Forward Invariance in Neural Network Controlled Systems

Akash Harapanahalli, *Student Member, IEEE*, Saber Jafarpour, *Member, IEEE*, and Samuel Coogan, *Senior Member, IEEE*

***Abstract*— We present a framework based on interval analysis and monotone systems theory to certify and search for forward invariant sets in nonlinear systems with neural network controllers. The framework (i) constructs localized first-order inclusion functions for the closed-loop system using Jacobian bounds and existing neural network verification tools; (ii) builds a dynamical embedding system where its evaluation along a single trajectory directly corresponds with a nested family of hyper-rectangles provably converging to an attractive set of the original system; (iii) utilizes linear transformations to build families of nested paralleletopes with the same properties. The framework is automated in Python using our interval analysis toolbox `npinterval`, in conjunction with the symbolic arithmetic toolbox `sympy`, demonstrated on an 8-dimensional leader-follower system.**

***Index Terms*— Neural networks, forward invariance**

I. Introduction

LEARNING enabled components are becoming increasingly prevalent in modern control systems. Their ease of computation and ability to outperform optimization-based approaches make them valuable for in-the-loop usage [1]. However, neural networks are known to be vulnerable to input perturbations—small changes in their input can yield wildly varying results. In safety-critical applications, under uncertainty in the system, it is paramount to verify safe system behavior for an infinite time horizon. Such behaviors are guaranteed through *robust forward invariant sets*, *i.e.*, a set for which a system will never leave under any uncertainty.

Forward invariant sets are useful for a variety of tasks. In monitoring, a safety specification extends to infinite time if the system is guaranteed to enter an invariant set. Additionally, for asymptotic behavior of a system, an invariant set can be used as a robustness margin to replace the traditional equilibrium viewpoint in the presence of disturbances. In designing controllers, one can induce infinite-time safe behavior by ensuring the existence of an invariant set containing all initial states of the system and excluding any unsafe regions. There are several classical techniques in the literature for certifying forward invariant sets, such as Lyapunov-based analysis [2], barrier-based methods [3], and set-based approaches [4]. However, a naïve application of these methods generally fails when confronted with high-dimensional and highly nonlinear neural network controllers in-the-loop.

Literature review: The problem of verifying the input-output behavior of standalone neural networks has been studied extensively [5]. There is a growing body of literature studying verification of neural networks applied in feedback loops, which presents unique challenges due to the accumulation of error in closed-loop, *i.e.*, the wrapping effect. For example, there are functional approaches such as POLAR [6], JuliaReach [7], and ReachMM [8], [9] for nonlinear systems, and linear (resp. semi-definite) programming ReachLP [10] (resp. Reach-SDP [11]) for linear systems. While these methods verify finite time safety, their guarantees do not readily extend to infinite time. In particular, it is not clear how to adapt these tools to search and certify forward invariant sets of neural network controlled systems. There are a handful of papers that directly study forward invariance for neural networks in dynamics: In [12] a set-based approach is used to study forward invariance of a specific class of control-affine systems with feedforward neural network controllers. In [13] an ellipsoidal inner-approximation of a region of attraction of neural network controlled system is obtained using Integral Quadratic Constraints (IQCs). In [14], a Lyapunov-based approach is used to study robust invariance of control systems modeled by neural networks. In [15], an adaptive template polytopic approach using MILP verifies RL-based controllers.

Contributions: In this letter, we propose a dynamical system approach for systematically finding nested families of robust forward invariant sets for nonlinear systems controlled by neural networks. Our method uses *localized* first-order inclusion functions to construct an embedding system, which evaluates the inclusion function separately on the edges of a hyper-rectangle. Our first result is Proposition 1, which certifies (and fully characterizes for some systems) the forward invariance of a hyper-rectangle through the embedding system's evaluation at a single point. Our main result is Theorem 1, which describes how a single trajectory of the dynamical embedding system can be used to construct a nested family of invariant and attracting hyper-rectangles. However, in many applications, hyper-rectangles are not suitable for capturing forward invariant regions, and a simple linear transformation can greatly improve results. In Proposition 2, we carefully

Akash Harapanahalli and Samuel Coogan are with the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, USA. {aharapan,sam.coogan}@gatech.edu.

Saber Jafarpour is with the Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder. saber.jafarpour@colorado.edu.

This work was supported in part by the National Science Foundation under grants 1749357 and 2219755 and the Air Force Office of Scientific Research under Grant FA9550-23-1-0303.

construct an accurate localized inclusion function for any linear transformation on the original system, which we use in Theorem 2 to find a nested family of forward invariant paralleletopes. Finally, we implement the framework in Python, demonstrating its applicability to an 8-dimensional leader-follower system.[1] In previous work [8], [9], we consider the online problem of efficiently overapproximating the reachable set of nonlinear learning-enabled systems. In this letter, we consider the offline problem of searching for invariant sets, which greatly benefit from the novel use of localization and state transformations.

Notation: Define the partial order $\leq$ on $\mathbb{R}^n$ as $x \leq y$ if and only if $x_i \leq y_i$ for every $i = 1, \dots, n$. For two vectors $\underline{x}, \overline{x} \in \mathbb{R}^n$ such that $\underline{x} \leq \overline{x}$, denote the (closed) interval $[\underline{x}, \overline{x}] = \{x : \underline{x} \leq x \leq \overline{x}\}$. The set of intervals of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$. For $[\underline{a}, \overline{a}], [\underline{b}, \overline{b}] \in \mathbb{IR}$ and $[\underline{A}, \overline{A}] \in \mathbb{IR}^{m \times p}, [\underline{B}, \overline{B}] \in \mathbb{IR}^{p \times n}$,

1) $[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] := [\underline{a} + \underline{b}, \overline{a} + \overline{b}]$ (also on $\mathbb{IR}^n$ element-wise);
2) $[\underline{a}, \overline{a}] \cdot [\underline{b}, \overline{b}] := [\min\{\underline{ab}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{ab}\}, \max\{\underline{ab}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{ab}\}]$;
3) $([\underline{A}, \overline{A}][\underline{B}, \overline{B}])_{i,j} := \sum_{k=1}^{p} [\underline{A}_{i,k}, \overline{A}_{i,k}] \cdot [\underline{B}_{k,j}, \overline{B}_{k,j}]$.

For two vectors $x, y \in \mathbb{R}^n$ and $i \in \{1, \dots, n\}$, let $x_{i:y} \in \mathbb{R}^n$ be the vector obtained by replacing the ith entry of x with that of y, *i.e.*, $(x_{i:y})_j = y_j$ if $i = j$ and otherwise $(x_{i:y})_j = x_j$.

The partial order $\leq$ on $\mathbb{R}^n$ induces the *southeast partial order* $\leq_{\text{SE}}$ on $\mathbb{R}^{2n}$ as $\left(\begin{smallmatrix} x \\ \widehat{x} \end{smallmatrix}\right) \leq_{\text{SE}} \left(\begin{smallmatrix} y \\ \widehat{y} \end{smallmatrix}\right)$ if and only if $x \leq y$ and $\widehat{y} \leq \widehat{x}$. Let $\mathcal{T}_{\geq 0}^{2n} = \{\left(\begin{smallmatrix} x \\ \widehat{x} \end{smallmatrix}\right) \in \mathbb{R}^{2n} : x \leq \widehat{x}\}$. Note that $\mathcal{T}_{\geq 0}^{2n} \simeq \mathbb{IR}^n$, and define $[\left(\begin{smallmatrix} \underline{x} \\ \overline{x} \end{smallmatrix}\right)] := [\underline{x}, \overline{x}]$. Given a mapping g and a set $\mathcal{X} \subseteq \text{dom}(g)$, define the set $g(\mathcal{X}) := \{g(x) : x \in \mathcal{X}\}$.

For the nonlinear system $\dot{x} = f(x, w)$ with initial condition $x_0 \in \mathcal{X}_0$ and disturbance $w \in \mathcal{W}$ for all time, define the *reachable set* as $\mathcal{R}_f(t, \mathcal{X}_0, \mathcal{W}) = \left\{ \begin{array}{l} \phi_f(t, x_0, \mathbf{w}), \ \forall x_0 \in \mathcal{X}, \\ \mathbf{w} : [0, \infty) \to \mathcal{W} \text{ PW cont.} \end{array} \right\}$ where $t \mapsto \phi_f(t, x_0, \mathbf{w})$ is the flow of the system from initial condition x_0 at time 0 under disturbance mapping $\mathbf{w}$. A set $\mathcal{X} \subseteq \mathbb{R}^n$ is $\mathcal{W}$-robustly forward invariant if for every $t \in \mathbb{R}_{\geq 0}$, we have $\mathcal{R}_f(t, \mathcal{X}, \mathcal{W}) \subseteq \mathcal{X}$. $\mathcal{X}$ is a $\mathcal{W}$-attracting set with region of attraction $\mathcal{Y} \subseteq \mathbb{R}^n$ if $\mathcal{X}$ is $\mathcal{W}$-robustly forward invariant, and for every $x_0 \in \mathcal{Y}$, every piecewise continuous $\mathbf{w} : [0, \infty) \to \mathcal{W}$, and every open neighborhood $N \supseteq \mathcal{X}$, there exists $T^* \geq 0$ such that $\phi_f(t, x_0, \mathbf{w}) \in N$, for every $t \geq T^*$.

II. Problem Statement

Consider a nonlinear dynamical system of the form

$$\dot{x} = f(x, u, w), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^p$ is the control input, $w \in \mathcal{W} \subset \mathbb{R}^q$ is a disturbance, and $f : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^n$ is a parameterized vector field. We assume that the state feedback to the system is defined by a continuously applied k-layer feed-forward neural network controller $N : \mathbb{R}^n \to \mathbb{R}^p$:

$$\begin{aligned} &\xi^{(0)} = x, \qquad u = N(x) := W^{(k)} \xi^{(k)}(x) + b^{(k)}, \\ &\xi^{(i)} = \phi^{(i-1)}(W^{(i-1)} \xi^{(i-1)} + b^{(i-1)}), \quad i \in \{1, \dots, k\} \end{aligned} \tag{2}$$

where n_i is the number of neurons in the ith layer, $W^{(i-1)} \in \mathbb{R}^{n_i \times n_{i-1}}$ is the weight matrix of the ith layer, $b^{(i-1)} \in \mathbb{R}^{n_i}$ is the bias vector of the ith layer, $\xi^{(i)}(y) \in \mathbb{R}^{n_i}$ is the ith layer hidden variable, and $\phi^{(i-1)} : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ is the ith layer diagonal activation function satisfying $0 \leq \frac{\phi_j^{(i-1)}(x) - \phi_j^{(i-1)}(y)}{x - y} \leq 1$ for every $j \in \{1, \dots, n_i\}$. A large class of activation functions including ReLU, leaky ReLU, sigmoid, and tanh satisfies this condition (after a possible re-scaling of their co-domain). In feedback with this controller, define the closed-loop neural network controlled system

$$\dot{x} = f^{\text{c}}(x, w) := f(x, N(x), w). \tag{3}$$

Our goal is to find sets $\mathcal{X}$ such that, starting inside $\mathcal{X}$, the reachable set of the closed-loop system remains inside $\mathcal{X}$ for all times $t \geq 0$ and for any disturbances in $\mathcal{W}$, *i.e.*, $\mathcal{W}$-robustly forward invariant sets of the closed-loop system (3).

III. Inclusion Functions for Neural Network Controlled Systems

A. Inclusion Functions

Interval analysis aims to provide interval bounds on the output of a function given an interval of possible inputs [16]. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, the function $\mathsf{F} = \left(\begin{smallmatrix} \underline{\mathsf{F}} \\ \overline{\mathsf{F}} \end{smallmatrix}\right) : \mathcal{T}_{\geq 0}^{2n} \to \mathcal{T}_{\geq 0}^{2m}$ is called an *inclusion function* for f if

$$\underline{\mathsf{F}}(\underline{x}, \overline{x}) \leq f(x) \leq \overline{\mathsf{F}}(\underline{x}, \overline{x}), \quad \text{for every } x \in [\underline{x}, \overline{x}], \tag{4}$$

for every interval $[\underline{x}, \overline{x}] \subset \mathbb{R}^n$, and is an *$\mathcal{S}$-localized inclusion function* if the bounds (4) are valid for every interval $[\underline{x}, \overline{x}] \subseteq \mathcal{S}$, in which case we instead write $\mathsf{F}_{\mathcal{S}}$. Additionally, an inclusion function is

1) *monotone* if $\mathsf{F}(\underline{x}, \overline{x}) \geq_{\text{SE}} \mathsf{F}(\underline{y}, \overline{y})$, for any $[\underline{x}, \overline{x}] \subseteq [\underline{y}, \overline{y}]$;
2) *thin* if for any x, we have $\underline{\mathsf{F}}(x, x) = f(x) = \overline{\mathsf{F}}(x, x)$;
3) *minimal* if F returns the tightest possible interval, *i.e.* for each $i \in \{1, \dots, m\}$,

$$\underline{\mathsf{F}}_i^{\min}(\underline{x}, \overline{x}) = \inf_{x \in [\underline{x}, \overline{x}]} f_i(x), \quad \overline{\mathsf{F}}_i^{\min}(\underline{x}, \overline{x}) = \sup_{x \in [\underline{x}, \overline{x}]} f_i(x).$$

Given the one-to-one correspondence $\mathcal{T}_{\geq 0}^{2n} \simeq \mathbb{IR}^n$, an inclusion function is often interpreted as a mapping from intervals to intervals—given an inclusion function $\mathsf{F} = \left(\begin{smallmatrix} \underline{\mathsf{F}} \\ \overline{\mathsf{F}} \end{smallmatrix}\right) : \mathcal{T}_{\geq 0}^{2n} \to \mathcal{T}_{\geq 0}^{2m}$, we use the notation $[\mathsf{F}] = [\underline{\mathsf{F}}, \overline{\mathsf{F}}] : \mathbb{IR}^n \to \mathbb{IR}^m$ to denote the equivalent interval-valued function with interval argument.

In this paper, we focus on two main methods to construct inclusion functions: (i) Given a composite function $f = f_1 \circ f_2 \circ \cdots \circ f_N$, and inclusion functions F_i for f_i for every $i \in \{1, \dots, N\}$, the *natural inclusion function*

$$\mathsf{F}^{\text{nat}}(\underline{x}, \overline{x}) := (\mathsf{F}_1 \circ \mathsf{F}_2 \cdots \circ \mathsf{F}_N)(\underline{x}, \overline{x}) \tag{5}$$

provides a simple but possibly conservative method to build inclusion functions using the inclusion functions of simpler mappings; (ii) Given a differentiable function f, an inclusion function J_x for its Jacobian, and a centering point $\mathring{x} \in [\underline{x}, \overline{x}]$, the *Jacobian-based inclusion function*

$$[\mathsf{F}^{\text{jac}}(\underline{x}, \overline{x})] := [\mathsf{J}_x(\underline{x}, \overline{x})]([\underline{x}, \overline{x}] - \mathring{x}) + f(\mathring{x}), \tag{6}$$

can provide better estimates by bounding the first order Taylor expansion of f around $\mathring{x}$. Both of these inclusion functions are

[1] All code for the numerical experiments can be found at https://github.com/gtfactslab/Harapanahalli_LCSS2024

monotone (resp. thin) assuming the inclusion functions used to build them are also monotone (resp. thin).

In previous work [17], we introduce the open source package `npinterval` which automates natural inclusion functions in `numpy`. When used with a symbolic toolbox like `sympy`, one can construct Jacobian-based inclusion functions.

B. Localized Closed-Loop Inclusion Functions

One of the biggest challenges in neural network controlled system verification is correctly capturing the interactions between the system and the controller. For invariance analysis, it is paramount to capture the stabilizing nature of the controller, which can easily be lost with naïve overbounding of the input-output interactions between the system and controller. We make the following assumption throughout.

Assumption 1 (Local affine bounds of neural network)**.** For the neural network (2), there exists an algorithm that, for any interval $[\underline{\xi}, \overline{\xi}]$, produces a local affine bound $(C_{[\underline{\xi},\overline{\xi}]}, \underline{d}_{[\underline{\xi},\overline{\xi}]}, \overline{d}_{[\underline{\xi},\overline{\xi}]})$ such that for every $x \in [\underline{\xi}, \overline{\xi}]$,

$$C_{[\underline{\xi},\overline{\xi}]}x + \underline{d}_{[\underline{\xi},\overline{\xi}]} \leq N(x) \leq C_{[\underline{\xi},\overline{\xi}]}x + \overline{d}_{[\underline{\xi},\overline{\xi}]}.$$

Many off-the-shelf neural network verification frameworks can produce the linear estimates required in Assumption 1, and in particular, we focus on CROWN [18]. For ReLU and otherwise piecewise linear networks, one can setup a mixed integer linear program similar to [19], which is tractable for small-sized networks. Frameworks like `auto_LiRPA` [20] operate on general computational graphs, and thus satisfy this assumption for a wide variety neural network architectures, *e.g.*, residual neural networks, recurrent neural networks, and convolutional neural networks.

The bounds from Assumption 1 can be used to construct a $[\underline{\xi}, \overline{\xi}]$-localized inclusion function for $N(x)$:

$$\begin{aligned}\underline{\mathsf{N}}_{[\underline{\xi},\overline{\xi}]}(\underline{x}, \overline{x}) &= C^+_{[\underline{\xi},\overline{\xi}]}\underline{x} + C^-_{[\underline{\xi},\overline{\xi}]}\overline{x} + \underline{d}_{[\underline{\xi},\overline{\xi}]},\\ \overline{\mathsf{N}}_{[\underline{\xi},\overline{\xi}]}(\underline{x}, \overline{x}) &= C^-_{[\underline{\xi},\overline{\xi}]}\underline{x} + C^+_{[\underline{\xi},\overline{\xi}]}\overline{x} + \overline{d}_{[\underline{\xi},\overline{\xi}]},\end{aligned} \tag{7}$$

where $(C^+)_{i,j} = \max(C_{i,j}, 0)$, and $C^- = C - C^+$.

Using the localized first-order bounds of the neural network, we propose a general framework for constructing closed-loop inclusion functions for neural network-controlled systems that capture the first-order stabilizing effects of the controller. First, assuming f is differentiable, with inclusion functions $\mathsf{J}_x, \mathsf{J}_u, \mathsf{J}_w$ for the Jacobians $D_x f, D_u f, D_w f$, one can construct a closed-loop Jacobian-based inclusion function F^{c}. Given an interval $[\underline{z}, \overline{z}]$, with $\mathsf{J}_x, \mathsf{J}_u, \mathsf{J}_w$ evaluated on the input $(\underline{z}, \overline{z}, \underline{\mathsf{N}}_{[\underline{z},\overline{z}]}(\underline{z}, \overline{z}), \overline{\mathsf{N}}_{[\underline{z},\overline{z}]}(\underline{z}, \overline{z}), \underline{w}, \overline{w})$, define

$$\begin{aligned}[\mathsf{F}^{\mathsf{c}}_{[\underline{z},\overline{z}]}(\underline{x}, \overline{x}, \underline{w}, \overline{w})] &= ([\mathsf{J}_x] + [\mathsf{J}_u]C_{[\underline{x},\overline{x}]})[\underline{x}, \overline{x}]\\ &\quad + [\mathsf{J}_u][\underline{d}_{[\underline{x},\overline{x}]}, \overline{d}_{[\underline{x},\overline{x}]}] + [\mathsf{R}_{[\underline{z},\overline{z}]}(\underline{w}, \overline{w})],\end{aligned} \tag{8}$$

where $\mathring{x} \in [\underline{x}, \overline{x}] \subseteq [\underline{z}, \overline{z}]$, $\mathring{u} = N(\mathring{x})$, $\mathring{w} \in [\underline{w}, \overline{w}]$, and $[\mathsf{R}_{[\underline{z},\overline{z}]}(\underline{w}, \overline{w})] := -[\mathsf{J}_x]\mathring{x} - [\mathsf{J}_u]\mathring{u} + [\mathsf{J}_w]([\underline{w}, \overline{w}] - \mathring{w}) + f(\mathring{x}, \mathring{u}, \mathring{w})$. Proposition 2 provides a more general result proving (8) is a $[\underline{z}, \overline{z}]$-localized inclusion function for f^{c} $(T = I)$.

In the case that f is not differentiable, or finding an inclusion function for its Jacobian is difficult, as long as a (monotone) inclusion function F for the open-loop system f is known, a (monotone) closed-loop inclusion function for f^{c} from (3) can be constructed using the natural inclusion approach in (5) with N from (7)

$$\mathsf{F}^{\mathsf{c}}(\underline{x}, \overline{x}, \underline{w}, \overline{w}) = \mathsf{F}(\underline{x}, \overline{x}, \underline{\mathsf{N}}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x}), \overline{\mathsf{N}}_{[\underline{x},\overline{x}]}(\underline{x}, \overline{x}), \underline{w}, \overline{w}). \tag{9}$$

Remark 1 (Interval observer)**.** In the case that the system state is not perfectly known but rather the output of an interval observer, one can modify (3) to $\dot{x} = f(x, N(x + v), w)$, where $v \in [\underline{v}, \overline{v}]$ is the interval observer uncertainty. One can incorporate this into either inclusion function by bloating the localization of calls to Assumption 1. For (8), as long as $[\underline{x} + \underline{v}, \overline{x} + \overline{v}] \subseteq [\underline{z}, \overline{z}]$, replace $(C_{[\underline{x},\overline{x}]}, \underline{d}_{[\underline{x},\overline{x}]}, \overline{d}_{[\underline{x},\overline{x}]})$ with $(C_{[\underline{x}+\underline{v},\overline{x}+\overline{v}]}, \underline{d}_{[\underline{x}+\underline{v},\overline{x}+\overline{v}]}, \overline{d}_{[\underline{x}+\underline{v},\overline{x}+\overline{v}]})$. For (9), replace $\mathsf{N}_{[\underline{x},\overline{x}]}$ with $\mathsf{N}_{[\underline{x}+\underline{v},\overline{x}+\overline{v}]}$.

IV. A Dynamical Approach to Set Invariance

Using the closed-loop inclusion functions developed in Section III, we embed the uncertain dynamical system (3) into a larger certain system that enables computationally tractable approaches to verify and compute families of invariant sets. Consider the closed-loop system (3) with an $\mathcal{S}$-localized inclusion function $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}} : \mathcal{T}^{2n}_{\geq 0} \times \mathcal{T}^{2q}_{\geq 0} \to \mathcal{T}^{2n}_{\geq 0}$ for f^{c} constructed via, *e.g.*, (8) or (9), with the disturbance set $\mathcal{W} \subseteq [\underline{w}, \overline{w}]$. Then $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$ *induces* an *embedding system* for (3) with state $\left(\begin{smallmatrix}\underline{x}\\ \overline{x}\end{smallmatrix}\right) \in \mathcal{T}^{2n}_{\geq 0}$ and dynamics defined by

$$\begin{aligned}\dot{\underline{x}}_i &= \left(\underline{\mathsf{E}}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}, \overline{x}, \underline{w}, \overline{w})\right)_i := \left(\underline{\mathsf{F}}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}, \overline{x}_{i:\underline{x}}, \underline{w}, \overline{w})\right)_i,\\ \dot{\overline{x}}_i &= \left(\overline{\mathsf{E}}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}, \overline{x}, \underline{w}, \overline{w})\right)_i := \left(\overline{\mathsf{F}}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}_{i:\overline{x}}, \overline{x}, \underline{w}, \overline{w})\right)_i,\end{aligned} \tag{10}$$

where $\mathsf{E}^{\mathsf{c}}_{\mathcal{S}} : \mathcal{T}^{2n}_{\geq 0} \times \mathcal{T}^{2q}_{\geq 0} \to \mathbb{R}^{2n}$. One of the key features of the embedding system, which evolves on $\mathcal{T}^{2n}_{\geq 0}$, is that the inclusion function is evaluated separately on each face of the hyper-rectangle $[\underline{x}, \overline{x}]$, represented by $[\underline{x}, \overline{x}_{i:\underline{x}}]$ and $[\underline{x}_{i:\overline{x}}, \overline{x}]$ for each $i \in \{1, \ldots, n\}$. In Proposition 1, this meshes nicely with Nagumo's Theorem [4], which allows us to guarantee forward invariance by checking the boundary of the hyper-rectangle through one evaluation of the embedding system (10).

Proposition 1 (Forward invariance in hyper-rectangles)**.** *Consider the neural network controlled system (3) with the disturbance set $\mathcal{W} = [\underline{w}, \overline{w}]$ and initial condition $x_0 \in [\underline{x}^\star, \overline{x}^\star]$. Given a set $\mathcal{S} \supseteq [\underline{x}^\star, \overline{x}^\star]$, let $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$ be a $\mathcal{S}$-localized inclusion function for f^{c},* e.g. *(8) or (9), and $\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}$ be the embedding system induced by $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$. If*

$$\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}^\star, \overline{x}^\star, \underline{w}, \overline{w}) \geq_{\mathrm{SE}} 0, \tag{11}$$

then $[\underline{x}^\star, \overline{x}^\star]$ is a $[\underline{w}, \overline{w}]$-robustly forward invariant set. Moreover, if $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$ is the minimal inclusion function of f^{c}, the condition (11) is also necessary for $[\underline{x}^\star, \overline{x}^\star]$ to be a $[\underline{w}, \overline{w}]$-robustly forward invariant set.

Proof. For brevity, since $[\underline{x}^\star, \overline{x}^\star] \subseteq \mathcal{S}$, we drop $\mathcal{S}$ from the notation. Consider the set $[\underline{x}^\star, \overline{x}^\star]$, and suppose that $\mathsf{E}^{\mathsf{c}}(\underline{x}^\star, \overline{x}^\star, \underline{w}, \overline{w}) \geq_{\mathrm{SE}} 0$. Therefore, for every $i \in \{1, \ldots, n\}$,

$$0 \leq \underline{\mathsf{F}}^{\mathsf{c}}_i(\underline{x}^\star, \overline{x}^\star_{i:\underline{x}^\star}, \underline{w}, \overline{w}) \leq \inf_{x \in [\underline{x}^\star, \overline{x}^\star_{i:\underline{x}^\star}], w \in [\underline{w}, \overline{w}]} f^{\mathsf{c}}(x, w).$$

This implies that $f^{\mathsf{c}}(x,w) \geq 0$ for every x on the i-th lower face of the hyperrectangle $[\underline{x}^\star, \overline{x}^\star_{i:\underline{x}^\star}]$. Similarly, $f^{\mathsf{c}}(x,w) \leq 0$ on the i-th upper face of the hyperrectangle $[\underline{x}^\star_{i:\overline{x}^\star}, \overline{x}^\star]$. Since this holds for every $i \in \{1,\dots,n\}$, by Nagumo's theorem [4, Theorem 3.1], the closed set $[\underline{x}^\star, \overline{x}^\star]$ is forward invariant since for every point x along its boundary $\bigcup_i([\underline{x}^\star, \overline{x}^\star_{i:\underline{x}^\star}] \cup [\underline{x}^\star_{i:\overline{x}^\star}, \overline{x}^\star])$, the vector field $f^{\mathsf{c}}(x,w)$ points into the set, for every $w \in [\underline{w}, \overline{w}]$. Now, suppose that E is the embedding system induced by the minimal inclusion function $\mathsf{F}^{\min}$ for f^{c}, and $[\underline{x}^\star, \overline{x}^\star]$ is a hyper-rectangle such that $\mathsf{E}(\underline{x}^\star, \overline{x}^\star, \underline{w}, \overline{w}) \not\geq_{\mathrm{SE}} 0$. Then there exists $i \in \{1,\dots,n\}$ such that either

$$\underline{\mathsf{F}}^{\min}_i(\underline{x}^\star, \overline{x}^\star_{i:\underline{x}^\star}, \underline{w}, \overline{w}) = \inf_{x\in[\underline{x}^\star, \overline{x}^\star_{i:\underline{x}^\star}], w\in[\underline{w},\overline{w}]} f(x,w) < 0, \text{ or}$$

$$\overline{\mathsf{F}}^{\min}_i(\underline{x}^\star_{i:\overline{x}^\star}, \overline{x}^\star, \underline{w}, \overline{w}) = \sup_{x\in[\underline{x}^\star_{i:\overline{x}^\star}, \overline{x}^\star], w\in[\underline{w},\overline{w}]} f(x,w) > 0.$$

If the first case holds, then there exists $x' \in [\underline{x}^\star, \overline{x}^\star_{i:\underline{x}}], w \in [\underline{w}, \overline{w}]$ such that $f^{\mathsf{c}}(x',w) < 0$ along the i-th lower face of the hyper-rectangle. If the second case holds, then there exists $x' \in [\underline{x}^\star_{i:\overline{x}}, \overline{x}^\star], w \in [\underline{w}, \overline{w}]$ such that $f^{\mathsf{c}}(x',w) > 0$ along the i-th upper face of the hyper-rectangle. Thus, by Nagumo's theorem, the set $[\underline{x}^\star, \overline{x}^\star]$ is not $[\underline{w}, \overline{w}]$-robustly forward invariant, as there exists a point along its boundary such that the vector field f^{c} points outside the set. □

Remark 2 (Linear systems with piecewise linear activations)**.** For the special case of the linear system $\dot{x} = Ax + Bu + Dw$ controlled by a neural network $u = N(x)$ with piecewise linear activations, *e.g.* ReLU or Leaky ReLU, one can compute the minimal inclusion function using a Mixed Integer Linear Program (MILP) similar to [19]. For $i \in \{1,\dots,n\}$,

$$\underline{\mathsf{F}}^{\min}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w}) = \min_{x\in[\underline{x},\overline{x}], w\in[\underline{w},\overline{w}]} (Ax + BN(x) + Dw)_i,$$

$$\overline{\mathsf{F}}^{\min}_i(\underline{x}, \overline{x}, \underline{w}, \overline{w}) = \max_{x\in[\underline{x},\overline{x}], w\in[\underline{w},\overline{w}]} (Ax + BN(x) + Dw)_i.$$

The next Theorem shows how monotonicity of the embedding dynamics define a family of nested robustly forward invariant sets using the condition from Proposition 1.

Theorem 1 (A nested family of invariant sets)**.** *Consider the neural network controlled system (3) with the disturbance set $\mathcal{W} = [\underline{w}, \overline{w}]$ and initial condition $x_0 \in [\underline{x}_0, \overline{x}_0]$. Given a set $\mathcal{S} \supseteq [\underline{x}_0, \overline{x}_0]$, let $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$ be a $\mathcal{S}$-localized monotone inclusion function for f^{c},* e.g. *(8) or (9), and $\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}$ be the embedding system induced by $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$. If*

$$\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}_0, \overline{x}_0, \underline{w}, \overline{w}) \geq_{\mathrm{SE}} 0,$$

then

i) $[\underline{x}(t), \overline{x}(t)]$ is a $[\underline{w}, \overline{w}]$-robustly forward invariant set for the system (3) for every $t \geq 0$, and for every $t \leq \tau$, $[\underline{x}(\tau), \overline{x}(\tau)] \subseteq [\underline{x}(t), \overline{x}(t)]$; and

ii) $\lim_{t\to\infty} \begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix} = \begin{pmatrix} \underline{x}^\star \\ \overline{x}^\star \end{pmatrix}$, where $\begin{pmatrix} \underline{x}^\star \\ \overline{x}^\star \end{pmatrix} \in \mathcal{T}^{2n}_{\geq 0}$ is an equilibrium point of the embedding system (10) and $[\underline{x}^\star, \overline{x}^\star]$ is a $[\underline{w}, \overline{w}]$-attracting set for the system (3) with region of attraction $[\underline{x}_0, \overline{x}_0]$,

where $t \mapsto \begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix}$ is the trajectory of (10) from $\begin{pmatrix} \underline{x}_0 \\ \overline{x}_0 \end{pmatrix}$.

Proof. (Monotonicity of $\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}$ dynamics). Consider any two points $\begin{pmatrix} \underline{x} \\ \overline{x} \end{pmatrix}, \begin{pmatrix} \underline{x}' \\ \overline{x}' \end{pmatrix} \in \mathcal{T}^{2n}_{\geq 0}$, such that $\begin{pmatrix} \underline{x} \\ \overline{x} \end{pmatrix} \leq_{\mathrm{SE}} \begin{pmatrix} \underline{x}' \\ \overline{x}' \end{pmatrix}$. This implies that $\begin{pmatrix} \underline{x} \\ \overline{x}_{i:\underline{x}} \end{pmatrix} \leq_{\mathrm{SE}} \begin{pmatrix} \underline{x}' \\ \overline{x}'_{i:\underline{x}'} \end{pmatrix}$ and $\begin{pmatrix} \underline{x}_{i:\overline{x}} \\ \overline{x} \end{pmatrix} \leq_{\mathrm{SE}} \begin{pmatrix} \underline{x}'_{i:\overline{x}'} \\ \overline{x}' \end{pmatrix}$. Since $\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}$ is a monotone inclusion function,

$$\left(\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}, \overline{x}_{i:\underline{x}}, \underline{w}, \overline{w})\right)_i \leq \left(\mathsf{F}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}', \overline{x}'_{i:\underline{x}'}, \underline{w}, \overline{w})\right)_i,$$
$$\left(\overline{\mathsf{F}}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}_{i:\underline{x}}, \overline{x}, \underline{w}, \overline{w})\right)_i \geq \left(\overline{\mathsf{F}}^{\mathsf{c}}_{\mathcal{S}}(\underline{x}'_{i:\underline{x}'}, \overline{x}', \underline{w}, \overline{w})\right)_i,$$

for every $i \in \{1,\dots,n\}$. This implies that the embedding system $\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}$ is monotone w.r.t. the southeast partial order $\leq_{\mathrm{SE}}$ [21], [22]. (Part (i)). Now using [23, Proposition 2.1], the set $\mathcal{P}_+ = \{\begin{pmatrix} \underline{x} \\ \overline{x} \end{pmatrix} : \begin{pmatrix} \underline{x} \\ \overline{x} \end{pmatrix} \geq_{\mathrm{SE}} 0\}$ is a forward invariant set for $\mathsf{E}^{\mathsf{c}}_{\mathcal{S}}$. Since $\begin{pmatrix} \underline{x}_0 \\ \overline{x}_0 \end{pmatrix} \in \mathcal{P}_+$, forward invariance implies $\begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix} \in \mathcal{P}_+$ for every $t \geq 0$. Therefore, using Proposition 1, $[\underline{x}(t), \overline{x}(t)]$ is forward invariant for the closed-loop system (3) for every $t \geq 0$. Additionally, the curve $t \mapsto \begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix}$ is nondecreasing with respect to the partial order $\leq_{\mathrm{SE}}$ [23, Proposition 2.1]. This means that for every $t \leq \tau$, $\begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix} \leq_{\mathrm{SE}} \begin{pmatrix} \underline{x}(\tau) \\ \overline{x}(\tau) \end{pmatrix}$, which implies that $[\underline{x}(\tau), \overline{x}(\tau)] \subseteq [\underline{x}(t), \overline{x}(t)]$. (Part (ii)). Since $t \mapsto \begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix}$ is nondecreasing w.r.t. $\leq_{\mathrm{SE}}$, for every $i \in \{1,\dots,n\}$, the curves $t \mapsto \underline{x}_i(t)$ (resp. $t \mapsto \overline{x}_i(t)$) are nondecreasing (resp. nonincreasing) w.r.t. $\leq$ and bounded on $\mathbb{R}$. This implies that there exists $\underline{x}^*_i$ and $\overline{x}^*_i$ such that $\lim_{t\to\infty} \underline{x}_i(t) = \underline{x}^*_i$ and $\lim_{t\to\infty} \overline{x}_i(t) = \overline{x}^*_i$. By defining the vector $\underline{x}^* = (\underline{x}^*_1, \dots, \underline{x}^*_n)^\top$ and $\overline{x}^* = (\overline{x}^*_1, \dots, \overline{x}^*_n)^\top$, we get $\lim_{t\to\infty} \begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix} = \begin{pmatrix} \underline{x}^\star \\ \overline{x}^\star \end{pmatrix}$. Moreover, since $\begin{pmatrix} \underline{x}(t) \\ \overline{x}(t) \end{pmatrix} \in \mathcal{T}^{2n}_{\geq 0}$ for every $t \geq 0$ and is a continuous curve in time t, we get $\begin{pmatrix} \underline{x}^\star \\ \overline{x}^\star \end{pmatrix} \in \mathcal{T}^{2n}_{\geq 0}$. Finally, $\mathcal{R}_{f^{\mathsf{c}}}(t, [\underline{x}_0, \overline{x}_0], [\underline{w}, \overline{w}]) \subseteq [\underline{x}(t), \overline{x}(t)]$ for every $t \geq 0$ [9, Proposition 5]. Thus, every trajectory of the system starting from $[\underline{x}_0, \overline{x}_0]$ converges to $[\underline{x}^\star, \overline{x}^\star]$. □

After finding one invariant set using Proposition 1, Theorem 1 obtains a nested family of invariant sets guaranteed to converge to some $[\underline{x}^\star, \overline{x}^\star]$, obtained by evolving the embedding system forwards in time. This is the smallest invariant set in the family, and is a $[\underline{w}, \overline{w}]$-attractive set with region of attraction $[\underline{x}_0, \overline{x}_0]$. Note that the embedding system can also be evolved backwards in time while $[\underline{x}(t), \overline{x}(t)] \geq_{\mathrm{SE}} 0$ and $[\underline{x}(t), \overline{x}(t)] \subseteq \mathcal{S}$ to expand the invariant sets. Additionally, Proposition 1 and Theorem 1 do not require thin inclusion functions, generalizing existing decomposition-based approaches for finding hyper-rectangular invariant sets [24].

V. Paralleletope Invariant Sets

Theorem 1 can be used to verify and search for hyper-rectangular invariant sets using the embedding system. In this section, we extend our framework to characterize a more general class of invariant paralleletopes. For some invertible matrix $T \in \mathbb{R}^{n\times n}$, define the T-transformed system

$$\begin{aligned} y &:= Tx := \Phi(x) \\ \dot{y} &= g^{\mathsf{c}}(y,w) = Tf(T^{-1}y, N'(y), w), \end{aligned} \tag{12}$$

where $N'(y) := N(T^{-1}y)$ is the neural network from (2), with an extra initial layer with weight matrix T^{-1} and linear activation $\phi(x) = x$. There is a one-to-one correspondence

between the transformed system (12) and the original system (3), in the sense that every trajectory $t \mapsto y(t)$ of (12) uniquely corresponds with the trajectory $t \mapsto \Phi^{-1}(y(t))$ of (3). Given an interval $[\underline{y}, \overline{y}]$, the set $\Phi^{-1}([\underline{y}, \overline{y}]) = \{T^{-1}y : y \in [\underline{y}, \overline{y}]\}$ defines a paralleletope in standard coordinates. We construct a localized closed-loop Jacobian-based inclusion function $\mathsf{G}^{\mathsf{c}}_{\Phi([\underline{z},\overline{z}])}$ for g^{c} as follows. Given an interval $[\underline{z}, \overline{z}]$ in standard coordinates, with inclusion functions $\mathsf{J}_x, \mathsf{J}_u, \mathsf{J}_w$ for the Jacobians $D_x f, D_u f, D_w f$ of the original system evaluated on the input $(\underline{z}, \overline{z}, \underline{\mathsf{N}}_{[\underline{z},\overline{z}]}(\underline{z}, \overline{z}), \overline{\mathsf{N}}_{[\underline{z},\overline{z}]}(\underline{z}, \overline{z}), \underline{w}, \overline{w})$, define

$$\begin{aligned}[\mathsf{G}^{\mathsf{c}}_{\Phi([\underline{z},\overline{z}])}(\underline{y}, \overline{y}, \underline{w}, \overline{w})] &= T([\mathsf{J}_x] + [\mathsf{J}_u](C'_{[\underline{y},\overline{y}]}T))T^{-1}[\underline{y}, \overline{y}] \\ &\quad + T[\mathsf{J}_u][\underline{d}'_{[\underline{y},\overline{y}]}, \overline{d}'_{[\underline{y},\overline{y}]}] + T[\mathsf{R}_{[\underline{z},\overline{z}]}(\underline{w}, \overline{w})], \end{aligned} \tag{13}$$

where $(C', \underline{d}', \overline{d}')$ are from Assumption 1 evaluated over $[\underline{y}, \overline{y}]$ on the transformed neural network $N'(y) = N(T^{-1}y)$, $\mathring{x} \in \Phi^{-1}([\underline{y}, \overline{y}]) \subseteq [\underline{z}, \overline{z}]$, $\mathring{u} \in \mathsf{N}_{[\underline{z},\overline{z}]}(\underline{z}, \overline{z})$, $\mathring{w} \in [\underline{w}, \overline{w}]$, and $[\mathsf{R}_{[\underline{z},\overline{z}]}(\underline{w}, \overline{w})] = -[\mathsf{J}_x]\mathring{x} - [\mathsf{J}_u]\mathring{u} + [\mathsf{J}_w]([\underline{w}, \overline{w}] - \mathring{w}) + f(\mathring{x}, \mathring{u}, \mathring{w})$.

Proposition 2. *Consider the neural network controlled system (3) and let $T \in \mathbb{R}^{n\times n}$ be an invertible matrix transforming the system into (12). Then (13) is a $\Phi([\underline{z}, \overline{z}])$-localized (monotone) inclusion function for g^{c}.*

Proof. For $x \in [\underline{z}, \overline{z}]$ (mean-value, see [16, Section 2.4.3]),

$$\begin{aligned} Tf^{\mathsf{c}}(T^{-1}y, w) &\in T[\mathsf{J}_x](T^{-1}y - \mathring{x}) + T[\mathsf{J}_w](w - \mathring{w}) \\ &\quad + T[\mathsf{J}_u](N(T^{-1}y) - \mathring{u}) + Tf(\mathring{x}, \mathring{u}, \mathring{w}). \end{aligned}$$

Considering any interval $[\underline{y}, \overline{y}] \subseteq \Phi([\underline{z}, \overline{z}])$ containing y, using $(C'_{[\underline{y},\overline{y}]}, \underline{d}'_{[\underline{y},\overline{y}]}, \overline{d}'_{[\underline{y},\overline{y}]})$ from Assumption 1 on N',

$$\begin{aligned} g^{\mathsf{c}}(y, w) &\in T[\mathsf{J}_x](T^{-1}y - \mathring{x}) + T[\mathsf{J}_w](w - \mathring{w}) \\ &\quad + T[\mathsf{J}_u](C'_{[\underline{y},\overline{y}]}y + [\underline{d}'_{[\underline{y},\overline{y}]}, \overline{d}'_{[\underline{y},\overline{y}]}] - \mathring{u}) + Tf(\mathring{x}, \mathring{u}, \mathring{w}), \\ g^{\mathsf{c}}(y, w) &\in T([\mathsf{J}_x] + [\mathsf{J}_u](C'_{[\underline{y},\overline{y}]}T))T^{-1}[\underline{y}, \overline{y}] \\ &\quad + T[\mathsf{J}_u][\underline{d}'_{[\underline{y},\overline{y}]}, \overline{d}'_{[\underline{y},\overline{y}]}] + T[\mathsf{R}_{[\underline{z},\overline{z}]}(\underline{w}, \overline{w})]. \end{aligned}$$

□

It is important to note that the inclusion functions $\mathsf{J}_x, \mathsf{J}_u, \mathsf{J}_w$ in (13) are evaluated in the original coordinates. Instead, one could symbolically write g as a new system and directly apply the closed-loop Jacobian-based inclusion function from (8). In practice, however, these transformed dynamics often have complicated expressions that lead to excessive conservatism when using natural inclusion functions, and are not suitable for characterizing invariant sets. In the next Theorem, we link forward invariant hyper-rectangles in transformed coordinates to forward invariant paralleletopes in standard coordinates.

Theorem 2 (Forward invariance in paralleletopes)**.** *Consider the neural network controlled system (3) with the disturbance set $\mathcal{W} = [\underline{w}, \overline{w}]$. Let $T \in \mathbb{R}^{n\times n}$ be an invertible matrix transforming the system into (12), with initial condition $y_0 \in [\underline{y}_0, \overline{y}_0]$. Given a set $\mathcal{S} \supseteq [\underline{y}_0, \overline{y}_0]$, let $\mathsf{G}^{\mathsf{c}}_{\mathcal{S}}$ be a $\mathcal{S}$-localized inclusion function for g^{c},* e.g. *(13), and let $\mathsf{E}^{\mathsf{c}}_{T,\mathcal{S}}$ be the embedding system (10) induced by $\mathsf{G}^{\mathsf{c}}_{\mathcal{S}}$. If*

$$\mathsf{E}^{\mathsf{c}}_{T,\mathcal{S}}(\underline{y}_0, \overline{y}_0, \underline{w}, \overline{w}) \geq_{\mathrm{SE}} 0,$$

then the paralleletope $\Phi^{-1}([\underline{y}_0, \overline{y}_0])$ is a $[\underline{w}, \overline{w}]$-robustly forward invariant set for the neural network controlled system (3).

Proof. Consider any trajectory $t \mapsto x(t)$ of the original system (3) starting from $x_0 \in \Phi^{-1}([\underline{y}_0, \overline{y}_0])$. Given the one-to-one correspondence between (3) and (12), the curve $t \mapsto \Phi(x(t))$ is the trajectory of the transformed system (12) starting from $y_0 = \Phi(x_0)$. Using Proposition 1, $\mathsf{E}^{\mathsf{c}}_{T,\mathcal{S}}(\underline{y}_0, \overline{y}_0, \underline{w}, \overline{w}) \geq_{\mathrm{SE}} 0$ implies that the hyper-rectangle $[\underline{y}_0, \overline{y}_0]$ is $[\underline{w}, \overline{w}]$-robustly forward invariant in the transformed system (12). This implies that $y(t) \in [\underline{y}_0, \overline{y}_0]$ for every $t \geq 0$. Since $x(t) = \Phi^{-1}(y(t))$, it follows that $x(t) \in \Phi^{-1}([\underline{y}_0, \overline{y}_0])$ for every $t \geq 0$. □

When using (13) as $\mathsf{G}^{\mathsf{c}}_{\mathcal{S}}$, the neural network verification step from Assumption 1 to find $(C', \underline{d}', \overline{d}')$ is evaluated separately on each face of the hyperrectangle $[\underline{y}_0, \overline{y}_0]$, *i.e.*, each face of the paralleletope $\Phi^{-1}([\underline{y}_0, \overline{y}_0])$. The dynamical approach from Theorem 1 yields a nested family of invariant hyperrectangles for the transformed system (12), corresponding to a nested family of invariant paralleletopes for the original system (3). There are principled ways of choosing T, *e.g.*, the Jordan decomposition of the linearization about an equilibrium.

VI. Numerical Experiments

Consider two vehicles L and F each with dynamics

$$\begin{aligned} \dot{p}^j_x &= v^j_x, & \dot{v}^j_x &= \sigma(u^j_x) + w^j_x, \\ \dot{p}^j_y &= v^j_y, & \dot{v}^j_y &= \sigma(u^j_y) + w^j_y, \end{aligned} \tag{14}$$

for $j \in \{\mathrm{L}, \mathrm{F}\}$, where $p^j = (p^j_x, p^j_y) \in \mathbb{R}^2$ is the displacement of the center of mass of j in the plane, $v^j = (v^j_x, v^j_y) \in \mathbb{R}^2$ is the velocity of the center of mass of j, $(u^j_x, u^j_y) \in \mathbb{R}^2$ are desired acceleration inputs limited by the nonlinear softmax operator $\sigma(u) = u_{\mathrm{lim}} \tanh(u/u_{\mathrm{lim}})$ with $u_{\mathrm{lim}} = 20$, and $w^j_x, w^j_y \in [-0.005, 0.005]$ are bounded disturbances on j. Denote the combined state of the system $x := (p^{\mathrm{L}}, v^{\mathrm{L}}, p^{\mathrm{F}}, v^{\mathrm{F}}) \in \mathbb{R}^8$. We consider a leader-follower structure for the system, where the leader vehicle L chooses its control $u = (u^{\mathrm{L}}_x, u^{\mathrm{L}}_y)$ as the output of a state feedback continuously applied neural network controller ($4 \times 100 \times 100 \times 2$, ReLU activations), with input $(p^{\mathrm{L}}_x, p^{\mathrm{L}}_y, v^{\mathrm{L}}_x, v^{\mathrm{L}}_y)$. The neural network was trained using imitation learning on 5.7M data points from an offline MPC control policy for the leader only, with control limits implemented as hard constraints rather than σ. The offline policy minimized a quadratic cost aiming to stabilize to the origin while avoiding four circular obstacles centered at $(\pm 4, \pm 4)$ with radius 2.25 each, implemented as hard constraints with 33% padding and a slack variable. The follower vehicle F follows the leader with a PD controller

$$u^{\mathrm{F}}_{\mathbf{d}} = k_p(p^{\mathrm{L}}_{\mathbf{d}} - p^{\mathrm{F}}_{\mathbf{d}}) + k_v(v^{\mathrm{L}}_{\mathbf{d}} - v^{\mathrm{F}}_{\mathbf{d}}), \tag{15}$$

for each $\mathbf{d} \in \{x, y\}$ with $k_p = 6$ and $k_v = 7$.

First, a trajectory of the undisturbed system is run until it reaches the equilibrium point $x^\star \approx [0.01, 0, 0, 0, 0.01, 0, 0, 0]^T$. Then, using `sympy`, the Jacobian matrices $D_{\mathbf{s}} f(x^\star, N(x^\star), 0)$ for $\mathbf{s} \in \{x, u, w\}$ are computed, along with CROWN (`same-slope`) using `auto_LiRPA` [20] along the interval $[\underline{z}, \overline{z}] := x^\star + [-0.06, 0.06]^4 \times [-0.25, 0.25]^2 \times [-0.325, 0.325]^2$ yielding $(C_{[\underline{z},\overline{z}]}, \underline{d}_{[\underline{z},\overline{z}]}, \overline{d}_{[\underline{z},\overline{z}]})$. Then, the Jordan decomposition $T^{-1}JT = (D_x f(x^\star, N(x^\star), 0) + D_u f(x^\star, N(x^\star), 0) C_{[\underline{z},\overline{z}]})$

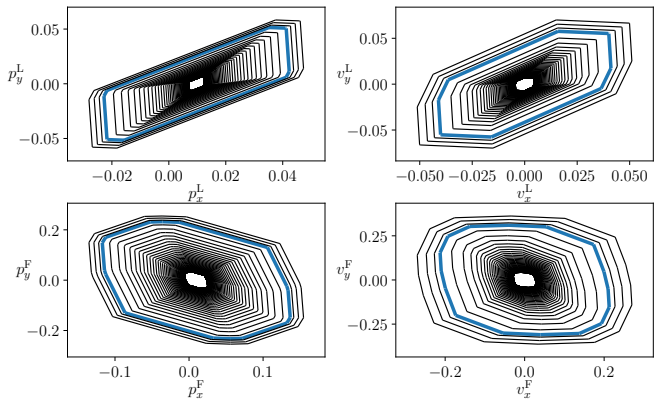

Fig. 1. A family of 93 paralleletope invariant sets for the leader-follower system (14) are visualized using projections from $\mathbb{R}^8$ onto 4 $\mathbb{R}^2$ planes. **Top Left:** projection onto leader's position $\boldsymbol{p_x^{\mathrm{L}}}, \boldsymbol{p_y^{\mathrm{L}}}$; **Top Right:** projection onto leader's velocity $\boldsymbol{v_x^{\mathrm{L}}}, \boldsymbol{v_y^{\mathrm{L}}}$; **Bottom Left:** projection onto follower's position $\boldsymbol{p_x^{\mathrm{F}}}, \boldsymbol{p_y^{\mathrm{F}}}$; **Bottom Right:** projection onto follower's velocity $\boldsymbol{v_x^{\mathrm{F}}}, \boldsymbol{v_y^{\mathrm{F}}}$. After one invariant set $\boldsymbol{\Phi^{-1}([\underline{y}_0, \overline{y}_0])}$ is found (blue line), the transformed embedding system is integrated forwards until approximate convergence, and backwards until Proposition 1 is violated, yielding a monotonically decreasing collection of nested invariant sets converging to an attractive set (innermost lines).

yields the transformation T, and the matrix $J \approx \mathrm{diag}(-6, -6, -4.12, -4.26, -0.93, -0.95, -1, -1)$ is filled with negative real eigenvalues, signifying that the equilibrium $x^\star$ is locally stable. The T-transformed system (12) is analyzed with the embedding system induced by (13), using `npinterval` [17] to compute the natural inclusion functions of the symbolic Jacobian matrices to obtain $\mathsf{J}_x, \mathsf{J}_u, \mathsf{J}_w$. The interval $[\underline{y}_0, \overline{y}_0] = Tx^\star + [-0.05, 0.05]^4 \times [-0.08, 0.08]^2 \times [-0.11, 0.11]^2$ yields $\Phi^{-1}([\underline{y}_0, \overline{y}_0]) \subseteq [\underline{z}, \overline{z}]$, and $\mathsf{E}_{T,\Phi([\underline{z},\overline{z}])}(\underline{y}_0, \overline{y}_0, \underline{w}, \overline{w}) \geq_{\mathrm{SE}} 0$. Thus, using Theorem 2, the paralleletope $\Phi^{-1}([\underline{y}_0, \overline{y}_0])$ is an invariant set of (14). The embedding system in y coordinates is simulated forwards using Euler integration with a step-size of 0.1 for 90 time steps, and at each step the localization $[\underline{z}, \overline{z}] = T^{-1}[\underline{y}(t), \overline{y}(t)]$ is refined. Starting from $\begin{pmatrix} \underline{y}_0 \\ \overline{y}_0 \end{pmatrix}$, the forward-time embedding system converges to a point $\begin{pmatrix} \underline{y}^\star \\ \overline{y}^\star \end{pmatrix}$, where $\mathsf{E}_{T,\Phi([\underline{z},\overline{z}])}(\underline{y}^\star, \overline{y}^\star, \underline{w}, \overline{w}) = 0$. The transformed embedding system is also simulated backwards using Euler integration with a step-size of 0.05 until the condition $\begin{pmatrix} \underline{y}(t) \\ \overline{y}(t) \end{pmatrix} \geq_{\mathrm{SE}} 0$ is violated (call the final time t'). Using Theorems 1 and 2, the collection $\{\Phi^{-1}([\underline{y}(t), \overline{y}(t)])\}_{t \geq t'}$ consists of nested invariant paralleletopes converging to the attractive set $\Phi^{-1}([\underline{y}^\star, \overline{y}^\star])$ with region of attraction $\Phi^{-1}([\underline{y}(t'), \overline{y}(t')])$. The initial paralleletope takes 0.38 seconds to verify, and the entire nested family of 93 paralleletopes takes 40.28 seconds to compute.

VII. Conclusions

Using interval analysis and neural network verification tools, we propose a framework for certifying hyper-rectangle and paralleletope invariant sets in neural network controlled systems. The key component of our approach is the dynamical embedding system, whose trajectories can be used to construct a nested family of invariant sets. This work opens up an avenue for future work in designing safe learning-enabled controllers.

References

[1] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 1520–1527.

[2] U. Topcu, A. Packard, and P. Seiler, "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, vol. 44, no. 10, pp. 2669–2675, 2008.

[3] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[4] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[5] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer *et al.*, "Algorithms for verifying deep neural networks," *Foundations and Trends® in Optimization*, vol. 4, no. 3-4, pp. 244–404, 2021.

[6] C. Huang, J. Fan, X. Chen, W. Li, and Q. Zhu, "POLAR: A polynomial arithmetic framework for verifying neural-network controlled systems," in *Automated Technology for Verification and Analysis*. Springer International Publishing, 2022, pp. 414–430.

[7] C. Schilling, M. Forets, and S. Guadalupe, "Verification of neural-network control systems by integrating Taylor models and zonotopes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[8] S. Jafarpour, A. Harapanahalli, and S. Coogan, "Interval reachability of nonlinear dynamical systems with neural network controllers," in *Learning for Dynamics and Control Conference*. PMLR, 2023.

[9] ——, "Efficient interaction-aware interval analysis of neural network feedback loops," *arXiv preprint arXiv:2307.14938*, 2023.

[10] M. Everett, G. Habibi, C. Sun, and J. How, "Reachability analysis of neural feedback loops," *IEEE Access*, vol. 9, pp. 163 938–163 953, 2021.

[11] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-SDP: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," in *59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 5929–5934.

[12] A. Saoud and R. G. Sanfelice, "Computation of controlled invariants for nonlinear systems: Application to safe neural networks approximation and control," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 91–96, 2021, conference on Analysis and Design of Hybrid Systems (ADHS).

[13] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1980–1987, 2022.

[14] H. Dai, L. Landry, B.and Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," *arXiv preprint arXiv:2109.14152*, 2021.

[15] E. Bacci, M. Giacobbe, and D. Parker, "Verifying reinforcement learning up to infinity," in *Proceedings of the International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2021.

[16] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer London, 2001.

[17] A. Harapanahalli, S. Jafarpour, and S. Coogan, "A toolbox for fast interval arithmetic in `numpy` with an application to formal verification of neural network controlled system," in *2nd ICML Workshop on Formal Verification of Machine Learning*, 2023.

[18] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems*, vol. 31, 2018, p. 4944–4953.

[19] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *International Conference on Learning Representations*, 2019.

[20] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1129–1141, 2020.

[21] D. Angeli and E. D. Sontag, "Monotone control systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 10, pp. 1684–1698, 2003.

[22] G. A. Enciso, H. L. Smith, and E. D. Sontag, "Nonmonotone systems decomposable into monotone systems with negative feedback," *Journal of Differential Equations*, vol. 224, no. 1, pp. 205–227, 2006.

[23] H. L. Smith, *Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems*, ser. Mathematical Surveys and Monographs. American Mathematical Society, 1995.

[24] M. Abate and S. Coogan, "Computing robustly forward invariant sets for mixed-monotone systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 4553–4559.