Interval Reachability of Nonlinear Dynamical Systems with Neural Network Controllers

Saber Jafarpour*[1], Akash Harapanahalli*[1], and Samuel Coogan [1]

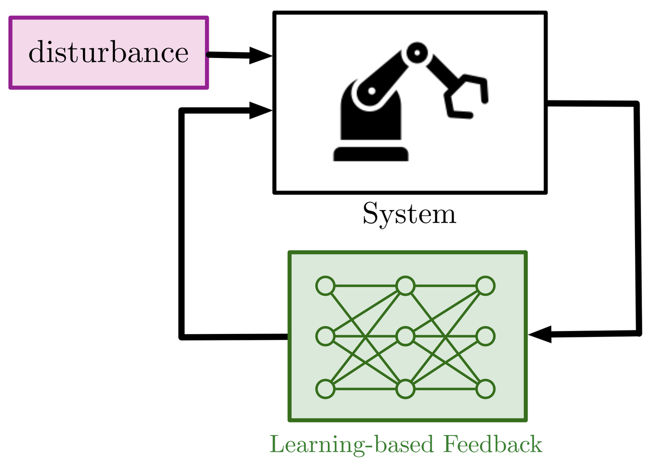
(1) Georgia Institute of Technology, {saber,aharapan,sam.coogan}@gatech.edu

Neural Network Controllers

- Neural networks are deployed as controllers in safety-critical applications (self driving vehicle and mobile robots).

Problem

Under uncertainty, ensure safety of the closed-loop system.

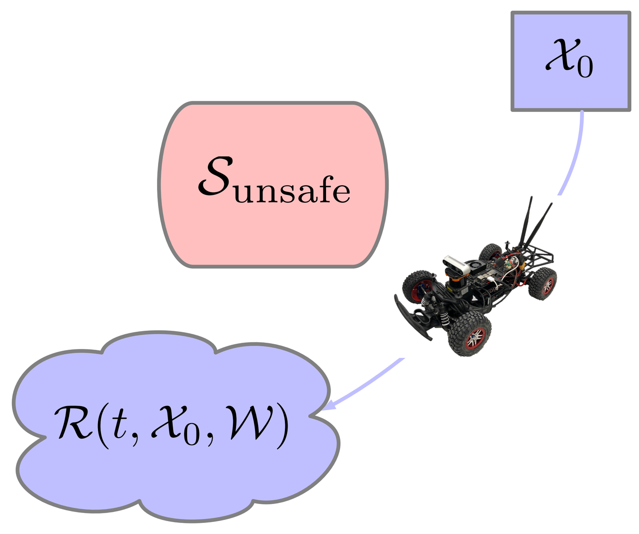


Challenges:

1. Neural networks are brittle with respect to input perturbations
2. The error can compound in the closed-loop interconnection.

Safety Verification via Reachability Analysis

- The disturbance $\mathcal{W}$
- The initial uncertainty $\mathcal{X}_0$
- The *reachable set* $\mathcal{R}(t, \mathcal{X}_0, \mathcal{W}) = \{x(t) \text{ is a trajectory}\}$
- Unsafe set $\mathcal{S}_{\text{unsafe}} \subseteq \mathbb{R}^n$



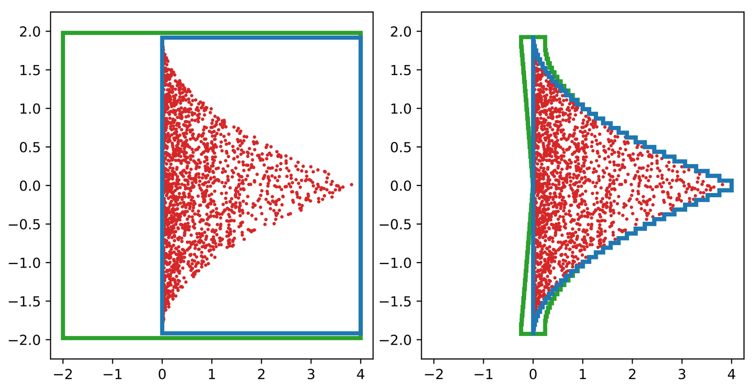
Find an over-approximation $\overline{\mathcal{R}}(t, \mathcal{X}_0, \mathcal{W})$, and check if

$$\overline{\mathcal{R}}(t, \mathcal{X}_0, \mathcal{W}) \bigcap \mathcal{S}_{\text{unsafe}} = \varnothing$$

Interval Analysis

Goal: over-approximate the output of a mapping using intervals.

- $\mathsf{G} = \left[\frac{\underline{\mathsf{G}}}{\overline{\mathsf{G}}}\right]$ is an *inclusion function* for g if for every $x \in [\underline{x}, \overline{x}]$,
$$g(x) \in [\underline{\mathsf{G}}(\underline{x}, \overline{x}), \overline{\mathsf{G}}(\underline{x}, \overline{x})].$$



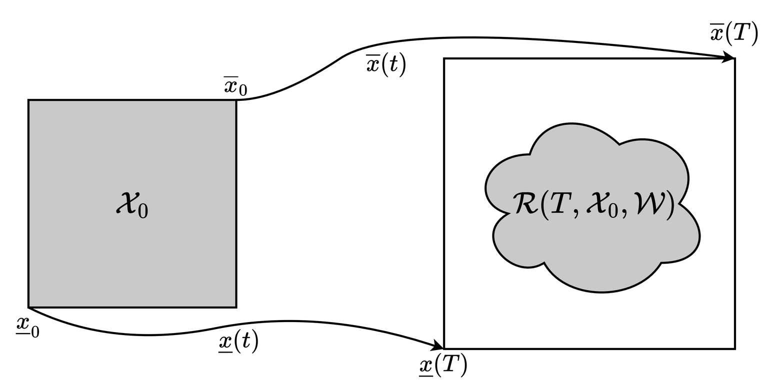
- Inclusion functions can capture localized behaviors of functions—they preserve the structure when the intervals are small.
- Different approaches exist for constructing inclusion functions.

Interval Reachability of Dynamical Systems

- Consider $\dot{x} = f(x, w)$ with an inclusion function $\mathsf{F} = \left[\frac{\underline{\mathsf{F}}}{\overline{\mathsf{F}}}\right]$ for f, the *embedding system* is

$$\dot{\underline{x}}_i = \underline{\mathsf{F}}_i(\underline{x}, \overline{x}_{i:\underline{x}}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}}_i = \overline{\mathsf{F}}_i(\underline{x}_{i:\overline{x}}, \overline{x}, \underline{w}, \overline{w})$$

A single trajectory of the embedding system provides lower bound $\underline{x}$ and upper bound $\overline{x}$ on reachable set of original system at time t.



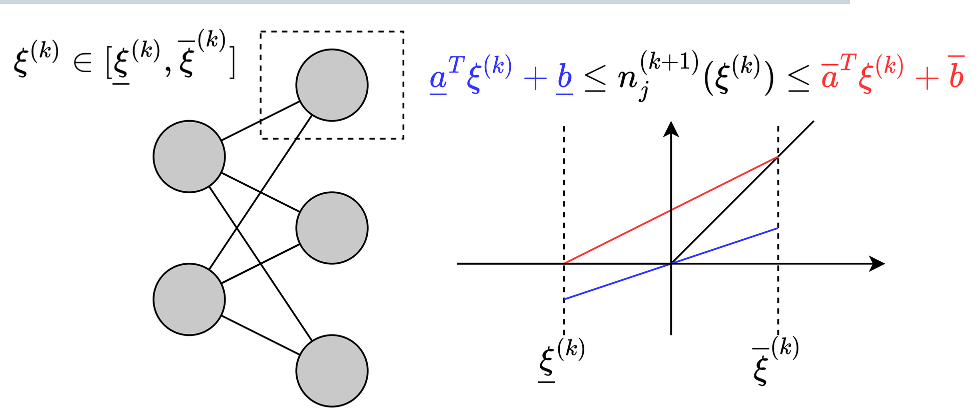
Main Question: How to construct an embedding system for the neural network controlled system?

Inclusion Functions for Neural Networks

Find $\underline{N}, \overline{N}$ such that for every $x \in [\underline{x}, \overline{x}] \subseteq [\underline{y}, \overline{y}]$,

$$\underline{N}_{[\underline{y},\overline{y}]}(\underline{x}, \overline{x}) \leq N(x) \leq \overline{N}_{[\underline{y},\overline{y}]}(\underline{x}, \overline{x}).$$

- $\underline{N}, \overline{N}$: neural network verification algorithms such as CROWN, IBP, LipSDP.
- CROWN [1] provides linear bounds $\underline{N}_{[\underline{y},\overline{y}]}$ and $\overline{N}_{[\underline{y},\overline{y}]}$.

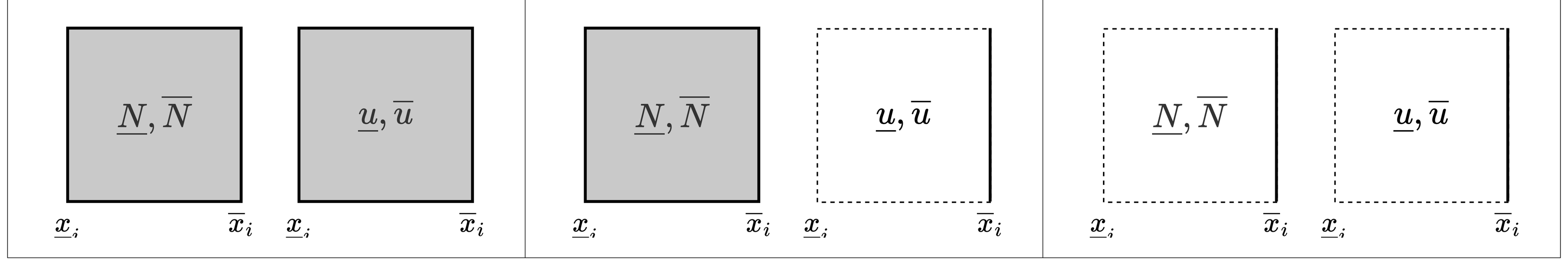


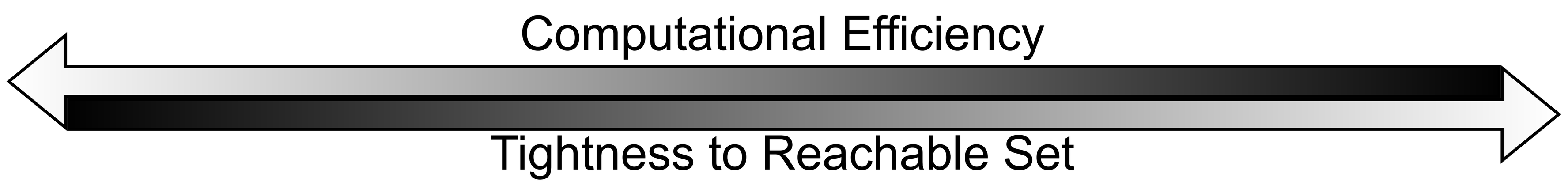
Compositional Reachability of the Closed-loop System

- open-loop system $\dot{x} = f(x, u, w)$ with inclusion function $\mathsf{F} = \left[\frac{\underline{\mathsf{F}}}{\overline{\mathsf{F}}}\right]$ for f,
- neural network controller $u = N(x)$ with inclusion function $\left[\frac{\underline{N}}{\overline{N}}\right]$
- the embedding system for the closed-loop system is

$$\dot{\underline{x}}_i = \underline{\mathsf{F}}_i(\underline{x}, \overline{x}_{i:\underline{x}}, \underline{\eta}_i, \overline{\eta}_i, \underline{w}, \overline{w}) := \underline{\mathsf{F}}^{\mathsf{S}}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w}),$$
$$\dot{\overline{x}}_i = \overline{\mathsf{F}}_i(\underline{x}_{i:\overline{x}}, \overline{x}, \underline{\nu}_i, \overline{\nu}_i, \underline{w}, \overline{w}) := \overline{\mathsf{F}}^{\mathsf{S}}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w}).$$

Global (F^{G})	Hybrid (F^{H})	Local (F^{L})
$\underline{\eta}_i = \underline{N}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x})$ $\overline{\eta}_i = \overline{N}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x})$	$\underline{\eta}_i = \underline{N}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x}_{i:\underline{x}})$ $\overline{\eta}_i = \overline{N}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x}_{i:\underline{x}})$	$\underline{\eta}_i = \underline{N}_{[\underline{x},\overline{x}_{i:\underline{x}}]}(\underline{x}, \overline{x}_{i:\underline{x}})$ $\overline{\eta}_i = \overline{N}_{[\underline{x},\overline{x}_{i:\underline{x}}]}(\underline{x}, \overline{x}_{i:\underline{x}})$
$\underline{\nu}_i = \underline{N}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x})$ $\overline{\nu}_i = \overline{N}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x})$	$\underline{\nu}_i = \underline{N}_{[\underline{x},\overline{x}]}(\underline{x}_{i:\overline{x}}, \overline{x})$ $\overline{\nu}_i = \overline{N}_{[\underline{x},\overline{x}]}(\underline{x}_{i:\overline{x}}, \overline{x})$	$\underline{\nu}_i = \underline{N}_{[\underline{x}_{i:\overline{x}},\overline{x}]}(\underline{x}_{i:\overline{x}}, \overline{x})$ $\overline{\nu}_i = \overline{N}_{[\underline{x}_{i:\overline{x}},\overline{x}]}(\underline{x}_{i:\overline{x}}, \overline{x})$





Theorem

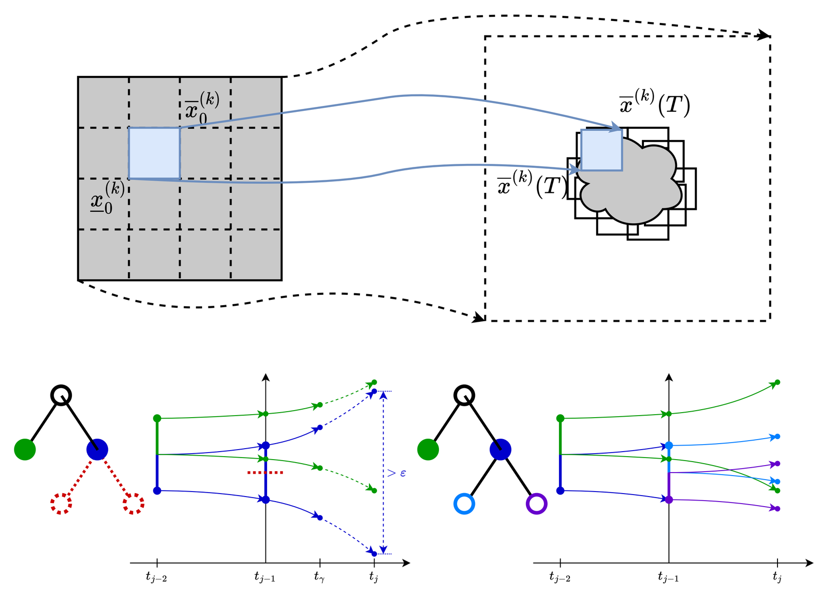
For disturbance $\mathcal{W} = [\underline{w}, \overline{w}]$ and initial set $\mathcal{X}_0 = [\underline{x}_0, \overline{x}_0]$,

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

where $(\underline{x}^{\mathsf{S}}(t), \overline{x}^{\mathsf{S}})$ is the trajectory of F^{S} for $\mathsf{S} \in \{\mathsf{G}, \mathsf{H}, \mathsf{L}\}$.

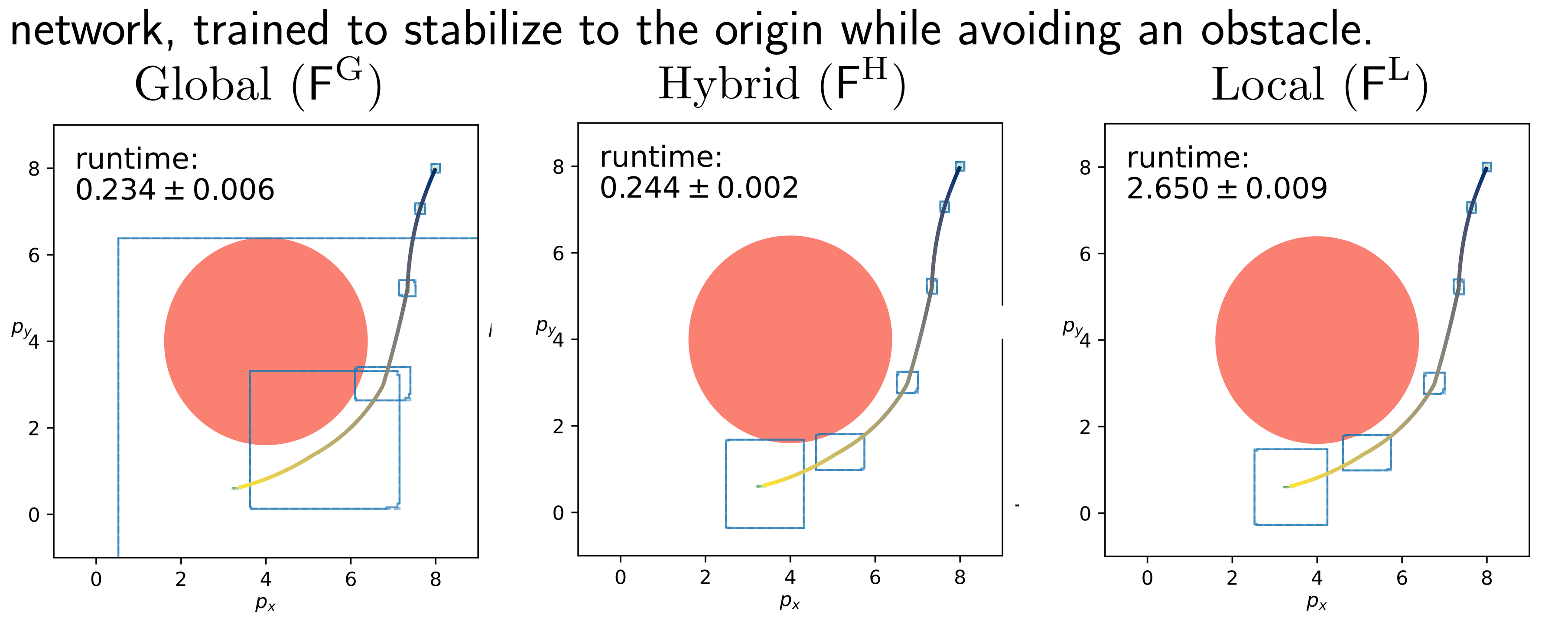
Numerical Experiments

- Partitioning improves the accuracy of interval analysis.
- separation between i) partitions that query neural network verification algorithm, and ii) partitions that only do integration.



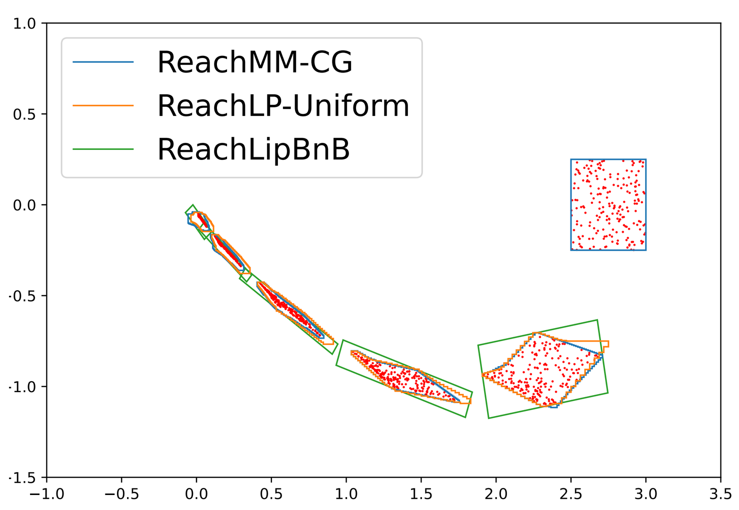
Vehicle Model:

Kinematic bicycle model, controlled by a $4 \times 100 \times 100 \times 2$ ReLU neural network, trained to stabilize to the origin while avoiding an obstacle.



Double Integrator Model:

Controlled by a $2 \times 10 \times 5 \times 1$ ReLU neural network, compare to [2,3].



Method	Runtime (s)	Area
ReachMM-CG	$\mathbf{0.079 \pm 0.001}$	$\mathbf{1.0 \cdot 10^{-1}}$
ReachLP-Unif	0.212 ± 0.002	$1.5 \cdot 10^{-1}$
ReachLP-GSG	0.913 ± 0.031	$5.3 \cdot 10^{-1}$
ReachLipBnB	0.956 ± 0.067	$5.4 \cdot 10^{-1}$

References

(1) H. Zhang et al., *Efficient neural network robustness certification with general activation function*, NeurIPS, 2018

(2) M. Everett et al., *Reachability analysis of neural feedback loops*, IEEE Access, 2021

(3) T. Entesari et al., *ReachLipBnB: A branch-and- bound method for reachability analysis of neural autonomous systems using lipschitz bounds*, arXiv, 2022