

Certifying and training reachable sets for neural network controlled systems

Sam Coogan

Associate professor

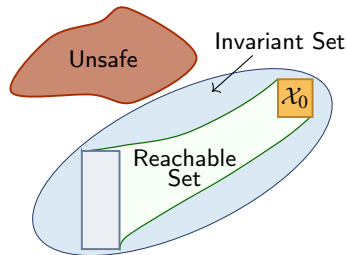
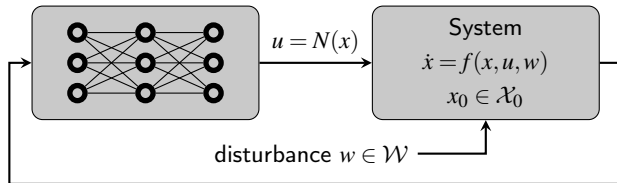
Georgia Tech



2024 Dynamical Systems and Control Theory Program Review

AFOSR FA9550-23-1-0303

Learning-based feedback controllers: Safety from reachability



- **Goal:** compute reachable sets of closed-loop system

Challenges: High-dimensional nonlinearities in system+controller, efficiency-vs-conservatism tradeoff

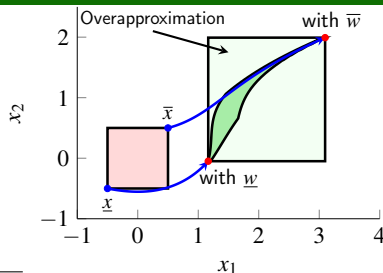
In this talk:

- Training NN controllers for forward invariant sets using monotonicity
- Efficient reachability on Lie groups from Lie algebra monotonicity

Reachability estimates for monotone systems

- The system $\dot{x} = f(x, w)$ is **monotone**¹ if
- $x_0 \preceq x'_0$ implies that $x(t) \preceq x'(t)$ for all time,
for any $w(\cdot)$ and $w'(\cdot)$ such that $w(t) \preceq w'(t)$ for all t .

Reachability analysis for monotone systems. For a monotone system,
Reachable set \subseteq [lower trajectory, upper trajectory].



¹D. Angeli and E. Sontag, "Monotone Control Systems", *IEEE TAC*, 2003

Nonmonotone systems embedded in a monotone system

- Given $\dot{x} = f(x, w)$, $x \in \mathbb{R}^n$, disturbance input $w \in [\underline{w}, \bar{w}] = \{w : \underline{w} \leq w \leq \bar{w}\} \subseteq \mathbb{R}^p$

Mixed monotone approach: Find *decomposition functions* \underline{d} , \bar{d} such that

- ① $\underline{d}(x, x, w, w) \leq f(x, w) \leq \bar{d}(x, x, w, w)$ for all x, w and
- ② the $2n$ dimensional *embedding system*²

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\bar{x}} \end{bmatrix} = \begin{bmatrix} \underline{d}(\underline{x}, \bar{x}, \underline{w}, \bar{w}) \\ \bar{d}(\underline{x}, \bar{x}, \underline{w}, \bar{w}) \end{bmatrix} =: E(\underline{x}, \bar{x}, \underline{w}, \bar{w})$$

is monotone w.r.t the *southeast order* \leq_{SE} on \mathbb{R}^{2n} defined as:

$$(x, \hat{x}) \leq_{\text{SE}} (y, \hat{y}) \text{ if and only if } x \leq y \text{ and } \hat{y} \leq \hat{x}.$$

²Enciso, Smith, Sontag, "Non-monotone systems decomposable into monotone systems with negative feedback", *Journal of Diff. Eq.*, 2006

Our approach to automated decomposition function construction

Given a map $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $(\underline{G}, \overline{G})$ is an *inclusion function* for g if

$$\underline{G}(x, \hat{x}) \leq g(z) \leq \overline{G}(x, \hat{x}) \quad \text{for every } x \leq \hat{x} \text{ and all } z \in [x, \hat{x}].$$

- ▶ Interval arithmetic allows computing inclusion functions with various techniques (natural, Taylor series-based, etc.)
- ▶ An interval inclusion of $f(x, w)$ leads to a valid decomposition
- ▶ Our toolbox `npinterval`³:
 - ▶ Implements intervals as native data-type in `numpy`
 - ▶ Uses symbolic manipulation for simplifications
 - ▶ Automates computation of decomposition functions from inclusion functions

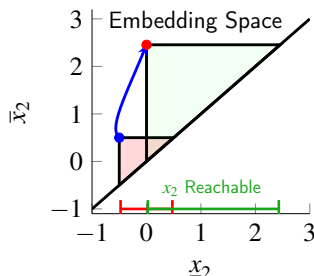
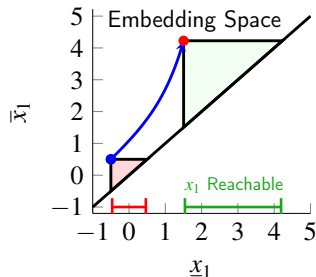
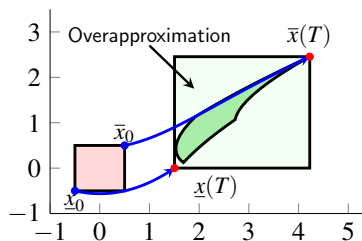
³github.com/gtfactslab/npinterval

Reachability from embedding system

Reachable set from embedding trajectory:

$$\text{Reach}(T; [\underline{x}_0, \bar{x}_0]) \subseteq [\underline{x}(T), \bar{x}(T)]$$

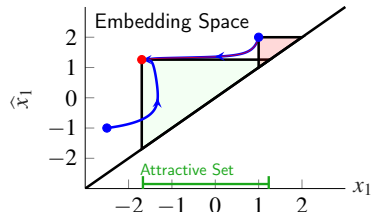
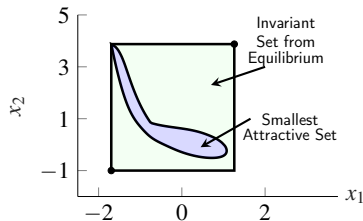
where $\underline{x}(t), \bar{x}(t)$ is embedding system solution with initial condition $(\underline{x}_0, \bar{x}_0)$



Robust invariance from stability in the embedding system

A nested family of invariant sets.⁴ If $E(\underline{x}_0, \bar{x}_0, \underline{w}, \bar{w}) \geq_{SE} 0$, then:

- 1 $[\underline{x}(t), \bar{x}(t)]$ is a robustly forward invariant set for all $t \geq 0$,
- 2 For every $t \leq \tau$, $[\underline{x}(\tau), \bar{x}(\tau)] \subseteq [\underline{x}(t), \bar{x}(t)]$,
- 3 $\lim_{t \rightarrow \infty} [\underline{x}(t), \bar{x}(t)] = [\underline{x}^*, \bar{x}^*]$ exists and is an attracting set from $[\underline{x}_0, \bar{x}_0]$.



⁴[Abate, Coogan, *IEEE TAC*, 2022]

Proposed approach for training controllers with an invariant set

Idea: for specified $\underline{x}_0, \bar{x}_0$, train a controller to satisfy $E(\underline{x}_0, \bar{x}_0, \underline{w}, \bar{w}) \geq_{SE} 0$, i.e.,

- ▶ Embedding system satisfies a sign constraint at a single point, $(\underline{x}_0, \bar{x}_0)$

Challenges:

- 1 Generalize beyond axis-aligned intervals
- 2 Need a tractable decomposition function for neural network controllers so that $E(\underline{x}_0, \bar{x}_0, \underline{w}, \bar{w}) \geq_{SE} 0$ is a trainable condition on controller parameters

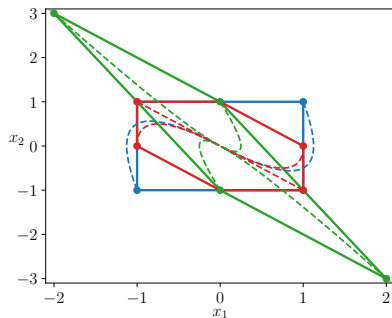
The limitation of axis-aligned intervals: An example

Double integrator:

$$\dot{x}_1 = x_2$$

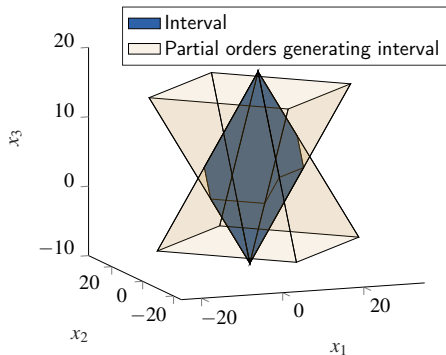
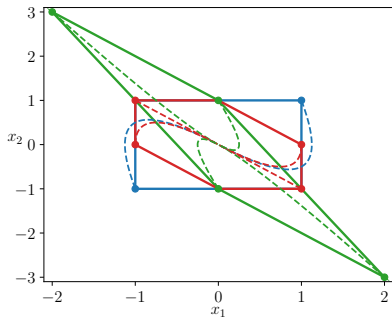
$$\dot{x}_2 = u = -2x_1 - 3x_2.$$

- ▶ No box is controlled invariant (blue)
- ▶ Could transform coordinates (green)
- ▶ Alternatively, introduce auxiliary $y = x_1 + x_2$ and consider boxes in (x_1, x_2, y) coordinates (red)



Auxiliary variables for increased flexibility

- ▶ Our goal: systematize and automate addition of auxiliary variables
- ▶ Closely related to monotonicity w.r.t. polyhedral cones⁵



⁵[Jafarpour, Coogan, *IEEE TAC*, to appear]

A lifted system from auxiliary variables

Consider

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

Let $H \in \mathbb{R}^{m \times n}$, $m \geq n$ be full rank and H^+ satisfy $H^+H = I$. With $y = Hx$,

$$\dot{y} = g(y, w) := Hf(H^+y, N(H^+y), w)$$

is the closed-loop (H, H^+) -*lifted system* of (\star) .

Lemma The linear subspace $\mathcal{H} = \{Hx \mid x \in \mathbb{R}^n\}$ is forward invariant for the lifted system.

- ▶ \mathcal{H} contains the original dynamics
- ▶ We propose embedding the lifted system instead (with one more tweak to come)

Parameterizing the lifted system

For fixed H , the collection of (H, H^+) -lifted systems is parameterized by the set of left inverses H^+ :

$$\left\{ H^+ \in \mathbb{R}^{n \times m} \mid H^+ = H^\dagger + \eta N^T, \eta \in \mathbb{R}^{n \times (m-n)} \right\}$$

where

- ▶ Columns of $N \in \mathbb{R}^{m \times (m-n)}$ are basis for left null space of H
- ▶ $H^\dagger = (H^T H)^{-1} H^T$ is Moore-Penrose Pseudoinverse

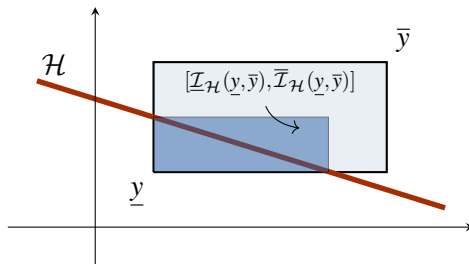
Matrix η parameterizes all lifted systems, and all lifted systems agree on the invariant subspace \mathcal{H} containing the original dynamics.

Interval refinement operators

Let $\mathcal{H} \subseteq \mathbb{R}^m$ be a subset. $\mathcal{I}_{\mathcal{H}} = (\underline{\mathcal{I}}_{\mathcal{H}}, \overline{\mathcal{I}}_{\mathcal{H}})$ is an *interval refinement operator* on \mathcal{H} if for every $[\underline{y}, \bar{y}] \subset \mathbb{R}^m$,

$$\mathcal{H} \cap [\underline{y}, \bar{y}] \subseteq [\underline{\mathcal{I}}_{\mathcal{H}}(\underline{y}, \bar{y}), \overline{\mathcal{I}}_{\mathcal{H}}(\underline{y}, \bar{y})] \subseteq [\underline{y}, \bar{y}]$$

- For affine spaces \mathcal{H} , optimal interval refinement is fast and algorithmic⁶



⁶Shen and Scott, *Computers & Chemical Engineering*, 2017.

A refined embedding system from a lifted system

The *refined embedding system* from a (H, H^+) -lifted system is

$$\begin{bmatrix} \underline{\dot{y}} \\ \underline{\dot{\bar{y}}} \end{bmatrix} = \begin{bmatrix} \underline{d}_{H, H^+}(\underline{y}, \bar{y}, \underline{w}, \bar{w}) \\ \bar{d}_{H, H^+}(\underline{y}, \bar{y}, \underline{w}, \bar{w}) \end{bmatrix} =: E_{H, H^+}(\underline{y}, \bar{y}, \underline{w}, \bar{w})$$

where \underline{d}_{H, H^+} and \bar{d}_{H, H^+} are constructed from composing an interval inclusion function of lifted dynamics and an interval refinement operator for $\mathcal{H} = \{Hx \mid x \in \mathbb{R}^n\}$.

- ▶ Unlike a full (nonrefined) embedding of the lifted system, only \mathcal{H} -restricted (i.e., original) dynamics are embedded
- ▶ Still monotone w.r.t. SE order

Theorem.⁷

Let $E_{H,H^+}(\underline{y}, \bar{y}, \underline{w}, \bar{w})$ be a refined embedding system for $\dot{x} = f(x, N(x), w)$, $w \in [\underline{w}, \bar{w}]$.

If

$$E_{H,H^+}(\underline{y}_0, \bar{y}_0, \underline{w}, \bar{w}) \geq_{SE} 0,$$

then the polytope

$$\{x \in \mathbb{R}^n \mid \underline{y}_0 \leq Hx \leq \bar{y}_0\}$$

is robustly forward invariant for the original system.

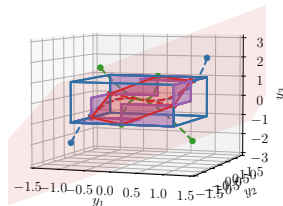
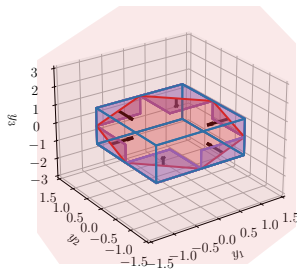
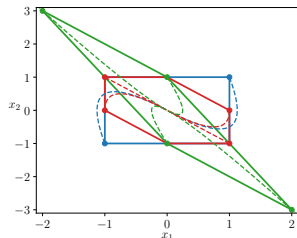
⁷[Harapanahalli, Coogan, in submission, arXiv:2408.01273.]

Example

Returning to $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -2x_1 - 3x_2 \end{bmatrix}$, take $H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$, $\underline{y} = -\mathbf{1}$, $\bar{y} = \mathbf{1}$. We have

$$E_{H,H^\dagger}(\underline{y}, \bar{y}) = (0, 1, 4/3, 0, -1, -4/3) \geq_{\text{SE}} 0,$$

and therefore the polytope $\{x \mid -\mathbf{1} \leq Hx \leq \mathbf{1}\}$ is forward invariant.



One last ingredient: Decomposition functions for NN-controlled systems

- ▶ Computation of refined embedding system requires inclusion function of closed-loop, neural network controlled system
- ▶ We have proposed methods for this⁸ (part of last year's talk)
- ▶ Critically, our methods retain autodifferentiation with respect to NN parameters

⁸[Harapanahalli, Jafarpour, Coogan, *IEEE TAC*, to appear]

Training for forward invariant sets

Algorithmic approach:

- ① Fix desired polytope $\{\underline{y}_0 \leq Hx \leq \bar{y}_0\}$ (could possibly allow to be trainable in the future)
 - ② Add condition $E_{H,H^+}(\underline{y}_0, \bar{y}_0, \underline{w}, \bar{w}) \geq_{SE} 0$ to any training procedure
 - ③ Train over neural network parameters and η parameterizing H^+
- ▶ Successful training *guarantees* forward invariance (beyond simply *promoting* invariance)
 - ▶ Completely automatable in modern frameworks such as JAX⁹ (autodiff, just-in-time compilation, parallelization to GPUs)

⁹Our implementation: github.com/gtfactslab/immrax

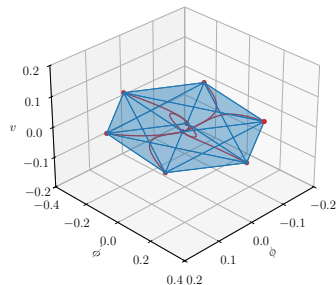
Case study 1: Segway¹⁰

$$\begin{bmatrix} \dot{\phi} \\ \dot{v} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \phi \\ \frac{\cos \phi (-1.8u + 11.5v + 9.8 \sin \phi) - 10.9u + 68.4v - 1.2\dot{\phi}^2 \sin \phi}{\cos \phi - 24.7} \\ \frac{(9.3u - 58.8v) \cos \phi + 38.6u - 234.5v - \sin \phi (208.3 + \dot{\phi}^2 \cos \phi)}{\cos^2 \phi - 24.7} \end{bmatrix}$$

- ▶ v velocity
- ▶ ϕ tilt angle
- ▶ u applied voltage

- ▶ Train to match locally optimal linear control
- ▶ Choose $H = T^{-1}$, T from diagonalizing linearized LQR-controlled system

Method	Volume	Runtime (s)		
		Setup (JIT)	Verified Training	Total
Ours	0.00152	139.	11.4	150.4
FI-ODE	0.158	—	2758	2758



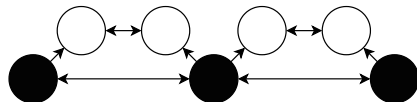
¹⁰Huang, Rodriguez, Zhang, Shi, Yue, "FI-ODE: Certifiably Robust Forward Invariance in Neural ODEs", arXiv:2210.16940v4.

Case study 2: Vehicle platoon

N vehicles, each with dynamics

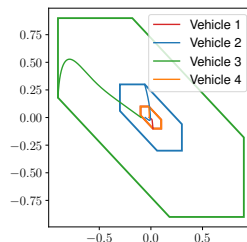
$$\dot{p}_j = v_j$$

$$\dot{v}_j = \sigma(u_j)(1 + w_j)$$



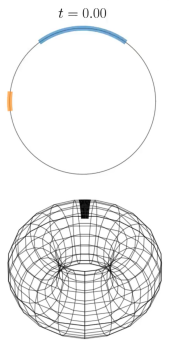
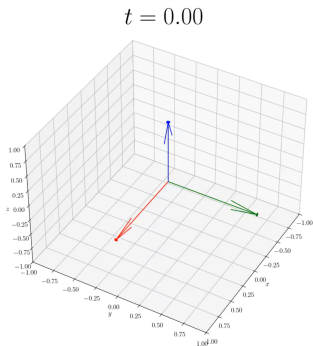
N	# States		Runtime (s)	
	Original	Lifted	Setup	Training (#iter)
4	8	12	35.8	6.90 (724)
10	20	30	64.4	57.7 (725)
16	32	48	128.	170. (807)
22	44	66	264.	408. (890)
28	56	84	478.	1040 (1267)

- ▶ All vehicles use same NN controller
- ▶ Followers (white) use relative state to two neighbors
- ▶ Leaders (black) use relative state to neighboring leaders and own state
- ▶ $H = I_N \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$



Other ongoing work: Interval reachable sets on Lie groups¹¹

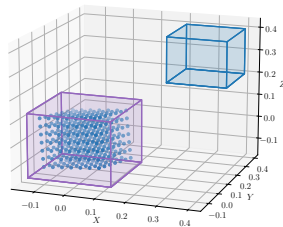
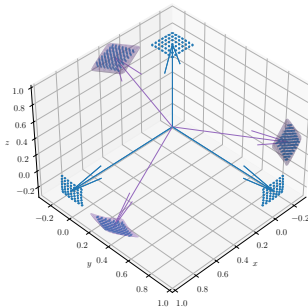
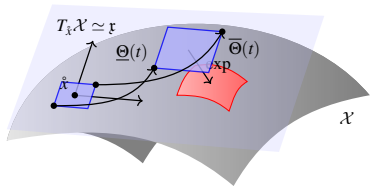
- ▶ Goal: Generalize monotonicity-based interval reachable/invariant sets to manifolds
- ▶ Idea: leverage tools from geometric numerical integration on Lie groups, which integrate in algebra then exponentiate to group



¹¹[Harapanahalli and Coogan, *IEEE CDC*, 2024]

Other ongoing work: Interval reachable sets on Lie groups¹¹

- Approach: Integrate intervals in Lie algebra, occasionally recenter intervals to new base points, implemented as Runge-Kutta scheme
- Key technical innovation: For left invariant cone fields, inclusion function of Baker-Campbell-Hausdorff formula ensures valid recentering



¹¹[Harapanahalli and Coogan, *IEEE CDC*, 2024]

Conclusions

In this talk:

- ▶ Monotone embedding system gives easy-to-check sufficient condition for forward invariance (vector field sign constraint at a single point)
- ▶ Lifting the dynamics introduces degrees of freedom that reduce conservatism
- ▶ Well-suited for high-dimensional settings, such as neural-networked controlled systems, and manifolds

Project philosophy: theory \implies elegant and useful computational tools.

`coogan.ece.gatech.edu` for papers and code

Thank you

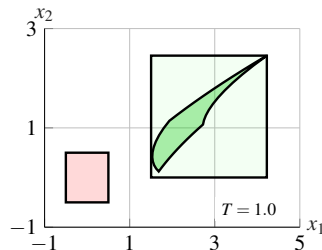
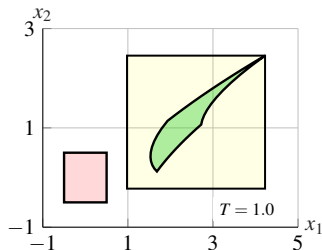
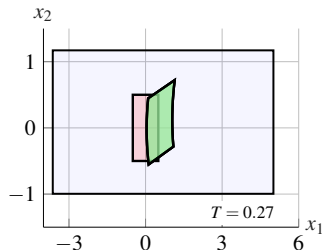
Decomposition functions from interval arithmetic and inclusion functions

For $\dot{x} = f(x, w)$, let $(\underline{F}, \overline{F})$ be an inclusion function for f . Then

$$\underline{d}_i(x, \hat{x}, w, \hat{w}) = \underline{F}_i(x, \hat{x}_{[i:x]}, w, \hat{w})$$

$$\overline{d}_i(x, \hat{x}, w, \hat{w}) = \underline{F}_i(x_{[i:\hat{x}]}, \hat{x}, w, \hat{w})$$

is a decomposition function, where $y_{[i:z]}$ is notation for the vector y with its i -th component replaced with that of z .



Computing decomposition functions for neural controlled systems

Assumption. Given a neural network $N : \mathbb{R}^n \rightarrow \mathbb{R}^p$, there exists an algorithm generating $(\underline{C}_{[\underline{x}, \bar{x}]}, \bar{C}_{[\underline{x}, \bar{x}]}, \underline{d}_{[\underline{x}, \bar{x}]}, \bar{d}_{[\underline{x}, \bar{x}]})$ where for every $x \in [\underline{x}, \bar{x}]$,

$$\underline{C}_{[\underline{x}, \bar{x}]}x + \underline{d}_{[\underline{x}, \bar{x}]} \leq N(x) \leq \bar{C}_{[\underline{x}, \bar{x}]}x + \bar{d}_{[\underline{x}, \bar{x}]}.$$

- There are several ML tools that achieve this with autodiff capabilities

Computing decomposition functions for neural controlled systems

Assumption. For the open-loop dynamics f , given centering points and intervals $\hat{x} \in [\underline{x}, \bar{x}]$, $\hat{u} \in [\underline{u}, \bar{u}]$, $\hat{w} \in [\underline{w}, \bar{w}]$, there exists interval mappings $[M_x], [M_u], [M_w]$ such that

$$\begin{aligned} f(x, u, w) \in & [M_x(\underline{x}, \bar{x}, \underline{u}, \bar{u}, \underline{w}, \bar{w})](x - \hat{x}) + [M_u(\underline{x}, \bar{x}, \underline{u}, \bar{u}, \underline{w}, \bar{w})](u - \hat{u}) \\ & + [M_w(\underline{x}, \bar{x}, \underline{u}, \bar{u}, \underline{w}, \bar{w})](w - \hat{w}), \end{aligned}$$

for any $x \in [\underline{x}, \bar{x}]$, $u \in [\underline{u}, \bar{u}]$, $w \in [\underline{w}, \bar{w}]$.

- ▶ Interval arithmetic+autodiff can be used to create such mappings
- ▶ For example, as interval inclusion of Jacobian matrices (mean value theorem)
- ▶ Alternative choices possible

Computing decomposition functions for neural controlled systems

Theorem.¹² Let

$$\begin{bmatrix} \underline{F}^N(\underline{x}, \bar{x}, \underline{w}, \bar{w}) \\ \bar{F}^N(\underline{x}, \bar{x}, \underline{w}, \bar{w}) \end{bmatrix} = \begin{bmatrix} \underline{H}^+ - \underline{M}_x & \underline{H}^- \\ \bar{H}^- & \bar{H}^+ \end{bmatrix} \begin{bmatrix} \underline{x} \\ \bar{x} \end{bmatrix} + \begin{bmatrix} -\underline{M}_w^- & \underline{M}_w^- \\ -\bar{M}_w^+ & \bar{M}_w^+ \end{bmatrix} \begin{bmatrix} \underline{w} \\ \bar{w} \end{bmatrix} + \begin{bmatrix} -\underline{M}_u \underline{u} + \underline{M}_u^+ \underline{d}_{[\underline{x}, \bar{x}]} + \underline{M}_u^- \bar{d}_{[\underline{x}, \bar{x}]} + f(\underline{x}, \underline{u}, \underline{w}) \\ -\bar{M}_u \underline{u} + \bar{M}_u^+ \bar{d}_{[\underline{x}, \bar{x}]} + \bar{M}_u^- \underline{d}_{[\underline{x}, \bar{x}]} + f(\underline{x}, \underline{u}, \underline{w}) \end{bmatrix},$$

$$\underline{H} = \underline{M}_x + \underline{M}_u^+ \underline{C}_{[\underline{x}, \bar{x}]} + \underline{M}_u^- \bar{C}_{[\underline{x}, \bar{x}]}, \quad \bar{H} = \bar{M}_x + \bar{M}_u^+ \bar{C}_{[\underline{x}, \bar{x}]} + \bar{M}_u^- \underline{C}_{[\underline{x}, \bar{x}]}.$$

Then $(\underline{F}^N, \bar{F}^N)$ is an inclusion function for $f(x, N(x), w)$, $w \in [\underline{w}, \bar{w}]$, i.e.,

$$\underline{F}^N(\underline{x}, \bar{x}, \underline{w}, \bar{w}) \leq f(x, N(x), w) \leq \bar{F}^N(\underline{x}, \bar{x}, \underline{w}, \bar{w})$$

for all $x \in [\underline{x}, \bar{x}]$ and $w \in [\underline{w}, \bar{w}]$. Moreover, a decomposition pair (\underline{d}, \bar{d}) is directly obtained from this inclusion function.

¹²[Harapanahalli, Jafarpour, Coogan, *IEEE TAC*, to appear.]