$$u = N(x),$$

safety of the closed-loop system

$$\dot{x} = f(x, N(x), w) := f^c(x, w)$$

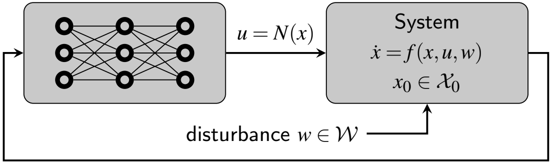


An open-loop nonlinear system with a neural network controller

$$\dot{x} = f(x,u,w),$$
$$u = N(x),$$

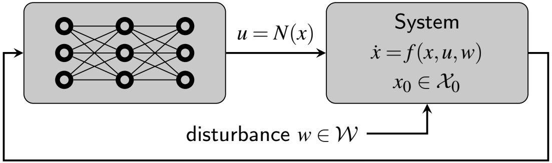
safety of the closed-loop system

$$\dot{x} = f(x,N(x),w) := f^c(x,w)$$



$u = N(x)$ is **pre-trained** feed-forward neural network with k-layer:

$$\xi^{(i)}(x) = \phi^{(i)}(W^{(i-1)}\xi^{(i-1)}(x) + b^{(i-1)})$$
$$x = \xi^{(0)}, \quad u = W^{(k)}\xi^{(k)}(x) + b^{(k)} := N(x),$$



An open-loop nonlinear system with a neural network controller

$$\dot{x} = f(x,u,w),$$
$$u = N(x),$$

safety of the closed-loop system

$$\dot{x} = f(x, N(x), w) := f^c(x,w)$$

$u = N(x)$ is **pre-trained** feed-forward neural network with k-layer:

$$\xi^{(i)}(x) = \phi^{(i)}(W^{(i-1)}\xi^{(i-1)}(x) + b^{(i-1)})$$
$$x = \xi^{(0)}, \quad u = W^{(k)}\xi^{(k)}(x) + b^{(k)} := N(x),$$

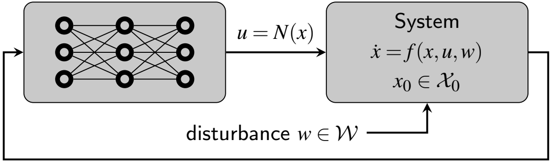
directly performing reachability on f^c is computationally challenging

An open-loop nonlinear system with a neural network controller

$$\dot{x} = f(x, u, w),$$
$$u = N(x),$$

safety of the closed-loop system

$$\dot{x} = f(x, N(x), w) := f^c(x, w)$$



$u = N(x)$ is **pre-trained** feed-forward neural network with k-layer:

$$\xi^{(i)}(x) = \phi^{(i)}(W^{(i-1)}\xi^{(i-1)}(x) + b^{(i-1)})$$
$$x = \xi^{(0)}, \quad u = W^{(k)}\xi^{(k)}(x) + b^{(k)} := N(x),$$

Rigorousness of control tools + effectiveness of ML tools

Combine our reachability frameworks with neural network verification methods

Input-output bounds: Given a neural network controller $u = N(x)$

$$\underline{u}_{[\underline{x},\overline{x}]} \leq N(x) \leq \overline{u}_{[\underline{x},\overline{x}]}, \qquad \text{for all } x \in [\underline{x}, \overline{x}]$$

[5] H. Zhang et al., NeurIPS 2018.

Input-output bounds: Given a neural network controller $u = N(x)$

$$\underline{u}_{[\underline{x},\overline{x}]} \leq N(x) \leq \overline{u}_{[\underline{x},\overline{x}]}, \qquad \text{for all } x \in [\underline{x}, \overline{x}]$$

Many neural network verification algorithms can produce these bounds.

ex. CROWN (H. Zhang et al., 2018), LipSDP (M. Fazlyab et al., 2019), IBP (S. Gowal et al., 2018).

[5] H. Zhang et al., NeurIPS 2018.

Input-output bounds: Given a neural network controller $u = N(x)$

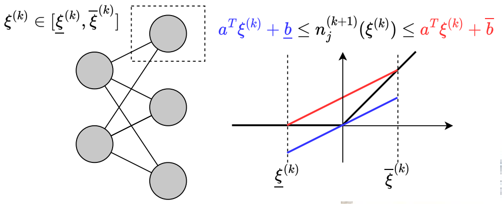
$$\underline{u}_{[\underline{x},\overline{x}]} \leq N(x) \leq \overline{u}_{[\underline{x},\overline{x}]}, \qquad \text{for all } x \in [\underline{x}, \overline{x}]$$

Many neural network verification algorithms can produce these bounds.

ex. CROWN (H. Zhang et al., 2018), LipSDP (M. Fazlyab et al., 2019), IBP (S. Gowal et al., 2018).

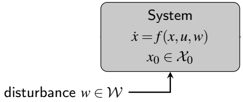
CROWN[5]

- Bounding the value of each neurons
- Linear upper and lower bounds on the activation function

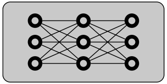


[5] H. Zhang et al., NeurIPS 2018.

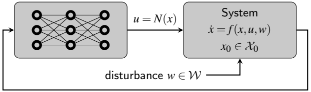
Reachability of open-loop system treating u as a parameter



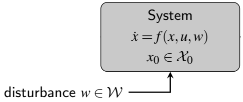
Neural network verification algorithm for bounds on u



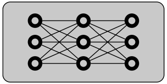
Reachability of open-loop system + Neural network verification bounds



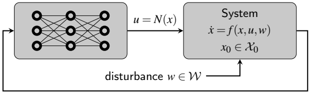
$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$



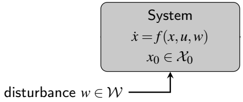
$$\underline{u}_{[\underline{x},\overline{x}]} \leq N(x) \leq \overline{u}_{[\underline{x},\overline{x}]} \quad \text{for every } x \in [\underline{x}, \overline{x}].$$



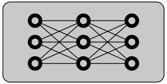
$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}_{[\underline{x},\overline{x}]}, \overline{u}_{[\underline{x},\overline{x}]}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}_{[\underline{x},\overline{x}]}, \overline{u}_{[\underline{x},\overline{x}]}, \underline{w}, \overline{w})$$



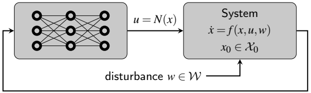
$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$



$$\underline{u}_{[\underline{x},\overline{x}]} \leq N(x) \leq \overline{u}_{[\underline{x},\overline{x}]} \quad \text{for every } x \in [\underline{x}, \overline{x}].$$



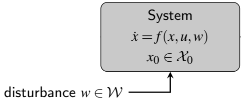
$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}_{[\underline{x},\overline{x}]}, \overline{u}_{[\underline{x},\overline{x}]}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}_{[\underline{x},\overline{x}]}, \overline{u}_{[\underline{x},\overline{x}]}, \underline{w}, \overline{w})$$



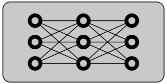
Composition approach over-approximation:

$$\mathcal{R}_{f^c}(t, \mathcal{X}_0, \mathcal{W}) \subseteq [\underline{x}(t), \overline{x}(t)]$$

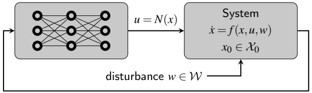
$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$



$$\underline{u}_{[\underline{x},\overline{x}]} \leq N(x) \leq \overline{u}_{[\underline{x},\overline{x}]} \quad \text{for every } x \in [\underline{x}, \overline{x}].$$



$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}_{[\underline{x},\overline{x}]}, \overline{u}_{[\underline{x},\overline{x}]}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}_{[\underline{x},\overline{x}]}, \overline{u}_{[\underline{x},\overline{x}]}, \underline{w}, \overline{w})$$



Composition approach over-approximation:
$$\mathcal{R}_{f^c}(t, \mathcal{X}_0, \mathcal{W}) \subseteq [\underline{x}(t), \overline{x}(t)]$$

It lead to overly-conservative estimates of reachable set

Stabilizing Effect of Neural Network Controllers

Issues with the compositional approach

Neural network controller as **disturbances** (worst-case scenario)
It does not capture the **stabilizing** effect of the neural network.

Neural network controller as **disturbances** (worst-case scenario)
It does not capture the **stabilizing** effect of the neural network.

An illustrative example

$\dot{x} = x + u + w$ with controller $u = -Kx$, for some unknown $1 < K \leq 3$.

Neural network controller as **disturbances** (worst-case scenario)
It does not capture the **stabilizing** effect of the neural network.

An illustrative example

$\dot{x} = x + u + w$ with controller $u = -Kx$, for some unknown $1 < K \leq 3$.

Compositional approach

First find the bounds $\underline{u} \leq Kx \leq \overline{u}$, then

$$\dot{\underline{x}} = \underline{x} + \underline{u} + \underline{w}$$
$$\dot{\overline{x}} = \overline{x} + \overline{u} + \overline{w}$$

This system is unstable.

Interaction-aware approach

First replace $u = Kx$ in the system, then

$$\dot{\underline{x}} = (1 - K)\underline{x} + \underline{w}$$
$$\dot{\overline{x}} = (1 - K)\overline{x} + \overline{w}$$

This system is stable.

Neural network controller as **disturbances** (worst-case scenario)
It does not capture the **stabilizing** effect of the neural network.

An illustrative example

$\dot{x} = x + u + w$ with controller $u = -Kx$, for some unknown $1 < K \leq 3$.

Compositional approach

First find the bounds $\underline{u} \leq Kx \leq \overline{u}$, then

$$\dot{\underline{x}} = \underline{x} + \underline{u} + \underline{w}$$
$$\dot{\overline{x}} = \overline{x} + \overline{u} + \overline{w}$$

This system is unstable.

Interaction-aware approach

First replace $u = Kx$ in the system, then

$$\dot{\underline{x}} = (1 - K)\underline{x} + \underline{w}$$
$$\dot{\overline{x}} = (1 - K)\overline{x} + \overline{w}$$

This system is stable.

We need to know the **functional** dependencies of neural network bounds

Functional bounds: Given a neural network controller $u = N(x)$

$$\underline{N}_{[\underline{x},\overline{x}]}(x) \leq N(x) \leq \overline{N}_{[\underline{x},\overline{x}]}(x), \qquad \text{for all } x \in [\underline{x}, \overline{x}]$$

[6] H. Zhang et al., NeurIPS 2018.

Functional bounds: Given a neural network controller $u = N(x)$

$$\underline{N}_{[\underline{x},\overline{x}]}(x) \leq N(x) \leq \overline{N}_{[\underline{x},\overline{x}]}(x), \qquad \text{for all } x \in [\underline{x},\overline{x}]$$

- Example: CROWN[6] can provide functional bounds.

CROWN functional bounds:

$$\underline{N}_{[\underline{x},\overline{x}]}(x) = \underline{A}_{[\underline{x},\overline{x}]}x + \underline{b}_{[\underline{x},\overline{x}]},$$
$$\overline{N}_{[\underline{x},\overline{x}]}(x) = \overline{A}_{[\underline{x},\overline{x}]}x + \overline{b}_{[\underline{x},\overline{x}]}$$

CROWN input-output bounds:

$$\underline{u}_{[\underline{x},\overline{x}]} = \underline{A}^{+}_{[\underline{x},\overline{x}]}\overline{x} + \overline{A}^{-}_{[\underline{x},\overline{x}]}\underline{x} + \underline{b}_{[\underline{x},\overline{x}]},$$
$$\overline{u}_{[\underline{x},\overline{x}]} = \overline{A}^{+}_{[\underline{x},\overline{x}]}\overline{x} + \underline{A}^{-}_{[\underline{x},\overline{x}]}\underline{x} + \overline{b}_{[\underline{x},\overline{x}]}$$

[6] H. Zhang et al., NeurIPS 2018.

Theorem[7]

Original system

$$\dot{x} = f(x, N(x), w)$$

Embedding system

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\overline{x}} \end{bmatrix} = \begin{bmatrix} [\underline{H}]^{+} - \underline{J}_{[x,\overline{x}]} & [\underline{H}]^{-} \\ [\overline{H}]^{+} - \overline{J}_{[x,\overline{x}]} & [\overline{H}]^{-} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \overline{x} \end{bmatrix} + \begin{bmatrix} -[\underline{J}_{[w,\overline{w}]}]^{-} & [\underline{J}_{[w,\overline{w}]}]^{+} \\ -[\overline{J}_{[w,\overline{w}]}]^{-} & [\overline{J}_{[w,\overline{w}]}]^{+} \end{bmatrix} \begin{bmatrix} \underline{w} \\ \overline{w} \end{bmatrix} + Q$$

$\underline{H}$ and $\overline{H}$ capture the effect of interactions between nonlinear system and neural network.

Interaction-aware over-approximation:

$$\mathcal{R}_{f^c}(t, \mathcal{X}_0, \mathcal{W}) \subseteq [\underline{x}(t), \overline{x}(t)]$$

[7] **SJ** and A. Harapanahalli and S. Coogan, under review, 2023

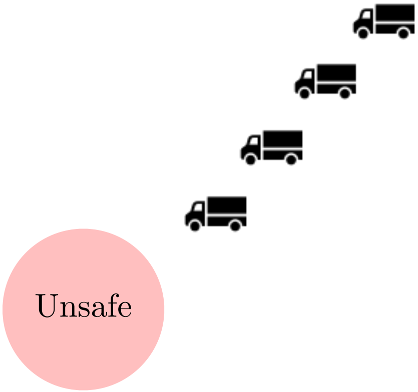
Case Study: Vehicle Platooning

Numerical Experiments

Dynamics of the jth vehicle

$$
\dot{p}_x^j = v_x^j, \qquad \dot{v}_x^j = \tanh(u_x^j) + w_x^j,
$$
$$
\dot{p}_y^j = v_y^j, \qquad \dot{v}_y^j = \tanh(u_y^j) + w_y^j,
$$

where $w_x^j, w_y^j \sim \mathcal{U}([-0.001, 0.001])$.



Case Study: Vehicle Platooning
Numerical Experiments

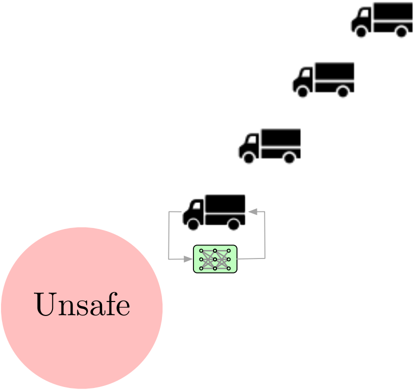
Dynamics of the jth vehicle

$$\dot{p}_x^j = v_x^j, \qquad \dot{v}_x^j = \tanh(u_x^j) + w_x^j,$$
$$\dot{p}_y^j = v_y^j, \qquad \dot{v}_y^j = \tanh(u_y^j) + w_y^j,$$

where $w_x^j, w_y^j \sim \mathcal{U}([-0.001, 0.001])$. First vehicle uses a neural network controller

$4 \times 100 \times 100 \times 2$, with ReLU activations

and is trained using trajectory data from an MPC controller for the first vehicle.



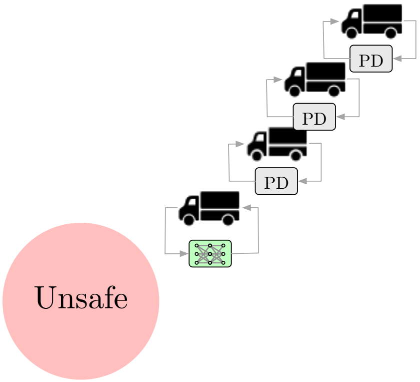
Dynamics of the jth vehicle

$$\dot{p}_x^j = v_x^j, \qquad \dot{v}_x^j = \tanh(u_x^j) + w_x^j,$$
$$\dot{p}_y^j = v_y^j, \qquad \dot{v}_y^j = \tanh(u_y^j) + w_y^j,$$

where $w_x^j, w_y^j \sim \mathcal{U}([-0.001, 0.001])$. Other vehicles use PD controller

$$u_d^j = k_p\left(p_d^{j-1} - p_d^j - r\frac{v_d^{j-1}}{\|v^{j-1}\|_2}\right) + k_v(v_d^{j-1} - v_d^j),$$

where $d \in \{x, y\}$.

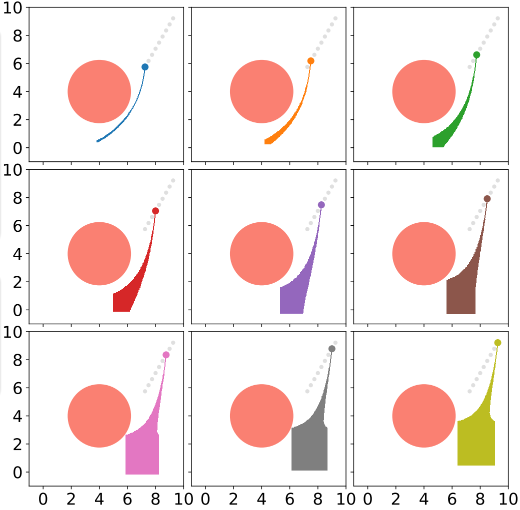


Dynamics of the jth vehicle

$$\dot{p}_x^j = v_x^j, \qquad \dot{v}_x^j = \tanh(u_x^j) + w_x^j,$$
$$\dot{p}_y^j = v_y^j, \qquad \dot{v}_y^j = \tanh(u_y^j) + w_y^j,$$

where $w_x^j, w_y^j \sim \mathcal{U}([-0.001, 0.001])$.

- compositional approach
- a platoon of 9 vehicles
- reachable overapproximations for $t \in [0, 1.5]$



Dynamics of the jth vehicle

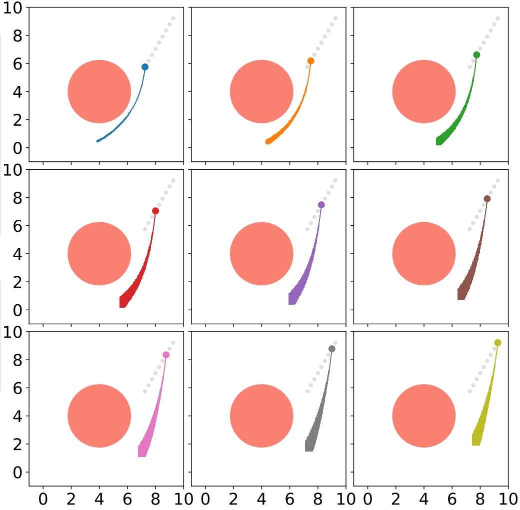
$$\dot{p}_x^j = v_x^j, \qquad \dot{v}_x^j = \tanh(u_x^j) + w_x^j,$$
$$\dot{p}_y^j = v_y^j, \qquad \dot{v}_y^j = \tanh(u_y^j) + w_y^j,$$

where $w_x^j, w_y^j \sim \mathcal{U}([-0.001, 0.001])$.

- interaction-aware approach
- a platoon of 9 vehicles
- reachable over-approximations for $t \in [0, 1.5]$

N (units)	# of states	Our Approach (s)	POLAR (s)	JuliaReach (s)
4	16	1.369	14.182	12.579
9	36	3.144	43.428	59.929
20	80	9.737	316.337	–
50	200	46.426	4256.435	–

Table: Run-time comparison



POLAR = C. Huang et al., ATVA 2022

JuliaReach = C. Schilling et al., AAAI 2022

Conclusions

Key takeaways

- reachability as a framework for safety certification of autonomous systems
- developed computationally efficient and scalable approaches for reachability: contraction-based and Interval-based
- run-time verification of neural network controlled systems
- capture stabilizing effect of learning-based components

Outline of this talk

- Reachability Analysis
- Neural Network Controlled Systems
- **Future Research Directions**

Data-assisted reachability of mechanical systems

[8] **SJ** and S. Coogan, “Monotonicity and Contraction on Polyhedral Cones”, submitted 2023

Data-assisted reachability of mechanical systems

Safety in manufacturing robotics

- complex tasks and operations
- interactions with human
- availability of data

Safe control of transportation systems

- nonlinear dynamics
- learning-enabled components
- large mobility data

[8] **SJ** and S. Coogan, "Monotonicity and Contraction on Polyhedral Cones", submitted 2023

Data-assisted reachability of mechanical systems

Safety in manufacturing robotics

- complex tasks and operations
- interactions with human
- availability of data

Safe control of transportation systems

- nonlinear dynamics
- learning-enabled components
- large mobility data

1. finite abstractions from reachability (formal methods)

[8] **SJ** and S. Coogan, "Monotonicity and Contraction on Polyhedral Cones", submitted 2023

Data-assisted reachability of mechanical systems

Safety in manufacturing robotics

- complex tasks and operations
- interactions with human
- availability of data

Safe control of transportation systems

- nonlinear dynamics
- learning-enabled components
- large mobility data

1. finite abstractions from reachability (formal methods)
2. physics-informed metrics for run-time monitoring[8]

[8] **SJ** and S. Coogan, “Monotonicity and Contraction on Polyhedral Cones”, submitted 2023

Data-assisted reachability of mechanical systems

Safety in manufacturing robotics

- complex tasks and operations
- interactions with human
- availability of data

Safe control of transportation systems

- nonlinear dynamics
- learning-enabled components
- large mobility data

1. finite abstractions from reachability (formal methods)
2. physics-informed metrics for run-time monitoring[8]
3. data to obtain suitable metrics for reachability analysis

funding: NSERC Alliance (possible partner: Electrans or LoopX AI)

[8] **SJ** and S. Coogan, "Monotonicity and Contraction on Polyhedral Cones", submitted 2023

Safe learning and control in learning-enabled feedback loops

[9] **SJ** and Y. Chen, "Probabilistic Reachability of Stochastic Systems", submitted 2024

Safe learning and control in learning-enabled feedback loops

Uncertainty learning and calibration

- learn uncertainties in run-time
- effect of feedback on uncertainty
- design a correction control

Safe control of feedback loop

- switch to back up controllers
- differentiable safety metrics
- correct-by-design training

[9] **SJ** and Y. Chen, "Probabilistic Reachability of Stochastic Systems", submitted 2024

Safe learning and control in learning-enabled feedback loops

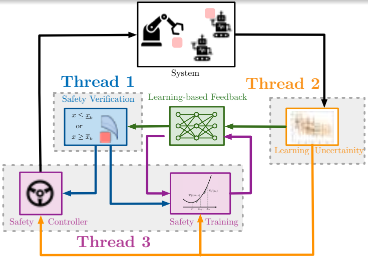
Uncertainty learning and calibration

- learn uncertainties in run-time
- effect of feedback on uncertainty
- design a correction control

Safe control of feedback loop

- switch to back up controllers
- differentiable safety metrics
- correct-by-design training

- utilize the statistical knowledge of uncertainty[9]
- reachability analysis to obtain differentiable safety metrics



funding: NSERC discovery

[9] **SJ** and Y. Chen, "Probabilistic Reachability of Stochastic Systems", submitted 2024

Detection and control in modern power grids

Detection and control in modern power grids

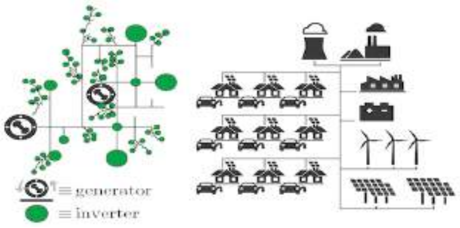
Far future grids $= 100\%$ penetration of renewables

Near future grids $=$ hybrid with both renewables and synchronous machines

Detection and control in modern power grids

Far future grids = 100% penetration of renewables

Near future grids = hybrid with both renewables and synchronous machines



Unique features of renewables

- fast dynamics
- stochastic generation/consumption

Goal: transient stability of the grid

1. fast and computationally efficient safety monitoring

funding: NSERC Alliance (possible partner: Canadian Solar Inc.)

Thank you for your attention!

Back up Slides

For $\|\cdot\|_{2,P}$ with a positive definite matrix P:

$$\mu_{2,P}(Df(t,x)) \leq c \iff PDf(t,x) + Df(t,x)^\top P \preceq 2cP$$

For $\|\cdot\|_{1,\mathrm{diag}(\eta)}$ with $\eta \in \mathbb{R}^n_{>0}$:

$$\mu_{1,\mathrm{diag}(\eta)}(Df(t,x)) \leq c \iff \eta^\top [Df(t,x)]^M \leq c\eta^\top$$
$$\mu_{\infty,\mathrm{diag}(\eta)}(Df(t,x)) \leq c \iff [Df(t,x)]^M \eta \leq c\eta$$

where $[A]^M$ is Metzler part of matrix A.

If f is polynomial in t and x,

1. for a fix c, search for P (or η) can be done using SOS programming
2. iterative bisection on c and SOS programming to find the minimum c

E. M. Aylward, P. A. Parrilo, and J.-J. E. Slotine. Stability and robustness analysis of nonlinear systems via contraction metrics and SOS programming. Automatica, 2008

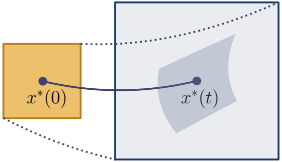
Assume $\mu_{\|\cdot\|}\left(\frac{\partial f}{\partial x}(x,w)\right) \leq c$ and $\left\|\frac{\partial f}{\partial w}(x,w)\right\| \leq \ell$ for almost all x, u

Theorem

If $\mathcal{X}_0 = B_{\|\cdot\|}(r_1, x_0^*)$ and $\mathcal{W} = B_{\|\cdot\|}(r_2, w^*)$, then

$$\mathcal{R}^f(t, \mathcal{X}_0) \subseteq B_{\|\cdot\|}(e^{ct} r_1 + \tfrac{\ell}{c}(e^{ct}-1) r_2, x^*(t))$$

where $x^*(\cdot)$ is the solution of $\dot{x} = f(x, w^*)$ with $x(0) = x_0^*$.



Proof: let $x(\cdot)$ be a traj of $\dot{x} = f(x,w)$. Using Taylor expansion, for $h \geq 0$

$$x(t+h) - x^*(t+h) = x(t) - x^*(t) + h \overbrace{\left(\int_0^1 D_x f(\tau x + (1-\tau)x^*) d\tau\right)}^{A(x,w)} (x(t) - x^*(t))$$

$$+ h \overbrace{\left(\int_0^1 D_w f(x, \tau w + (1-\tau) w^*) d\tau\right)}^{B(x,w)} (w - w^*) + \mathcal{O}(h^2)$$

Contraction-based Reachability

Proof continued

$$D^+\|x(t)-x^*(t)\| = \limsup_{h\to 0^+} \frac{\|x(t+h)-x^*(t+h)\| - \|x(t)-x^*(t)\|}{h}$$

$$= \limsup_{h\to 0^+} \frac{\| \left(I_n + hA(x,w)\right)\left(x(t)-x^*(t)\right) + hB(x,w)(w-w^*)\| - \|x(t)-x^*(t)\|}{h}$$

$$\leq \limsup_{h\to 0^+} \frac{\| \left(I_n + hA(x,w)\right)\left(x(t)-x^*(t)\right)\| + h\|B(x,w)\|\|w-w^*\| - \|x(t)-x^*(t)\|}{h}$$

$$\leq \mu_{\|\cdot\|}(A(x,w))\|x(t)-x^*(t)\| + \|B(x,w)\|\|w-w^*\|$$

$$\leq c\|x(t)-x^*(t)\| + \ell\|w-w^*\|$$

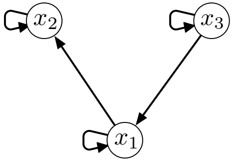
- generalized version of Grönwall's lemma
- overly conservative since c and ℓ are defined globally

Embedding System for Linear Dynamical System

A structure preserving decomposition

- **Metzler/non-Metzler** decomposition: $A = \lceil A \rceil^{\mathrm{Mzl}} + \lfloor A \rfloor^{\mathrm{Mzl}}$
- Example: $A = \begin{bmatrix} 2 & 0 & -1 \\ 1 & -3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \implies \lceil A \rceil^{\mathrm{Mzl}} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & -3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \lfloor A \rfloor^{\mathrm{Mzl}} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Linear systems

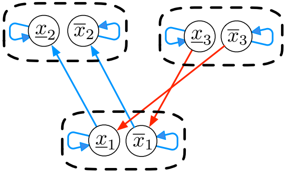


Original system

$$\dot{x} = Ax + Bw$$

Embedding system

$$\dot{\underline{x}} = \lceil A \rceil^{\mathrm{Mzl}} \underline{x} + \lfloor A \rfloor^{\mathrm{Mzl}} \overline{x} + B^{+} \underline{w} + B^{-} \overline{w}$$
$$\dot{\overline{x}} = \lceil A \rceil^{\mathrm{Mzl}} \overline{x} + \lfloor A \rfloor^{\mathrm{Mzl}} \underline{x} + B^{+} \overline{w} + B^{-} \underline{w}$$



For a scalar vector field $f : \mathbb{R} \to \mathbb{R}$, we show that $\underline{d}(\underline{x}, \overline{x}) = f(\underline{x}) + \left[\min_{z\in[\underline{x},\overline{x}]} \frac{\partial f}{\partial x}\right]^{-} (\overline{x} - \underline{x})$ is

1. cooperative in $\underline{x}$
2. competitive in $\overline{x}$

$$\frac{\partial}{\partial \underline{x}}\underline{d}(\underline{x}, \overline{x}) = \frac{\partial}{\partial \underline{x}} f(\underline{x}) - \left[\min_{z\in[\underline{x},\overline{x}]} \frac{\partial f}{\partial x}\right]^{-} = \frac{\partial f}{\partial x}\Big|_{x=\underline{x}} - \left[\min_{z\in[\underline{x},\overline{x}]} \frac{\partial f}{\partial x}\right]^{-} \geq 0.$$

Similarly,

$$\frac{\partial}{\partial \overline{x}}\underline{d}(\underline{x}, \overline{x}) = \left[\min_{z\in[\underline{x},\overline{x}]} \frac{\partial f}{\partial x}\right]^{-} \leq 0$$

Case Study: Bicycle Model

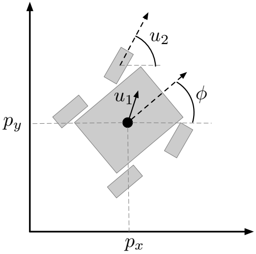
A naive compositional approach

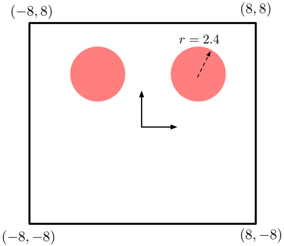
Dynamics of bicycle

$$\dot{p_x} = v\cos(\phi + \beta(u_2)) \qquad \dot{\phi} = \frac{v}{\ell_r}\sin(\beta(u_2))$$

$$\dot{p_y} = v\sin(\phi + \beta(u_2)) \qquad \dot{v} = u_1$$

$$\beta(u_2) = \arctan\left(\frac{l_r}{l_f + l_r}\tan(u_2)\right)$$



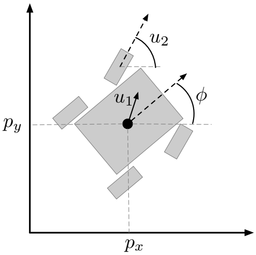


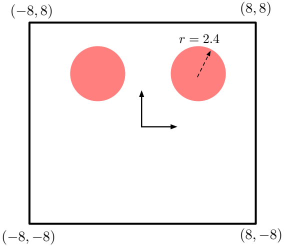
Dynamics of bicycle

$$\dot{p_x} = v\cos(\phi + \beta(u_2)) \qquad \dot{\phi} = \frac{v}{\ell_r}\sin(\beta(u_2))$$

$$\dot{p_y} = v\sin(\phi + \beta(u_2)) \qquad \dot{v} = u_1$$

$$\beta(u_2) = \arctan\left(\frac{l_r}{l_f + l_r}\tan(u_2)\right)$$





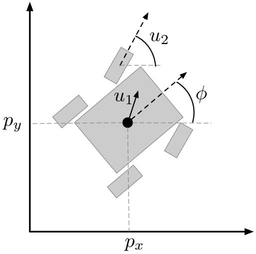
Goal: steer the bicycle to the origin avoiding the obstacles

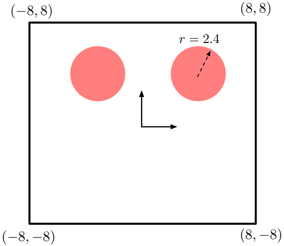
Dynamics of bicycle

$$\dot{p_x} = v\cos(\phi + \beta(u_2)) \qquad \dot{\phi} = \frac{v}{\ell_r}\sin(\beta(u_2))$$

$$\dot{p_y} = v\sin(\phi + \beta(u_2)) \qquad \dot{v} = u_1$$

$$\beta(u_2) = \arctan\left(\frac{l_r}{l_f + l_r}\tan(u_2)\right)$$





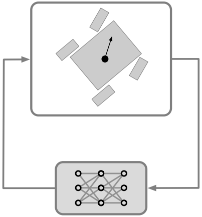
Goal: steer the bicycle to the origin avoiding the obstacles

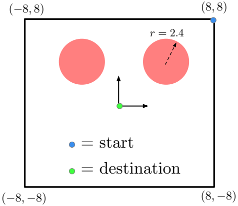
- train a feedforward neural network $4 \mapsto 100 \mapsto 100 \mapsto 2$ using data from model predictive control

- start from $(8,8)$ toward $(0,0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

$$\underline{x}_0 = \begin{pmatrix} 7.95 & 7.95 & -\frac{\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$
$$\overline{x}_0 = \begin{pmatrix} 8.05 & 8.05 & -\frac{\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

- CROWN for verification of neural network





Embedding system:

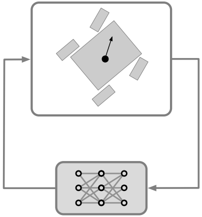
$$\dot{\underline{x}} = \underline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$
$$\dot{\overline{x}} = \overline{d}(\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{w}, \overline{w})$$

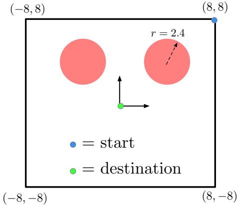
$\underline{u} \leq N(x) \leq \overline{u}$, for every $x \in [\underline{x}, \overline{x}]$.

- start from $(8, 8)$ toward $(0, 0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

$$\underline{x}_0 = \begin{pmatrix} 7.95 & 7.95 & -\frac{\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$
$$\overline{x}_0 = \begin{pmatrix} 8.05 & 8.05 & -\frac{\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

- CROWN for verification of neural network

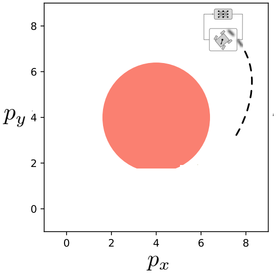


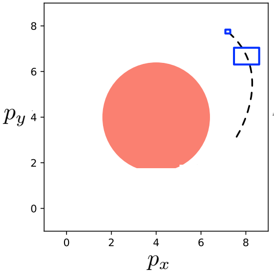


Euler integration with step h:

$$\underline{x}_1 = \underline{x}_0 + h\underline{d}(\underline{x}_0, \overline{x}_0, \underline{u}_0, \overline{u}_0, \underline{w}, \overline{w})$$
$$\overline{x}_1 = \overline{x}_0 + h\overline{d}(\underline{x}_0, \overline{x}_0, \underline{u}_0, \overline{u}_0, \underline{w}, \overline{w})$$

$\underline{u}_0 \leq N(x) \leq \overline{u}_0$, for every $x \in [\underline{x}_0, \overline{x}_0]$.





Reachability of Closed-loop System

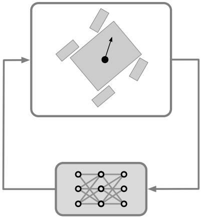
Case Study: Bicycle Model

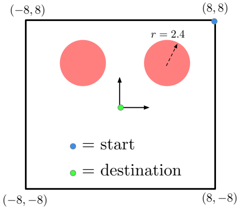
- start from $(8,8)$ toward $(0,0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

$$\underline{x}_0 = \begin{pmatrix} 7.95 & 7.95 & -\frac{\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$

$$\overline{x}_0 = \begin{pmatrix} 8.05 & 8.05 & -\frac{\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

- CROWN for verification of neural network



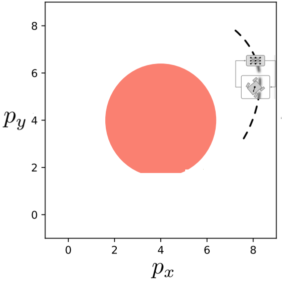


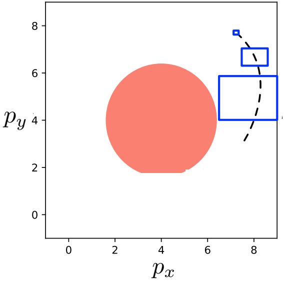
Euler integration with step h:

$$\underline{x}_2 = \underline{x}_1 + h\underline{d}(\underline{x}_1, \overline{x}_1, \underline{u}_1, \overline{u}_1, \underline{w}, \overline{w})$$

$$\overline{x}_2 = \overline{x}_1 + h\overline{d}(\underline{x}_1, \overline{x}_1, \underline{u}_1, \overline{u}_1, \underline{w}, \overline{w})$$

$\underline{u}_1 \leq N(x) \leq \overline{u}_1$, for every $x \in [\underline{x}_1, \overline{x}_1]$.

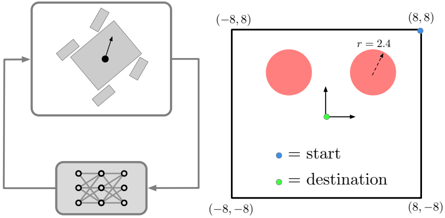




- start from $(8,8)$ toward $(0,0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

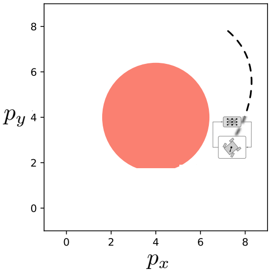
$$\underline{x}_0 = \begin{pmatrix} 7.95 & 7.95 & -\frac{\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$
$$\overline{x}_0 = \begin{pmatrix} 8.05 & 8.05 & -\frac{\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

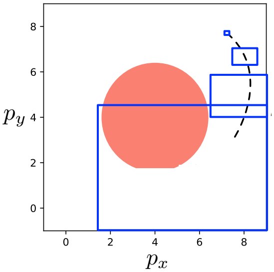
- CROWN for verification of neural network



Euler integration with step h:

$$\underline{x}_3 = \underline{x}_2 + h\underline{d}(\underline{x}_2, \overline{x}_2, \underline{u}_2, \overline{u}_2, \underline{w}, \overline{w})$$
$$\overline{x}_3 = \overline{x}_2 + h\overline{d}(\underline{x}_2, \overline{x}_2, \underline{u}_2, \overline{u}_2, \underline{w}, \overline{w})$$





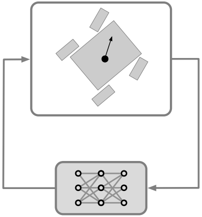
$\underline{u}_2 \leq N(x) \leq \overline{u}_2$, for every $x \in [\underline{x}_2, \overline{x}_2]$.

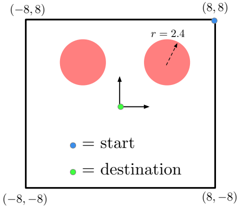
- start from $(8, 7)$ toward $(0, 0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

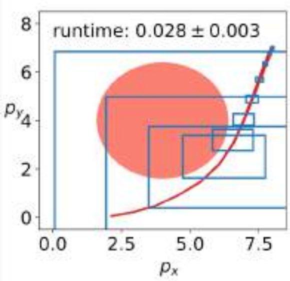
$$\underline{x}_0 = \begin{pmatrix} 7.95 & 6.95 & -\frac{2\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$

$$\overline{x}_0 = \begin{pmatrix} 8.05 & 7.05 & -\frac{2\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

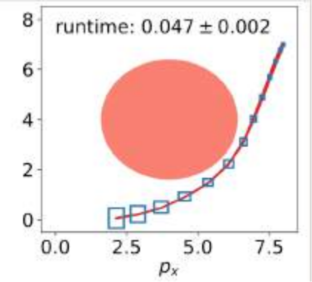
- CROWN for verification of neural network







Naive interconnection approach



interaction approach