Reachability Analysis and Safety Verification of Neural Feedback Systems via Hybrid Zonotopes

Yuhao Zhang and Xiangru Xu

***Abstract*— Hybrid zonotopes generalize constrained zonotopes by introducing additional binary variables and possess some unique properties that make them convenient to represent nonconvex sets. This paper presents novel hybrid zonotope-based methods for the reachability analysis and safety verification of neural feedback systems. Algorithms are proposed to compute the input-output relationship of each layer of a feed-forward neural network, as well as the exact reachable sets of neural feedback systems. It is shown that a ReLU-activated feed-forward neural network can be exactly represented by a hybrid zonotope. In addition, a sufficient and necessary condition is formulated as a mixed-integer linear program to certify whether the trajectories of a neural feedback system can avoid unsafe regions. The proposed approach is shown to yield a formulation that provides the tightest convex relaxation for the reachable sets of the neural feedback system. Complexity reduction techniques for the reachable sets are developed to balance the computation efficiency and approximation accuracy. Two numerical examples demonstrate the superior performance of the proposed approach compared to other existing methods.**

I. Introduction

Artificial neural networks have shown their extraordinary performance in many fields such as auto-driving systems [1] and mobile robots [2]. Implementation of neural networks in such controlled systems also raises safety concerns as even a small chance of failure may cause catastrophic consequences. Therefore, it is critical to find an efficient method to verify the safety properties of controlled systems with neural network components before real implementations. However, analyzing properties of neural networks is notoriously difficult due to their highly non-convex and nonlinear natures [3].

Various methods have been proposed to perform reachability analysis and safety verification for the *neural feedback systems* (i.e., feedback systems with neural network controllers) [4], [5], [6], [7], [8]. Based on quadratic constraints, a reachable set over-approximation method was proposed in [9], [10] using Semi-Definite Programming (SDP). A fast reachability method was introduced in [11] by relaxing the SDP into Linear Programming (LP). Learning-based reachability methods were also developed in [12], [13] for neural feedback systems with probabilistic guarantees on the correctness of the approximated reachable sets. Set-based methods were also proposed to compute the exact reachable sets of neural feedback systems using star sets [14] and constrained zonotopes [15]. Despite their interesting results, these two methods can only deal with convex set representations which limit their usage for complex safety verification problems. Besides, the computation complexity increases rapidly for deep neural networks.

Yuhao Zhang and Xiangru Xu are with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI, USA. Email: {yuhao.zhang2,xiangru.xu}@wisc.edu.

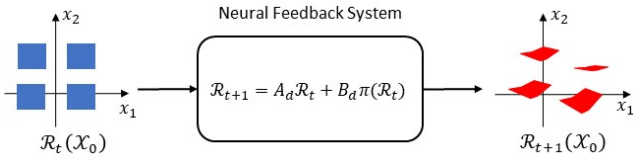

Fig. 1: The neural feedback system is given as $\boldsymbol{x}(t+1) = A_d\boldsymbol{x}(t) + B_d\boldsymbol{u}(t)$ where the state feedback controller $\boldsymbol{u}(t) = \pi(\boldsymbol{x}(t))$ is a given ℓ-layer FNN with the ReLU activation function. At each time step, the neural feedback system maps a hybrid zonotope as the input set to another hybrid zonotope as the output set. The initial set is $\mathcal{X}_0$ and the reachable set at time t from $\mathcal{X}_0$ is $\mathcal{R}_t(\mathcal{X}_0)$.

Recently, a new set representation named the *hybrid zonotope* was introduced in [16]. Through the addition of binary generators, hybrid zonotope can represent non-convex sets with flat faces. And the reachability analysis based on hybrid zonotopes will lead to the formulation of Mixed-Integer Linear Programs (MILPs), for which many state-of-the-art solvers such as Gurobi [17] and learning-based solver MLOPT [18] can be utilized to accelerate the computation.

In this work, we present hybrid zonotope-based methods for reachability analysis and safety verification of neural feedback systems with ReLU-activated Feed-forward Neural Network (FNN) controllers (see Figure 1). The contributions of this paper are fourfold: (i) Through analytical analysis, it is shown that a FNN with ReLU activation functions can be exactly represented by a hybrid zonotope; (ii) For neural feedback systems with hybrid zonotopes as the input sets, a novel approach is presented to compute the nonconvex exact reachable sets represented as hybrid zonotopes; (iii) Based on the convex relaxation property of the computed reachable sets and the properties of hybrid zonotopes, heuristic reduction methods are proposed to reduce the complexity growth of the hybrid zonotope sets; (iv) Using the computed reachable sets, an MILP-based condition is provided to verify the unsafe region avoidance of neural feedback systems, for which off-the-shelf solvers can be employed. The efficiency of the proposed methods is demonstrated through two numerical examples.

II. Preliminaries & Problem Statement

A. Hybrid Zonotopes

Definition 1: Let $\mathcal{Z}, \mathcal{Z}_c, \mathcal{Z}_h \subset \mathbb{R}^n$. $\mathcal{Z}$ is a *zonotope* if (1) holds [19], $\mathcal{Z}_c$ is a *constrained zonotope* if (2) holds [20],

and $\mathcal{Z}_h$ is a *hybrid zonotope* if (3) holds [16]:

$$\exists(\mathbf{G},\mathbf{c}) \in \mathbb{R}^{n\times n_g}\times\mathbb{R}^n : \mathcal{Z} = \{\mathbf{G}\boldsymbol{\xi}+\mathbf{c} \mid \|\boldsymbol{\xi}\|_\infty \le 1\}, \quad (1)$$

$$\exists(\mathbf{G},\mathbf{c},\mathbf{A},\mathbf{b}) \in \mathbb{R}^{n\times n_g}\times\mathbb{R}^n\times\mathbb{R}^{n_c\times n_g}\times\mathbb{R}^{n_c} : \mathcal{Z}_c = \{\mathbf{G}\boldsymbol{\xi}+\mathbf{c} \mid \|\boldsymbol{\xi}\|_\infty \le 1, \mathbf{A}\boldsymbol{\xi}=\mathbf{b}\}, \quad (2)$$

$$\exists(\mathbf{G}^c,\mathbf{G}^b,\mathbf{c},\mathbf{A}^c,\mathbf{A}^b,\mathbf{b}) \in \mathbb{R}^{n\times n_g}\times\mathbb{R}^{n\times n_b}\times\mathbb{R}^n\times\mathbb{R}^{n_c\times n_g} \times\mathbb{R}^{n_c\times n_b}\times\mathbb{R}^{n_c} : \quad (3)$$

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

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 notations of the zonotope, constrained zonotope and hybrid zonotope are given by $\mathcal{Z} = Z\langle\mathbf{G},\mathbf{c}\rangle$, $\mathcal{Z}_c = CZ\langle\mathbf{G},\mathbf{c},\mathbf{A},\mathbf{b}\rangle$, and $\mathcal{Z}_h = HZ\langle\mathbf{G}^c,\mathbf{G}^b,\mathbf{c},\mathbf{A}^c,\mathbf{A}^b,\mathbf{b}\rangle$, respectively.

Note that a hybrid zonotope degenerates into a constrained zonotope when $n_b = 0$, and a constrained zonotope degenerates into a zonotope when $n_c = 0$. A hybrid zonotope with $n_b > 0$ is equivalent to the union of 2^{n_b} corresponding constrained zonotopes [16, Theorem 5]. For a given hybrid zonotope, 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* (or simply *generators* if binary generators are not present). For simplicity, we define the set $\mathcal{B}(\mathbf{A}^c,\mathbf{A}^b,\mathbf{b}) = \{(\boldsymbol{\xi}^c,\boldsymbol{\xi}^b) \in \mathcal{B}_\infty^{n_g}\times\{-1,1\}^{n_b} \mid \mathbf{A}^c\boldsymbol{\xi}^c+\mathbf{A}^b\boldsymbol{\xi}^b=\mathbf{b}\}$. We denote $\mathbf{G}[:,i]$ as the i-th column of a matrix $\mathbf{G}$. The complexity of a hybrid zonotope is described by its *degrees-of-freedom order* or simply *order* $o_h = (n_g+n_b-n_c)/n$.

Identities to compute the linear map, intersection and union operation of hybrid zonotopes are given in [16, Proposition 7] and [21, Proposition 1]. The emptiness of a hybrid zonotope can be checked by solving an MILP [16].

Lemma 1: Given $\mathcal{Z}_h = HZ\langle\mathbf{G}^c,\mathbf{G}^b,\mathbf{c},\mathbf{A}^c,\mathbf{A}^b,\mathbf{b}\rangle \subset \mathbb{R}^n$, $\mathcal{Z}_h \neq \emptyset$ 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 a discrete-time linear system:

$$\boldsymbol{x}(t+1) = A_d\boldsymbol{x}(t)+B_d\boldsymbol{u}(t) \quad (4)$$

where $\boldsymbol{x}(t)\in\mathbb{R}^n$, $\boldsymbol{u}(t)\in\mathbb{R}^m$ are the state and the control input. $A_d\in\mathbb{R}^{n\times n}$, $B_d\in\mathbb{R}^{n\times m}$ are the state matrix and the input matrix, respectively.

We assume a state-feedback controller $\boldsymbol{u}(t) = \pi(\boldsymbol{x}(t))$, which is parameterized by an ℓ-layer FNN with the Rectified Linear Unit (ReLU) activation function. The closed-loop system is denoted as:

$$\boldsymbol{x}(t+1) = \boldsymbol{f}_{cl}(\boldsymbol{x}(t)) \triangleq A_d\boldsymbol{x}(t)+B_d\pi(\boldsymbol{x}(t)). \quad (5)$$

For the closed-loop system (5), we denote $\mathcal{R}_t(\mathcal{X}_0) \triangleq \{\boldsymbol{x}(t)\in\mathbb{R}^n \mid \boldsymbol{x}(0)\in\mathcal{X}_0, \boldsymbol{x}(k+1) = \boldsymbol{f}_{cl}(\boldsymbol{x}(k)), k = 0,1,\dots,t-1\}$ the (forward) reachable set at time t from a given set of initial conditions $\mathcal{X}_0\subset\mathbb{R}^n$.

For the ℓ-layer FNN controller, let $\mathbf{W}^{(k-1)}$ be the k-th layer weight matrix and $\mathbf{v}^{(k-1)}$ be the k-th layer bias vector, for $k = 1,\dots,\ell$. Denote $\boldsymbol{x}^{(k)}$ as the neurons of the k-th layer, then, for $k = 1,\dots,\ell-1$, we have

$$\boldsymbol{x}^{(k)} = ReLU(\mathbf{W}^{(k-1)}\boldsymbol{x}^{(k-1)}+\mathbf{v}^{(k-1)}) \quad (6)$$

where $\boldsymbol{x}^{(0)} = \boldsymbol{x}(t)$ and $ReLU(\boldsymbol{x}) = \max\{0,\boldsymbol{x}\}$. Only the linear map is applied in the last layer, i.e., $\pi(\boldsymbol{x}(t)) = \boldsymbol{x}^{(\ell)} = \mathbf{W}^{(\ell-1)}\boldsymbol{x}^{(\ell-1)}+\mathbf{v}^{(\ell-1)}$.

We assume the initial set and the unsafe set for the closed-loop system (7) are both represented by hybrid zonotopes. In this paper, we will investigate the following two problems.

Problem 1: (Reachability analysis) Given an initial set $\mathcal{X}_0$ that is represented as a hybrid zonotope, the parameters of the FNN controller π and a time horizon $T\in\mathbb{Z}_{>0}$, compute the reachable set $\mathcal{R}_t(\mathcal{X}_0)$ for the closed-loop system (5) where $t = 1,\dots,T$.

Problem 2: (Safety verification) Given unsafe set $\mathcal{O}$ represented by a hybrid zonotope, verify whether the state trajectories of the closed-loop system (5) can avoid the unsafe region for $t = 1,\dots,T$.

III. Exact Reachability analysis and Safety Verification

In this section, we consider Problem 1 and Problem 2 for the closed-loop system with an FNN controller as in (5).

A. Output Analysis of Standalone FNN

Firstly, we will present an algorithm to compute the exact output set of a given FNN as in (6) with an input set represented as a hybrid zonotope.

From the definition of the FNN in (6), the output of layer k is the input of layer $k+1$, for $k = 1,\dots,\ell-1$. Therefore, the output set of an FNN can be derived layer by layer and we will focus on finding the input-output relationship for one layer. Using Proposition 7 in [16], we can pass an input set as a hybrid zonotope $\mathcal{Z}_h = HZ\langle\mathbf{G}^c,\mathbf{G}^b,\mathbf{c},\mathbf{A}^c,\mathbf{A}^b,\mathbf{b}\rangle$ through a linear map as $\mathbf{W}\mathcal{Z}_h+\mathbf{v} = HZ\langle\mathbf{W}\mathbf{G}^c,\mathbf{W}\mathbf{G}^b,\mathbf{W}\mathbf{c}+\mathbf{v},\mathbf{A}^c,\mathbf{A}^b,\mathbf{b}\rangle$. Thus, the only difficulty remaining is to find the output of a ReLU activation function for a hybrid zonotope.

Inspired by the output analysis algorithm for FNN using the star sets representation in [22], we present Algorithm 1 to compute the exact output set for one layer of FNN using hybrid zonotopes. Given the weight matrix $\mathbf{W}^{(k-1)}$ and bias vector $\mathbf{b}^{(k-1)}$ for the k-th layer ($k = 1,\dots,\ell-1$), Line 2 computes the linear map of the input set $\mathcal{Z}_h$. Recall that $\boldsymbol{x}^{(k)}$ denotes the neurons of the k-th layer. Based on the definition, we know the ReLU activation function will only change the value of a neuron if it is negative. Thus, Line 3-4 of Algorithm 1 is used to compute the range of each neuron and identify the neurons with lower-bound in the negative half-space. Then, StepReLU function in Algorithm 1 applies the ReLU activation function on each of the identified neurons. Note that in Line 13-14, $\mathcal{H}_-^i = \{\boldsymbol{x}\in\mathbb{R}^n \mid \mathbf{e}_i^T\boldsymbol{x}\le 0\}$ and $\mathcal{H}_+^i = \{\boldsymbol{x}\in\mathbb{R}^n \mid \mathbf{e}_i^T\boldsymbol{x}\ge 0\}$ denote the half-spaces with i-th canonical vector $\mathbf{e}_i$, for $i = 1,\dots,n$. Algorithm

1 reveals that when the input set $\mathcal{Z}_h$ to the FNN π is a hybrid zonotope, the exact output of the FNN can also be represented as a hybrid zonotope.

Algorithm 1: Exact output analysis for k-th layer of FNN via hybrid zonotopes

Input: weight matrix $\mathbf{W}^{(k-1)}$, bias vector $\mathbf{v}^{(k-1)}$, hybrid zonotope input sets $\mathcal{Z}_h$
Output: exact output set $\mathcal{R}$ as a hybrid zonotope

Function $\mathcal{R}$ = `ReachNN(`$\mathcal{Z}_h$,$\mathbf{W}^{(k-1)}$,$\mathbf{v}^{(k-1)}$`)`**:**
  $\mathcal{R} = \mathbf{W}^{(k-1)}\mathcal{Z}_h + \mathbf{v}^{(k-1)}$; `// linear map`
  $[lb \quad up] \leftarrow$ range of $\boldsymbol{x}^{(k)}$ in $\mathcal{I}$; `// MILP`
  $map = find(lb < 0)$;
  for i *in* map **do**
    $\mathcal{R} =$ `StepReLU(`$\mathcal{R}, i, lb[i], up[i]$`)` ;
  return $\mathcal{R}$
Function $\tilde{\mathcal{R}}$ = `StepReLU(`$\mathcal{R}$,i,lb_i,up_i`)`**:**
  $\mathbf{E}_i = [\mathbf{e}_1 \ \cdots \ \mathbf{e}_{i-1} \ \mathbf{0} \ \mathbf{e}_{i+1} \ \cdots \ \mathbf{e}_{n_I}]$;
  if $up_i \leq 0$ **then**
    $\mathcal{I} = \mathbf{E}_i\mathcal{R}$; `// linear map`
  if $lb_i < 0$ & $up_i > 0$ **then**
    $\mathcal{I}_+ = \mathcal{R} \cap \mathcal{H}^i_+$; $\mathcal{I}_- = \mathcal{R} \cap \mathcal{H}^i_-$;
    $\mathcal{I} = \mathcal{I}_+ \cup \mathbf{E}_i\mathcal{I}_-$;
  return $\tilde{\mathcal{R}} = \mathcal{I}$

Algorithm 2: Compute $\mathbf{G}_1, \mathbf{G}_2$ in Theorem 1

Input: continuous generator matrix $\mathbf{G}^c \in \mathbb{R}^{n\times n_g}$ and binary generator matrix $\mathbf{G}^b \in \mathbb{R}^{n\times n_b}$ from hybrid zonotope $\mathcal{Z}_h$, n_b^π - the number of binary generators of $\mathcal{Z}_h^\pi$
Output: matrices $\mathbf{G}_1, \mathbf{G}_2$

$\mathbf{G}_1 \longleftarrow \mathbf{G}^c$; $\mathbf{G}_2 \longleftarrow \mathbf{G}^b$;
$k \longleftarrow \log_2(n_b^\pi + 1) - \log_2(n_b + 1)$;
repeat
  $\mathbf{G}_1 \longleftarrow [\mathbf{G}_1 \ \mathbf{0}_{n\times 1}]$;
  $\mathbf{G}_1 \longleftarrow [\mathbf{G}_1 \ \mathbf{G}_1]$; $\mathbf{G}_2 \longleftarrow [\mathbf{G}_2 \ \mathbf{G}_2]$;
  $m \longleftarrow 2*$(# columns of $\mathbf{G}_1$+# columns of $\mathbf{G}_2$);
  $\mathbf{G}_1 \longleftarrow [\mathbf{G}_1 \ \mathbf{0}_{n\times m}]$; $\mathbf{G}_2 \longleftarrow [\mathbf{G}_2 \ \mathbf{0}_{n\times 1}]$;
  $k \longleftarrow k - 1$;
until $k \leq 0$;
return $\mathbf{G}_1, \mathbf{G}_2$

B. Exact Reachable Set for Neural Feedback System

Next, we consider the reachability analysis for the closed-loop system (5). Recall that $\boldsymbol{f}_{cl}(\boldsymbol{x}) = A_d\boldsymbol{x} + B_d\pi(\boldsymbol{x})$. Note that a conservative over-approximation of the exact reachable set can be obtained by trivially adding the two terms of $\boldsymbol{f}_{cl}$ with the Minkowski sum. The following theorem provides the *exact* form of $\boldsymbol{f}_{cl}(\mathcal{Z}_h) = \{\boldsymbol{f}_{cl}(\boldsymbol{x}) | \boldsymbol{x} \in \mathcal{Z}_h\}$ for a given hybrid zonotope $\mathcal{Z}_h$.

Theorem 1: Given any hybrid zonotope $\mathcal{Z}_h = HZ\langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle \subset \mathbb{R}^n$ where $\mathbf{G}^c \in \mathbb{R}^{n\times n_g}$, $\mathbf{G}^b \in \mathbb{R}^{n\times n_b}$, $\mathbf{A}^c \in \mathbb{R}^{n_c\times n_g}$ and $\mathbf{A}^b \in \mathbb{R}^{n_c\times n_b}$, let $\pi(\mathcal{Z}_h) = HZ\langle \mathbf{G}^c_\pi, \mathbf{G}^b_\pi, \mathbf{c}_\pi, \mathbf{A}^c_\pi, \mathbf{A}^b_\pi, \mathbf{b}_\pi\rangle \triangleq \mathcal{Z}_h^\pi$ be the computed output set using Algorithm 1. Then, $\boldsymbol{f}_{cl}(\mathcal{Z}_h) = HZ\langle \mathbf{G}^c_{cl}, \mathbf{G}^b_{cl}, \mathbf{c}_{cl}, \mathbf{A}^c_{cl}, \mathbf{A}^b_{cl}, \mathbf{b}_{cl}\rangle \triangleq \mathcal{Z}_h^{cl}$, where

$$\mathbf{G}^c_{cl} = A_d\mathbf{G}_1 + B_d\mathbf{G}^c_\pi,\ \mathbf{G}^b_{cl} = A_d\mathbf{G}_2 + B_d\mathbf{G}^b_\pi,$$
$$\mathbf{c}_{cl} = A_d(\mathbf{c} + (\frac{n_b^\pi + 1}{n_b + 1} - 1)\mathbf{G}^b\mathbf{1}) + B_d\mathbf{c}_\pi,$$
$$\mathbf{A}^c_{cl} = \mathbf{A}^c_\pi,\ \mathbf{A}^b_{cl} = \mathbf{A}^b_\pi,\ \mathbf{b}_{cl} = \mathbf{b}_\pi,$$

matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are given by Algorithm 2, n_b and n_b^π are the numbers of binary generators of $\mathcal{Z}_h$ and $\mathcal{Z}_h^\pi$, respectively.

The detailed proof of Theorem 1 is omitted due to space limitation and can be found in [23]. Note that the construction of matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ is used to preserve the mapping implied by the closed-loop system (5).

Based on Theorem 1, the exact reachable sets of closed-loop system (5) can be computed as follows:

$$\mathcal{R}_0 = \mathcal{X}_0,\ \mathcal{R}_t = \boldsymbol{f}_{cl}(\mathcal{R}_{t-1}),\ t = 1, \ldots, T. \tag{7}$$

The reachable sets computed by (7) are *exact* as long as the initial set can be represented by a hybrid zonotope. The price of accuracy, however, is that the complexity order (i.e., the numbers of continuous and binary generators - n_g and n_b) of the hybrid zonotope reachable sets will grow exponentially. If n_π is the total number of neurons in π, then, in the worst case, n_b will increase in the order of $2^{n_\pi} - 1$ and n_g will increase in the order of $4^{n_\pi} - 1$. Thus, complexity reduction techniques are needed to reduce the computation burden, which will be introduced in the next section.

Remark 1: In our prior work [15], a method based on constrained zonotopes was proposed to compute exact reachable sets of neural feedback systems. Different from the exact reachability analysis in this section, the input set considered in [15] is limited to a single constrained zonotope, which is unable to represent *non-convex sets* as the hybrid zonotope does. Although one may convert the unions of constrained zonotopes into hybrid zonotopes using Proposition 1 in [21], this will result in a set with a larger complexity order as it will take much more union operations than Algorithm 1 of this work. Numerical comparisons of these two methods will be demonstrated by examples in Section V.

Remark 2: Although only linear feedback systems are considered in this work, the proposed approach can be readily extended to general nonlinear feedback systems by abstracting nonlinear dynamics with a set of optimally tight piecewise linear bounds as in [24].

C. Safety Verification

Denote the exact reachable set from initial set $\mathcal{X}_0$ at time t computed by (7) be $\mathcal{R}_t(\mathcal{X}_0) = HZ\langle \mathbf{G}^c_t, \mathbf{G}^b_t, \mathbf{c}_t, \mathbf{A}^c_t, \mathbf{A}^b_t, \mathbf{b}_t\rangle$ for $t = 1, \ldots, T$. Assume the unsafe region is represented by a hybrid zonotope $\mathcal{O} = HZ\langle \mathbf{G}^c_o, \mathbf{G}^b_o, \mathbf{c}_o, \mathbf{A}^c_o, \mathbf{A}^b_o, \mathbf{b}_o\rangle$. The following result provides a sufficient and necessary condition on the safety verification of the closed-loop system (5).

Proposition 1: Given the reachable sets $\mathcal{R}_1, \dots, \mathcal{R}_T$ and unsafe set $\mathcal{O}$ defined above, the state trajectories of the closed-loop system (5) will not enter the unsafe region if and only if the following condition is satisfied for $t \in \{1, \dots, T\}$:

$$\min \left\{ \|\boldsymbol{\xi}^c\|_\infty \left| \begin{bmatrix} \mathbf{A}_t^c & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_o^c \\ \mathbf{G}_t^c & -\mathbf{G}_o^c \end{bmatrix} \boldsymbol{\xi}^c + \begin{bmatrix} \mathbf{A}_t^b & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_o^b \\ \mathbf{G}_t^b & -\mathbf{G}_o^b \end{bmatrix} \boldsymbol{\xi}^b \right. \right. \\ \left. \left. = \begin{bmatrix} \mathbf{b}_t \\ \mathbf{b}_o \\ \mathbf{c}_o - \mathbf{c}_t \end{bmatrix}, \boldsymbol{\xi}^c \in \mathbb{R}^{n_{g,t}}, \boldsymbol{\xi}^b \in \{-1, 1\}^{n_{b,t}} \right\} > 1. \quad (8)$$

Avoiding unsafe regions can be equivalently expressed as none of the reachable sets intersect with the unsafe set. Since Proposition 1 is a straight-forward application of Proposition 7 in [16] and Lemma 1, the proof is omitted due to space limitation.

Remark 3: The safety verification problem is formulated as T MILPs (8) with $n_{g,t}$ continuous variables and $n_{b,t}$ binary variables. Although MILPs are well known to be NP-hard problems in general, some common commercial MILP solvers such as Gurobi [17] have shown promising performance in both average solving time and wide ranges of solvable problems. The fast development of these MILP solvers enables us to incorporate these off-the-shelf tools into our verification problem.

IV. Complexity Reduction

Due to the intersection and union operation in Algorithm 1, both the number of continuous generators and the number of binary generators will increase fast. As mentioned in Section III-B, in the worst case, these two numbers will grow exponentially which makes Theorem 1 computationally heavy. In this section, we will introduce two order reduction techniques that can provide over-approximated reachable sets with fewer continuous and binary generators.

A. Reducing the Number of Binary Generators

Given a hybrid zonotope, it is possible that the set can be represented by another hybrid zonotope with fewer binary generators. A rigorous approach is proposed in [16] to remove the redundant binary generators by exploring the independent feasible solutions for the binary variable $\boldsymbol{\xi}^b$. Although this approach can reduce the complexity of the hybrid zonotope representation, one major limitation is that it can not further reduce the binary generators without altering the set. In this subsection, however, we explore the relationship between the union and convex hull operations of hybrid zonotopes, and provide a novel method to reduce the number of binary generators while guaranteeing an over-approximation.

Let's first consider two constrained zonotopes $\mathcal{Z}_c = CZ\langle \mathbf{G}_z, \mathbf{c}_z, \mathbf{A}_z, \mathbf{b}_z \rangle \subset \mathbb{R}^n$ and $\mathcal{W}_c = CZ\langle \mathbf{G}_w, \mathbf{c}_w, \mathbf{A}_w, \mathbf{b}_w \rangle \subset \mathbb{R}^n$.

For any set $\mathcal{X} \subset \mathcal{R}^n$, we denote the convex hull of $\mathcal{X}$ as $conv(\mathcal{X})$ [25]. According to Theorem 5 in [26], we can compute the convex hull of $\mathcal{Z} \cup \mathcal{W}$ as a constrained zonotope $\mathcal{C}_c = conv(\mathcal{Z}_c \cup \mathcal{W}_c) = CZ\langle \mathbf{G}_{co}, \mathbf{c}_{co}, \mathbf{A}_{co}, \mathbf{b}_{co} \rangle$ where

$$\mathbf{G}_{co} = \begin{bmatrix} \mathbf{G}_z & \mathbf{G}_w & \frac{\mathbf{c}_z - \mathbf{c}_w}{2} & \mathbf{0} \end{bmatrix}, \ \mathbf{c}_{co} = \frac{\mathbf{c}_z + \mathbf{c}_w}{2}, \quad (9)$$

$$\mathbf{A}_{co} = \begin{bmatrix} \mathbf{A}_z & \mathbf{0} & -\frac{\mathbf{b}_z}{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_w & \frac{\mathbf{b}_w}{2} & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{A}_{3,1} & I \end{bmatrix}, \ \mathbf{b}_{co} = \begin{bmatrix} \frac{1}{2}\mathbf{b}_z \\ \frac{1}{2}\mathbf{b}_w \\ -\frac{1}{2}\mathbf{1} \end{bmatrix}, \quad (10)$$

$$\mathbf{A}_{3,1} = \begin{bmatrix} I \\ -I \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \mathbf{A}_{3,2} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ I \\ -I \end{bmatrix}, \ \mathbf{A}_{3,0} = \begin{bmatrix} -\frac{1}{2}\mathbf{1} \\ -\frac{1}{2}\mathbf{1} \\ \frac{1}{2}\mathbf{1} \\ \frac{1}{2}\mathbf{1} \end{bmatrix}. \quad (11)$$

According to Proposition 1 in [21], we can compute the union of $\mathcal{Z}_c$ and $\mathcal{W}_c$ as a hybrid zonotope: $\mathcal{U}_h = \mathcal{Z}_c \cup \mathcal{W}_c = HZ\langle \mathbf{G}_u^c, \mathbf{G}_u^b, \mathbf{c}_u, \mathbf{A}_u^c, \mathbf{A}_u^b, \mathbf{b}_u \rangle$ where

$$\mathbf{G}_u^c = \begin{bmatrix} \mathbf{G}_z & \mathbf{G}_w & \mathbf{0} \end{bmatrix}, \mathbf{G}_u^b = \frac{\mathbf{c}_z - \mathbf{c}_w}{2}, \mathbf{c}_u = \frac{\mathbf{c}_z + \mathbf{c}_w}{2}, \quad (12)$$

$$\mathbf{A}_u^c = \begin{bmatrix} \mathbf{A}_z & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_w & \mathbf{0} \\ & \mathbf{A}_3^c & I \end{bmatrix}, \mathbf{A}_u^b = \begin{bmatrix} \frac{-\mathbf{b}_z}{2} \\ \frac{\mathbf{b}_w}{2} \\ \mathbf{A}_3^b \end{bmatrix}, \mathbf{b}_u = \begin{bmatrix} \frac{\mathbf{b}_z}{2} \\ \frac{\mathbf{b}_w}{2} \\ \mathbf{b}_3 \end{bmatrix}, \quad (13)$$

$$\mathbf{A}_3^c = \begin{bmatrix} I & \mathbf{0} \\ -I & \mathbf{0} \\ \mathbf{0} & I \\ \mathbf{0} & -I \end{bmatrix}, \mathbf{A}_3^b = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \frac{-1}{2}\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \frac{-1}{2}\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \frac{1}{2}\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \frac{1}{2}\mathbf{1} \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} \frac{-1}{2}\mathbf{1} \\ \frac{-1}{2}\mathbf{1} \\ \frac{-1}{2}\mathbf{1} \\ \frac{-1}{2}\mathbf{1} \end{bmatrix}. \quad (14)$$

The relationship between the union and convex hull of two constrained zonotopes is summarized below.

Lemma 2: Consider two constrained zonotopes $\mathcal{Z}_c, \mathcal{W}_c \subset \mathbb{R}^n$, and let $\mathcal{U}_h = \mathcal{Z}_c \cup \mathcal{W}_c$ be a hybrid zonotope computed as in (12)-(14) and $\mathcal{C}_c = conv(\mathcal{Z}_c \cup \mathcal{W}_c)$ be a constrained zonotope computed as in (9)-(11). If the binary variable constraint in $\mathcal{U}_h$ is relaxed to continuous variable constraint, i.e., replace $\xi^b \in \{-1, 1\}$ with $\xi^b \in [-1, 1]$, to get a relaxed constrained zonotope $\mathcal{U}_c$, then $\mathcal{U}_c$ is equivalent to $\mathcal{C}_c$.

Proof: It is easy to check that $\mathbf{G}_{co} = [\mathbf{G}_z \ \ \mathbf{G}_w \ \ \mathbf{G}_u^b \ \ \mathbf{0}]$ and $\mathbf{A}_{co} = \begin{bmatrix} \mathbf{A}_z & \mathbf{0} & -\frac{\mathbf{b}_z}{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_w & \frac{\mathbf{b}_w}{2} & \mathbf{0} \\ & \mathbf{A}_3^c & \mathbf{b}_3 & I \end{bmatrix}$. Therefore, we have $\mathcal{C}_c = CZ\langle \mathbf{G}_{co}, \mathbf{c}_{co}, \mathbf{A}_{co}, \mathbf{b}_{co} \rangle = CZ\langle \begin{bmatrix} \mathbf{G}_u^c & \mathbf{G}_u^b \end{bmatrix}, \mathbf{c}_u, \begin{bmatrix} \mathbf{A}_u^c & \mathbf{A}_u^b \end{bmatrix}, \mathbf{b} \rangle = \mathcal{U}_c$. ∎

The following theorem extends Lemma 2 to the reachable sets computed by (7).

Theorem 2: Given any hybrid zonotope $\mathcal{Z}_h = \langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b} \rangle$ from the reachable set computation (7), the convex hull of $\mathcal{Z}_h$ can be constructed as the constrained zonotope $\mathcal{Z}_c = CZ\langle \begin{bmatrix} \mathbf{G}^c & \mathbf{G}^b \end{bmatrix}, \mathbf{c}, \begin{bmatrix} \mathbf{A}^c & \mathbf{A}^b \end{bmatrix}, \mathbf{b} \rangle$, i.e., $\mathcal{Z}_c = conv(\mathcal{Z}_h)$.

Proof: Using Theorem 5 in [16], we know that $\mathcal{Z}_h$ can be represented by the union of a finite number of constrained zonotopes. Without loss of generality, assume $\mathcal{Z}_h = \mathcal{Z}_{c,1} \cup \mathcal{Z}_{c,2} \cup \cdots \cup \mathcal{Z}_{c,N}$ with $\mathcal{Z}_{c,i}, , i = 1, \dots, N$ being constrained zonotopes. It can be observed that using Lemma 2, we can eliminate one binary variable each time by replacing the union of two constrained zonotopes with their convex hull. Based on properties of convex hull, by repeating the same procedure for $N - 1$ times, we can get $\mathcal{Z}_c = CZ\langle \begin{bmatrix} \mathbf{G}^c & \mathbf{G}^b \end{bmatrix}, \mathbf{c}, \begin{bmatrix} \mathbf{A}^c & \mathbf{A}^b \end{bmatrix}, \mathbf{b} \rangle = conv(\cdots conv(\mathcal{Z}_{c,1} \cup \mathcal{Z}_{c,2}) \cdots \cup \mathcal{Z}_{c,N}) = conv(\mathcal{Z}_h)$. ∎

Remark 4: Note that Theorem 2 is not true for an arbitrary hybrid zonotope. Although for any hybrid zonotope, relaxing the binary constraints into linear constraints leads to an over-approximation of the hybrid zonotope, it is not guaranteed to be the tightest convex relaxation. However, Theorem 2 shows that our reachable set formulation computed by (7) can provide the *tightest* convex relaxation of the neural feedback systems with ReLU-activated FNN controllers. This property is similar to the *ideal formulation* for MILPs in [27].

Using Theorem 2, we can reduce the desired number of binary generators of a hybrid zonotope by replacing them with the same number of continuous generators. For example, given a hybrid zonotope $\mathcal{Z}_h$ with n_g continuous generators and n_b binary generators, we can reduce $\hat{n}_b$ binary generators and get an over-approximated hybrid zonotope $\hat{\mathcal{Z}}_h$ with $n_g + \hat{n}_b$ continuous generators and $n_b - \hat{n}_b$ binary generators. When $\hat{n}_b = n_b$, $\hat{\mathcal{Z}}_h$ becomes a constrained zonotope which is also the convex hull of $\mathcal{Z}_h$.

B. Reducing the Number of Continuous Generators

In this subsection, we introduce two methods to reduce the number of continuous generators. For a zonotope, generator reduction can be done by identifying parallel generators and combining parallel generators through addition [28]. The same approach can be applied to a constrained zonotope $\mathcal{Z}_c = CZ\langle \mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\rangle$ if the lifted zonotope $\mathcal{Z}^+ = Z\langle \begin{bmatrix}\mathbf{G}\\ \mathbf{A}\end{bmatrix}, \begin{bmatrix}\mathbf{c}\\ \mathbf{b}\end{bmatrix}\rangle \triangleq Z\langle \mathbf{G}^+, \mathbf{c}^+\rangle$ has parallel generators $\mathbf{G}^+[:, i] \,||\, \mathbf{G}^+[:, j]$ [20].

This *lift-then-reduce* strategy can also be extended to hybrid zonotopes. The following proposition is inspired by similar results for constrained zonotopes in [20].

Proposition 2: Consider a hybrid zonotope $\mathcal{Z}_h = HZ\langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle \subset \mathbb{R}^n$ and a partition $\begin{bmatrix}\mathbf{A}^c & \mathbf{A}^b & \mathbf{b}\end{bmatrix} = \begin{bmatrix}\mathbf{A}^c_1 & \mathbf{A}^b_1 & \mathbf{b}_1\\ \mathbf{A}^c_2 & \mathbf{A}^b_2 & \mathbf{b}_2\end{bmatrix}$. For every $\boldsymbol{z} \in \mathbb{R}^n$, $\boldsymbol{z} \in \mathcal{Z}_h$ if and only if $\begin{bmatrix}\boldsymbol{z}\\ \mathbf{0}\end{bmatrix} \in \mathcal{Z}_h^+ \triangleq HZ\left\langle \begin{bmatrix}\mathbf{G}^c\\ \mathbf{A}^c_1\end{bmatrix}, \begin{bmatrix}\mathbf{G}^b\\ \mathbf{A}^b_1\end{bmatrix}, \begin{bmatrix}\mathbf{c}\\ -\mathbf{b}_1\end{bmatrix}, \mathbf{A}^c_2, \mathbf{A}^b_2, \mathbf{b}_2\right\rangle$.

For a hybrid zonotope $\mathcal{Z}_h = HZ\langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle$, we can form a lifted hybrid zonotope $\mathcal{Z}_h^+$ by Proposition 2:

$$\mathcal{Z}_h^+ = HZ\left\langle \begin{bmatrix}\mathbf{G}^c\\ \mathbf{A}^c\end{bmatrix}, \begin{bmatrix}\mathbf{G}^b\\ \mathbf{A}^b\end{bmatrix}, \begin{bmatrix}\mathbf{c}\\ -\mathbf{b}\end{bmatrix}, \emptyset, \emptyset, \emptyset\right\rangle. \tag{15}$$

It is obvious that this lifted hybrid zonotope $\mathcal{Z}_h^+$ is equivalent to a union of lifted zonotopes with the same group of generators and shifted centers, i.e., $\mathcal{Z}_h = \mathcal{Z}_1 \cup \mathcal{Z}_2 \cup \cdots \cup \mathcal{Z}_{2^{n_b}}$, where $\mathcal{Z}_i = Z\left\langle \begin{bmatrix}\mathbf{G}^c\\ \mathbf{A}^c\end{bmatrix}, \begin{bmatrix}\mathbf{c} + \mathbf{G}^b\boldsymbol{\xi}^b_i\\ -\mathbf{b} + \mathbf{A}^b\boldsymbol{\xi}^b_i\end{bmatrix}\right\rangle$ and $\boldsymbol{\xi}^b_i \in \{-1, 1\}^{n_b}$ for $i = 1, \ldots, 2^{n_b}$.

Therefore, if there exist parallel generators for any of the lifted zonotopes, all the other lifted zonotopes have the same set of parallel generators. We then combine the parallel generators for the lifted zonotopes and use Proposition 2 to transform the reduced lifted zonotopes back to a reduced hybrid zonotope with fewer continuous generators.

The approach described above can be used to remove the continuous generators based on the generator directions. In what follows, we provide another method to reduce a continuous generator and an equality constraint at the same time. The following proposition extends Proposition 5 in [20] and is required in the complexity reduction algorithm.

Proposition 3: Let $\mathcal{Z}_h = HZ\langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle$. The set $\tilde{\mathcal{Z}}_h \triangleq HZ\langle \mathbf{G}^c - \Lambda_G\mathbf{A}^c, \mathbf{G}^b - \Lambda_G\mathbf{A}^b, \mathbf{c} + \Lambda_G\mathbf{b}, \mathbf{A}^c - \Lambda_A\mathbf{A}^c, \mathbf{A}^b - \Lambda_A\mathbf{A}^b, \mathbf{b} - \Lambda_A\mathbf{b}\rangle$ satisfies $\mathcal{Z}_h \subseteq \tilde{\mathcal{Z}}_h$ for every $\Lambda_G \in \mathbb{R}^{n\times n_c}$ and $\Lambda_A \in \mathbb{R}^{n_c\times n_c}$.

The proof of Proposition 3 is omitted due to space limitation. Proposition 3 allows us to choose any Λ_A and Λ_G to get an over-approximation of a hybrid zonotope. Next, we will introduce a heuristic approach to select proper Λ_A and Λ_G that leads to a less conservative over-approximation.

Consider the equality constraint of the hybrid zonotope $\mathbf{A}^c\boldsymbol{\xi}^c + \mathbf{A}^b\boldsymbol{\xi}^b = \mathbf{b}$. The i-th row ($i = 1, \ldots, n_c$) of the constraint can be written as

$$\sum_{j=1,\ldots,n_g} \mathbf{A}^c[i, j]\boldsymbol{\xi}^c[j] + \mathbf{A}^b[i, j]\boldsymbol{\xi}^b[j] = \mathbf{b}[i]. \tag{16}$$

Following the procedure in [20], choose

$$\Lambda_G = \mathbf{G}^c\mathbf{E}_{c,r}(\mathbf{A}^c[r, c])^{-1}, \Lambda_A = \mathbf{A}^c\mathbf{E}_{c,r}(\mathbf{A}^c[r, c])^{-1}, \tag{17}$$

where $\mathbf{E}_{c,r} \in \mathbb{R}^{n_g\times n_c}$ is zero except for a one in the (c, r) position and $\mathbf{A}^c[r, c]$ is the entry of $\mathbf{A}^c$ in the (r, c) position. With $\tilde{\mathcal{Z}}_h = HZ\langle \tilde{\mathbf{G}}^c, \tilde{\mathbf{G}}^b, \tilde{\mathbf{c}}, \tilde{\mathbf{A}}^c, \tilde{\mathbf{A}}^b, \tilde{\mathbf{b}}\rangle = HZ\langle \mathbf{G}^c - \Lambda_G\mathbf{A}^c, \mathbf{G}^b - \Lambda_G\mathbf{A}^b, \mathbf{c} + \Lambda_G\mathbf{b}, \mathbf{A}^c - \Lambda_A\mathbf{A}^c, \mathbf{A}^b - \Lambda_A\mathbf{A}^b, \mathbf{b} - \Lambda_A\mathbf{b}\rangle$, this transformation uses the r-th row of (16) to solve for $\boldsymbol{\xi}^c[c]$ in terms of $\boldsymbol{\xi}^c[k], k \in \{1, \ldots, c-1, c+1, \ldots, n_g\}$. This yields that $\tilde{\mathbf{G}}^c$ and $\tilde{\mathbf{A}}^c$ have identical zero c-th columns and $\tilde{\mathbf{A}}^c$, $\tilde{\mathbf{A}}^b$ and $\tilde{\mathbf{b}}$ have identically zero r-th rows. Removing these columns and rows results in a hybrid zonotope with one less continuous generator and one less equality constraint.

This strategy ensures that the removed r-th equality constraint is still imposed in the reduced hybrid zonotope but the ability to constraint the c-th continuous variable is lost, i.e., $|\boldsymbol{\xi}^c[c]| \leq 1$. In order to select which continuous variable to eliminate, we consider the Hausdorff error introduced by reduction $d_H(r, c, \mathcal{Z}_h) = \max_{\tilde{\boldsymbol{z}}\in\tilde{\mathcal{Z}}_h} \min_{\boldsymbol{z}\in\mathcal{Z}_h} \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|_2$.

C. Reduction Algorithm

Algorithm 3 summarizes the procedures in this section for reducing the numbers of continuous and binary generators of hybrid zonotopes computed by the reachability analysis (7). In this algorithm, Line 1-5 perform binary generator reduction, Line 6-7 are used to remove redundant parallel continuous generators, and Line 8-11 implement Proposition 3 to further reduce continuous generators.

V. Simulation

In this section, two simulation examples are provided to demonstrate the performance of the proposed hybrid zonotope-based reachability analysis method.

Example 1: Consider a double integrator model [10], [11]:

$$\boldsymbol{x}(t+1) = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\boldsymbol{x}(t) + \begin{bmatrix}0.5\\ 1\end{bmatrix}\boldsymbol{u}(t).$$

Algorithm 3: Complexity reduction for hybrid zonotopes computed by (7)

Input: hybrid zonotope $\mathcal{Z}_h = HZ\langle \mathbf{G}^c, \mathbf{G}^b, \mathbf{c}, \mathbf{A}^c, \mathbf{A}^b, \mathbf{b}\rangle$, $\hat{n}_g$ - number of continuous generators to reduce, $\hat{n}_b$ - number of continuous generators to reduce

Output: reduced hybrid zonotope $\hat{\mathcal{Z}}_h$

$\begin{bmatrix}\mathbf{G}_1^b & \mathbf{G}_2^b\end{bmatrix} \longleftarrow \mathbf{G}^b$; // Partition by $\hat{n}_b$

$\begin{bmatrix}\mathbf{A}_1^b & \mathbf{A}_2^b\end{bmatrix} \longleftarrow \mathbf{A}^b$; // Partition by $\hat{n}_b$

$\hat{\mathbf{G}}^c \longleftarrow \begin{bmatrix}\mathbf{G}^c & \mathbf{G}_1^b\end{bmatrix}$, $\hat{\mathbf{A}}^c \longleftarrow \begin{bmatrix}\mathbf{A}^c & \mathbf{A}_1^b\end{bmatrix}$

$\hat{\mathbf{G}}^b \longleftarrow \mathbf{G}_2^b$, $\hat{\mathbf{A}}^b \longleftarrow \mathbf{A}_2^b$, $\hat{\mathbf{c}} \longleftarrow \mathbf{c}$, $\hat{\mathbf{b}} \longleftarrow \mathbf{b}$

$\hat{\mathcal{Z}}_h \longleftarrow HZ\langle \hat{\mathbf{G}}^c, \hat{\mathbf{G}}^b, \hat{\mathbf{c}}, \hat{\mathbf{A}}^c, \hat{\mathbf{A}}^b, \hat{\mathbf{b}}\rangle$

$\hat{\mathcal{Z}}_h^+ \longleftarrow$ lift $\hat{\mathcal{Z}}_h$ using (15)

$\hat{\mathcal{Z}}_h \longleftarrow$ remove parallel generators in $\hat{\mathcal{Z}}_h^+$ and unlift

for $i \in \{1, \ldots, \max\{n_c, \hat{n}_g\}\}$ **do**

  $(r, c) \longleftarrow \text{argmin}_{r,c} d_H(r, c, \hat{\mathcal{Z}}_h)$

  $(\Lambda_G, \Lambda_A) \longleftarrow$ (17)

  $\hat{\mathcal{Z}}_h \longleftarrow HZ\langle \hat{\mathbf{G}}^c - \Lambda_G \hat{\mathbf{A}}^c, \hat{\mathbf{G}}^b - \Lambda_G \hat{\mathbf{A}}^b, \hat{\mathbf{c}} + \Lambda_G \hat{\mathbf{b}}, \hat{\mathbf{A}}^c - \Lambda_A \hat{\mathbf{A}}^c, \hat{\mathbf{A}}^b - \Lambda_A \hat{\mathbf{A}}^b, \hat{\mathbf{b}} - \Lambda_A \hat{\mathbf{b}}\rangle$

return $\hat{\mathcal{Z}}_h$

We use the same 3-layer FNN with ReLU activation functions in [11] as the feedback controller. Algorithm 1 is implemented to get exact output sets of the FNN and then utilized to compute the reachable sets of the closed-loop system for $T = 2$ time steps based on Theorem 1 and Algorithm 2. The initial set is given by $\mathcal{X}_0 = HZ\left\langle \begin{bmatrix}0.2 & 0\\ 0 & 0.2\end{bmatrix}, \begin{bmatrix}0.25\\ 0\end{bmatrix}, \begin{bmatrix}2.5\\ 0\end{bmatrix}, \emptyset, \emptyset, \emptyset \right\rangle$.

We denote the proposed exact reachability analysis method based on (7) and Theorem 1 as Reach-HZ. We compare the proposed method with the Reach-CZ algorithm ([15]), the Reach-LP algorithm ([11]), and the Reach-SDP algorithm ([10]). For the latter two algorithms, we also test the version with initial set partition, i.e., Reach-LP-Partition and Reach-SDP-Partition. Table I summarizes the computation times and set over-approximation errors for the proposed method and other state-of-the-art methods. The approximation errors are computed based on the difference ratio of sizes of computed reachable sets and exact reachable sets at the last time step. Note that although both the proposed Reach-HZ method and the Reach-CZ method can return the exact reachable sets, Reach-HZ only takes about half of the time of Reach-CZ. This results from the fact that Reach-CZ computes each reachable set as multiple constrained zonotopes while our Reach-HZ represents each reachable set compactly as a single hybrid zonotope. Our algorithms are implemented in Python with Gurobi [17]. The computer used for all the algorithms has a 3.7GHz CPU and 32GB memory.

Figure 2 illustrates reachable sets of the double integrator system using different methods. It can be observed that both our method and Reach-CZ provide more accurate reachable sets for all the time steps compared with other methods. For the safety verification, we consider a star-shaped unsafe region as plotted in Figure 2, which is represented by a hybrid zonotope. We also compare the times used to solve the safety verification conditions between the MILP-based method in Proposition 1 and the LP-based method in [15]. The commercial solver Gurobi takes 0.016 seconds to solve the MILP-based conditions while the runtime is 0.063 seconds for MOSEK to solve the LP-based conditions [29]. This is due to the fact that the LP-based conditions require solving multiple LPs for each time step, while in our approach, only one MILP is solved for each time step.

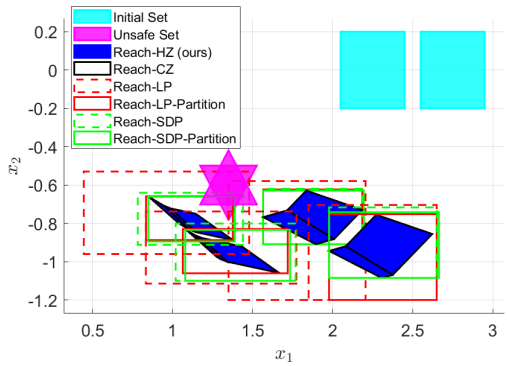

Fig. 2: Reachable sets computed for the double integrator example. The initial set $\mathcal{X}_0$ is shown in cyan and the star-shaped unsafe region is in magenta. Reachable set computed by Reach-HZ is plotted in blue. Reachable set computed by Reach-CZ ([15]) is in black, the LP-based method ([11]) is in red, and the SDP-based method ([10]) is in green.

Algorithm	Runtime [s]	Approx. Error
Reach-HZ (ours)	0.146	0
Reach-CZ [15]	0.312	0
Reach-LP [11]	0.032	3.34
Reach-LP-Partition	2.297	0.23
Reach-SDP [10]	108.77	0.79
Reach-SDP-Partition	5222.91	0.33

TABLE I: Comparison of different reachability methods for Example 1. Reach-HZ returns exact reachable sets within a shorter time compared with Reach-CZ.

Example 2: Consider a 4-D lateral dynamics model:

$$\boldsymbol{x}(t+1) = \begin{bmatrix}0 & 1 & 5 & 0\\ 0 & -5 & 0 & -9.5\\ 0 & 0 & 0 & 1\\ 0 & 0.05 & 0 & -2.8\end{bmatrix}\boldsymbol{x}(t) + \begin{bmatrix}0\\ 25\\ 0\\ 50\end{bmatrix}\boldsymbol{u}(t).$$

A 2-layer FNN is employed as the feedback controller and the initial set is given by $\mathcal{X}_0 = [0.1, 0.9] \times [-0.9, -0.1] \times [0.05, 0.15] \times [0.05, 0.15]$.

We implement the proposed Reach-HZ method and the relaxed version denoted as Reach-HZ-Relax which combine Reach-HZ with the complexity reduction method in Algorithm 3. We also run the Reach-CZ and Reach-CZ-Approx algorithms from [15] for comparison. Figure 3 shows the one-step reachable sets of the lateral dynamics system computed by four different methods and Table II summarizes their runtimes. Similar to the previous example, our Reach-HZ method provides the exact reachable set in a shorter

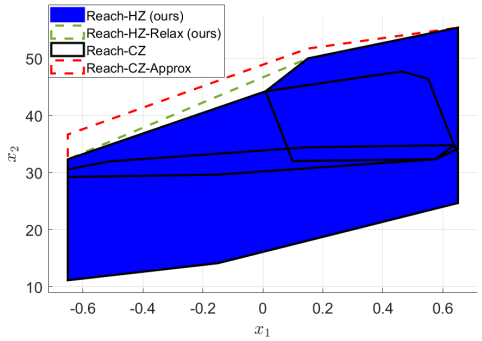

Fig. 3: Reachable sets computed for the lateral dynamics example. Reachable set computed by Reach-HZ is plotted in blue and the Reach-CZ method ([15]) is in black. Our Reach-HZ-Relax computes the tightest convex relaxation (green), while the Reach-CZ-Approx method ([15]) only provides a more conservative convex relaxation (red).

Algorithm	Runtime [s]	Approx. Error
Reach-HZ (ours)	0.062	0
Reach-HZ-Relax (ours)	0.079	0.08
Reach-CZ [15]	0.110	0
Reach-CZ-Approx [15]	0.047	0.20

TABLE II: Comparison of different reachability-based methods for the lateral dynamics example.

time compared with Reach-CZ. Furthermore, our Reach-HZ-Relax method provides the tightest convex relaxation (the convex hull) of the reachable set while Reach-CZ-Approx returns a more conservative convex relaxation.

VI. Conclusion

In this work, we introduce a novel approach for reachability analysis of neural feedback systems based on hybrid zonotopes. We show that a ReLU-activated neural network can be exactly represented by a hybrid zonotope. When the input set of a neural feedback system is a hybrid zonotope, it is also proven that the exact reachable sets can be compactly represented by hybrid zonotopes. Based on the reachability analysis, an MILP-based condition is presented for safety verification of the neural feedback system. Complexity reduction techniques are also proposed for the hybrid zonotopes to reduce the computation burden. As demonstrated in two numerical examples, the proposed approach outperforms other methods for reachability analysis and safety verification of neural feedback systems.

References

[1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[2] D. A. Pomerleau, *Neural network perception for mobile robot guidance*. Springer Science & Business Media, 2012, vol. 239.

[3] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 97–117.

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

[5] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: Verifying safety properties of hybrid systems with neural network controllers," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 169–178.

[6] 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, 2021.

[7] N. Kochdumper, C. Schilling, M. Althoff, and S. Bak, "Open-and closed-loop neural network verification using polynomial zonotopes," *arXiv preprint arXiv:2207.02715*, 2022.

[8] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," in *Proceedings of Robotics: Science and Systems*, 2021.

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

[10] 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 *IEEE 59th Conference on Decision and Control*, 2020, pp. 5929–5934.

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

[12] A. Chakrabarty, C. Danielson, S. Di Cairano, and A. Raghunathan, "Active learning for estimating reachable sets for systems with unknown dynamics," *IEEE Transactions on Cybernetics*, 2020.

[13] A. Devonport and M. Arcak, "Data-driven reachable set computation using adaptive Gaussian process classification and Monte Carlo methods," in *American Control Conference*. IEEE, 2020, pp. 2629–2634.

[14] H.-D. Tran, F. Cai, M. L. Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, "Safety verification of cyber-physical systems with reinforcement learning control," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 5s, pp. 1–22, 2019.

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

[16] 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," *arXiv preprint arXiv:2106.14831*, 2021.

[17] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2022. [Online]. Available: https://www.gurobi.com

[18] D. Bertsimas and B. Stellato, "Online mixed-integer optimization in milliseconds," *INFORMS Journal on Computing*, 2022.

[19] P. McMullen, "On zonotopes," *Transactions of the American Mathematical Society*, vol. 159, pp. 91–109, 1971.

[20] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, "Constrained zonotopes: A new tool for set-based estimation and fault detection," *Automatica*, vol. 69, pp. 126–136, 2016.

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

[22] H.-D. Tran, D. Manzanas Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Star-based reachability analysis of deep neural networks," in *International Symposium on Formal Methods*. Springer, 2019, pp. 670–686.

[23] Y. Zhang and X. Xu, "Reachability analysis and safety verification of neural feedback systems via hybrid zonotopes," *arXiv preprint arXiv:2210.03244*, 2022.

[24] C. Sidrane, A. Maleki, A. Irfan, and M. J. Kochenderfer, "Overt: An algorithm for safety verification of neural network control policies for nonlinear systems," *Journal of Machine Learning Research*, vol. 23, no. 117, 2022.

[25] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[26] V. Raghuraman and J. P. Koeln, "Set operations and order reductions for constrained zonotopes," *Automatica*, vol. 139, p. 110204, 2022.

[27] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma, "Strong mixed-integer programming formulations for trained neural networks," *Mathematical Programming*, vol. 183, no. 1, pp. 3–39, 2020.

[28] M. Althoff, "On computing the Minkowski difference of zonotopes," *arXiv preprint arXiv:1512.02794*, 2015.

[29] MOSEK ApS, "The MOSEK Fusion API for Python manual. Version 9.3.18," 2019. [Online]. Available: http://docs.mosek.com/9.3/pythonfusion/index.html