Exploiting Structure in Feedback Systems with Learning-based Components

Saber Jafarpour

Decision and Control Laboratory
Georgia Institute of Technology

April 26, 2023

Power grids

Transportation networks

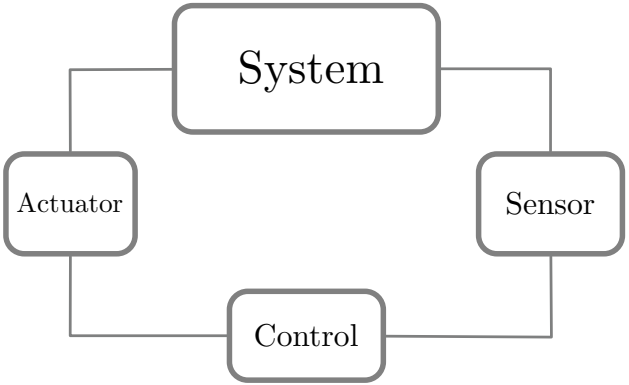
Learning-based systems

- 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

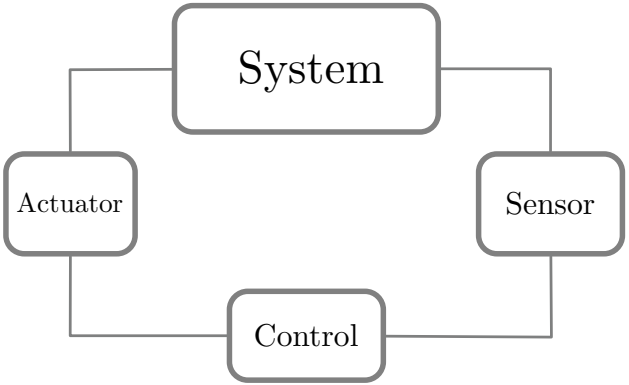
reconsider the traditional approaches for **monitoring** and **control** of autonomous systems

Feedback is a central paradigm in control theory

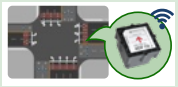


Magic of feedback[a]: robustness, shape behavior, command tracking, etc.

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



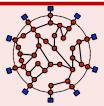
Agents have wide range of **communication** capabilities

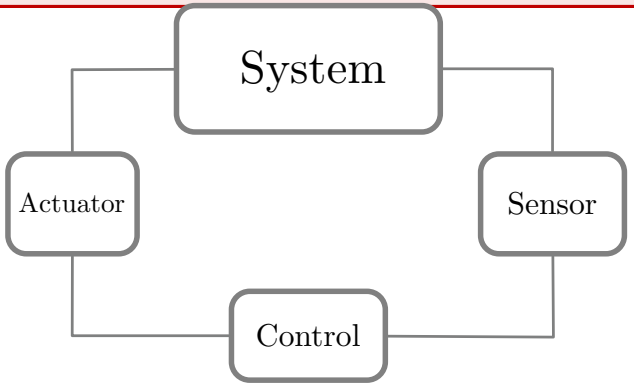


Large number of measurement devices for sensing

Enhanced processing units allow new **computational** approaches

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

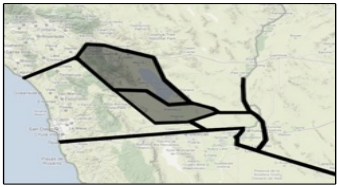




Controllers contain **high-dimensional**, **learning-based**, and **complex** parts

A critical task

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



2011 US Southwest blackout

Traffic congestion in Beijing

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

Analysis 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)

Nonlinear controllability

- 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)

Learning algorithms

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

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

Feedback controller or some elements of it are learned from data

Aerial vehicles

Self-driving cars

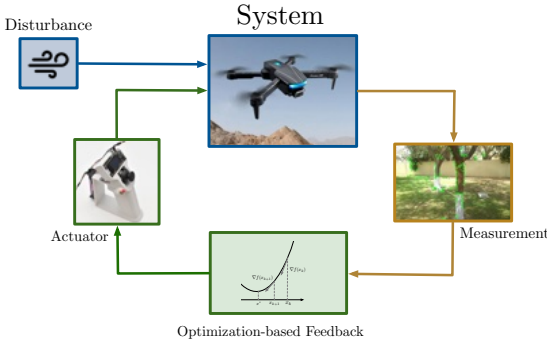
GaTech A1 robot

Why data-driven feedback?

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

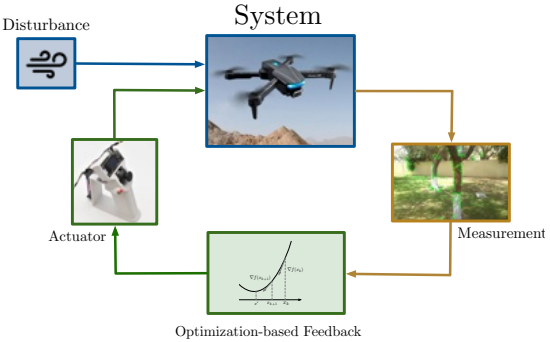
Assumption: An (approximate) model of the system is available

Optimization-based control



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

Optimization-based control



- Example method: Model Predictive Control (MPC)

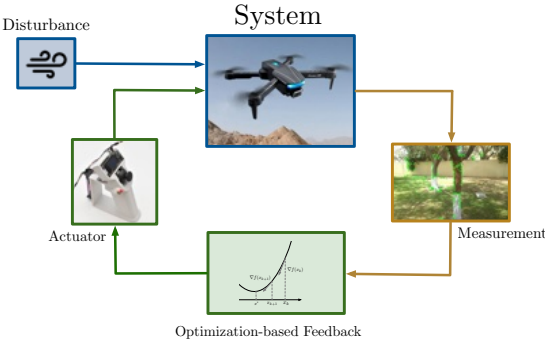
$$\min_{u(0),\ldots,u(N-1)} \quad \sum_{i=0}^{N-1} \ell(x(t),u(t)) + \phi(x(N)),$$
$$x(t+1) = x(t) + \alpha f(x(t),u(t)),$$
$$x(t) \in \mathcal{X}, \quad t \in \{1,\ldots,N\}$$
$$u(t) \in \mathcal{U}, \quad t \in \{0,\ldots,N-1\}$$
$$x(0) = x$$

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

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

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

Optimization-based control



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

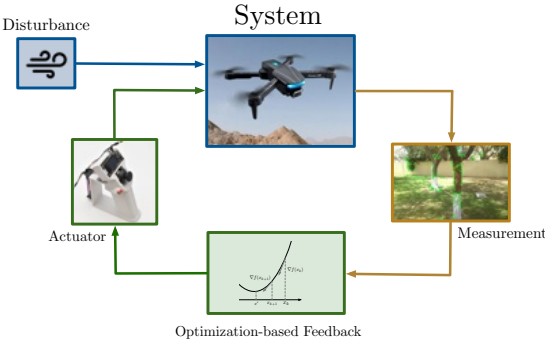
$$\min_{u(0),\ldots,u(N-1)} \quad \sum_{i=0}^{N-1} \ell(x(t),u(t)) + \phi(x(N)),$$
$$x(t+1) = x(t) + \alpha f(x(t),u(t)),$$
$$x(t) \in \mathcal{X}, \quad t \in \{1,\ldots,N\}$$
$$u(t) \in \mathcal{U}, \quad t \in \{0,\ldots,N-1\}$$
$$x(0) = x$$

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

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

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

Optimization-based control



- Example issues: the optimization problem is computationally complicated

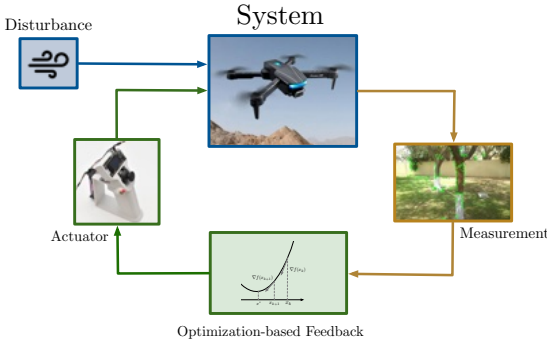
$$\min_{u(0),\ldots,u(N-1)} \quad \sum_{i=0}^{N-1} \ell(x(t),u(t)) + \phi(x(N)),$$
$$x(t+1) = x(t) + \alpha f(x(t),u(t)),$$
$$x(t) \in \mathcal{X}, \quad t \in \{1,\ldots,N\}$$
$$u(t) \in \mathcal{U}, \quad t \in \{0,\ldots,N-1\}$$
$$x(0) = x$$

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

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

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

Optimization-based control

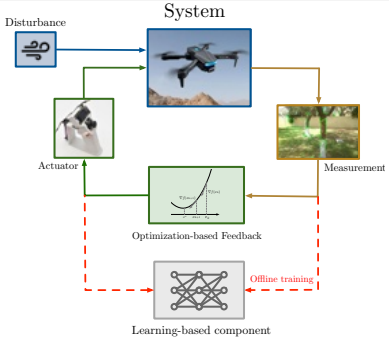


- full knowledge of environment
- computationally complexity

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

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

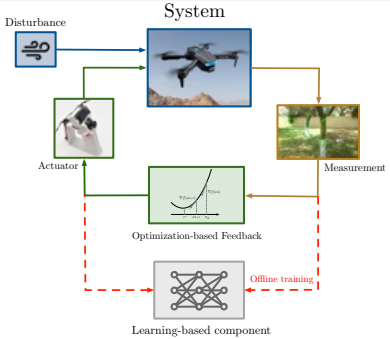
Offline training



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

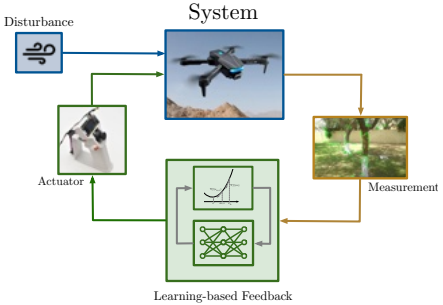
Assumption: An (approximate) model of the system is available

Offline training



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

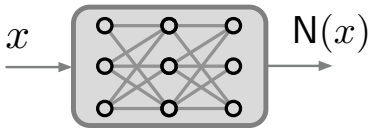
Online implementation



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

Isolated learning component

- Robustness of learning algorithms



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

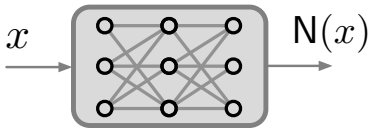
Output perturbations

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

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

Isolated learning component

- Robustness of learning algorithms



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

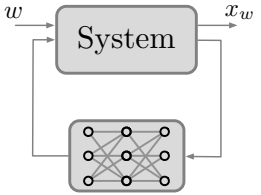
Output perturbations

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

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 $\mathcal{R}(\mathcal{W}, t) \subset \mathcal{S}$

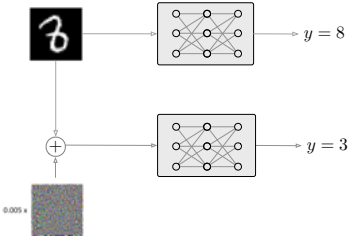
1. learning algorithms are fragile wrt input perturbations

Adversarial Perturbations

Small changes in the input

↓

Large changes in the output



C. Szegedy and et. al. Intriguing properties of neural networks. In *ICLR*, 2014

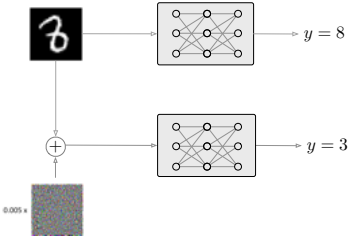
1. learning algorithms are fragile wrt input perturbations

Adversarial Perturbations

Small changes in the input

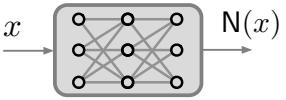
↓

Large changes in the output



C. Szegedy and et. al. Intriguing properties of neural networks. In *ICLR*, 2014

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



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

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

not scalable to large-scale systems

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

not scalable to large-scale systems

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

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

not scalable to large-scale systems

- Reachability of learning algorithms is more recent: $\sim$ 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

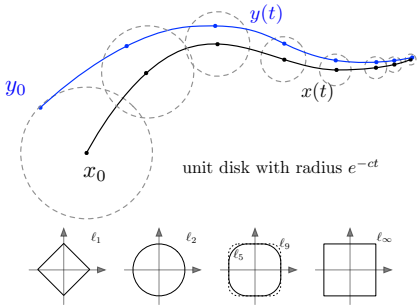
- **Contractivity**, to ensure computational efficiency
- **Mixed monotonicity**, a key property of neural network loops

Outline of this talk

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

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 $\|\cdot\|$

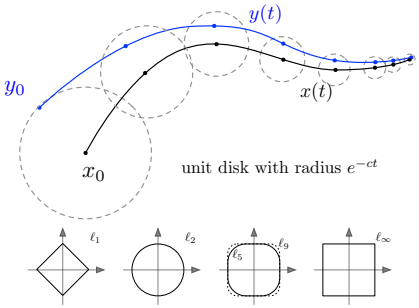


Definition (Contraction)

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



Differential and **integral** characterization of contractivity

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,

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

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

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,

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

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

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
- $[\![x, x]\!] = \|x\|^2$

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

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

Theorem[2]

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

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

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

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

Theorem

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

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

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

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

f is a contracting vector field wrt to $\|\cdot\|_2$
iff
$-f$ is a strongly monotone operator wrt to the inner product $\langle\cdot,\cdot\rangle$.

Theorem

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

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

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

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

f is a contracting vector field wrt to $\|\cdot\|$
iff
$-f$ is a strongly monotone operator wrt to the weak pairing $[\![\cdot, \cdot]\!]$.

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: $(\overline{x}, \overline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$

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: $(\overline{x}, \overline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$

- f locally Lipschitz $\implies$ mixed monotonicity

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: $(\overline{x}, \overline{u}) \mapsto g(\underline{x}, \overline{x}, \underline{u}, \overline{u})$

- f locally Lipschitz $\implies$ mixed monotonicity
- g is not unique: structure of the system to construct one.

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.

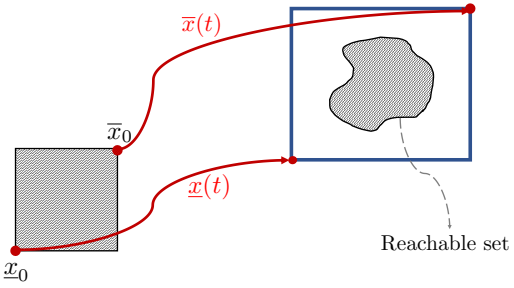
[1] S. Coogan, M. Arcak, HSCC 2015

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

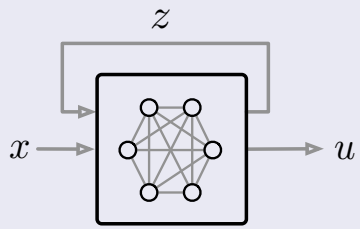
$$\dot{\underline{x}} = g(\underline{x}, \overline{x}, \underline{u}, \overline{u}), \qquad \underline{x}(0) = \underline{x}_0$$
$$\dot{\overline{x}} = g(\overline{x}, \underline{x}, \overline{u}, \underline{u}), \qquad \overline{x}(0) = \overline{x}_0$$



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

Outline of this talk

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

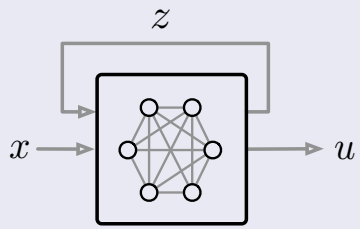


- Generalized neural networks:

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

- $\Phi(y_1, \ldots, y_n) = (\phi_1(y_1), \ldots, \phi_n(y_n))^\top$ 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



- 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

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?

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	$\iff$	Dynamical system
$z = \Phi(Az + Bx + b)$		$\dot{z} = -z + \Phi(Az + Bx + b)$
fixed-points	$\iff$	**equilibrium points**
robustness	$\iff$	**forward reachability** $(t = \infty)$

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

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-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.

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

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

Fixed-point of	$\iff$	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.

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

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

Fixed-point of	$\iff$	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.

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

Theorem[4]

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

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

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

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

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

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

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

Theorem[5]

The neural network is mixed monotone with a decomposition function:

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

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

Theorem[6]

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

Theorem[6]

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

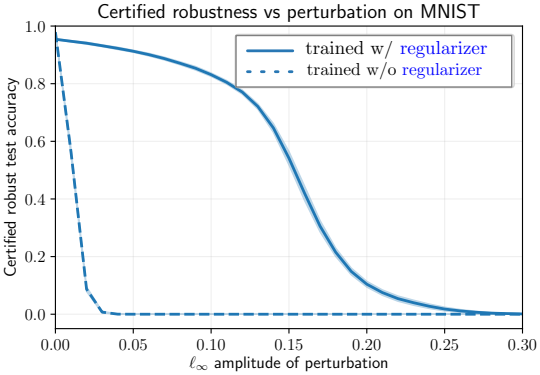
1. $z = \Phi(Az + Bx + b)$ has a unique solution z_u^*
2. $\begin{bmatrix} \underline{z} \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathsf{G}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) \\ \mathsf{G}(\overline{z}, \underline{z}, \overline{x}, \underline{x}) \end{bmatrix}$ has a unique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

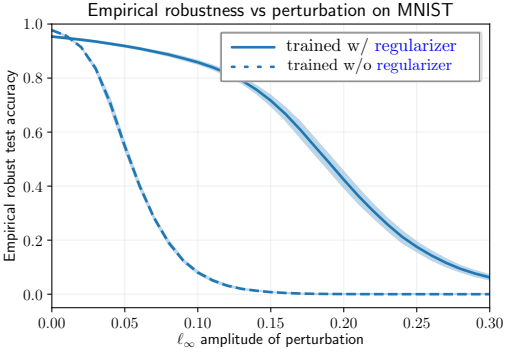
- **Verification:** find robustness margin of generalized neural networks
- **Training:** design robust generalized neural networks
  - $a_{ii} + \sum_{j \neq i} |a_{ij}| < 1$ as a constraint to the training problem
  - a regularization term $\mathcal{R}(\underline{u}, \overline{u})$ to the training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

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

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

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



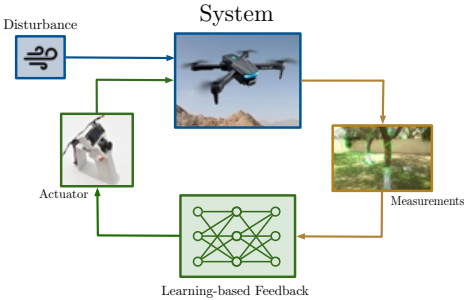


- Certified robustness = all the elements of $[\underline{u}(\epsilon), \overline{u}(\epsilon)]$ classify as the correct digit
- Empirical robustness = Projected Gradient Descent (PGD) attack

Outline of this talk

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

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



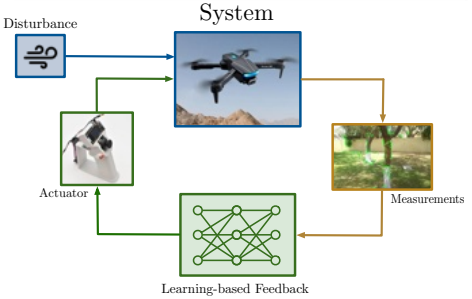
Closed-loop safety for $t \mapsto t + T$

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

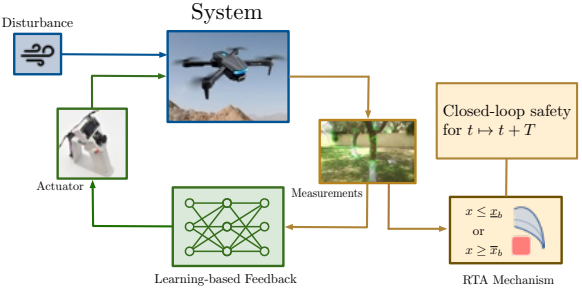
RTA Mechanism

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

Closed-loop system without RTA

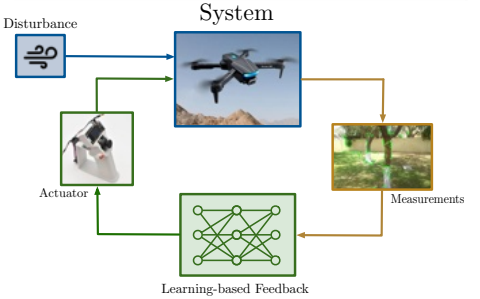


Closed-loop system with RTA

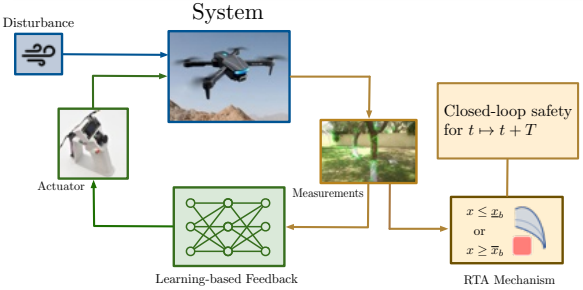


Run-Time Assurance mechanism (RTA): monitor + predict

Closed-loop system without RTA



Closed-loop system with RTA



Mixed monotonicity offers a computationally efficient framework

System dynamics is mixed monotone with a decomposition function g

System dynamics is mixed monotone with a decomposition function g

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

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

System dynamics is mixed monotone with a decomposition function g

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

Embedding system

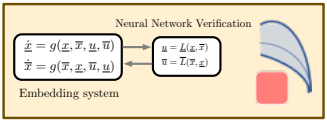
A neural network verification algorithm, for all $x \in [\underline{x}, \overline{x}]$,

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

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

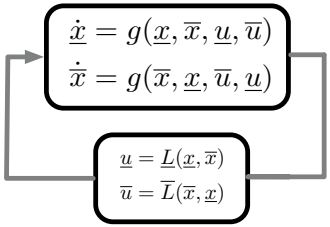
Neural Network Verification

RTA mechanism = Embedding system + Neural network verification



Naive compositional approach:

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



In practice this approach is overly-conservative

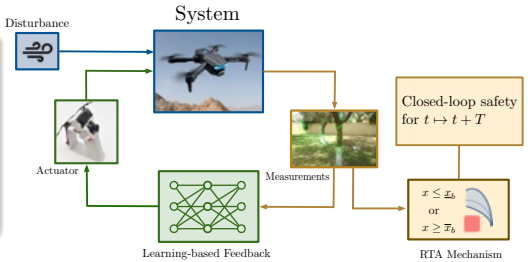
New approach: closed-loop perspective toward learning-based system

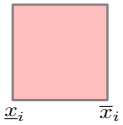
Idea: find a decomposition function for closed-loop system

Theorem[7]

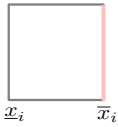
The closed-loop system is mixed-monotone with a decomposition function

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





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



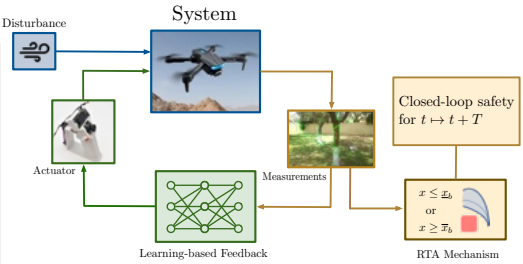
$$\underline{u} = \underline{L}(\underline{x}^{\underline{x}_i \mapsto \overline{x}_i},\overline{x}) \le N(x) \quad \text{for all } x \in \left[\begin{pmatrix}\overline{x}_i\\ \underline{x}_{-i}\end{pmatrix}, \begin{pmatrix}\overline{x}_i\\ \overline{x}_{-i}\end{pmatrix}\right]$$
$$N(x) \le \overline{L}(\underline{x}^{\underline{x}_i \mapsto \overline{x}_i},\overline{x}) = \overline{u} \quad \text{for all } x \in \left[\begin{pmatrix}\overline{x}_i\\ \underline{x}_{-i}\end{pmatrix}, \begin{pmatrix}\overline{x}_i\\ \overline{x}_{-i}\end{pmatrix}\right]$$

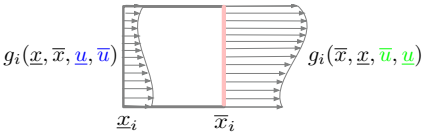
For the dynamical system

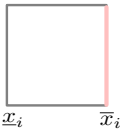
$$\dot{\underline{x}}_i = g_i(\underline{x}, \overline{x}, \quad \underline{L}(\underline{x}, \overline{x}), \quad \overline{L}(\underline{x}, \overline{x})) \quad \underline{x}_i(0) = (\underline{x}_0)_i \qquad \overline{x}_i \mapsto \underline{x}_i \quad \overline{x}_i \mapsto \underline{x}_i$$

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

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







$$\underline{u} = \underline{L}(\underline{x}, \overline{x}) \leq N(x) \quad \text{for all} \quad x \in \left[\begin{pmatrix} \overline{x}_i \\ \underline{x}_{-i} \end{pmatrix}, \begin{pmatrix} \overline{x}_i \\ \overline{x}_{-i} \end{pmatrix} \right] \qquad \underline{x}_i \mapsto \overline{x}_i$$

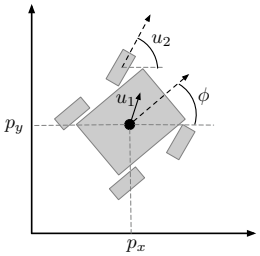
$$N(x) \leq \overline{L}(\underline{x}, \overline{x}) = \overline{u} \quad \text{for all} \quad x \in \left[\begin{pmatrix} \overline{x}_i \\ \underline{x}_{-i} \end{pmatrix}, \begin{pmatrix} \overline{x}_i \\ \overline{x}_{-i} \end{pmatrix} \right] \qquad \underline{x}_i \mapsto \overline{x}_i$$

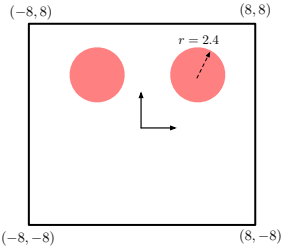
Dynamics of vehicle

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

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

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





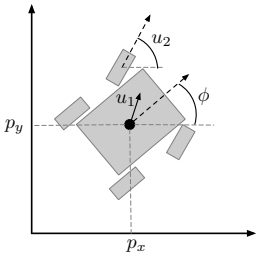
Goal: steer the vehicle to the origin avoiding the obstacles

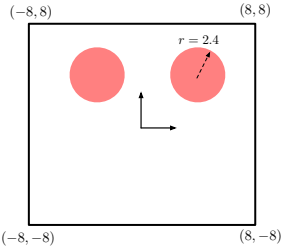
Dynamics of vehicle

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

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

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





Goal: steer the vehicle to the origin avoiding the obstacles

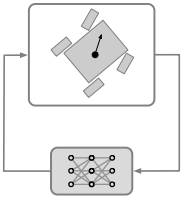
- **offline controller:** MPC with hard constraint to avoid the obstacles
- run MPC for 65000 randomly chosen initial condition (20 sample per trajectory)
- train a feedforward neural network $4 \mapsto 100 \mapsto 100 \mapsto 2$ with this data

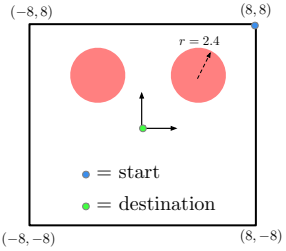
- start from $(8, 8)$ toward $(0, 0)$
- $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$ with

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

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

- CROWN for verification of neural network

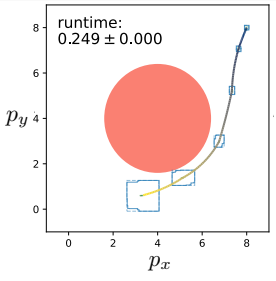


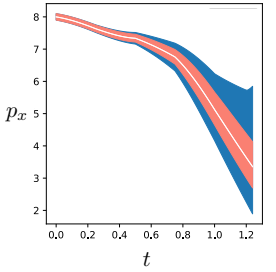


- partition the states to improve accuracy
- very small increase in computational time
- significant improvement in accuracy
- mixed monotone approach **certify** that closed-loop system is avoiding the obstacle

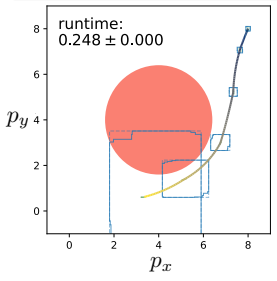
blue = reachable set of **naive compositional**
red= reachable set of **mixed monotone**

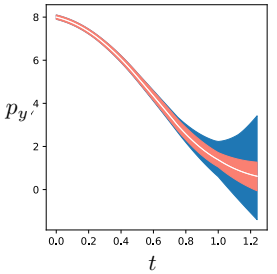
mixed monotone approach





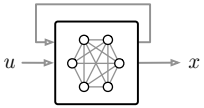
naive compositional approach

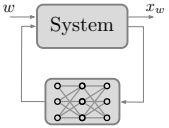




Conclusions

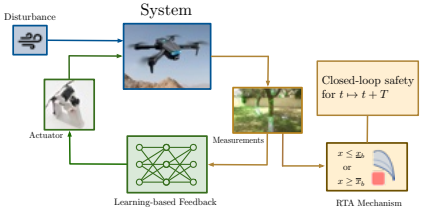
- A computationally efficient framework for reachability of dynamical systems
- Exploit the structure in **isolated neural networks** and **neural network feedback loops**





- This framework naturally leads to tractable analysis of generalized neural networks
  - Sufficient conditions for their well-posedness
  - Hyper-rectangular over-approximation of reachable sets

- Mixed monotone approach to design run-time assurance algorithm for closed-loop systems with neural network controllers



Outline of this talk

- Contraction theory and mixed monotonicity
- Isolated learning algorithms
- Learning-based feedback loop
- Future research directions

Future research directions

Scalable analysis and control of large-scale Cyber-Physical Systems (CPS)

contraction theory as a unifying framework

Verification and design of closed-loop learning-based systems

closed-loop dynamical system perspective

Security and fault protection in large-scale modern power grids

a network perspective to penetration of renewable generators to the grid

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!

- Origins:

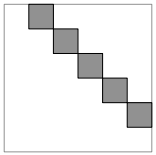
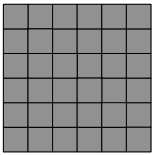
S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In *NeurIPS*, 2019

L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Y. Tsai. Implicit deep learning. *SIMODS*, 2019

A. Kag, Z. Zhang, and V. Saligrama. RNNs incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients? In *ICLR*, 2020

- Generalizing feedforward neural networks to fully-connected synaptic matrices

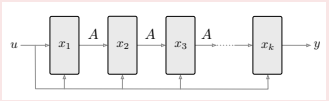
Intuition: $z^{i+1} = \phi_i(A_i z^i + b_i) \quad \Longleftrightarrow \quad z = \Phi(Ax + Bu + b)$, where A has upper diagonal structure.

$A_{\text{upper-diagonal}} =$  $A_{\text{complete}} =$ 

- comparable accuracy to traditional neural networks with significant memory reduction

  S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In *NeurIPS*, 2019

Intuition: generalized neural network = weight-tied infinite-layer network



$z^{i+1} = \phi_i(Az^i + B_i x + b_i) \implies \lim_{i\to\infty} z^i = x^*$ solution to the generalized neural network

- suitable for learning constrained optimization problems

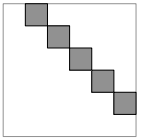
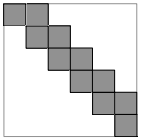
  A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. In *NeurIPS*, 2019

Intuition: casting KKT condition as an implicit layer

- vanishing and exploding gradient

A. Kag, Z. Zhang, and V. Saligrama. RNNs incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients? In *ICLR*, 2020

Intuition: the notion of "autapse" (time-delayed self-feedback) from neuroscience

$A_{\text{upper-diagonal}} =$  $A_{\text{Autapse}} =$ 

- suitable for learning stiff problems or problems with discontinuity

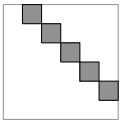
S. Pfrommer, M. Halm, and M. Posa. ContactNets: Learning discontinuous contact dynamics with smooth, implicit representations. *arXiv preprint*, 2020

- Feedforward neural networks:

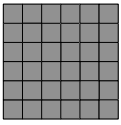
$$z^{(\ell+1)} = \Phi(A_\ell z^{(\ell)} + b_\ell), \quad z^{(0)} = x$$
$$u = A_k z^{(k)} + b_k$$

- Generalized neural networks:

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

$z =$ Φ  $z +$  $x + b$

$u =$  $z + b_k$

$z =$ Φ  $z +$  $x + b$

$u =$  $z + c$

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

Training generalized neural networks

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

$$z_i = \Phi(Az_i + B\widehat{u}_i + b),$$

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

Training FFNNs

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

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

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

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

Training generalized neural networks

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

$$z_i = \Phi(Az_i + B\widehat{u}_i + b),$$

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

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

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

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

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