Backward Reachability Analysis of Neural Feedback Systems Using Hybrid Zonotopes

Yuhao Zhang, *Graduate Student Member, IEEE*, Hang Zhang, and Xiangru Xu, *Member, IEEE*

***Abstract*—The proliferation of neural networks in safety-critical applications necessitates the development of effective methods to ensure their safety. This letter presents a novel approach for computing the exact backward reachable sets of neural feedback systems with known linear system models based on hybrid zonotopes. It is shown that the input-output relationship imposed by a ReLU-activated neural network can be exactly described by a hybrid zonotope-represented graph set. Based on that, the one-step exact backward reachable set of a neural feedback system is computed as a hybrid zonotope in the closed form. In addition, a necessary and sufficient condition is formulated as a mixed-integer linear program to certify whether the trajectories of a neural feedback system can avoid unsafe regions in finite time. Numerical examples are provided to demonstrate the efficiency of the proposed approach.**

***Index Terms*—Backward reachable set, neural networks, safety verification, hybrid zonotope.**

I. Introduction

NEURAL Networks (NNs) have become increasingly prevalent in autonomous systems. However, it has been shown that NNs are highly sensitive to even small perturbations in the input space, despite performing well in nominal scenarios [1]. Given the potential safety risks associated with using NNs in safety-critical systems, there is a pressing need for developing efficient tools to provide safety guarantees for control systems with NN components.

Reachability analysis of neural feedback systems, which are systems with NN controllers in the feedback loop, has been investigated in recent works [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. The majority of these results focus on forward reachability, which estimate the set of states that can be reached from an initial set [2], [3], [4], [5], [6], [7]. On the contrary, the backward reachability problem is to compute a set of states, known as the Backward Reachable Set (BRS), from which the system's trajectories can reach a specified *target set* within a finite time. For a safety-critical system (e.g., aircraft and autonomous vehicles), backward reachability analysis can identify the states that lead to safety violations when the target set is the set of unsafe states; for example, in the aircraft collision avoidance protocol, the unsafe target set would contain all the states where two aircraft are within the minimum separation distance [12].

Although various techniques have been developed for backward reachability analysis on systems without NNs [12], [13], [14], they are not directly applicable to neural feedback systems due to the highly nonlinear and nonconvex nature of NNs. In [11], a method was presented to compute the exact BRS of a ReLU-activated NN by determining the activation pattern, but it is only applicable to NNs in isolation, not neural feedback systems. In [8], an algorithm was proposed to over-approximate the BRS of a linear neural feedback system using the convex relaxation of NNs. The result was generalized in [9] for nonlinear system models with a guided partition algorithm to reduce the conservatism induced by the relaxation. A hybrid partition scheme was presented in [10] to further reduce conservatism. Note that the BRSs computed in these works are inexact, even for linear systems.

This letter aims to compute the *exact* BRS of a neural feedback system where the controller is a Feedforward Neural Network (FNN) with Rectified Linear Unit (ReLU) activation functions. The main mathematical tool used is *Hybrid Zonotope* (HZ), which can compactly represent a finite union of polytopic sets [15], [16], [17], [18]. This letter builds on our previous work [18], which shows that an FNN with ReLU activation functions can be exactly represented by an HZ and provides algorithms to compute the exact and approximated forward reachable sets of neural feedback systems. The contributions of this letter are at least threefold: (i) An algorithm with a linear set complexity growth rate is provided to represent the exact input-output relationship of a ReLU-activated FNN as an HZ, which is an improvement on the exponential set complexity growth rate given in [18]; (ii) Based on the reachability analysis of FNNs in isolation, an algorithm is proposed to compute the exact BRS of neural feedback systems represented by HZs; (iii) A necessary and sufficient condition formulated as a Mixed-Integer Linear Program (MILP) is provided to certify the safety properties of neural feedback systems. The performance of the proposed method is demonstrated through two numerical examples.

Manuscript received 17 March 2023; revised 19 May 2023; accepted 8 June 2023. Date of publication 26 June 2023; date of current version 12 July 2023. This work was supported in part by NSF under Grant CNS-2222541 and Grant CMMI-2237850. Recommended by Senior Editor S. Dey. *(Corresponding author: Xiangru Xu.)*

The authors are with the Department of Mechanical Engineering, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: yuhao.zhang2@wisc.edu; hang.zhang@wisc.edu; xiangru.xu@wisc.edu).

Digital Object Identifier 10.1109/LCSYS.2023.3289572

Notation: The i-th component of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by x_i with $i \in \{1, \dots, n\}$. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, $\mathbf{A}[i:j,:]$ denotes the matrix constructed by the i-th to j-th rows of $\mathbf{A}$. The identity matrix is denoted as $\boldsymbol{I}$ and $\boldsymbol{e}_i$ is the i-th column of $\boldsymbol{I}$. The vectors and matrices whose entries are all 0 (resp. 1) are denoted as $\mathbf{0}$ (resp. $\mathbf{1}$). Given sets $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{Z} \subset \mathbb{R}^m$ and a matrix $\boldsymbol{R} \in \mathbb{R}^{m\times n}$, the Cartesian product of $\mathcal{X}$ and $\mathcal{Z}$ is $\mathcal{X} \times \mathcal{Z} = \{(\boldsymbol{x}, \boldsymbol{z}) \mid \boldsymbol{x} \in \mathcal{X}, \boldsymbol{z} \in \mathcal{Z}\}$, the generalized intersection of $\mathcal{X}$ and $\mathcal{Z}$ under $\boldsymbol{R}$ is $\mathcal{X} \cap_{\boldsymbol{R}} \mathcal{Z} = \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{R}\boldsymbol{x} \in \mathcal{Z}\}$, and the k-ary Cartesian power of $\mathcal{X}$ is $\mathcal{X}^k = \mathcal{X} \times \cdots \times \mathcal{X}$.

II. Preliminaries & Problem Statement

A. Hybrid Zonotopes

Definition 1 [15, Definition 3]: The set $\mathcal{Z} \subset \mathbb{R}^n$ is a *hybrid zonotope* if there exist $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{G}^c \in \mathbb{R}^{n\times n_g}$, $\mathbf{G}^b \in \mathbb{R}^{n\times n_b}$, $\mathbf{A}^c \in \mathbb{R}^{n_c\times n_g}$, $\mathbf{A}^b \in \mathbb{R}^{n_c\times n_b}$, $\mathbf{b} \in \mathbb{R}^{n_c}$ such that

$$\mathcal{Z} = \left\{ \begin{bmatrix}\mathbf{G}^c & \mathbf{G}^b\end{bmatrix}\begin{bmatrix}\boldsymbol{\xi}^c \\ \boldsymbol{\xi}^b\end{bmatrix} + \mathbf{c} \;\middle|\; \begin{array}{c} \begin{bmatrix}\boldsymbol{\xi}^c \\ \boldsymbol{\xi}^b\end{bmatrix} \in \mathcal{B}_\infty^{n_g} \times \{-1,1\}^{n_b}, \\ \begin{bmatrix}\mathbf{A}^c & \mathbf{A}^b\end{bmatrix}\begin{bmatrix}\boldsymbol{\xi}^c \\ \boldsymbol{\xi}^b\end{bmatrix} = \mathbf{b} \end{array} \right\},$$

where $\mathcal{B}_\infty^{n_g} = \{\boldsymbol{x} \in \mathbb{R}^{n_g} \mid \|\boldsymbol{x}\|_\infty \le 1\}$ is the unit hypercube in $\mathbb{R}^{n_g}$. The shorthand notation of the hybrid zonotope is given by $\mathcal{Z} = \langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle$.

Given an HZ $\mathcal{Z} = \langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle$, the vector $\mathbf{c}$ is called the *center*, the columns of $\mathbf{G}^b$ are called the *binary generators*, and the columns of $\mathbf{G}^c$ are called the *continuous generators*. For simplicity, we define the set $\mathcal{B}(\mathbf{A}^c, \mathbf{A}^b, \mathbf{b}) = \{(\boldsymbol{\xi}^c, \boldsymbol{\xi}^b) \in \mathcal{B}_\infty^{n_g} \times \{-1,1\}^{n_b} \mid \mathbf{A}^c\boldsymbol{\xi}^c + \mathbf{A}^b\boldsymbol{\xi}^b = \mathbf{b}\}$.

Identities are provided to compute the linear map and generalized intersection [15, Proposition 7], union operation [19, Proposition 1], and Cartesian product of HZs [20, Proposition 3.2.5]. The emptiness of an HZ can be verified by solving an MILP [15].

Lemma 1: Given $\mathcal{Z} = \langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle \subset \mathbb{R}^n$, $\mathcal{Z} \neq \emptyset$ if and only if $\min\{\|\boldsymbol{\xi}^c\|_\infty \mid \mathbf{A}^c\boldsymbol{\xi}^c + \mathbf{A}^b\boldsymbol{\xi}^b = \mathbf{b}, \boldsymbol{\xi}^c \in \mathbb{R}^{n_g}, \boldsymbol{\xi}^b \in \{-1,1\}^{n_b}\} \le 1$.

B. Problem Statement

Consider the following discrete-time linear system:

$$\boldsymbol{x}(t+1) = \boldsymbol{A}_d\boldsymbol{x}(t) + \boldsymbol{B}_d\boldsymbol{u}(t), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$ are the state and the control input, respectively. We assume $\boldsymbol{x} \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^n$ is called the state set and the controller is given as $\boldsymbol{u}(t) = \pi(\boldsymbol{x}(t))$, where π is an ℓ-layer FNN with ReLU activation functions. The neural feedback system consisting of system (1) and controller π is a closed-loop system denoted as:

$$\boldsymbol{x}(t+1) = \boldsymbol{f}_{cl}(\boldsymbol{x}(t)) \triangleq \boldsymbol{A}_d\boldsymbol{x}(t) + \boldsymbol{B}_d\pi(\boldsymbol{x}(t)). \tag{2}$$

Given a target set $\mathcal{T} \subset \mathcal{X}$ for the closed-loop system (2), the set of states in $\mathcal{X}$ that can be mapped into the target set $\mathcal{T}$ by (2) in exactly t steps is defined as the t-step BRS and denoted as $\mathcal{P}_t(\mathcal{T}) \triangleq \{\boldsymbol{x}(0) \in \mathcal{X} | \boldsymbol{x}(t) \in \mathcal{T}, \boldsymbol{x}(k) = \boldsymbol{f}_{cl}(\boldsymbol{x}(k_1)), k = 1, 2, \dots, t\}$. Note that the t-step BRS is always a subset of the state set $\mathcal{X}$, i.e., $\mathcal{P}_t(\mathcal{T}) \subseteq \mathcal{X}$. For simplicity, the one-step BRS is also denoted as $\mathcal{P}(\mathcal{T})$, i.e., $\mathcal{P}(\mathcal{T}) = \mathcal{P}_1(\mathcal{T})$.

The equivalence of an HZ and a union of constrained zonotopes [15, Th. 5] shows that an HZ can compactly represent non-convex sets with flat faces. In this letter, we assume both the state set $\mathcal{X}$ and the target set $\mathcal{T}$ are represented by HZs. This assumption enables us to handle sets and system dynamics using a unified HZ-based approach.

For the ℓ-layer FNN controller π, the k-th layer weight matrix and bias vector are denoted as $\boldsymbol{W}^{(k-1)}$ and $\boldsymbol{v}^{(k-1)}$, respectively, where $k = 1, \dots, \ell$. Denote $\boldsymbol{x}^{(k)}$ as the neurons of the k-th layer and n_k as the dimension of $\boldsymbol{x}^{(k)}$. Then, for $k = 1, \dots, \ell - 1$, we have $\boldsymbol{x}^{(k)} = \phi(\boldsymbol{W}^{(k-1)}\boldsymbol{x}^{(k-1)} + \boldsymbol{v}^{(k-1)})$, where $\boldsymbol{x}^{(0)} = \boldsymbol{x}(t)$ and ϕ is the *vector-valued* activation function constructed by component-wise repetition of ReLU function, i.e., $\phi(\boldsymbol{x}) \triangleq [ReLU(x_1), \dots, ReLU(x_n)]^T$. Only the linear map is applied in the last layer, i.e., $\pi(\boldsymbol{x}(t)) = \boldsymbol{x}^{(\ell)} = \boldsymbol{W}^{(\ell-1)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{v}^{(\ell-1)}$. The total number of hidden neurons is denoted as $N_\pi = n_1 + \cdots + n_{\ell-1}$.

The following problem will be investigated in this letter.

Problem 1: Given a target set $\mathcal{T} \subset \mathcal{X}$ represented as an HZ and a time horizon $T \in \mathbb{Z}_{>0}$, compute the exact BRS $\mathcal{P}_t(\mathcal{T})$ of the neural feedback system (2), for $t = 1, 2, \dots, T$.

III. Exact Backward Reachability Analysis

In this section, we first present a technique that can represent the exact input-output relationship of a ReLU-activated FNN as an HZ-based graph set that has a linear set complexity growth rate. Then, based on that, we show if the target set is given as an HZ, the exact BRS of the system (2) can be also represented as HZs in closed form.

A. Representation of the Graph of FNNs via HZs

The problem of computing the BRS and invariant set of controlled dynamical systems has been studied in many works, such as [21], [22], [23]. A commonly-used technique is to abstract the constraints imposed by the dynamic system in the input-output space. For example, state-update sets are proposed in [16] to compute successor and precursor sets for hybrid systems. For neural feedback systems, the imposed constraints can be identified by finding a proper representation of the input-output relationship of the NN controllers.

One of the major difficulties in analyzing NNs is the composition of nonlinear activation functions [6]. To simplify the analysis of NNs, quadratic constraints have been utilized to abstract the constraints imposed by the NNs on the pre- and post-activation signals [6], [24]. Building upon these methodologies, our approach employs an HZ to capture the constraints imposed by NNs in an exact manner. Specifically, we denote $\mathcal{G}(\pi, \mathcal{X}) = \{(\boldsymbol{x}, \boldsymbol{u}) \mid \boldsymbol{u} = \pi(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\} \subset \mathbb{R}^{n+m}$ as the *graph* of the ReLU-activated FNN π over the state space domain $\mathcal{X}$, and we will show that there exists an HZ $\mathcal{H}_\pi = \langle \mathbf{G}_\pi^c, \mathbf{G}_\pi^b, \mathbf{c}_\pi, \mathbf{A}_\pi^c, \mathbf{A}_\pi^b, \mathbf{b}_\pi\rangle$, such that $\mathcal{G}(\pi, \mathcal{X}) = \mathcal{H}_\pi$.

To that end, we first consider the representation of a scalar-valued ReLU function $x = ReLU(z) = \max\{z, 0\}$ over an interval domain $[-\alpha, \beta]$ where $\alpha, \beta \in \mathbb{R}_{>0}$. The graph of the ReLU function over the interval domain is plotted in Fig. 1.

It is obvious that the set of points satisfying the ReLU function over $[-\alpha, \beta]$ form two line segments which can be exactly

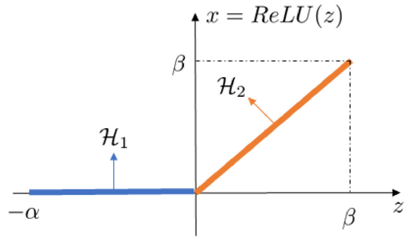

Fig. 1. The graph of the ReLU function as the union of two HZs $\mathcal{H}_1$ and $\mathcal{H}_2$, over the interval domain $[-\alpha, \beta]$.

represented as two HZs $\mathcal{H}_1$ and $\mathcal{H}_2$ given as follows:

$$\mathcal{H}_1 = \left\langle \begin{bmatrix} \frac{\alpha}{2} \\ 0 \end{bmatrix}, \emptyset, \begin{bmatrix} \frac{-\alpha}{2} \\ 0 \end{bmatrix}, \emptyset, \emptyset, \emptyset \right\rangle, \mathcal{H}_2 = \left\langle \begin{bmatrix} \frac{\beta}{2} \\ \frac{\beta}{2} \end{bmatrix}, \emptyset, \begin{bmatrix} \frac{\beta}{2} \\ \frac{\beta}{2} \end{bmatrix}, \emptyset, \emptyset, \emptyset \right\rangle.$$

The union of $\mathcal{H}_1$ and $\mathcal{H}_2$ can be directly computed as a single HZ $\mathcal{H}$ using [19, Proposition 1]. We use the approach presented in [18, Algorithm 3] to identify two redundant continuous generators. We then apply [18, Proposition 3] with proper transformation matrices to remove the redundant continuous generators and obtain

$$\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 = \langle \mathbf{G}_h^c, \mathbf{G}_h^b, \mathbf{c}_h, \mathbf{A}_h^c, \mathbf{A}_h^b, \mathbf{b}_h \rangle, \tag{3}$$

where

$$\mathbf{G}_h^c = \begin{bmatrix} -\frac{\alpha}{2} & -\frac{\beta}{2} & 0 & 0 \\ 0 & -\frac{\beta}{2} & 0 & 0 \end{bmatrix}, \mathbf{G}_h^b = \begin{bmatrix} -\frac{\alpha}{2} \\ 0 \end{bmatrix}, \mathbf{c}_h = \begin{bmatrix} \frac{\beta}{2} \\ \frac{\beta}{2} \end{bmatrix},$$
$$\mathbf{A}_h^c = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \mathbf{A}_h^b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{b}_h = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Note that a similar formulation was presented in [25, Lemma 1] to represent the ReLU as an HZ for forward reachability analysis. Their formulation was built on the relationship of three zonotopes while the graph set $\mathcal{H}$ in (3) was computed based on the union of two line segments.

Using (3), we get the exact representation of the graph of the ReLU function over $[-\alpha, \beta]$ using HZ, i.e., $\mathcal{H} = \{(z, x) \in \mathbb{R}^2 \mid x = ReLU(z), z \in [-\alpha, \beta]\}$. Note that the graph of a ReLU function can be also linearly approximated using intervals, symbolic intervals, and polytopes, as stated in [26]; however, the nonlinear nature of the ReLU function makes it impossible for these convex relaxation-based set representations to exactly represent its graph.

In the following lemma, the analysis described above on the ReLU function is extended to the vector-valued activation function ϕ over a domain represented as an HZ.

Lemma 2: Given a domain represented as an HZ $\mathcal{Z} \subset \mathbb{R}^{n_k}$, the graph of the k-th layer's vector-valued activation function $\phi : \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}$ over $\mathcal{Z}$ can be exactly represented by the following HZ:

$$\mathcal{G}(\phi, \mathcal{Z}) = (\boldsymbol{P} \cdot \mathcal{H}^{n_k}) \cap_{[\boldsymbol{I}\ \mathbf{0}]} \mathcal{Z}, \tag{4}$$

where $\boldsymbol{P} = [\boldsymbol{e}_2\ \boldsymbol{e}_4, \cdots, \boldsymbol{e}_{2n_k}\ \boldsymbol{e}_1\ \boldsymbol{e}_3, \cdots, \boldsymbol{e}_{2n_k-1}]^T \in \mathbb{R}^{2n_k \times 2n_k}$ is a permutation matrix and $\mathcal{H}$ is given in (3).

Proof: Since the HZ $\mathcal{Z}$ is a closed set, we can always find large enough scalars $\alpha, \beta \in \mathbb{R}_{>0}$ such that the interval $\mathcal{I} = [-\alpha\mathbf{1}, \beta\mathbf{1}] \subset \mathbb{R}^{n_k}$ is an enclosure of $\mathcal{Z}$, i.e., $\mathcal{Z} \subseteq \mathcal{I}$. Let $\boldsymbol{z}^{(k)}$ denote the input of function ϕ and $\boldsymbol{x}^{(k)}$ denote the output. The graph of ϕ over the domain $\mathcal{I}$ is $\mathcal{G}(\phi, \mathcal{I}) = \{(\boldsymbol{z}^{(k)}, \boldsymbol{x}^{(k)}) \mid \boldsymbol{x}^{(k)} = \phi(\boldsymbol{z}^{(k)}), \boldsymbol{z}^{(k)} \in \mathcal{I}\} \subset \mathbb{R}^{2n_k}$. As the vector-valued activation function ϕ is constructed by component-wise repetition of ReLU functions, i.e., $x_i^{(k)} = ReLU(z_i^{(k)})$, we have $[z_1^{(k)}\ x_1^{(k)}\ z_2^{(k)}\ x_2^{(k)}\ \cdots\ z_{n_k}^{(k)}\ x_{n_k}^{(k)}]^T \in \mathcal{H}^{n_k} \subset \mathbb{R}^{2n_k}$. To reassemble the pairs of input and output elements in the same order of $[\boldsymbol{z}^{(k)T}, \boldsymbol{x}^{(k)T}]^T$, we use the permutation matrix $\boldsymbol{P}$ and get $[\boldsymbol{z}^{(k)T}, \boldsymbol{x}^{(k)T}]^T = [z_1^{(k)}, \ldots, z_{n_k}^{(k)}\ x_1^{(k)}, \ldots, x_{n_k}^{(k)}]^T = \boldsymbol{P}[z_1^{(k)}\ x_1^{(k)}\ \cdots,\ z_{n_k}^{(k)}\ x_{n_k}^{(k)}]^T$. Since HZs are closed under the linear map and generalized intersection [15, Proposition 7], the graph of ϕ over the interval $\mathcal{I}$ is an HZ as $\mathcal{G}(\phi, \mathcal{I}) = \boldsymbol{P} \cdot \mathcal{H}^{n_k}$. Then, we have $\mathcal{G}(\phi, \mathcal{Z}) = \{(\boldsymbol{z}^{(k)}, \boldsymbol{x}^{(k)}) \mid \boldsymbol{x}^{(k)} = \phi(\boldsymbol{z}^{(k)}), \boldsymbol{z}^{(k)} \in \mathcal{Z}\} = \mathcal{G}(\phi, \mathcal{I}) \cap_{[\boldsymbol{I}\ \mathbf{0}]} \mathcal{Z} = (\boldsymbol{P} \cdot \mathcal{H}^{n_k}) \cap_{[\boldsymbol{I}\ \mathbf{0}]} \mathcal{Z}$, which is also an HZ. This completes the proof. ■

Remark 1: Lemma 2 shows that the graph set of a vector-valued ReLU activation function can be exactly represented by an HZ. Reference [6, Lemma 4] abstracts the input-output relationship of the ReLU function using quadratic constraints. However, their proposed approach will only provide an over-approximation of the graph set.

From the structure of the FNN π, it is obvious that each layer is a composition of the activation function ϕ and the linear map with weight matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{v}$. Therefore, to construct the HZ representation $\mathcal{H}_\pi = \mathcal{G}(\pi, \mathcal{X})$ for the graph of the entire network π, we can repeat the procedures described in Lemma 2 layer-by-layer and connect the input of the k-th layer $\boldsymbol{z}^{(k)}$ and the output of the $(k-1)$-th layer $\boldsymbol{x}^{(k-1)}$ with the linear map $\boldsymbol{z}^{(k)} = \boldsymbol{W}^{(k-1)}\boldsymbol{x}^{(k-1)} + \boldsymbol{v}^{(k-1)}$. The details on the iterative construction of the HZ $\mathcal{H}_\pi$ are summarized in Algorithm 1.

Theorem 1: Given an ℓ-layer ReLU-activated FNN $\pi : \mathbb{R}^n \to \mathbb{R}^m$ and an HZ $\mathcal{X} \subset \mathbb{R}^n$, the output of Algorithm 1 $\mathcal{H}_\pi$ is an HZ that can exactly represent the graph set of π over the domain $\mathcal{X}$, i.e., $\mathcal{H}_\pi = \mathcal{G}(\pi, \mathcal{X})$.

Proof: For the ℓ-layer ReLU-activated FNN π, it is easy to check that the input set $\mathcal{Z}^{(k-1)}$, graph set $\mathcal{G}^{(k)}$ and output

Algorithm 1: Exact Graph Set Computation of FNN via HZs

```
Input: HZ domain $\mathcal{X}$, number of layers $\ell$, weight
       matrices $\{\boldsymbol{W}^{(k-1)}\}_{k=1}^{\ell}$, bias vectors $\{\boldsymbol{v}^{(k-1)}\}_{k=1}^{\ell}$,
       large scalars $\alpha, \beta > 0$
Output: exact graph set as an HZ $\mathcal{H}_\pi = \mathcal{G}(\pi, \mathcal{X})$
1  $\mathcal{X}^{(0)} \leftarrow \mathcal{X} = \langle \mathbf{G}_x^c, \mathbf{G}_x^b, \mathbf{c}_x, \mathbf{A}_x^c, \mathbf{A}_x^b, \mathbf{b}_x \rangle$;
2  $\mathcal{H} \leftarrow$ compute the graph of ReLU using (3);
3  for $k \in \{1, 2, \ldots, \ell-1\}$ do
4      $\mathcal{Z}^{(k-1)} \leftarrow \boldsymbol{W}^{(k-1)}\mathcal{X}^{(k-1)} + \boldsymbol{v}^{(k-1)}$;   // Input set
5      $\mathcal{G}^{(k)} \leftarrow (\boldsymbol{P} \cdot \mathcal{H}^{n_k}) \cap_{[\boldsymbol{I}\ \mathbf{0}]} \mathcal{Z}^{(k-1)}$;   // Using (4)
6      $\mathcal{X}^{(k)} \leftarrow [\mathbf{0}\ \boldsymbol{I}] \cdot \mathcal{G}^{(k)}$ ;   // Output set
7  $\mathcal{X}^{(\ell)} \leftarrow \boldsymbol{W}^{(\ell-1)}\mathcal{X}^{(\ell-1)} + \boldsymbol{v}^{(\ell-1)}$;   // Last layer
8  $\langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b} \rangle \leftarrow \mathcal{X}^{(\ell)}$;
   // Stack input and output
9  $\mathcal{H}_\pi \leftarrow \langle \begin{bmatrix} \mathbf{G}_x^c & \mathbf{0} \\ & \mathbf{G}^c \end{bmatrix}, \begin{bmatrix} \mathbf{G}_x^b & \mathbf{0} \\ & \mathbf{G}^b \end{bmatrix}, \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c} \end{bmatrix}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b} \rangle$;
10 return $\mathcal{H}_\pi$
```

set $\mathcal{X}^{(k)}$ of the k-th layer activation function ϕ are computed iteratively for $k = 1, \ldots, \ell-1$ in Line 4-6 of Algorithm 1. For the last layer, only a linear map is applied and the output set of FNN π is computed as $\mathcal{X}^{(\ell)}$ in Line 7. Note that from the construction, the equality constraints in the domain set $\mathcal{X}$ are included in $\mathcal{X}^{(\ell)}$. In Line 8, $\mathcal{H}_\pi$ stacks the input and output of π as $\mathcal{H}_\pi = \{(\boldsymbol{x}, \boldsymbol{u}) \mid \boldsymbol{x} \in \mathcal{X}, \boldsymbol{u} = \pi(\boldsymbol{x})\} = \mathcal{G}(\pi, \mathcal{X})$, which is an exact representation of the graph set of π over $\mathcal{X}$. ■

Denote $n_{g,x}$, $n_{b,x}$ and $n_{c,x}$ as the number of continuous generators, binary generators and equality constraints of the HZ $\mathcal{X}$, respectively. The set complexity growth of the graph set $\mathcal{H}_\pi$ is given by $n_{g,\pi} = n_{g,x} + 4N_\pi$, $n_{b,\pi} = n_{b,x} + N_\pi$, $n_{c,\pi} = n_{c,x} + 3N_\pi$. The output set $\mathcal{X}^{(\ell)}$ of the FNN π computed in Algorithm 1 has the same set complexity as $\mathcal{H}_\pi$. In our previous work [18], it has been shown that a ReLU-activated FNN can be exactly represented by an HZ; however, the set complexity of the computed HZ there will grow exponentially with the number of neurons in the FNN. In comparison. the HZ representation of FNN produced by Algorithm 1 has a linear set complexity growth rate.

B. Computation of Exact BRS for Neural Feedback Systems

In this subsection, we will consider the computation of exact BRS for the neural feedback system (2). Inspired by the precursor set formulation for Mixed Logical Dynamical (MLD) systems given by [16, Th. 2], the following theorem provides the closed-form of the one-step BRS, $\mathcal{P}(\mathcal{T})$ with a given target set represented by an HZ, $\mathcal{T}$.

Theorem 2: Given any HZ $\mathcal{X} \subset \mathbb{R}^n$, let $\mathcal{H}_\pi = \langle \mathbf{G}^c_\pi, \mathbf{G}^b_\pi, \mathbf{c}_\pi, \mathbf{A}^c_\pi, \mathbf{A}^b_\pi, \mathbf{b}_\pi \rangle$ be the computed graph set of the FNN π over the domain $\mathcal{X}$ using Algorithm 1, i.e., $\mathcal{H}_\pi = \mathcal{G}(\pi, \mathcal{X})$. Let $\boldsymbol{D} = \begin{bmatrix}\boldsymbol{A}_d & \boldsymbol{B}_d\end{bmatrix}$. Then, for any target set represented by an HZ $\mathcal{T} = \langle \mathbf{G}^c_\tau, \mathbf{G}^b_\tau, \mathbf{c}_\tau, \mathbf{A}^c_\tau, \mathbf{A}^b_\tau, \mathbf{b}_\tau \rangle \subset \mathbb{R}^n$, the one-step BRS of the neural feedback system (2) is an HZ given as

$$\mathcal{P}(\mathcal{T}) = \langle \mathbf{G}^c_p, \mathbf{G}^b_p, \mathbf{c}_p, \mathbf{A}^c_p, \mathbf{A}^b_p, \mathbf{b}_p \rangle, \tag{5}$$

where

$$\mathbf{G}^c_p = \begin{bmatrix}\mathbf{G}^c_\pi[1:n,:] & \mathbf{0}\end{bmatrix},\ \mathbf{G}^b_p = \begin{bmatrix}\mathbf{G}^b_\pi[1:n,:] & \mathbf{0}\end{bmatrix},$$
$$\mathbf{A}^c_p = \begin{bmatrix}\mathbf{A}^c_\pi & \mathbf{0}\\ \mathbf{0} & \mathbf{A}^c_\tau\\ \boldsymbol{D}\mathbf{G}^c_\pi & -\mathbf{G}^c_\tau\end{bmatrix},\ \mathbf{A}^b_p = \begin{bmatrix}\mathbf{A}^b_\pi & \mathbf{0}\\ \mathbf{0} & \mathbf{A}^b_\tau\\ \boldsymbol{D}\mathbf{G}^b_\pi & -\mathbf{G}^b_\tau\end{bmatrix},$$
$$\mathbf{c}_p = \mathbf{c}_\pi[1:n,:],\quad \mathbf{b}_p = \begin{bmatrix}\mathbf{b}_\pi\\ \mathbf{b}_\tau\\ \mathbf{c}_\tau - \boldsymbol{D}\mathbf{c}_\pi\end{bmatrix}.$$

Proof: By the definition of the one-step BRS, we have $\mathcal{P}(\mathcal{T}) = \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{f}_{cl}(\boldsymbol{x}) \in \mathcal{T}\} = \{\boldsymbol{x} \mid \boldsymbol{A}_d\boldsymbol{x} + \boldsymbol{B}_d\boldsymbol{u} \in \mathcal{T}, \boldsymbol{u} = \pi(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\} = \{\boldsymbol{x} \mid \boldsymbol{D}[\boldsymbol{x}^T\ \boldsymbol{u}^T]^T \in \mathcal{T}, [\boldsymbol{x}^T\ \boldsymbol{u}^T]^T \in \mathcal{H}_\pi\}$. Denote the right-hand side of (5) as $\mathcal{H}_p$. We will first prove that $\mathcal{P}(\mathcal{T}) \subseteq \mathcal{H}_p$. Let $\boldsymbol{x}$ be any element of set $\mathcal{P}(\mathcal{T})$. Then, there exist $\boldsymbol{\xi}^c_\pi, \boldsymbol{\xi}^b_\pi, \boldsymbol{\xi}^c_\tau$ and $\boldsymbol{\xi}^b_\tau$ such that $(\boldsymbol{\xi}^c_\pi, \boldsymbol{\xi}^b_\pi) \in \mathcal{B}(\mathbf{A}^c_\pi, \mathbf{A}^b_\pi, \mathbf{b}_\pi)$, $(\boldsymbol{\xi}^c_\tau, \boldsymbol{\xi}^b_\tau) \in \mathcal{B}(\mathbf{A}^c_\tau, \mathbf{A}^b_\tau, \mathbf{b}_\tau)$, $[\boldsymbol{x}^T\ \pi(\boldsymbol{x})^T]^T = \mathbf{G}^c_\pi\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi$ and $\boldsymbol{D}[\boldsymbol{x}^T\ \pi(\boldsymbol{x})^T]^T = \mathbf{G}^c_\tau\boldsymbol{\xi}^c_\tau + \mathbf{G}^b_\tau\boldsymbol{\xi}^b_\tau + \mathbf{c}_\tau$. Therefore, we have $\boldsymbol{x} = [\boldsymbol{I}_n\ \ \mathbf{0}] \cdot [\boldsymbol{x}^T\ \pi(\boldsymbol{x})^T]^T = [\boldsymbol{I}_n\ \ \mathbf{0}] \cdot (\mathbf{G}^c_\pi\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi) = \mathbf{G}^c_\pi[1:n,:]\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi[1:n,:]\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi[1:n,:]$ and $\boldsymbol{D}(\mathbf{G}^c_\pi\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi) = \mathbf{G}^c_\tau\boldsymbol{\xi}^c_\tau + \mathbf{G}^b_\tau\boldsymbol{\xi}^b_\tau + \mathbf{c}_\tau$.

Let $\boldsymbol{\xi}^c = [(\boldsymbol{\xi}^c_\pi)^T\ (\boldsymbol{\xi}^c_\tau)^T]^T$ and $\boldsymbol{\xi}^b = [(\boldsymbol{\xi}^b_\pi)^T\ (\boldsymbol{\xi}^b_\tau)^T]^T$. Then, it's easy to check that $\boldsymbol{x} = \begin{bmatrix}\mathbf{G}^c_\pi[1:n,:] & \mathbf{0}\end{bmatrix}\boldsymbol{\xi}^c + \begin{bmatrix}\mathbf{G}^b_\pi[1:n,:] & \mathbf{0}\end{bmatrix}\boldsymbol{\xi}^b + \mathbf{c}_\pi[1:n,:] = \mathbf{G}^c_p\boldsymbol{\xi}^c + \mathbf{G}^b_p\boldsymbol{\xi}^b + \mathbf{c}_p$ and $(\boldsymbol{\xi}^c, \boldsymbol{\xi}^b) \in \mathcal{B}(\mathbf{A}^c_p, \mathbf{A}^b_p, \mathbf{b}_p)$. Thus, we have $\boldsymbol{x} \in \mathcal{H}_p$. And since $\boldsymbol{x}$ is arbitrary, we know that $\mathcal{P}(\mathcal{T}) \subseteq \mathcal{H}_p$. Next, we will show that $\mathcal{H}_p \subseteq \mathcal{P}(\mathcal{T})$. Let $\boldsymbol{x} \in \mathcal{H}_p$. Then, there exist $\boldsymbol{\xi}^c$ and $\boldsymbol{\xi}^b$ such that $(\boldsymbol{\xi}^c, \boldsymbol{\xi}^b) \in \mathcal{B}(\mathbf{A}^c_p, \mathbf{A}^b_p, \mathbf{b}_p)$ and $\boldsymbol{x} = \mathbf{G}^c_p\boldsymbol{\xi}^c + \mathbf{G}^b_p\boldsymbol{\xi}^b + \mathbf{c}_p$. Partitioning $\boldsymbol{\xi}^c$ as $\boldsymbol{\xi}^c = [(\boldsymbol{\xi}^c_\pi)^T\ (\boldsymbol{\xi}^c_\tau)^T]^T$ and $\boldsymbol{\xi}^b$ as $\boldsymbol{\xi}^b = [(\boldsymbol{\xi}^b_\pi)^T\ (\boldsymbol{\xi}^b_\tau)^T]^T$, it follows that $(\boldsymbol{\xi}^c_\pi, \boldsymbol{\xi}^b_\pi) \in \mathcal{B}(\mathbf{A}^c_\pi, \mathbf{A}^b_\pi, \mathbf{b}_\pi)$, $(\boldsymbol{\xi}^c_\tau, \boldsymbol{\xi}^b_\tau) \in \mathcal{B}(\mathbf{A}^c_\tau, \mathbf{A}^b_\tau, \mathbf{b}_\tau)$ and $\boldsymbol{x} = \mathbf{G}^c_\pi[1:n,:]\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi[1:n,:]\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi[1:n,:]$. Choose $\boldsymbol{u} = \mathbf{G}^c_\pi[n+1:m+n,:]\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi[n+1:m+n,:]\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi[n+1:m+n,:]$. Then, we can get $[\boldsymbol{x}^T\ \boldsymbol{u}^T]^T = \mathbf{G}^c_\pi\boldsymbol{\xi}^c_\pi + \mathbf{G}^b_\pi\boldsymbol{\xi}^b_\pi + \mathbf{c}_\pi$ and $\boldsymbol{D}[\boldsymbol{x}^T\ \boldsymbol{u}^T]^T = \mathbf{G}^c_\tau\boldsymbol{\xi}^c_\tau + \mathbf{G}^b_\tau\boldsymbol{\xi}^b_\tau + \mathbf{c}_\tau$. Thus, $\boldsymbol{x} \in \mathcal{P}(\mathcal{T})$. Since $\boldsymbol{x}$ is arbitrary, $\mathcal{H}_p \subseteq \mathcal{P}(\mathcal{T})$. Therefore, we conclude that $\mathcal{P}(\mathcal{T}) = \mathcal{H}_p$. ■

To the best of our knowledge, Theorem 2 is the first result that can compute the exact BRS of a neural feedback system that consists of a linear model and an FNN controller.

Based on Theorem 2, the exact T-step BRS of system (2) can be computed iteratively as follows:

$$\mathcal{P}_0(\mathcal{T}) = \mathcal{T},\quad \mathcal{P}_t(\mathcal{T}) = \mathcal{P}(\mathcal{P}_{t-1}(\mathcal{T})),\ t = 1, \ldots, T. \tag{6}$$

Assuming that the target set $\mathcal{T}$ has $n_{g,\tau}$ continuous generators, $n_{b,\tau}$ binary generators and $n_{c,\tau}$ equality constraints, the set complexity of the T-step BRS computed using (6) and Theorem 2 is given by $n_{g,p} = T \cdot (n_{g,x} + 4N_\pi) + n_{g,\tau}$, $n_{b,p} = T \cdot (n_{b,x} + N_\pi) + n_{b,\tau}$, $n_{c,p} = T \cdot (n_{c,x} + 3N_\pi + n) + n_{c,\tau}$, where the subscript p represents the one-step BRS $\mathcal{P}(\mathcal{T})$.

Remark 2: Linear Programming (LP)-based methods were proposed in [8], [9], [10] to over-approximate the BRS for neural feedback systems. Our method relies on the exact HZ representation of the nonlinearities of ReLU-activated FNNs and can compute exact BRSs without the necessity of partition when the system model is linear. For general nonlinear feedback systems, our approach can be readily extended by abstracting nonlinear dynamics with piece-wise linear bounds as in [27]. In the presence of modeling error and measurement noise, our method remains applicable provided that these uncertainties are bounded by hybrid zonotopes.

Remark 3: In [16], a novel HZ-based approach was proposed to compute the precursor set of MLD systems by constructing the state-update sets. Although one might apply the results there for the backward reachability analysis of neural feedback systems, this will require the transformation from the ReLU-activated FNN to an equivalent MLD system using the big-M formulation. However, this transformation will induce additional auxiliary variables and result in a more complex hybrid zonotope representation than Theorem 2.

Remark 4: The analysis in the preceding subsections can be readily extended to neural feedback systems with saturated control inputs, using techniques similar to [4]. Specifically, assume that the system (1) has interval control input constraints, i.e., $\boldsymbol{u} \in \mathcal{U} = [\underline{\boldsymbol{u}}, \overline{\boldsymbol{u}}]$. Then the closed-loop system (2) becomes $\boldsymbol{x}(t+1) = \boldsymbol{A}_d\boldsymbol{x}(t) + \boldsymbol{B}_d\ sat^{\overline{\boldsymbol{u}}}_{\underline{\boldsymbol{u}}}(\pi(\boldsymbol{x}(t)))$, where the saturation function can be equivalently described by the ReLU functions as $sat^{\overline{\boldsymbol{u}}}_{\underline{\boldsymbol{u}}}(\boldsymbol{u}) = \min\{\max\{\boldsymbol{u}, \overline{\boldsymbol{u}}\}, \underline{\boldsymbol{u}}\} =$

$ReLU(-ReLU(\bar{\boldsymbol{u}} - \boldsymbol{u}) + \underline{\boldsymbol{u}} - \bar{\boldsymbol{u}}) + \underline{\boldsymbol{u}}$. Therefore, the saturated NN controller $\hat{\pi}(\boldsymbol{x}) = sat_{\underline{\boldsymbol{u}}}^{\bar{\boldsymbol{u}}}(\pi(\boldsymbol{x}))$ is an $(\ell + 2)$-layer ReLU-activated FNN. Then, all preceding results can be directly applied to this modified FNN.

IV. Safety Verification for Neural Feedback Systems via BRS

In this section, the backward reachability analysis in the preceding section will be utilized for the safety verification of neural feedback systems.

Consider an initial state set $\mathcal{X}_0 \subset \mathcal{X}$ and an unsafe region $\mathcal{O} \subset \mathcal{X}$, both of which are represented as HZs. We consider the unsafe set $\mathcal{O}$ as the target set in Section III and suppose that the exact t-step BRS of $\mathcal{O}$ can be computed as $\mathcal{P}_t(\mathcal{O})$ by (6), where $t = 1, \ldots, T$ with T an arbitrary positive integer. Clearly, if $\mathcal{X}_0$ does not intersect with any $\mathcal{P}_t$ for $t = 1, \ldots, T$, any state trajectory that starts from $\mathcal{X}_0$ will not enter into the unsafe region $\mathcal{O}$ within T time steps, in other words, the neural feedback system (2) is safe within T steps. By [15, Proposition 7] and Lemma 1, checking the emptiness of the intersection of $\mathcal{X}_0$ and $\mathcal{P}_t$ is equivalent to solving an MILP.

The safety verification of neural feedback systems via BRSs is summarized in the following proposition whose proof is omitted due to space limitations.

Proposition 1: Suppose that an initial state set $\mathcal{X}_0 = \langle \mathbf{G}_0^c, \mathbf{G}_0^b, \mathbf{c}_0, \mathbf{A}_0^c, \mathbf{A}_0^b, \mathbf{b}_0 \rangle \subset \mathcal{X}$ and an unsafe set $\mathcal{O} \subset \mathcal{X}$ are both HZs, and $\mathcal{P}_t = \langle \mathbf{G}_t^c, \mathbf{G}_t^b, \mathbf{c}_t, \mathbf{A}_t^c, \mathbf{A}_t^b, \mathbf{b}_t \rangle$ is the exact t-step BRS of $\mathcal{O}$ where $t = 1, \ldots, T$ with T an arbitrary positive integer. Then, the state trajectories of the neural feedback system (2) starting from $\mathcal{X}_0$ can avoid the unsafe region $\mathcal{O}$ within T steps, if and only if the following condition holds for $t = 1, \ldots, T$:

$$\min \left\{ \|\boldsymbol{\xi}^c\|_\infty \middle| \begin{bmatrix} \mathbf{A}_t^c & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_0^c \\ \mathbf{G}_t^c & -\mathbf{G}_0^c \end{bmatrix} \boldsymbol{\xi}^c + \begin{bmatrix} \mathbf{A}_t^b & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_0^b \\ \mathbf{G}_t^b & -\mathbf{G}_0^b \end{bmatrix} \boldsymbol{\xi}^b = \begin{bmatrix} \mathbf{b}_t \\ \mathbf{b}_0 \\ \mathbf{c}_0 - \mathbf{c}_t \end{bmatrix}, \boldsymbol{\xi}^c \in \mathbb{R}^{n_{g,t}}, \boldsymbol{\xi}^b \in \{-1, 1\}^{n_{b,t}} \right\} > 1. \quad (7)$$

Remark 5: Denote the number of continuous generators, binary generators and equality constraints of the HZ $\mathcal{O}$ (resp. $\mathcal{X}_0$) as $n_{g,o}$, $n_{b,o}$ and $n_{c,o}$ (resp. $n_{g,0}$, $n_{b,0}$ and $n_{c,0}$), respectively. The T MILPs in (7) include $n_{g,t}$ continuous variables, $n_{b,t}$ binary variables, and $n_{c,t}$ linear constraints, where $n_{g,t} = t \cdot (n_{g,x} + 4N_\pi) + n_{g,o} + n_{g,0}$, $n_{b,t} = t \cdot (n_{b,x} + N_\pi) + n_{b,o} + n_{b,0}$ and $n_{c,t} = t \cdot (n_{c,x} + 3N_\pi + n) + n_{c,o} + n_{c,0} + n$. Commercial solvers such as Gurobi have shown promising performance in solving MILPs. To further reduce the computation burden, we can use [18, Lemma 5] to get the tightest convex relaxation of the exact BRS $\mathcal{P}_t$ by replacing the binary generators with continuous generators. If relaxed BRSs are used in Proposition 1, (7) will degenerate into linear programs which are much easier to solve.

V. Simulation Examples

In this section, two simulation examples will be presented to demonstrate the effectiveness of the proposed method. The method proposed in this letter is implemented in MATLAB R2022a and executed on a desktop with an Intel Core i7-8700k CPU and 32GB of RAM.

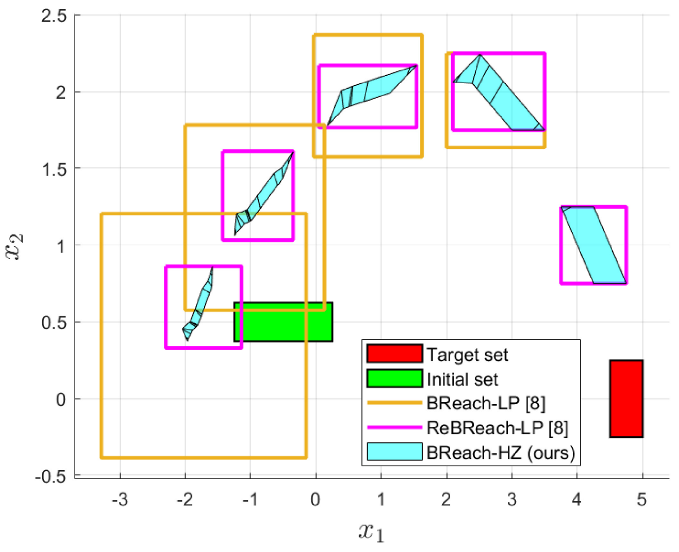

Fig. 2. Simulation results in Example 1. The exact BRSs computed by our HZ-based approach are shown in cyan. Over-approximated BRSs computed by BReach-LP and ReBReach-LP algorithms in [8] are bounded by orange and magenta lines, respectively. The target set as the unsafe region is in red and the initial set is in green.

Example 1: Consider the discrete-time double integrator model given in [6]: $\boldsymbol{x}(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \boldsymbol{u}(t)$. The NN controller $\boldsymbol{u}(t) = \pi(\boldsymbol{x}(t))$ has two hidden layers with ReLU activation functions and [10, 5] neurons. Similar to [8], this NN controller was trained using the dataset generated by an MPC controller. The saturation bounds $\underline{\boldsymbol{u}} = -1, \bar{\boldsymbol{u}} = 1$ was imposed on the controller, i.e., $\boldsymbol{u}(t) \in \mathcal{U} = [-1, 1]$. We chose the initial set as $\mathcal{X}_0 = [-1.25, 0.25] \times [0.4, 0.6]$, the unsafe region as $\mathcal{O} = [4.5, 5.0] \times [-0.25, 0.25]$, the state region as $\mathcal{X} = [-40, 40] \times [-40, 40]$, and $\alpha = \beta = 400$.

We denote our method based on Theorem 2 and equations (6) as BReach-HZ. To facilitate comparison with other BRS computation methods, we implement BReach-HZ to compute 5 exact BRSs $\mathcal{P}_1(\mathcal{T}), \ldots, \mathcal{P}_5(\mathcal{T})$ which are shown by the sets in cyan in Figure 2. The set complexity of the last-step BRS $\mathcal{P}_5(\mathcal{T})$ is given by: $n_g = 352$, $n_b = 85$ and $n_c = 265$. We also verified that condition (7) in Proposition 1 holds true, which implies the safety of the neural feedback system. The time for computing the BRSs is 0.0452 seconds, and the time for solving the MILPs given in (7) via the commercial solver Gurobi is 0.639 seconds. For comparison, we also ran the BReach-LP and ReBReach-LP algorithms proposed in [8], which were implemented in Python with default parameters provided by the authors of [8]. The times for computing the BRSs using BReach-LP and ReBReach-LP are 1.23 seconds and 11.7 seconds, respectively. The computed BRSs are shown by the rectangles with orange and magenta lines in Figure 2. It can be observed that our method provides more accurate BRSs for all the time steps compared with the BReach-LP and the ReBReach-LP algorithms. In addition, the exact BRSs computed by our method certify safety in this scenario, while the over-approximated BRSs computed by the two algorithms given in [8] lead to false unsafe detection.

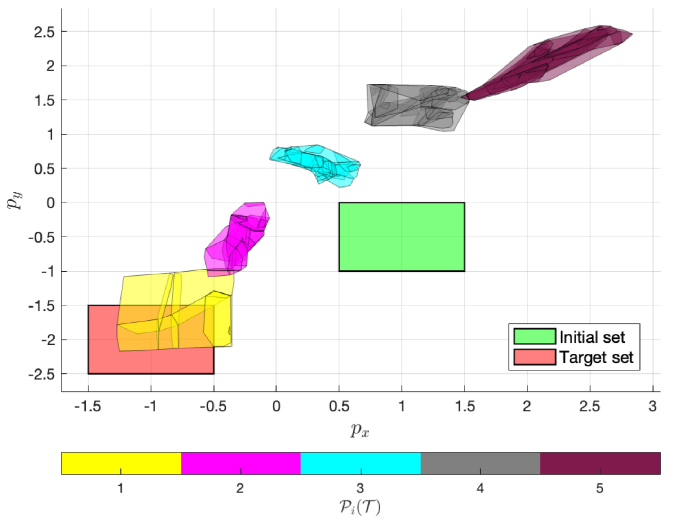

Fig. 3. Simulation results in Example 2. Note that there are set overlapping areas when projecting the BRSs onto the $x - y$ plane.

Example 2: Consider the 4-dimensional linearized ground robot model described by two integrators corresponding to the $x - y$ plane: $\boldsymbol{x}(t+1) = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{I}_2 \\ \boldsymbol{0} & \boldsymbol{I}_2 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0.5 \cdot \boldsymbol{I}_2 \\ \boldsymbol{I}_2 \end{bmatrix} \boldsymbol{u}(t)$, where $\boldsymbol{x} = [p_x, p_y, v_x, v_y]^\top$ is the position and velocity in the $x - y$ plane. The target set is $\mathcal{T} = [-1.5, -0.5] \times [-2.5, -1.5] \times [-0.1, 0.1] \times [-0.1, 0.1]$ and the initial set is $\mathcal{X}_0 = [0.5, 1.5] \times [-1, 0] \times [-1.1, -0.9] \times [-0.1, 0.1]$.

Similar to Example 1, we trained a ReLU-activated neural network comprising $[10, 5]$ neurons to learn an MPC policy while adhering to the saturation constraints of $\underline{\boldsymbol{u}} = -\boldsymbol{1}$ and $\overline{\boldsymbol{u}} = 1$. Figure 3 shows the projections of the computed exact BRSs $\mathcal{P}_1(\mathcal{T}), \ldots, \mathcal{P}_5(\mathcal{T})$ on the $x - y$ plane by using our proposed method. The set complexity of the last-step BRS $\mathcal{P}_5(\mathcal{T})$ is given by: $n_g = 404$, $n_b = 95$ and $n_c = 305$. The time for computing BRSs is 0.0116 seconds, and the time for solving MILPs to verify the safety is 0.745 seconds, which has the same order of magnitude as that in Example 1.

VI. Conclusion

We proposed a novel HZ-based approach to compute the exact BRSs of neural feedback systems. We showed that the input-output relationship of a ReLU-activated FNN can be exactly described by its graph set represented by an HZ. We provided an exact HZ formulation for the BRSs of neural feedback systems and extended the result to the saturated input case. We also proposed a sufficient and necessary condition in the form of MILPs for the safety verification of neural feedback systems via BRSs. The performance of the proposed approach was compared with state-of-the-art using two numerical examples.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Rep.*, 2015, pp. 1–8.

[2] S. Dutta, X. Chen, and S. Sankaranarayanan, "Reachability analysis for neural feedback systems using regressive polynomial rule inference," in *Proc. 22nd ACM Int. Conf. Hybrid Syst. Comput. Control*, 2019, pp. 157–168.

[3] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "ReachNN: Reachability analysis of neural-network controlled systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–22, 2019.

[4] M. Everett, G. Habibi, C. Sun, and J. P. How, "Reachability analysis of neural feedback loops," *IEEE Access*, vol. 9, pp. 163938–163953, 2021.

[5] H.-D. Tran et al., "NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems," in *Proc. 32nd Int. Conf. Comput.-Aided Verification*, 2020, pp. 3–17.

[6] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 1–15, Jan. 2022.

[7] Y. Zhang and X. Xu, "Safety verification of neural feedback systems based on constrained zonotopes," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 2737–2744.

[8] N. Rober, M. Everett, and J. P. How, "Backward reachability analysis for neural feedback loops," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 2897–2904.

[9] N. Rober et al., "Backward reachability analysis of neural feedback loops: Techniques for linear and nonlinear systems," *IEEE Open J. Control Syst.*, vol. 2, pp. 108–124, 2023.

[10] N. Rober, M. Everett, S. Zhang, and J. P. How, "A hybrid partitioning strategy for backward reachability of neural feedback loops," in *Proc. IEEE Amer. Control Conf.*, 2023, pp. 3523–3528.

[11] J. A. Vincent and M. Schwager, "Reachable polyhedral marching (RPM): A safety verification algorithm for robotic systems with deep neural network components," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 9029–9035.

[12] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[13] M. Althoff, G. Frehse, and A. Girard, "Set propagation techniques for reachability analysis," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 4, pp. 369–395, May 2021.

[14] L. Yang, H. Zhang, J.-B. Jeannin, and N. Ozay, "Efficient backward reachability using the Minkowski difference of constrained zonotopes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 11, pp. 3969–3980, Nov. 2022.

[15] T. J. Bird, H. C. Pangborn, N. Jain, and J. P. Koeln, "Hybrid zonotopes: A new set representation for reachability analysis of mixed logical dynamical systems," *Automatica*, vol. 154, Aug. 2023, Art. no. 111107.

[16] J. A. Siefert, T. J. Bird, J. P. Koeln, N. Jain, and H. C. Pangborn, "Robust successor and precursor sets of hybrid systems using hybrid zonotopes," *IEEE Control Syst. Lett.*, vol. 7, pp. 355–360, 2022.

[17] T. J. Bird, N. Jain, H. C. Pangborn, and J. P. Koeln, "Set-based reachability and the explicit solution of linear MPC using hybrid zonotopes," in *Proc. Amer. Control Conf.*, 2022, pp. 158–165.

[18] Y. Zhang and X. Xu, "Reachability analysis and safety verification of neural feedback systems via hybrid zonotopes," in *Proc. IEEE Amer. Control Conf.*, 2023, pp. 1915–1921.

[19] T. J. Bird and N. Jain, "Unions and complements of hybrid zonotopes," *IEEE Control Syst. Lett.*, vol. 6, pp. 1778–1783, 2021.

[20] T. J. Bird, *Hybrid Zonotopes: A Mixed-Integer Set Representation for the Analysis of Hybrid Systems*, Purdue Univ., West Lafayette, IN, USA, 2022.

[21] S. Keerthi and E. Gilbert, "Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints," *IEEE Trans. Autom. Control*, vol. AC-32, no. 5, pp. 432–435, May 1987.

[22] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Boston, MA, USA: Birkhäuser, 2008.

[23] T. Anevlavis and P. Tabuada, "Computing controlled invariant sets in two moves," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 6248–6254.

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

[25] J. Ortiz, A. Vellucci, J. Koeln, and J. Ruths, "Hybrid zonotopes exactly represent ReLU neural networks," 2023, *arXiv:2304.02755*.

[26] A. Rössig and M. Petkovic, "Advances in verification of ReLU neural networks," *J. Global Optim.*, vol. 81, pp. 109–152, Sep. 2021.

[27] C. Sidrane, A. Maleki, A. Irfan, and M. J. Kochenderfer, "Overt: An algorithm for safety verification of neural network control policies for nonlinear systems," *J. Mach. Learn. Res.*, vol. 23, no. 117, pp. 1–45, 2022.