Exploiting Structure in Feedback Systems with Learning-based Components

Saber Jafarpour

Decision and Control Laboratory
Georgia Institute of Technology

January 18, 2023

Modern societal autonomous systems

Introduction

Power grids

Transportation networks

Learning-based systems

- large penetration of distributed renewable units in power grids
- unprecedented demand is pushing transportation networks to their capacity
- increasing deployment of learning algorithms in safety-critical systems

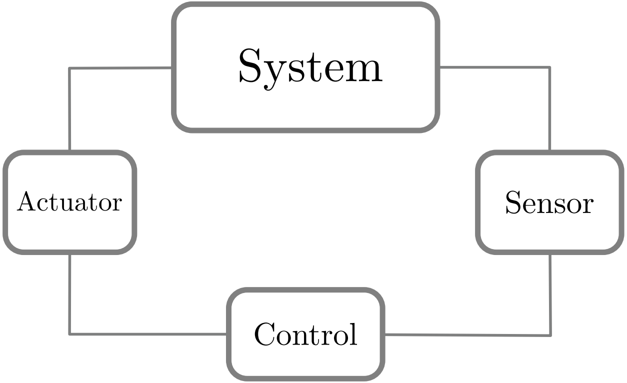
societal autonomous systems are becoming **large-scale** with **interconnected** and **complex** components

reconsider the traditional approaches for **monitoring** and **control** of autonomous systems

Feedback control of autonomous systems

Opportunities and challenges

Feedback is a central paradigm in control theory



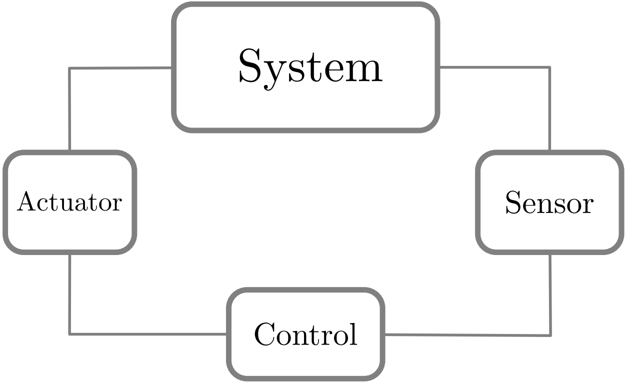
Magic of feedback[a]: robustness, shape behavior, command tracking, etc.

---

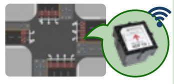
[a] Karl J. Astrom, Automatic Control - A Perspective, 2019

Feedback control of autonomous systems

Opportunities and challenges



Agents have wide range of **communication** capabilities



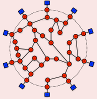
Enhanced processing units allow new **computational** approaches

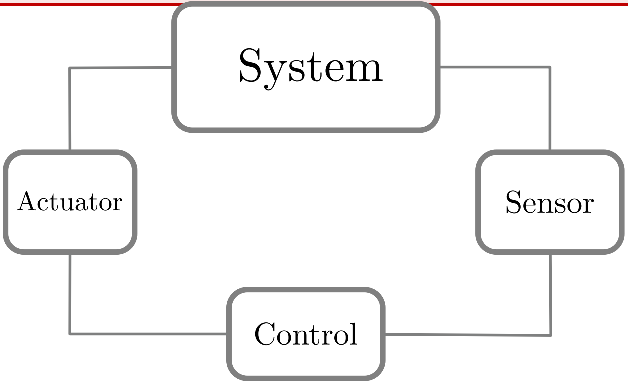
Large number of measurement devices for sensing

Feedback control of autonomous systems

Opportunities and challenges

Systems are becoming **large-scale** with **heterogenous** and **interconnected** components





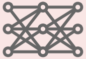
System

Actuator

Sensor

Control

Controllers contain **high-dimensional**, **learning-based**, and **complex** parts

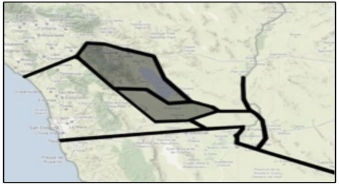


My research

Safety and robustness in control of autonomous systems

A critical task

Desired performance while ensuring their **safety** and **robustness**.



2011 US Southwest blackout

Traffic congestion in Beijing

[redacted]

Self-driving car accident

My Contribution

Exploit **structure** to ensure safety and resilience in control of large-scale autonomous systems

Tools: control theory, dynamical systems, optimization

Research summary

My past and current research

Stability of large-scale power grids

- threshold of frequency synchronization (TAC 2018, SICON 2019)
- multi-stability via partitioning the state-space (SIAM Review 2021, Nature Com 2022)
- dynamic stability of low-inertia power grids (TCNS 2019)

Geometric control

- small time local controllability (SICON 2020)
- locally convex topologies and control theory (MCSS 2016)

Contraction theory

- weak and semi-contraction theory (TAC 2021)
- non-Euclidean contraction theory (TAC 2022)
- time-varying optimization (TAC 2021)
- non-Euclidean monotone operator theory (CDC 2022)

Robustness of learning algorithms

- implicit neural networks (NeurIPS 2021, L4DC 2022)
- interval reachability of neural networks (L4DC 2022)
- safety verification of feedback loops

Learning-based feedback

Introduction

Learning-based feedback

Feedback controller or some elements of it are learned from data

Aerial vehicles

Self-driving cars

GaTech A1 robot

Why data-driven feedback?

- models are complicated or not available
- environment is unknown or varying
- traditional methods are cumbersome

Learning-based feedback

Introduction

Learning-based feedback

Feedback controller or some elements of it are learned from data

Aerial vehicles

Self-driving cars

GaTech A1 robot

Why data-driven feedback?

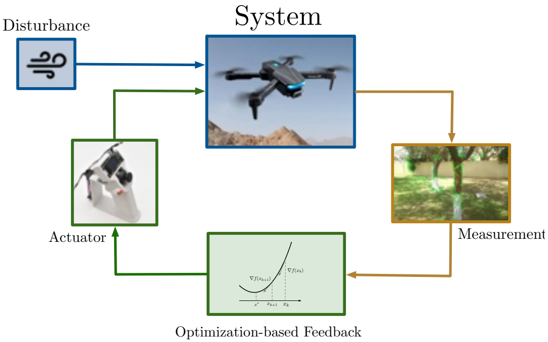
- models are complicated or not available
- environment is unknown or varying
- traditional methods are cumbersome

Learning-based feedback

A data-driven approach to controller design

Assumption: An (approximate) model of the system is available

Optimization-based control



- Example method: Model Predictive Control (MPC)

$$\begin{aligned}\min_{u(0),\dots,u(N-1)} \quad & \sum_{i=0}^{N-1} \ell(x(t),u(t)) + \phi(x(N)),\\ & x(t+1) = x(t) + \alpha f(x(t),u(t)),\\ & x(t) \in \mathcal{X}, \quad t \in \{1,\dots,N\}\\ & u(t) \in \mathcal{U}, \quad t \in \{0,\dots,N-1\}\\ & x(0) = x\end{aligned}$$

- $\mathcal{X}$ and $\mathcal{U}$ are the safety constraints

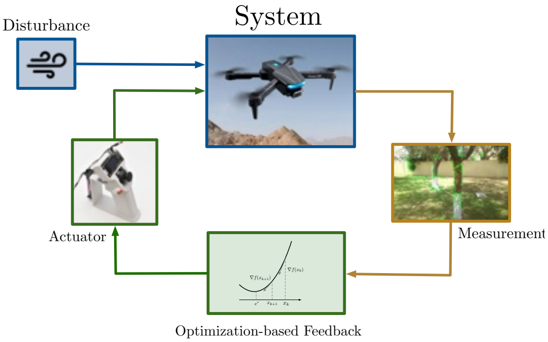
Feedback law: $u(0) = K(x)$

Learning-based feedback

A data-driven approach to controller design

Assumption: An (approximate) model of the system is available

Optimization-based control



- Example issues: set $\mathcal{X}$ is learned online

$$
\begin{aligned}
\min_{u(0),\ldots,u(N-1)} \quad & \sum_{i=0}^{N-1} \ell(x(t),u(t)) + \phi(x(N)), \\
& x(t+1) = x(t) + \alpha f(x(t),u(t)), \\
& x(t) \in \mathcal{X}, \quad t \in \{1,\ldots,N\} \\
& u(t) \in \mathcal{U}, \quad t \in \{0,\ldots,N-1\} \\
& x(0) = x
\end{aligned}
$$

- $\mathcal{X}$ and $\mathcal{U}$ are the safety constraints

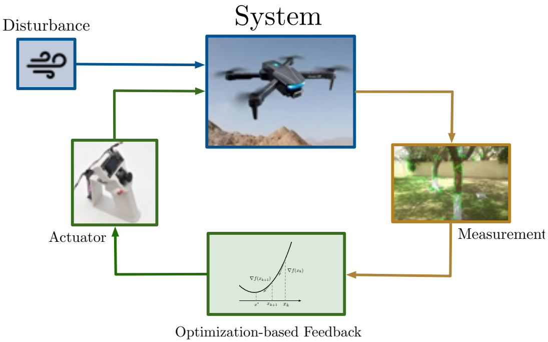
Feedback law: $u(0) = K(x)$

Learning-based feedback

A data-driven approach to controller design

Assumption: An (approximate) model of the system is available

Optimization-based control



- Example issues: the optimization problem is computationally complicated

$$
\begin{aligned}
\min_{u(0),\dots,u(N-1)} \quad & \sum_{i=0}^{N-1} \ell(x(t),u(t)) + \phi(x(N)),\\
& x(t+1) = x(t) + \alpha f(x(t),u(t)),\\
& x(t) \in \mathcal{X}, \quad t \in \{1,\dots,N\}\\
& u(t) \in \mathcal{U}, \quad t \in \{0,\dots,N-1\}\\
& x(0) = x
\end{aligned}
$$

- $\mathcal{X}$ and $\mathcal{U}$ are the safety constraints

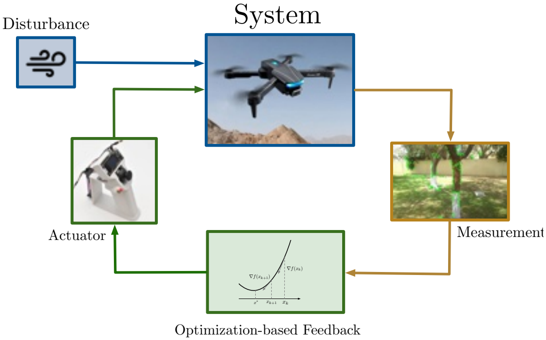
Feedback law: $u(0) = K(x)$

Learning-based feedback

A data-driven approach to controller design

Assumption: An (approximate) model of the system is available

Optimization-based control



- full knowledge of environment
- computationally complexity

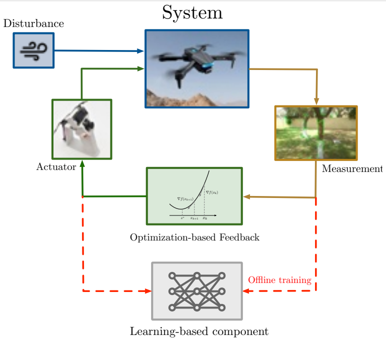
replace (some part of) the controller with a learning-based component

Learning-based feedback

A data-driven approach to controller design

Assumption: An (approximate) model of the system is available

Offline training



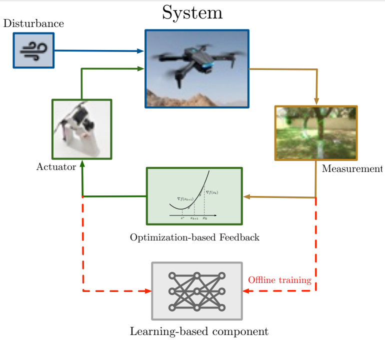
- overly conservative constraints
- solve the optimization offline
- data to train the learning algorithm

Learning-based feedback

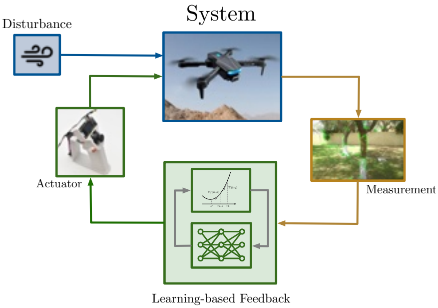
A data-driven approach to controller design

Assumption: An (approximate) model of the system is available

Offline training



Online implementation



- overly conservative safety guarantees
- solve the optimization offline
- data to train the learning algorithm

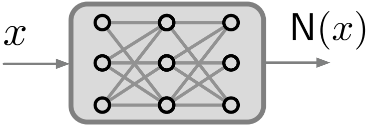
- efficient implementation
- partial knowledge of environment
- **limited** safety guarantees

Reachability analysis

A paradigm for safety assurance

Isolated learning component

- Robustness of learning algorithms



- An input perturbation set $\mathcal{X}$
- Safe output domain $\mathcal{Y}$

Output perturbations

$$\mathsf{N}(\mathcal{X}) = \{\mathsf{N}(x) \mid x \in \mathcal{X}\}$$

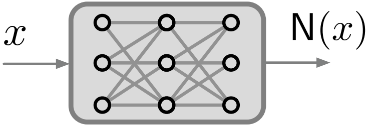
Goal: ensure that $\mathsf{N}(\mathcal{X}) \subset \mathcal{Y}$.

Reachability analysis

A paradigm for safety assurance

Isolated learning component

- Robustness of learning algorithms



- An input perturbation set $\mathcal{X}$
- Safe output domain $\mathcal{Y}$

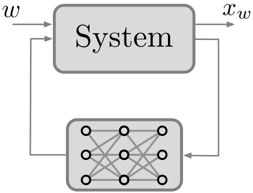
Output perturbations

$$\mathsf{N}(\mathcal{X}) = \{\mathsf{N}(x) \mid x \in \mathcal{X}\}$$

Goal: ensure that $\mathsf{N}(\mathcal{X}) \subset \mathcal{Y}$.

Interconnected learning-based system

- Safety of closed-loop system



- An input perturbation set $\mathcal{W}$
- Safe output domain $\mathcal{S}$

Reachable sets

$$\mathcal{R}(\mathcal{W}, t) = \{x_w(t) \mid w \in \mathcal{W}\}$$

Goal: ensure that $\mathcal{R}(\mathcal{W}, t) \subset \mathcal{S}$

Robustness of learning algorithms

Verification and training

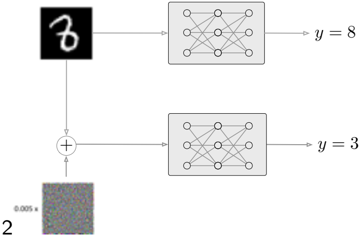
1. learning algorithms are fragile wrt input perturbations

Adversarial Perturbations

Small changes in the input

↓

Large changes in the output



C. Szegedy and et. al. Intriguing properties of neural networks. In *ICLR*, 2

Robustness of learning algorithms

Verification and training

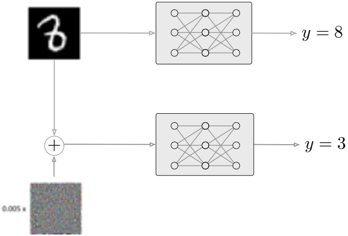
1. learning algorithms are fragile wrt input perturbations

Adversarial Perturbations

Small changes in the input

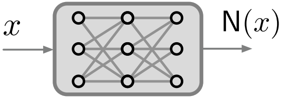
$\downarrow$

Large changes in the output



C. Szegedy and et. al. Intriguing properties of neural networks. In *ICLR*, 2

2. learning algorithms have large number of parameters and are highly nonlinear



Reachability of learning-based systems

The role of the structure

- Reachability of dynamical system is an old problem: $\sim$ 1980
  - Example approaches: Hamilton-Jacobi, Ellipsoidal methods

Reachability of learning-based systems

The role of the structure

- Reachability of dynamical system is an old problem: ~ 1980
  - Example approaches: Hamilton-Jacobi, Ellipsoidal methods

not scalable to large-scale systems

Reachability of learning-based systems

The role of the structure

- Reachability of dynamical system is an old problem: ~ 1980
  - Example approaches: Hamilton-Jacobi, Ellipsoidal methods

not scalable to large-scale systems

- Reachability of learning algorithms is more recent: ~ 2010
  - Example approaches: Interval arithmetic, Semi-definite programing

Reachability of learning-based systems

The role of the structure

- Reachability of dynamical system is an old problem: ~ 1980
  - Example approaches: Hamilton-Jacobi, Ellipsoidal methods

not scalable to large-scale systems

- Reachability of learning algorithms is more recent: ~ 2010
  - Example approaches: Interval arithmetic, Semi-definite programing

1. structure of the learning algorithm
2. Interconnection structure of the system

Structure lead to tractable algorithms

- **Contractivity**, to ensure computational efficiency
- **Mixed monotonicity**, a key property of neural network loops

Outline of this talk

- <u>Contraction theory and mixed monotonicity</u>
- Isolated learning algorithms
- Learning-based feedback loops

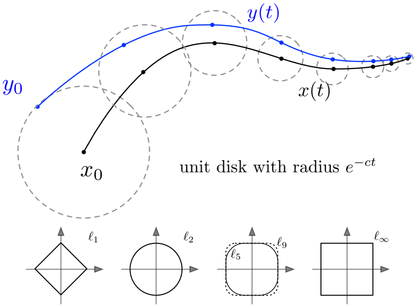
Tool #1: Contraction theory

A framework for stability analysis

Definition (Contraction)

$\dot{x} = f(x, u)$ is contracting wrt $\|\cdot\|$ if

the distance between every two trajectory is decreasing exponentially with rate c wrt $\|\cdot\|$



Tool #1: Contraction theory

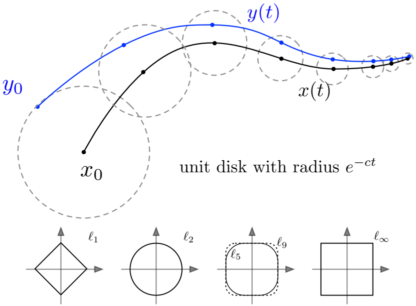
A framework for stability analysis

Definition (Contraction)

$\dot{x} = f(x,u)$ is contracting wrt $\|\cdot\|$ if
the distance between every two trajectory is decreasing exponentially with rate c wrt $\|\cdot\|$

Transient and asymptotic behavior:

- unique globally exponential stable equilibrium
- efficient equilibrium point computation
- input-output robustness
- modularity and interconnection properties
- . . .



Tool #1: Contraction theory

Characterization for Euclidean norms

Main Result

$\dot{x} = f(x,u)$ is contracting wrt $\|\cdot\|_2$ with rate c iff

Differential: $D_x f(x,u)^\top + D_x f(x,u) + 2cI \preceq 0,$ for all x, u

Integral: $\langle f(x,u) - f(y,u), x - y\rangle \leq -c\|x-y\|_2^2,$ for all x, y, u

Tool #1: Contraction theory

Characterization for Euclidean norms

Main Result

$\dot{x} = f(x, u)$ is contracting wrt $\|\cdot\|_2$ with rate c iff

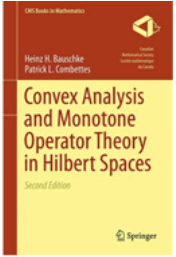
Differential: $D_x f(x,u)^\top + D_x f(x,u) + 2cI \preceq 0,$ for all x, u

Integral: $\langle f(x,u) - f(y,u), x - y \rangle \leq -c\|x - y\|_2^2,$ for all x, y, u

- Connection between **Euclidean contraction theory** and **monotone operator theory**

f is contracting with respect to $\|\cdot\|_2$
iff
$-f$ is a monotone operator with respect to the inner product $\langle\cdot,\cdot\rangle$.

- How about general norms?



CMS Books in Mathematics

Heinz H. Bauschke
Patrick L. Combettes

Convex Analysis and Monotone Operator Theory in Hilbert Spaces

Second Edition

Springer

Tool #1: Contraction theory

Logarithmic norm and weak pairings

Differential condition

Logarithmic norm

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,

$$\mu_2(A) = \tfrac{1}{2}\lambda_{\mathsf{max}}(A + A^\top)$$

$$\mu_1(A) = \max_j \left(a_{jj} + \sum\nolimits_{i\neq j} |a_{ij}|\right)$$

$$\mu_\infty(A) = \max_i \left(a_{ii} + \sum\nolimits_{j\neq i} |a_{ij}|\right)$$

---

[1] A. Davydov, S. Jafarpour, F. Bullo, TAC 2022

Tool #1: Contraction theory

Logarithmic norm and weak pairings

Differential condition

Logarithmic norm

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,

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

Integral condition

Weak pairing[1]

Given a norm $\|\cdot\|$, the associated weak pairing is $[\![\cdot,\cdot]\!] : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$:

- Subadditive and weakly homogeneity
- Positive definite
- Cauchy-Schwarz inequality
- $[\![x,x]\!] = \|x\|^2$

$$[\![x,y]\!]_2 = y^\top x$$
$$[\![x,y]\!]_1 = \mathrm{sign}(y)^\top x$$
$$[\![x,y]\!]_\infty = \max\nolimits_{i\in I_\infty(x)} x_i y_i$$

$$I_\infty(x) = \{i \mid |x_i| = \|x\|_\infty\}$$

---

[1] A. Davydov, S. Jafarpour, F. Bullo, TAC 2022

Tool #1: Contraction theory

Characterization for Non-Euclidean norms

Main Result

$\dot{x} = f(x,u)$ is contracting wrt $\|\cdot\|_2$ with rate c iff

Differential: $$D_x f(x,u)^\top + D_x f(x,u) + 2cI \preceq 0, \qquad \text{for all } x, u$$

Integral: $$\langle f(x,u) - f(y,u), x - y\rangle \leq -c\|x-y\|_2^2, \qquad \text{for all } x, y, u$$

---

[2] A. Davydov, S. Jafarpour, F. Bullo, TAC 2022

Tool #1: Contraction theory

Characterization for Non-Euclidean norms

Theorem[2]

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

Differential: $\mu_{\|\cdot\|}(D_x f(x,u)) \leq -c,$ for all x, u

Integral: $[\![f(x,u) - f(y,u), x - y]\!] \leq -c\|x-y\|^2,$ for all x, y, u

Why non-Euclidean?

- well suited for conservative systems
- computational advantages
- structural robustness

Non-Euclidean monotone operators

f is contracting with respect to $\|\cdot\|$
iff
$-f$ is a monotone operator with respect to $[\![\cdot,\cdot]\!]$.

---

[2] A. Davydov, S. Jafarpour, F. Bullo, TAC 2022

Tool #2: Mixed monotonicity

Cooperative and competitive dynamics

Original system

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

Embedding system

$$\begin{aligned}\dot{\underline{x}} &= g(\underline{x}, \overline{x}, \underline{u}, \overline{u}),\\ \dot{\overline{x}} &= g(\overline{x}, \underline{x}, \overline{u}, \underline{u})\end{aligned}$$

g is a **decomposition function** s.t.

1. $f(x, u) = g(x, x, u, u)$ for every x, u
2. cooperative: $(\underline{x}, \underline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$
3. competitive: $(\overline{x}, \overline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$

Tool #2: Mixed monotonicity

Cooperative and competitive dynamics

Original system

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

Embedding system

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

g is a **decomposition function** s.t.

1. $f(x, u) = g(x, x, u, u)$ for every x, u
2. cooperative: $(\underline{x}, \underline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$
3. competitive: $(\overline{x}, \overline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$

Embedding system is monotone with respect to $\leq_{\mathrm{SE}}$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq_{\mathrm{SE}} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$
iff
$x_1 \leq y_1$ and $y_2 \leq x_2$

- f locally Lipschitz $\Longrightarrow$ mixed monotonicity

Monotone system are being studied by Hirsch, Smith, Sontag, Angeli, ...

Tool #2: Mixed monotonicity

Reachability analysis

Theorem[3]

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

---

[3] S. Coogan, M. Arcak, HSCC 2015

Tool #2: Mixed monotonicity

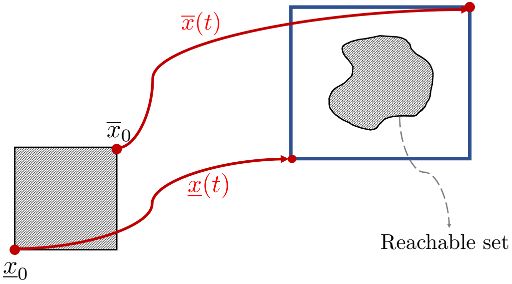
Reachability analysis

Theorem[3]

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

Embedding system

$$\begin{aligned} \dot{\underline{x}} &= g(\underline{x}, \overline{x}, \underline{u}, \overline{u}), \qquad & \underline{x}(0) &= \underline{x}_0 \\ \dot{\overline{x}} &= g(\overline{x}, \underline{x}, \overline{u}, \underline{u}), \qquad & \overline{x}(0) &= \overline{x}_0 \end{aligned}$$



---

[3] S. Coogan, M. Arcak, HSCC 2015

Tool #2: Mixed monotonicity

Tight decomposition functions

How to find a decomposition function?

- decomposition function might not be unique
- different approaches exist for certain class of systems
    - Jacobian of f
    - polynomial structure of f

Tool #2: Mixed monotonicity

Tight decomposition functions

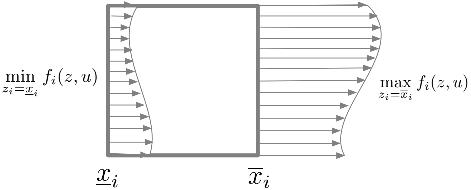
How to find a decomposition function?

- decomposition function might not be unique
- different approaches exist for certain class of systems
  - Jacobian of f
  - polynomial structure of f

The sharpest bounds = **tight decomposition function**

$$g_i(\underline{x}, \overline{x}, \underline{u}, \overline{u}) = \min_{\substack{z\in[\underline{x},\overline{x}], z_i=\underline{x}_i \\ w\in[\underline{u},\overline{u}]}} f_i(z, w)$$

$$g_i(\overline{x}, \underline{x}, \overline{u}, \underline{u}) = \max_{\substack{z\in[\underline{x},\overline{x}], z_i=\overline{x}_i \\ w\in[\underline{u},\overline{u}]}} f_i(z, w)$$



Tool #2: Mixed monotonicity

Connection with contraction theory

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

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

Theorem[4]

The original system is contracting wrt to $\|\cdot\|_\infty$ with rate c
iff
the **tight** embedding system is contracting wrt to $\|\cdot\|_\infty$ with rate c

---

[4] S. Jafarpour, S. Coogan, arXiv 2022

Tool #2: Mixed monotonicity

Connection with contraction theory

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

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

Theorem[4]

The original system is contracting wrt to $\|\cdot\|_\infty$ with rate c

iff

the **tight** embedding system is contracting wrt to $\|\cdot\|_\infty$ with rate c

The unique role of non-Euclidean ℓ_∞-norm

---

[4] S. Jafarpour, S. Coogan, arXiv 2022

Tool #2: Mixed monotonicity

Connection with contraction theory

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

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

Theorem[4]

The original system is contracting wrt to $\|\cdot\|_\infty$ with rate c

iff

the **tight** embedding system is contracting wrt to $\|\cdot\|_\infty$ with rate c

The unique role of non-Euclidean ℓ_∞-norm

Corollary (for contracting systems)

Mixed-monotone reachability is **sharper** than global input-to-state bounds:

$$\|x(t) - y(t)\|_\infty \leq e^{-ct}\|x_0 - y_0\|_\infty + \tfrac{\ell(1-e^{-ct})}{c}\|u - w\|_\infty.$$

---

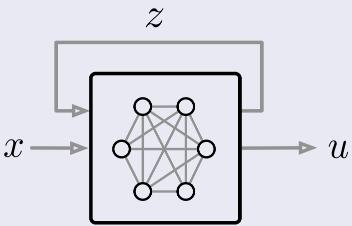
[4] S. Jafarpour, S. Coogan, arXiv 2022

Outline of this talk

- Contraction theory and mixed monotonicity
- Isolated learning algorithms
- Learning-based feedback loops

Generalized neural networks

A general learning model via fixed-point equations



- Generalized neural networks:

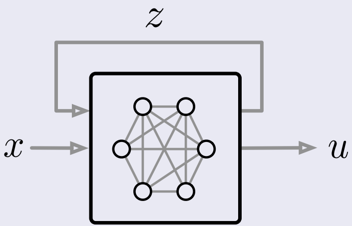
$$z = \Phi(Az + Bx + b)$$
$$u = Cz + c$$

- $\Phi(y_1, \dots, y_n) = (\phi_1(y_1), \dots, \phi_n(y_n))^\top$ with ϕ_i satisfies $0 \leq \frac{\phi_i(x) - \phi_i(y)}{x - y} \leq 1$.

- S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In *NeurIPS*, 2019
- L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Y. Tsai. Implicit deep learning. *SIMODS*, 2019

Generalized neural networks

A general learning model via fixed-point equations



- Generalized neural networks:

$$z = \Phi(Az + Bx + b)$$
$$u = Cz + c$$

Notion of layer

Output is an **implicit** function of input

(*e.g.*, fixed-point equation, differential equations, optimization problem)

Why implicit models?

- Representation
- Performance
- Memory

- $\Phi(y_1, \ldots, y_n) = (\phi_1(y_1), \ldots, \phi_n(y_n))^\top$ with ϕ_i satisfies $0 \leq \frac{\phi_i(x) - \phi_i(y)}{x-y} \leq 1$.

- S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In *NeurIPS*, 2019
- L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Y. Tsai. Implicit deep learning. *SIMODS*, 2019

Neural Networks

A dynamical system perspective

Main Questions

$$z = \Phi(Az + Bx + b)$$
$$u = Cz + c$$

1. Existence and computation of solutions?
2. How to estimate the input-output $x \mapsto u$ robustness?

Neural Networks

A dynamical system perspective

Main Questions

$$z = \Phi(Az + Bx + b)$$
$$u = Cz + c$$

① Existence and computation of solutions?

② How to estimate the input-output $x \mapsto u$ robustness?

Key insight

Fixed-point equation $\Longleftrightarrow$ Dynamical system

$z = \Phi(Az + Bx + b)$ $\qquad$ $\dot{z} = -z + \Phi(Az + Bx + b)$

fixed-points $\Longleftrightarrow$ **equilibrium points**

robustness $\Longleftrightarrow$ **forward reachability** $(t = \infty)$

- We can use tools from dynamical systems to study generalized neural networks

Fixed-points of neural network

A non-Euclidean contracting approach

Fixed-point of	$\Longleftrightarrow$	Equilibrium point of
$z = \Phi(Az + Bx + b)$		$\dot{z} = -z + \Phi(Az + Bx + b)$

- **Contraction theory**: Sufficient condition for existence a globally stable equilibrium point.

---

[5] S. Jafarpour, A. Davydov, A. Proskurnikov, F. Bullo, NeurIPS 2022

Fixed-points of neural network

A non-Euclidean contracting approach

Fixed-point of $\iff$ Equilibrium point of

$$z = \Phi(Az + Bx + b) \qquad \dot{z} = -z + \Phi(Az + Bx + b)$$

- **Contraction theory**: Sufficient condition for existence a globally stable equilibrium point.

$$[\![\Phi(Az_1 + Bx + b) - \Phi(Az_2 + Bx + b), z_1 - z_2]\!]_\infty < \|z_1 - z_2\|_\infty^2$$

---

[5] S. Jafarpour, A. Davydov, A. Proskurnikov, F. Bullo, NeurIPS 2022

Fixed-points of neural network

A non-Euclidean contracting approach

Fixed-point of $z = \Phi(Az + Bx + b)$ $\iff$ Equilibrium point of $\dot{z} = -z + \Phi(Az + Bx + b)$

- **Contraction theory**: Sufficient condition for existence a globally stable equilibrium point.

$$a_{ii} + \sum_{j\neq i} |a_{ij}| < 1 \quad \implies \quad [\![\Phi(Az_1 + Bx + b) - \Phi(Az_2 + Bx + b), z_1 - z_2]\!]_{\infty} < \|z_1 - z_2\|_{\infty}^{2}$$

---

[5] S. Jafarpour, A. Davydov, A. Proskurnikov, F. Bullo, NeurIPS 2022

Fixed-points of neural network

A non-Euclidean contracting approach

Fixed-point of $z = \Phi(Az + Bx + b)$ $\iff$ Equilibrium point of $\dot{z} = -z + \Phi(Az + Bx + b)$

- **Contraction theory**: Sufficient condition for existence a globally stable equilibrium point.

$$a_{ii} + \sum_{j \neq i} |a_{ij}| < 1 \quad \Longrightarrow \quad [\![\Phi(Az_1 + Bx + b) - \Phi(Az_2 + Bx + b), z_1 - z_2]\!]_\infty < \|z_1 - z_2\|_\infty^2$$

Theorem[5]

If $a_{ii} + \sum_{j \neq i} |a_{ij}| < 1$ then

1. $z = \Phi(Az + Bx + b)$ has a unique solution z_x^*
2. z_x^* can be computed using average iterations for $z = \Phi(Az + Bx + b)$

---

[5] S. Jafarpour, A. Davydov, A. Proskurnikov, F. Bullo, NeurIPS 2022

Robustness of neural network

A mixed monotone contracting approach

robustness of $\iff$ forward reachability of

$z = \Phi(Az + Bx + b)$ $\qquad$ $\dot{z} = -z + \Phi(Az + Bx + b)$

---

[6] S.Jafarpour, M. Abate, A. Davydov, F. Bullo, S. Coogan, L4DC 2022

Robustness of neural network

A mixed monotone contracting approach

robustness of	$\iff$	forward reachability of
$z = \Phi(Az + Bx + b)$		$\dot{z} = -z + \Phi(Az + Bx + b)$

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

---

[6] S.Jafarpour, M. Abate, A. Davydov, F. Bullo, S. Coogan, L4DC 2022

Robustness of neural network

A mixed monotone contracting approach

robustness of	$\Longleftrightarrow$	forward reachability of
$z = \Phi(Az + Bx + b)$		$\dot{z} = -z + \Phi(Az + Bx + b)$

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

Theorem[6]

The neural network is mixed monotone with the tight decomposition function:

$$\mathsf{G}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) = -\underline{z} + \Phi(\lceil A \rceil^{\mathrm{Mzl}} \underline{z} + \lfloor A \rfloor^{\mathrm{Mzl}} \overline{z} + [B]^{+} \underline{x} + [B]^{-} \overline{x} + b)$$

---

[6] S.Jafarpour, M. Abate, A. Davydov, F. Bullo, S. Coogan, L4DC 2022

Robustness of neural network

A mixed monotone contracting approach

Theorem[7]

If $a_{ii} + \sum_{j\neq i} |a_{ij}| < 1$ and $x \in [\underline{x}, \overline{x}]$

1. $z = \Phi(Az + Bx + b)$ has a unique solution z_u^*
2. $\begin{bmatrix}\underline{z}\\ \overline{z}\end{bmatrix} = \begin{bmatrix}\mathsf{G}(\underline{z}, \overline{z}, \underline{x}, \overline{x})\\ \mathsf{G}(\overline{z}, \underline{z}, \overline{x}, \underline{x})\end{bmatrix}$ has a unique solution $\begin{bmatrix}\underline{z}^*\\ \overline{z}^*\end{bmatrix}$
3. $\underbrace{([C]^+ \ [C]^-) \begin{bmatrix}\underline{z}^*\\ \overline{z}^*\end{bmatrix} + c}_{\underline{u}} \leq u \leq \underbrace{([C]^- \ [C]^+) \begin{bmatrix}\underline{z}^*\\ \overline{z}^*\end{bmatrix} + c}_{\overline{u}}$

---

[7] S.Jafarpour, M. Abate, A. Davydov, F. Bullo, S. Coogan, L4DC 2022

Robustness of neural network

A mixed monotone contracting approach

Theorem[7]

If $a_{ii} + \sum_{j\neq i} |a_{ij}| < 1$ and $x \in [\underline{x}, \overline{x}]$

1. $z = \Phi(Az + Bx + b)$ has a unique solution z_u^*
2. $\begin{bmatrix} \underline{z} \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathsf{G}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) \\ \mathsf{G}(\overline{z}, \underline{z}, \overline{x}, \underline{x}) \end{bmatrix}$ has a unique solution $\begin{bmatrix} \underline{z}^* \\ \overline{z}^* \end{bmatrix}$
3. $\underbrace{([C]^+ \ [C]^-) \begin{bmatrix} \underline{z}^* \\ \overline{z}^* \end{bmatrix} + c}_{\underline{u}} \leq u \leq \underbrace{([C]^- \ [C]^+) \begin{bmatrix} \underline{z}^* \\ \overline{z}^* \end{bmatrix} + c}_{\overline{u}}$

- **Verification:** find robustness margin of generalized neural networks
- **Training:** design robust generalized neural networks
  - $a_{ii} + \sum_{j\neq i} |a_{ij}| < 1$ as a constraint to the training problem
  - a regularization term $\mathcal{R}(\underline{u}, \overline{u})$ to the training cost

---

[7] S.Jafarpour, M. Abate, A. Davydov, F. Bullo, S. Coogan, L4DC 2022

Numerical experiments

MNIST dataset classification

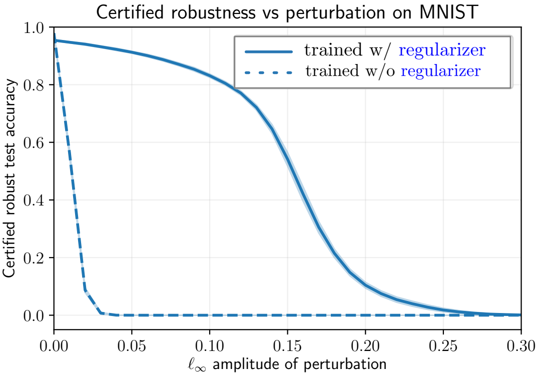
- MNIST dataset: 28×28 pixel handwritten digits between $0 - 9$.
- hidden layer of neural network $n = 100$
- $\epsilon =$ size of perturbation, $\mathcal{X} = [x - \epsilon\mathbb{1}_{784}, x + \epsilon\mathbb{1}_{784}]$.

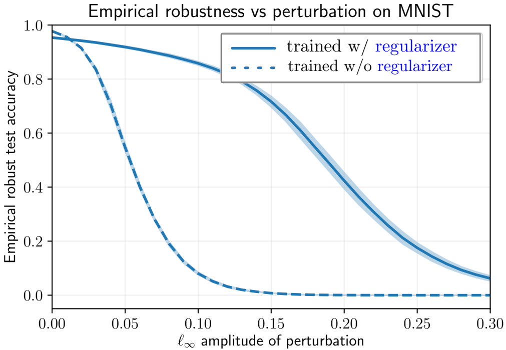
Numerical experiments

MNIST dataset classification

- MNIST dataset: 28×28 pixel handwritten digits between $0 - 9$.
- hidden layer of neural network $n = 100$

- $\epsilon =$ size of perturbation, $\mathcal{X} = [x - \epsilon \mathbb{1}_{784}, x + \epsilon \mathbb{1}_{784}]$.





- Certified robustness = all the elements of $[\underline{u}(\epsilon), \overline{u}(\epsilon)]$ classify as the correct digit
- Empirical robustness = Projected Gradient Descent (PGD) attack

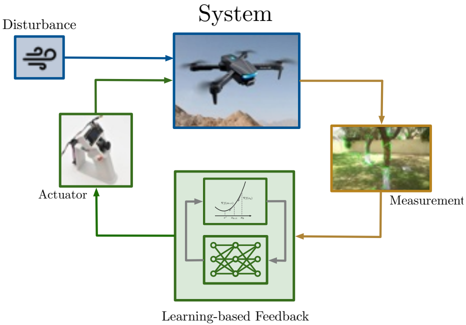
Outline of this talk

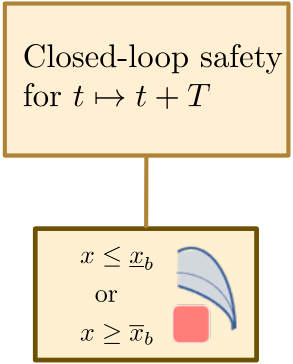
- Contraction theory and mixed monotonicity
- Isolated learning algorithms
- Learning-based feedback loops

Learning-based feedback

Safety guarantees for the feedback loops

Run-Time Assurance mechanism (RTA): monitor + predict




Closed-loop safety for $t \mapsto t + T$

$x \leq \underline{x}_b$
or
$x \geq \overline{x}_b$

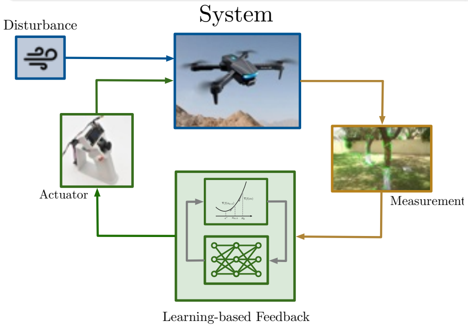
Online Safety Mechanism

Learning-based feedback

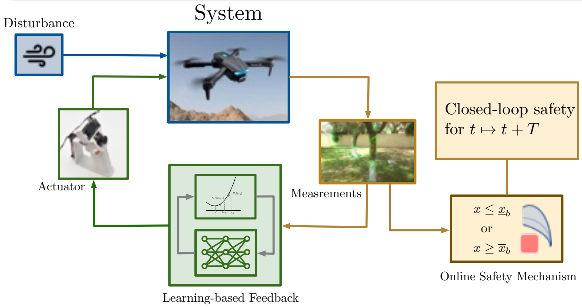
Safety guarantees for the feedback loops

Run-Time Assurance mechanism (RTA): monitor + predict

Closed-loop system without RTA



Closed-loop system with RTA

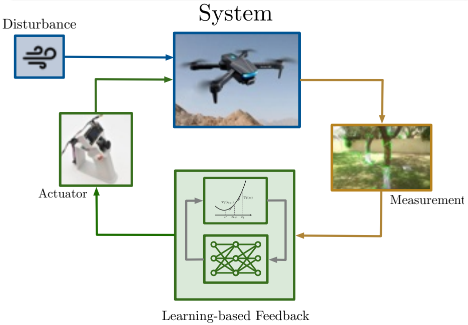


Learning-based feedback

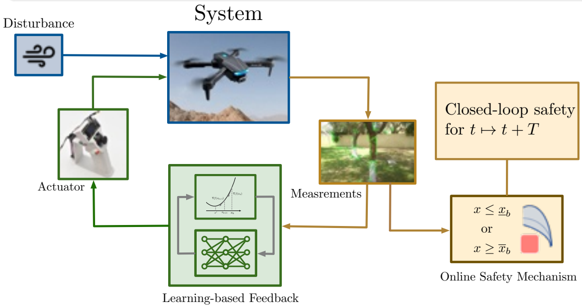
Safety guarantees for the feedback loops

Run-Time Assurance mechanism (RTA): monitor + predict

Closed-loop system without RTA



Closed-loop system with RTA



Mixed monotonicity offers a computationally efficient framework

Design of RTA mechanism

A mixed monotone compositional approach

Idea: find a decomposition function for closed-loop system

Design of RTA mechanism

A mixed monotone compositional approach

Idea: find a decomposition function for closed-loop system

System dynamics is mixed monotone with a decomposition function g

Design of RTA mechanism

A mixed monotone compositional approach

Idea: find a decomposition function for closed-loop system

System dynamics is mixed monotone with a decomposition function g

A neural network verification algorithm, for all $x \in [\underline{x}, \overline{x}]$,

$$\underline{L}(\underline{x}, \overline{x}) \leq \mathsf{N}(x) \leq \overline{L}(\underline{x}, \overline{x})$$

Design of RTA mechanism

A mixed monotone compositional approach

Idea: find a decomposition function for closed-loop system

System dynamics is mixed monotone with a decomposition function g

$$\boxed{\begin{aligned}\dot{\underline{x}} &= g(\underline{x}, \overline{x}, \underline{u}, \overline{u}) \\ \dot{\overline{x}} &= g(\overline{x}, \underline{x}, \overline{u}, \underline{u})\end{aligned}}$$

Embedding system

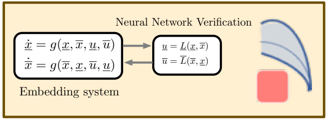
A neural network verification algorithm, for all $x \in [\underline{x}, \overline{x}]$,

$$\underline{L}(\underline{x}, \overline{x}) \leq \mathsf{N}(x) \leq \overline{L}(\underline{x}, \overline{x})$$

$$\boxed{\begin{aligned}\underline{u} &= \underline{L}(\underline{x}, \overline{x}) \\ \overline{u} &= \overline{L}(\overline{x}, \underline{x})\end{aligned}}$$

Neural Network Verification

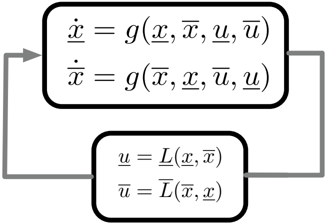
RTA mechanism = Embedding system + Neural network verification



Design of RTA mechanism

A mixed monotone compositional approach

For $x \in [\underline{x}, \overline{x}]$ feed the output of neural network verification algorithm into the embedding system



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

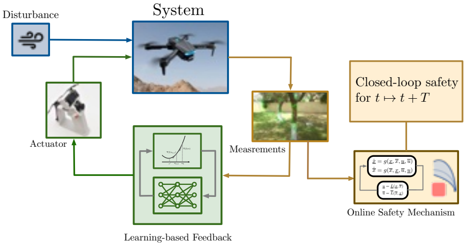
$$\underline{u} = \underline{L}(\underline{x}, \overline{x})$$
$$\overline{u} = \overline{L}(\overline{x}, \underline{x})$$

Theorem[8]

The mapping

$$h(\underline{x}, \overline{x}) = g(\underline{x}, \overline{x}, \underline{L}(\underline{x}, \overline{x}), \overline{L}(\underline{x}, \overline{x}))$$

is a decomposition function for closed-loop system



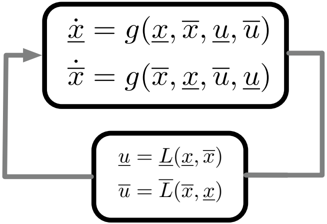
---

[8] S.Jafarpour, A. Harapanahalli, S. Coogan, arXiv 2022

Design of RTA mechanism

A mixed monotone compositional approach

For $x \in [\underline{x}, \overline{x}]$ feed the output of neural network verification algorithm into the embedding system



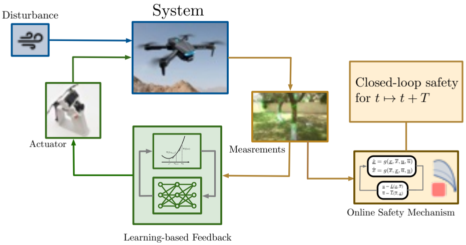
$$\dot{\underline{x}} = g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$$
$$\dot{\overline{x}} = g(\overline{x}, \underline{x}, \overline{u}, \underline{u})$$

$$\underline{u} = \underline{L}(\underline{x}, \overline{x})$$
$$\overline{u} = \overline{L}(\overline{x}, \underline{x})$$

For the dynamical system

$$\begin{aligned} \dot{\underline{x}} &= g(\underline{x}, \overline{x}, \underline{L}(\underline{x}, \overline{x}), \overline{L}(\underline{x}, \overline{x})) \qquad & \underline{x}(0) = \underline{x}_0 \\ \dot{\overline{x}} &= g(\overline{x}, \underline{x}, \overline{L}(\underline{x}, \overline{x}), (\underline{x}, \overline{x})) \qquad & \overline{x}(0) = \overline{x}_0 \end{aligned}$$

we have $\mathcal{R}([\underline{x}_0, \overline{x}_0], t) \subseteq [\underline{x}(t), \overline{x}(t)]$ for all $t \geq 0$.



---

[8] S.Jafarpour, A. Harapanahalli, S. Coogan, arXiv 2022

Vehicle experiment

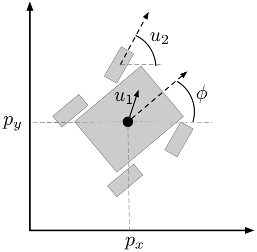
Neural network controller

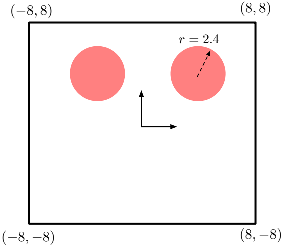
Dynamics of vehicle

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

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

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





Goal: steer the vehicle to the origin avoiding the obstacles

Vehicle experiment

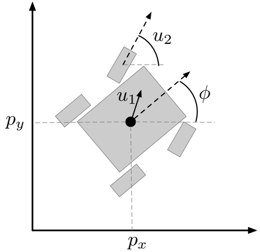
Neural network controller

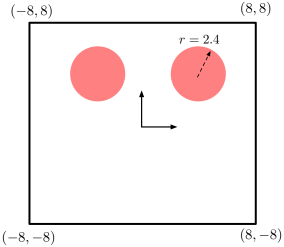
Dynamics of vehicle

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

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

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





Goal: steer the vehicle to the origin avoiding the obstacles

- **offline controller:** MPC with hard constraint to avoid the obstacles
- run MPC for 65000 randomly chosen initial condition (20 sample per trajectory)
- train a feedforward neural network $4 \mapsto 100 \mapsto 100 \mapsto 2$ with this data

Vehicle experiment

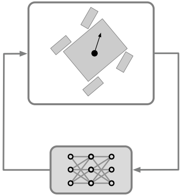
Design of RTA mechanism

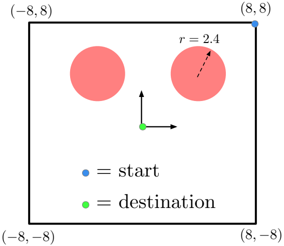
- start from $(8, 8)$ toward $(0, 0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

$$\underline{x}_0 = \begin{pmatrix} 7.9 & 7.9 & -\frac{2\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$

$$\overline{x}_0 = \begin{pmatrix} 8.1 & 8.1 & -\frac{2\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

- CROWN[9] for verification of neural network
- partition the states to improve accuracy





---

[9] H. Zhang, T-W. Weng, P-Y. Chen, C-J. Hsieh, L. Daniel, NeurIPS 2018

Vehicle experiment

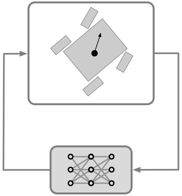
Design of RTA mechanism

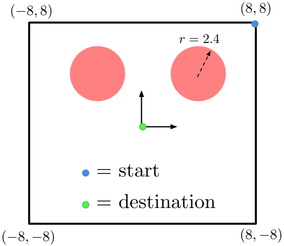
- start from $(8,8)$ toward $(0,0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

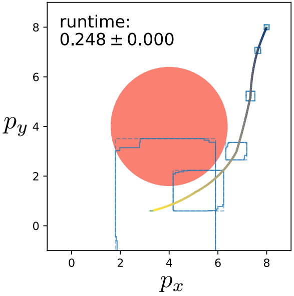
$$\underline{x}_0 = \begin{pmatrix} 7.9 & 7.9 & -\frac{2\pi}{3} - 0.01 & 1.99 \end{pmatrix}^\top$$

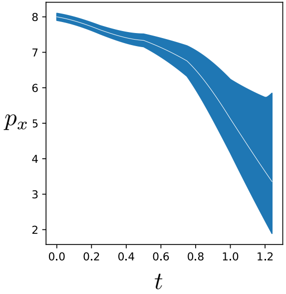
$$\overline{x}_0 = \begin{pmatrix} 8.1 & 8.1 & -\frac{2\pi}{3} + 0.01 & 2.01 \end{pmatrix}^\top$$

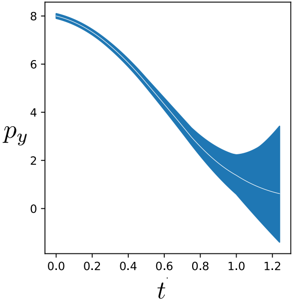
- CROWN[9] for verification of neural network
- partition the states to improve accuracy









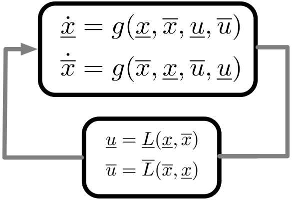


---

[9] H. Zhang, T-W. Weng, P-Y. Chen, C-J. Hsieh, L. Daniel, NeurIPS 2018

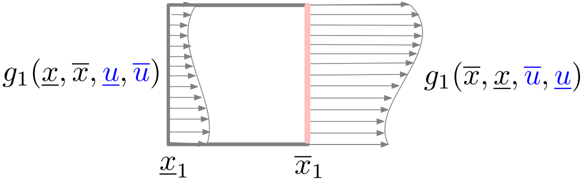
Design of RTA mechanism

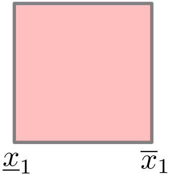
The source of conservativeness



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

$$\underline{u} = \underline{L}(\underline{x}, \overline{x})$$
$$\overline{u} = \overline{L}(\overline{x}, \underline{x})$$



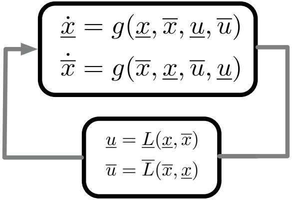


$$\underline{u} = \underline{L}(\underline{x}, \overline{x}) \leq N(x), \qquad \text{for all} \quad x \in [\underline{x}, \overline{x}]$$

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

Design of RTA mechanism

The source of conservativeness

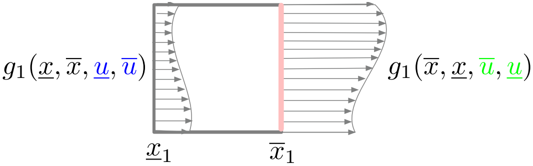


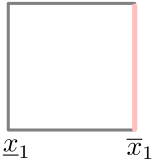
$$\dot{\underline{x}} = g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$$

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

$$\underline{u} = \underline{L}(\underline{x}, \overline{x})$$

$$\overline{u} = \overline{L}(\overline{x}, \underline{x})$$





$$\underline{u} = \underline{L}(\overset{\underline{x}_i \mapsto \overline{x}_i}{\cancel{\underline{x}}}, \overline{x}) \leq N(x) \quad \text{for all} \quad x \in \left[\begin{pmatrix} \overline{x}_1 \\ \underline{x}_2 \end{pmatrix}, \begin{pmatrix} \overline{x}_1 \\ \overline{x}_2 \end{pmatrix}\right]$$

$$N(x) \leq \overline{L}(\overset{\underline{x}_i \mapsto \overline{x}_i}{\cancel{\underline{x}}}, \overline{x}) = \overline{u} \quad \text{for all} \quad x \in \left[\begin{pmatrix} \overline{x}_1 \\ \underline{x}_2 \end{pmatrix}, \begin{pmatrix} \overline{x}_1 \\ \overline{x}_2 \end{pmatrix}\right]$$

Design of RTA mechanism

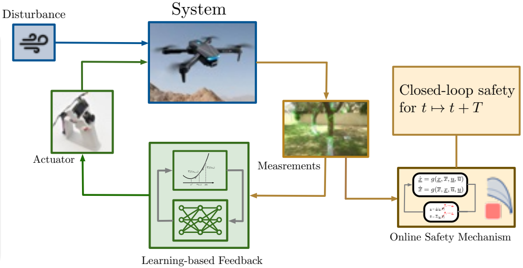
Revised compositional approach

Theorem[10]

The mapping

$$h_i(\underline{x},\overline{x}) = g_i\Big(\underline{x},\overline{x},\quad \underline{L}(\underline{x},\overset{\overline{x}_i \mapsto \underline{x}_i}{\cancel{\overline{x}}}),\quad \overline{L}(\underline{x},\overset{\overline{x}_i \mapsto \underline{x}_i}{\cancel{\overline{x}}})\Big)$$

is a decomposition function for closed-loop system



---

[10] S.Jafarpour, A. Harapanahalli, S. Coogan, arXiv 2022

Design of RTA mechanism

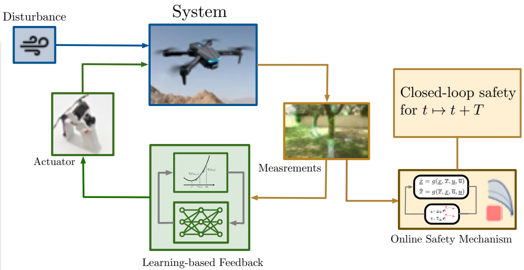
Revised compositional approach

For the dynamical system

$$\dot{\underline{x}}_i = g_i(\underline{x}, \overline{x}, \quad \underline{L}(\underline{x}, \cancelto{\overline{x}_i \mapsto \underline{x}_i}{\overline{x}}), \quad \overline{L}(\underline{x}, \cancelto{\overline{x}_i \mapsto \underline{x}_i}{\overline{x}})) \quad \underline{x}_i(0) = (\underline{x}_0)_i$$

$$\dot{\overline{x}}_i = g_i(\overline{x}, \underline{x}, \quad \overline{L}(\cancelto{\underline{x}_i \mapsto \overline{x}_i}{\underline{x}}, \overline{x}), \quad \underline{L}(\cancelto{\underline{x}_i \mapsto \overline{x}_i}{\underline{x}}, \overline{x})) \quad \overline{x}_i(0) = (\overline{x}_0)_i$$

we have $\mathcal{R}(\mathcal{X}_0, t) \subseteq [\underline{x}(t), \overline{x}(t)]$ for all $t \geq 0$.



---

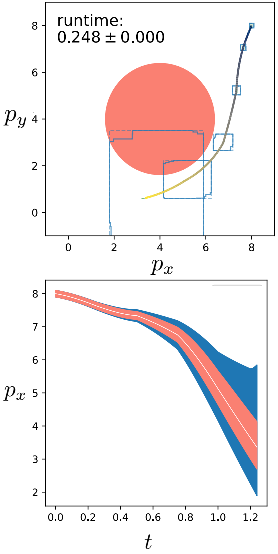
[10] S.Jafarpour, A. Harapanahalli, S. Coogan, arXiv 2022

Vehicle experiment revisited

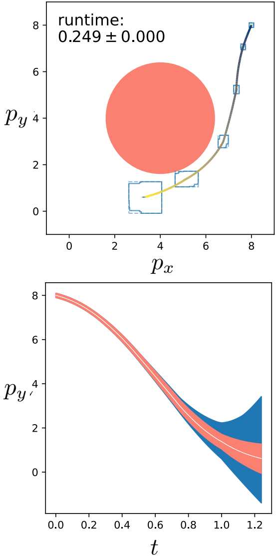
Design of RTA mechanism

- CROWN to verify the neural network
- partition the states to improve accuracy
- blue $=$ reachable set of **old**
- red$=$ reachable set of **new**
- very small increase in computational time
- significant improvement in accuracy
- New decomposition function **certify** that closed-loop system is avoiding the obstacle

Old decomposition function



New decomposition function



Conclusions

- A computationally efficient framework for reachability of dynamical systems
- Exploit the structure in **isolated neural networks** and **Neural network feedback loops**
- Suitable for generalized neural networks
    - Sufficient conditions for their well-posedness
    - Hyper-rectangular over-approximation of reachable sets
- Safety assurance mechanism for monitoring neural network feedback loops
- Mixed monotonicity to design a computationally efficient run-time assurance algorithm

Acknowledgment

Collaborators

Alexander Davydov

UCSB

Matthew Abate

Georgia Tech

Pedro Cisneros-Velarde

UIUC

Akash Harapanahalli

Georgia Tech

Anton Proskurnikov

Politecnico di Torino

Francesco Bullo

UCSB

Samuel Coogan

Georgia Tech

Thank you for your attention!

Generalized Structure

Comparison with feedforward neural networks

- Feedforward neural networks:

$$x^{(\ell+1)} = \Phi(A_\ell x^{(\ell)} + b_\ell), \quad x^{(0)} = u$$
$$y = A_k x^{(k)} + b_k$$

- Generalized neural networks:

$$x = \Phi(Ax + Bu + b)$$
$$y = Cx + c$$

$$\begin{bmatrix} x^{(k)} \\ x^{(k-1)} \\ \vdots \\ x^{(2)} \\ x^{(1)} \end{bmatrix} = \Phi\left(\begin{bmatrix} 0 & A_{k-1} & 0 & \dots & 0 \\ 0 & 0 & A_{k-2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A_1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}\begin{bmatrix} x^{(k)} \\ x^{(k-1)} \\ \vdots \\ x^{(2)} \\ x^{(1)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ A_0 \end{bmatrix} u + \begin{bmatrix} b_{k-1} \\ b_{k-2} \\ \vdots \\ b_1 \\ b_0 \end{bmatrix}\right)$$

$$y = \begin{bmatrix} A_k & 0 & 0 & \dots & 0 \end{bmatrix}\begin{bmatrix} x^{(k)} \\ x^{(k-1)} \\ \vdots \\ x^{(2)} \\ x^{(1)} \end{bmatrix} + b_k$$

$$\begin{bmatrix} x^{(k)} \\ x^{(k-1)} \\ \vdots \\ x^{(2)} \\ x^{(1)} \end{bmatrix} = \Phi\left(\begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1k} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{(k-1)1} & A_{(k-1)2} & A_{(k-1)3} & \dots & A_{(k-1)k} \\ A_{k1} & A_{k2} & A_{k3} & \dots & A_{kk} \end{bmatrix}\begin{bmatrix} x^{(k)} \\ x^{(k-1)} \\ \vdots \\ x^{(2)} \\ x^{(1)} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{k-1} \\ B_k \end{bmatrix} u + \begin{bmatrix} b_{k-1} \\ b_{k-2} \\ \vdots \\ b_1 \\ b_0 \end{bmatrix}\right)$$

$$y = \begin{bmatrix} C_1 & C_2 & C_3 & \dots & C_k \end{bmatrix}\begin{bmatrix} x^{(k)} \\ x^{(k-1)} \\ \vdots \\ x^{(2)} \\ x^{(1)} \end{bmatrix} + c$$

Generalized Structure

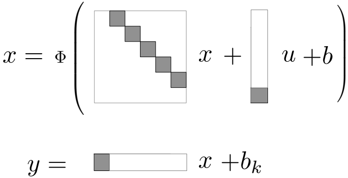
Comparison with feedforward neural networks

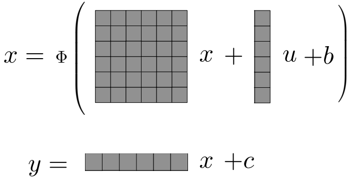
- Feedforward neural networks:

$$x^{(\ell+1)} = \Phi(A_\ell x^{(\ell)} + b_\ell), \quad x^{(0)} = u$$
$$y = A_k x^{(k)} + b_k$$

- Generalized neural networks:

$$x = \Phi(Ax + Bu + b)$$
$$y = Cx + c$$





Training of INNs

Promoting robustness via regularization

1. loss function $\mathcal{L}$ and training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^N$
2. $\epsilon =$ size of ℓ_∞-perturbation in input: $\mathcal{U} = [\underbrace{u - \epsilon \mathbb{1}_r}_{\underline{u}}, \underbrace{u + \epsilon \mathbb{1}_r}_{\overline{u}}]$

Training INNs

$$\min_{A,B,b,c} \sum_{i=1}^N \mathcal{L}(\widehat{y}_i, Cx_i + c)$$

$$x_i = \Phi(Ax_i + B\widehat{u}_i + b),$$

$$a_{ii} + \sum_{j=1} |a_{ij}| \leq \gamma \quad \text{well-posedness}$$

Training FFNNs

$$\min_{A,B,b,c} \sum_{i=1}^N \mathcal{L}(\widehat{y}_i, Cx_i^{(k)} + c)$$

$$x_i^{(\ell+1)} = \Phi(A_\ell x_i^{(\ell)} + b_\ell), \quad \ell \in \{1, \dots, k-1\}$$

Training of INNs

Promoting robustness via regularization

1. loss function $\mathcal{L}$ and training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^{N}$
2. $\epsilon =$ size of ℓ_∞-perturbation in input: $\mathcal{U} = [\underbrace{u - \epsilon \mathbb{1}_r}_{\underline{u}}, \underbrace{u + \epsilon \mathbb{1}_r}_{\overline{u}}]$

Our main result

output $y \in [\underline{y}(\epsilon), \overline{y}(\epsilon)]$

Training INNs

$$\min_{A,B,b,c} \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i + c) + \kappa \underbrace{\mathcal{R}(\underline{y}_i(\epsilon), \overline{y}_i(\epsilon))}_{\text{robustness}}$$

$$x_i = \Phi(Ax_i + B\widehat{u}_i + b),$$

$$a_{ii} + \sum_{j=1} |a_{ij}| \leq \gamma < 1 \quad \text{well-posedness}$$

Training FFNNs (S. Gowal, et. al., 2018)

$$\min_{A,B,b,c} \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i^{(k)} + c) + \kappa \underbrace{\mathcal{R}(\underline{y}_i(\epsilon), \overline{y}_i(\epsilon))}_{\text{robustness}}$$

$$x_i^{(\ell+1)} = \Phi(A_\ell x_i^{(\ell)} + b_\ell), \quad \ell \in \{1, \dots, k-1\}$$

- $\mathcal{R}(\underline{y}(\epsilon), \overline{y}(\epsilon))$ uses $\underline{y}(\epsilon)$ and $\overline{y}(\epsilon)$ to estimate robustness margin
- κ, ϵ, γ are hyperparameters