Goal-Reaching Control Synthesis for Neural Network Control Systems via Backward Reachability

Hang Zhang, Abigail J. Winn, Yuhao Zhang and Xiangru Xu

***Abstract*— This letter investigates goal-reaching control synthesis for neural network control systems. A backward reachability framework is developed based on constrained zonotopes, in which the graph set of a ReLU-activated feedforward neural network is encoded as a finite union of constrained zonotopes. Using this representation, under-approximations of backward reachable sets are computed for systems with nonlinear plant models, ensuring the feasibility of the goal-reaching task. Control sequences are then synthesized through an optimization procedure that exploits the under-approximated set. A numerical example demonstrates the effectiveness of the proposed approach.**

***Index Terms*— Neural networks, control synthesis, backward reachable set, constrained zonotope.**

I. Introduction

With the increasing prevalence of neural networks (NNs), many dynamical systems now incorporate an NN either as a controller [1], [2] or as a model for complex nonlinear effects [3], [4], giving rise to neural network control systems (NNCSs). Control synthesis for NNCSs, such as steering the system from a given initial state to a desired target set known as goal-reaching, remains a challenging problem. The nonlinearities of both the NN and the plant dynamics make it difficult to provide formal guarantees on the feasibility of control synthesis, which are crucial for safety-critical applications. Existing approaches include embedding NNs into a model predictive control (MPC) framework to compute control inputs via optimization [4], [5], although feasibility guarantees remain unresolved.

The backward reachable set (BRS) of a discrete-time control system is the set of initial states from which the system can be driven into a target region within a finite number of steps under some control sequence. Backward reachability has been studied using constrained zonotopes (CZs) [6], which support efficient set operations and enable a linear program (LP)-based formulation for safety verification [7], [8]; however, these methods do not extend to NNCSs. On the other hand, many techniques for BRS computation in NNCSs have been proposed [9], [10], [11], [12], [13], though most of them focus on safety verification with a fixed controller.

The authors are with the Department of Mechanical Engineering, University of Wisconsin–Madison, Madison, WI, USA. Email: {hang.zhang, ajwinn, yuhao.zhang2, xiangru.xu}@wisc.edu.

In this work, we address the goal-reaching control synthesis problem for NNCSs using backward reachability with CZs. Our main contributions are threefold: (i) We present a CZ-based algorithm that encodes the graph set of a ReLU-activated feedforward neural network (FNN) as a finite union of CZs. (ii) We develop a CZ-based backward reachability framework for ReLU-activated NNCSs with nonlinear plant models; the method computes both over- and under-approximations of the BRSs, with the under-approximation providing feasibility guarantees for goal-reaching from a given initial state. (iii) Leveraging the under-approximated BRS, we synthesize feasible control sequences that drive the NNCS into a target set within a finite time. These innovations differ fundamentally from our prior verification-oriented work (e.g., [11], [12], [13], [14]) in both problem formulation and technical approach.

Notation. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, x_i denotes the i-th component of $\boldsymbol{x}$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, $\boldsymbol{A}[i\!:\!j, r\!:\!s]$ denotes the submatrix of $\boldsymbol{A}$ consisting of rows i through j and columns r through s; if i or j (resp. r or s) is omitted, the range is understood to extend from the first row (resp. column) to j (resp. s), or from i (resp. r) to the last row (resp. column); $\boldsymbol{A}[-i\!:\!,:]$ (resp. $\boldsymbol{A}[:,-i\!:]$) denotes the submatrix formed by the last i rows (resp. columns). Given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ with compatible dimensions, $[\boldsymbol{A},\boldsymbol{B}]$ and $[\boldsymbol{A};\boldsymbol{B}]$ denote their horizontal and vertical concatenations, respectively. Given sets $\mathcal{X},\mathcal{Z} \subset \mathbb{R}^n$, $\mathcal{X} \oplus \mathcal{Z} = \{\boldsymbol{x}+\boldsymbol{z} \mid \boldsymbol{x}\in\mathcal{X}, \boldsymbol{z}\in\mathcal{Z}\}$ is the Minkowski sum, and $\mathcal{X} \ominus \mathcal{Z} = \{\boldsymbol{s}\in\mathbb{R}^n \mid \boldsymbol{s}+\mathcal{Z}\subseteq\mathcal{X}\}$ is the Minkowski difference. The affine transformation of $\mathcal{X}$ given a matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ is $\boldsymbol{A}\mathcal{X}+\boldsymbol{b} = \{\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b} \mid \boldsymbol{x}\in\mathcal{X}\}$. The identity matrix in $\mathbb{R}^{n\times n}$ is denoted as $\boldsymbol{I}_n$. The vectors and matrices whose entries are all 0 (resp. 1) are denoted as $\boldsymbol{0}$ (resp. $\boldsymbol{1}$).

II. Preliminaries & Problem Statement

Definition 1: [6] A set $\mathcal{Z}\subset\mathbb{R}^n$ is a *constrained zonotope* if there exists $\langle \boldsymbol{G},\boldsymbol{c},\boldsymbol{A},\boldsymbol{b}\rangle \in \mathbb{R}^{n\times n_g}\times\mathbb{R}^n\times\mathbb{R}^{n_c\times n_g}\times\mathbb{R}^{n_c}$ such that $\mathcal{Z} = \{\boldsymbol{G}\boldsymbol{\xi}+\boldsymbol{c} \mid \boldsymbol{\xi}\in\mathcal{B}_\infty^{n_g}, \boldsymbol{A}\boldsymbol{\xi}=\boldsymbol{b}\}$ where $\mathcal{B}_\infty^{n_g} = \{\boldsymbol{x}\in\mathbb{R}^{n_g} \mid \|\boldsymbol{x}\|_\infty \le 1\}$ is the unit hypercube in $\mathbb{R}^{n_g}$.

Each column of $\boldsymbol{G}$ is called a generator of $\mathcal{Z}$. The constrained generator representation of $\mathcal{Z}$ is $\mathcal{Z} = \langle \boldsymbol{G},\boldsymbol{c},\boldsymbol{A},\boldsymbol{b}\rangle$. Denote $\mathcal{B}_\infty(\boldsymbol{A},\boldsymbol{b}) = \{\boldsymbol{\xi}\in\mathcal{B}_\infty \mid \boldsymbol{A}\boldsymbol{\xi}=\boldsymbol{b}\}$. A CZ without the equality constraint $\boldsymbol{A}\boldsymbol{\xi}=\boldsymbol{b}$ reduces to a *zonotope*, defined as $\mathcal{Z} = \{\boldsymbol{G}\boldsymbol{\xi}+\boldsymbol{c} \mid \boldsymbol{\xi}\in\mathcal{B}_\infty^{n_g}\}$, with the generator representation $\mathcal{Z} = \langle \boldsymbol{c},\boldsymbol{G}\rangle$. CZs are closed under affine transformations,

Cartesian products, and generalized intersections. The emptiness of a CZ can be checked by solving an LP: for a CZ $\mathcal{Z} = \langle \boldsymbol{G}, \boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b} \rangle \subset \mathbb{R}^n$, $\mathcal{Z} \neq \emptyset \Leftrightarrow \min\{\|\boldsymbol{\xi}\|_\infty \mid \boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b}\} \leq 1$ [6].

The discrete-time NNCS considered in this work is:

$$\boldsymbol{x}(t+1) = \underbrace{\boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{B}_\pi \boldsymbol{\pi}(\boldsymbol{x}(t)) + \boldsymbol{B}\boldsymbol{u}(t)}_{\boldsymbol{f}_{\text{cl}}(\boldsymbol{x}(t), \boldsymbol{u}(t))} \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$ are the state and the control input, respectively; $\boldsymbol{f}$ is a twice-differentiable nonlinear function; $\boldsymbol{\pi}$ is an ℓ-layer ReLU-activated FNN, which may represent the model residual or nominal NN-based controller. We assume $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{u} \in \mathcal{U}$, where $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ are the state and input domains, respectively, which are both represented as CZs.

For the FNN $\boldsymbol{\pi}$, its k-th layer weight matrix and bias vector are denoted as $\boldsymbol{W}^{(k-1)}$ and $\boldsymbol{v}^{(k-1)}$, respectively, where $k = 1, \ldots, \ell$. Denote $\boldsymbol{x}^{(k)}$ as the output of the neurons in the k-th layer, and n_k as the dimension of $\boldsymbol{x}^{(k)}$ (i.e., the number of neurons in the k-th layer). Then, for $k = 1, \ldots, \ell-1$, we have $\boldsymbol{x}^{(k)} = \boldsymbol{\phi}(\boldsymbol{z}^{(k)}) = \boldsymbol{\phi}(\boldsymbol{W}^{(k-1)}\boldsymbol{x}^{(k-1)} + \boldsymbol{v}^{(k-1)})$, where $\boldsymbol{z}^{(k)}$ is referred to as the *pre-activation* of the neurons $\boldsymbol{x}^{(k)}$, $\boldsymbol{x}^{(0)} = \boldsymbol{x}(t)$, and $\boldsymbol{\phi}$ is the vector-valued activation function constructed by component-wise repetition of the ReLU function, i.e., $\boldsymbol{\phi}(\boldsymbol{z}^{(k)}) \triangleq [\text{ReLU}(z_1^{(k)}); \cdots; \text{ReLU}(z_{n_k}^{(k)})]$. In the last layer, only the linear map is applied, i.e., $\boldsymbol{\pi}(\boldsymbol{x}(t)) = \boldsymbol{x}^{(\ell)} = \boldsymbol{W}^{(\ell-1)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{v}^{(\ell-1)}$. The total number of hidden neurons of $\boldsymbol{\pi}$ is denoted as $N_{\boldsymbol{\pi}} \triangleq n_1 + \cdots + n_{\ell-1}$.

Definition 2: Given a target set $\mathcal{T} \subset \mathcal{X}$, the K-step BRS of the NNCS (1), denoted as $\mathcal{P}_K(\mathcal{T})$, is defined as $\mathcal{P}_K(\mathcal{T}) \triangleq \{\boldsymbol{x}(0) \in \mathcal{X} \mid \exists \boldsymbol{u}(0), \cdots, \boldsymbol{u}(K-1) \in \mathcal{U} : \boldsymbol{x}(t) = \boldsymbol{f}_{\text{cl}}(\boldsymbol{x}(t-1), \boldsymbol{u}(t-1)), \boldsymbol{x}(t) \in \mathcal{X}, \boldsymbol{x}(K) \in \mathcal{T}, t = 1, 2, \ldots, K\}$. For simplicity, denote $\mathcal{P}(\mathcal{T}) \triangleq \mathcal{P}_1(\mathcal{T})$.

The following problem will be investigated.

Problem 1: Given a target set $\mathcal{T} \subset \mathcal{X}$ represented as either a single CZ or a finite union of CZs, an initial condition $\boldsymbol{x}(0)$, and a time horizon $K \in \mathbb{Z}_{>0}$, determine whether there exists a feasible control sequence $\boldsymbol{u}(0), \boldsymbol{u}(1), \cdots, \boldsymbol{u}(K-1)$ such that $\boldsymbol{x}(K) \in \mathcal{T}$; if so, compute such a control sequence.

The main challenge in Problem 1 arises from the nonlinearities in $\boldsymbol{f}$ and $\boldsymbol{\pi}$. To address this, we compute over- and under-approximations of the BRSs for the NNCS (1), since computing the exact set $\mathcal{P}_K(\mathcal{T})$ is generally intractable. Specifically, we construct a K-step over-approximated BRS, $\overline{\mathcal{P}}_K(\mathcal{T})$, and a K-step under-approximated BRS, $\underline{\mathcal{P}}_K(\mathcal{T})$. These sets provide two key insights: *i) if $\boldsymbol{x}(0) \notin \overline{\mathcal{P}}_K(\mathcal{T})$, then no control sequence can render $\boldsymbol{x}(K) \in \mathcal{T}$; ii) if $\boldsymbol{x}(0) \in \underline{\mathcal{P}}_K(\mathcal{T})$, then there exists a feasible control sequence $\boldsymbol{u}(0), \boldsymbol{u}(1), \cdots, \boldsymbol{u}(K-1) \in \mathcal{U}$ such that $\boldsymbol{x}(K) \in \mathcal{T}$.* The following sections detail the computation of these approximations and the feasible control sequence.

III. Construction of FNN Graph Set via CZs

In this section, we describe how to construct the graph set of the ReLU-activated FNN $\boldsymbol{\pi}$ via CZs, which will be used for backward reachability analysis in the next section.

The graph set of $\boldsymbol{\pi}$ over $\mathcal{X}$ is defined as $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X}) \triangleq \{[\boldsymbol{x}; \boldsymbol{x}_{\boldsymbol{\pi}}] \mid \boldsymbol{x}_{\boldsymbol{\pi}} = \boldsymbol{\pi}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\} \subset \mathbb{R}^{n+m}$. It is shown in [11, Theorem 1] that $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$ can be represented exactly by a single hybrid zonotope; however, this representation is often highly redundant as many binary assignments fail to satisfy the associated equality constraints. To mitigate this redundancy, we represent $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$ as a finite union of CZs.

Algorithm 1: FNN Graph Set Construction via CZs

Input: CZ domain $\mathcal{X}$, FNN $\boldsymbol{\pi}$, a large scalar α
Output: $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \text{GetNNGraphSet}(\mathcal{X}, \boldsymbol{\pi}, \alpha)$ as a finite union of CZs

1 $\mathcal{Z}_1 \leftarrow \mathcal{X}$, $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) \leftarrow \emptyset$, $n_0 \leftarrow n$;
2 **for** $k = 1, 2, \ldots, \ell - 1$ **do**
3 $\quad \{\mathcal{AP}_k^1, \mathcal{AP}_k^2, \cdots, \mathcal{AP}_k^{s_k}\} \leftarrow$ Find AP Candidates;
4 $\quad r \leftarrow 0$;
5 $\quad$ **for** $j \in \{1, 2, \cdots, s_k\}$ **do**
6 $\quad\quad \langle \boldsymbol{G}_k, \boldsymbol{c}_k, \boldsymbol{A}_k, \boldsymbol{b}_k \rangle \leftarrow \mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k^j)$; // use (2)
7 $\quad\quad$ **for** $\mathcal{Z}_{k,i} = \langle \boldsymbol{G}_i, \boldsymbol{c}_i, \boldsymbol{A}_i, \boldsymbol{b}_i \rangle$ *in* $\mathcal{Z}_k$ **do**
8 $\quad\quad\quad \boldsymbol{G}_a \leftarrow \boldsymbol{G}_k[1 : n_k, :]; \boldsymbol{c}_a \leftarrow \boldsymbol{c}_k[1 : n_k, :]$;
9 $\quad\quad\quad \boldsymbol{G}_b \leftarrow \boldsymbol{G}_i[-n_{k-1} :, :]; \boldsymbol{c}_b \leftarrow \boldsymbol{c}_i[-n_{k-1} :, :]$;
10 $\quad\quad\quad \boldsymbol{A}_{\text{aug}} \leftarrow [\text{diag}(\boldsymbol{A}_i, \boldsymbol{A}_k); [-\boldsymbol{W}^{(k-1)}\boldsymbol{G}_b, \boldsymbol{G}_a]]$;
11 $\quad\quad\quad \boldsymbol{b}_{\text{aug}} \leftarrow [\boldsymbol{b}_i; \boldsymbol{b}_k; \boldsymbol{W}^{(k-1)}\boldsymbol{c}_b + \boldsymbol{v}^{(k-1)} - \boldsymbol{c}_a]$;
12 $\quad\quad\quad \mathcal{H}_{\text{aug}} \leftarrow \langle \text{diag}(\boldsymbol{G}_i, \boldsymbol{G}_k), [\boldsymbol{c}_i; \boldsymbol{c}_k], \boldsymbol{A}_{\text{aug}}, \boldsymbol{b}_{\text{aug}} \rangle$;
13 $\quad\quad\quad \mathcal{H}_k \leftarrow \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{I}_{n_k} \end{bmatrix} \mathcal{H}_{\text{aug}}$;
14 $\quad\quad\quad$ **if** $\mathcal{H}_k \neq \emptyset$ **then**
15 $\quad\quad\quad\quad r \leftarrow r + 1$; $\mathcal{Z}_{k+1,r} \leftarrow \mathcal{H}_k$;
16 $\quad \mathcal{Z}_{k+1} \leftarrow \cup_{i=1}^{r} \mathcal{Z}_{k+1,i}$;
17 **for** $\mathcal{Z}_{\ell,i} \in \mathcal{Z}_\ell$ **do**
18 $\quad \mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) \leftarrow \mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) \cup (\text{diag}(\boldsymbol{I}_n, \boldsymbol{W}^{(\ell-1)})\mathcal{Z}_{\ell,i} + [\boldsymbol{0}; \boldsymbol{v}^{(\ell-1)}])$
19 **return** $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X})$

Given an input to $\boldsymbol{\pi}$, the *activation pattern* of $\boldsymbol{\pi}$ is an assignment of a binary value to each neuron, indicating whether it is active (1) or inactive (0) [15]. Specifically, the activation pattern of $\boldsymbol{\pi}$ is defined as $\mathcal{AP} \triangleq [\mathcal{AP}_1; \mathcal{AP}_2; \cdots; \mathcal{AP}_{\ell-1}] \in \{0,1\}^{N_{\boldsymbol{\pi}}}$ where $\mathcal{AP}_k \triangleq [\mathcal{AP}_{k,1}; \mathcal{AP}_{k,2}; \cdots; \mathcal{AP}_{k,n_k}] \in \{0,1\}^{n_k}$; here, $\mathcal{AP}_{k,j} = 1$ if $z_j^{(k)} \geq 0$ and $\mathcal{AP}_{k,j} = 0$ otherwise, where $z_j^{(k)}$ is the pre-activation of the j-th neuron in the k-th layer. For a given $\mathcal{AP}_k$ and a constant $\alpha > 0$, the graph set of the vector-valued ReLU function for the k-th layer, denoted as $\mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k)$, represents the relationship between $\boldsymbol{z}^{(k)}$ and $\boldsymbol{x}^{(k)} = \boldsymbol{\phi}(\boldsymbol{z}^{(k)})$ in the $[\boldsymbol{z}^{(k)}; \boldsymbol{x}^{(k)}]$ space:

$$\mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k) = \boldsymbol{P} \cdot \left(\times_{j=1}^{n_k} \mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_{k,j})\right) \tag{2}$$

where $\mathcal{H}_{\text{ReLU}}(\alpha, 1) = \langle [\alpha/2; \alpha/2], [\alpha/2; \alpha/2], \emptyset, \emptyset \rangle$, $\mathcal{H}_{\text{ReLU}}(\alpha, 0) = \langle [\alpha/2; 0], [-\alpha/2; 0], \emptyset, \emptyset \rangle$, and $\boldsymbol{P}$ is a permutation matrix satisfying $[z_1; \cdots; z_{n_k}; x_1; \cdots; x_{n_k}] = \boldsymbol{P}[z_1; x_1; \cdots; z_{n_k}; x_{n_k}]$. Note that the graph of a scalar ReLU function over $[-\alpha, \alpha]$ is $\mathcal{H}_{\text{ReLU}}(\alpha, 1) \cup \mathcal{H}_{\text{ReLU}}(\alpha, 0)$ in the $[z; x]$ plane. Since affine transformations and Cartesian products of CZs are closed, $\mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k)$ in (2) is also a CZ, denoted as $\mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k) = \langle \boldsymbol{G}_k, \boldsymbol{c}_k, \boldsymbol{A}_k, \boldsymbol{b}_k \rangle$.

Algorithm 1 constructs the graph set $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$ using CZs. The procedure begins by finding all candidate activation patterns for the k-th layer over the input space $\mathcal{X}$ (Line 3). This can be achieved either by exhaustively enumerating all 2^{n_k}

possible activation patterns, or by using NN verifiers [16] or Algorithm 2 in [14] to over-approximate the neuron ranges and eliminate activation patterns that violate these bounds from the initial set of 2^{n_k} candidates. For each candidate $\mathcal{AP}_k^j$, we compute $\mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k^j)$ and propagate the layer transformation from $[\boldsymbol{x}^{(0)}; \boldsymbol{x}^{(k-1)}]$ to the $[\boldsymbol{x}^{(0)}; \boldsymbol{x}^{(k)}]$ space for each non-empty CZ $\mathcal{Z}_{k,i}$. This is done by combining $\mathcal{Z}_{k,i}$ and $\mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_k^j)$ with affine equality constraints (Lines 8-13). The resulting CZ is then projected onto $[\boldsymbol{x}^{(0)}; \boldsymbol{x}^{(k)}]$ to yield $\mathcal{H}_k$. If $\mathcal{H}_k$ is non-empty, indicating the activation pattern is feasible, it is stored and passed to the next layer (Lines 14-16). The final affine transformation is applied at the last layer to obtain the final graph set (Lines 17-18). Because the number of activation patterns is finite, this algorithm is guaranteed to terminate. Note that α can be arbitrarily chosen as long as $z_j^{(k)} \in [-\alpha, \alpha]$, thereby preserving the exact graph set representation.

The following theorem formally proves that $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X})$ returned by Algorithm 1 is exactly $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$.

Theorem 1: The output of Algorithm 1, $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X})$, is a finite union of CZs that exactly represent the graph set of $\boldsymbol{\pi}$ over the state domain $\mathcal{X}$, i.e. $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$.

Proof: We only provide a proof sketch due to the space limit. Define $\mathcal{H}_{0,k} \triangleq \{[\boldsymbol{x}^{(0)}; \boldsymbol{x}^{(k)}] \mid \boldsymbol{x}^{(0)} \in \mathcal{X}, [\boldsymbol{z}^{(j)}; \boldsymbol{x}^{(j)}] \in \mathcal{H}_{\text{ReLU}}(\alpha, \mathcal{AP}_j), \boldsymbol{z}^{(j)} = \boldsymbol{W}^{(j-1)}\boldsymbol{x}^{(j-1)} + \boldsymbol{v}^{(j-1)}, j = 1, 2, \cdots, k\}$. It is easy to show $\mathcal{H}_k = \mathcal{H}_{0,k}$ holds for all $k \in \{1, 2, \ldots, \ell - 1\}$ by induction where $\mathcal{H}_k$ is computed by Line 13. From Line 3 and Lines 14-15, all infeasible activation patterns over $\mathcal{X}$ are eliminated. Thus, from Lines 15-16, $\mathcal{Z}_\ell = \cup_{i=1}^r \mathcal{Z}_{\ell,i}$ exactly represents the relationship between the input $\boldsymbol{x}$ and $\boldsymbol{x}^{(\ell-1)}$ for all feasible activation patterns. For the last layer, only the affine transformation in Lines 17-18 is applied, which yields $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$. ∎

Theorem 1 shows that $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$ can be represented as a finite union of CZs, where each individual CZ corresponds to a *feasible activation pattern* of $\boldsymbol{\pi}$ over $\mathcal{X}$. Therefore, the number of CZs in $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$, denoted as n_π, is equal to the total number of feasible activation patterns of $\boldsymbol{\pi}$ over $\mathcal{X}$.

The graph set $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X})$ returned by Algorithm 1 possesses useful properties for computing the BRS over-approximation.

Proposition 1: Consider a CZ-represented state domain $\mathcal{X} = \langle \boldsymbol{G}_x, \boldsymbol{c}_x, \boldsymbol{A}_x, \boldsymbol{b}_x \rangle$ where $\boldsymbol{G}_x \in \mathbb{R}^{n \times n_{g_x}}$ and $\boldsymbol{A}_x \in \mathbb{R}^{n_{c_x} \times n_{g_x}}$. Let $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \cup_{i=1}^{n_\pi} \mathcal{H}_{\boldsymbol{\pi}_i} \triangleq \cup_{i=1}^{n_\pi} \langle \boldsymbol{G}_{\boldsymbol{\pi}_i}, \boldsymbol{c}_{\boldsymbol{\pi}_i}, \boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{b}_{\boldsymbol{\pi}_i} \rangle$ be the output of Algorithm 1. For any $i \in \{1, \cdots, n_\pi\}$, the following properties hold: (i) $\boldsymbol{A}_{\boldsymbol{\pi}_i}[1 : n_{c_x}, :] = [\boldsymbol{A}_x, \boldsymbol{0}]$, $\boldsymbol{b}_{\boldsymbol{\pi}_i}[1 : n_{c_x}, :] = \boldsymbol{b}_x$, $\boldsymbol{G}_{\boldsymbol{\pi}_i}[1 : n, :] = [\boldsymbol{G}_x, \boldsymbol{0}]$, and $\boldsymbol{c}_{\boldsymbol{\pi}_i}[1 : n, :] = \boldsymbol{c}_x$; (ii) $\ker(\boldsymbol{G}_x) \cap \ker(\boldsymbol{A}_x) \subseteq \ker(\boldsymbol{G}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}]) \cap \ker(\boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}])$ where $\ker$ denotes the matrix kernel.

Proof: We provide a proof sketch due to the space limit. From the construction of $\mathcal{H}_k$ (Lines 8-13) in Algorithm 1, it is easy to check property (i). For (ii), let $\boldsymbol{q} \in \ker(\boldsymbol{G}_x) \cap \ker(\boldsymbol{A}_x)$, then $\boldsymbol{G}_x\boldsymbol{q} = \boldsymbol{0}$ and $\boldsymbol{A}_x\boldsymbol{q} = \boldsymbol{0}$. From Line 12 and Line 13, it is easy to verify that $\boldsymbol{G}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}] = [\boldsymbol{G}_x; \boldsymbol{0}]$. From Line 10 and Line 12, it is easy to obtain that $\boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}] = [\boldsymbol{A}_x; -\boldsymbol{W}^{(0)}\boldsymbol{G}_x; \boldsymbol{0}]$. Hence, $\boldsymbol{G}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}]\boldsymbol{q} = \boldsymbol{0}$ and $\boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}]\boldsymbol{q} = \boldsymbol{0}$, which proves property (ii). ∎

IV. Goal-Reaching Control Synthesis using Backward Reachability

A. BRS Over-Approximation

In this subsection, we compute $\overline{\mathcal{P}}_K(\mathcal{T})$, the K-step over-approximated BRS, for the NNCS (1).

Similar to [12], we define the graph set of $\boldsymbol{f}$ over $\mathcal{X}$ as $\mathcal{G}_{\boldsymbol{f}}(\mathcal{X}) \triangleq \{[\boldsymbol{x}; \boldsymbol{y}] \mid \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\}$. We can use function approximation techniques, such as Special Ordered Set [17] or OVERT [18], to construct a finite union of CZs, denoted as $\mathcal{H}_f(\mathcal{X})$, such that $\mathcal{G}_{\boldsymbol{f}}(\mathcal{X}) \subseteq \mathcal{H}_{\boldsymbol{f}}(\mathcal{X}) \triangleq \cup_{j=1}^{n_f} \mathcal{H}_{\boldsymbol{f}_j}$.

Assumption 1: For all $j \in \{1, 2, \cdots, n_f\}$, $\mathcal{H}_{\boldsymbol{f}_j} \triangleq \langle \boldsymbol{G}_{\boldsymbol{f}_j}, \boldsymbol{c}_{\boldsymbol{f}_j}, \boldsymbol{A}_{\boldsymbol{f}_j}, \boldsymbol{b}_{\boldsymbol{f}_j} \rangle$ satisfies $\boldsymbol{A}_{\boldsymbol{f}_j}[1 : n_{c_x}, :] = [\boldsymbol{A}_x, \boldsymbol{0}]$, $\boldsymbol{b}_{\boldsymbol{f}_j}[1 : n_{c_x}, :] = \boldsymbol{b}_x$, $\boldsymbol{G}_{\boldsymbol{f}_j}[1 : n, :] = [\boldsymbol{G}_x, \boldsymbol{0}]$, and $\boldsymbol{c}_{\boldsymbol{f}_j}[1 : n, :] = \boldsymbol{c}_x$.

Remark 1: Assumption 1 is readily satisfied when $\mathcal{H}_{\boldsymbol{f}_j}$ is constructed in graph-set form using the CZ generalized intersection operation. For instance, consider a CZ $\hat{\mathcal{H}}_{\boldsymbol{f}_j} \triangleq \langle \hat{\boldsymbol{G}}_{\boldsymbol{f}_j}, \hat{\boldsymbol{c}}_{\boldsymbol{f}_j}, \hat{\boldsymbol{A}}_{\boldsymbol{f}_j}, \hat{\boldsymbol{b}}_{\boldsymbol{f}_j} \rangle$ constructed by geometrically bounding the graph of $\boldsymbol{f}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \hat{\mathcal{X}} \supset \mathcal{X}$. In this case, the graph-set over-approximation $\mathcal{H}_{\boldsymbol{f}_j}$ can be computed as $\hat{\mathcal{H}}_{\boldsymbol{f}_j} \cap_{[\boldsymbol{I}_n, \boldsymbol{0}]} \mathcal{X}$. Applying [6, Proposition 1], we have $\boldsymbol{A}_{\boldsymbol{f}_j} = [[\hat{\boldsymbol{A}}_{\boldsymbol{f}_j}, \boldsymbol{0}]; [\boldsymbol{0}, \boldsymbol{A}_x]; [\hat{\boldsymbol{G}}_{\boldsymbol{f}_j}[1 : n, :], -\boldsymbol{G}_x]]$, $\boldsymbol{b}_{\boldsymbol{f}_j} = [\hat{\boldsymbol{b}}_{\boldsymbol{f}_j}; \boldsymbol{b}_x; \boldsymbol{c}_x - \hat{\boldsymbol{c}}_{\boldsymbol{f}_j}[1 : n, :]]$, $\boldsymbol{G}_{\boldsymbol{f}_j} = [\hat{\boldsymbol{G}}_{\boldsymbol{f}_j}, \boldsymbol{0}]$, $\boldsymbol{c}_{\boldsymbol{f}_j} = \hat{\boldsymbol{c}}_{\boldsymbol{f}_j}$. Let $\boldsymbol{\xi}_{\boldsymbol{f}_j} = [\hat{\boldsymbol{\xi}}_{\boldsymbol{f}_j}; \boldsymbol{\xi}_x]$. Then we have $\hat{\boldsymbol{A}}_{\boldsymbol{f}_j}\hat{\boldsymbol{\xi}}_{\boldsymbol{f}_j} = \hat{\boldsymbol{b}}_{\boldsymbol{f}_j}$, $\boldsymbol{A}_x\boldsymbol{\xi}_x = \boldsymbol{b}_x$, and $\hat{\boldsymbol{G}}_{\boldsymbol{f}_j}[1 : n, :]\hat{\boldsymbol{\xi}}_{\boldsymbol{f}_j} + \hat{\boldsymbol{c}}_{\boldsymbol{f}_j}[1 : n, :] = \boldsymbol{G}_x\boldsymbol{\xi}_x + \boldsymbol{c}_x$. By reordering $\boldsymbol{\xi}_{\boldsymbol{f}_j} = [\boldsymbol{\xi}_x; \hat{\boldsymbol{\xi}}_{\boldsymbol{f}_j}]$ we equivalently obtain $\boldsymbol{A}_{\boldsymbol{f}_j} = [[\boldsymbol{A}_x, \boldsymbol{0}]; [\boldsymbol{0}, \hat{\boldsymbol{A}}_{\boldsymbol{f}_j}]; [-\boldsymbol{G}_x, \hat{\boldsymbol{G}}_{\boldsymbol{f}_j}[1 : n, :]]]$, $\boldsymbol{b}_{\boldsymbol{f}_j} = [\boldsymbol{b}_x; \hat{\boldsymbol{b}}_{\boldsymbol{f}_j}; \boldsymbol{c}_x - \hat{\boldsymbol{c}}_{\boldsymbol{f}_j}[1 : n, :]]$, $\boldsymbol{G}_{\boldsymbol{f}_j} = [[\boldsymbol{G}_x, \boldsymbol{0}]; [\boldsymbol{0}, \hat{\boldsymbol{G}}_{\boldsymbol{f}_j}[-n :, :]]]$, and $\boldsymbol{c}_{\boldsymbol{f}_j} = [\boldsymbol{c}_x; \hat{\boldsymbol{c}}_{\boldsymbol{f}_j}[-n :,]]$, implying Assumption 1 holds.

The next theorem presents a constructive method for computing the one-step BRS over-approximation of the NNCS (1), represented as a finite union of CZs. Recall that n_{g_x} and n_{c_x} denote the numbers of generators and equality constraints in $\mathcal{X}$, respectively.

Theorem 2: Given $\mathcal{X} = \langle \boldsymbol{G}_x, \boldsymbol{c}_x, \boldsymbol{A}_x, \boldsymbol{b}_x \rangle$, let $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \cup_{i=1}^{n_\pi} \mathcal{H}_{\boldsymbol{\pi}_i} \triangleq \cup_{i=1}^{n_\pi} \langle \boldsymbol{G}_{\boldsymbol{\pi}_i}, \boldsymbol{c}_{\boldsymbol{\pi}_i}, \boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{b}_{\boldsymbol{\pi}_i} \rangle$ be the output of Algorithm 1, and $\mathcal{H}_{\boldsymbol{f}}(\mathcal{X}) = \cup_{j=1}^{n_f} \mathcal{H}_{\boldsymbol{f}_j} \triangleq \cup_{j=1}^{n_f} \langle \boldsymbol{G}_{\boldsymbol{f}_j}, \boldsymbol{c}_{\boldsymbol{f}_j}, \boldsymbol{A}_{\boldsymbol{f}_j}, \boldsymbol{b}_{\boldsymbol{f}_j} \rangle$ be an over-approximation of the graph set $\mathcal{G}_{\boldsymbol{f}}(\mathcal{X})$, i.e., $\mathcal{G}_{\boldsymbol{f}}(\mathcal{X}) \subseteq \mathcal{H}_{\boldsymbol{f}}(\mathcal{X})$. Assume Assumption 1 holds. For a CZ-represented target set, $\mathcal{T} \subset \mathbb{R}^n$, define $\langle \boldsymbol{G}_\tau, \boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{b}_\tau \rangle = \mathcal{T} \oplus (-\boldsymbol{B}\mathcal{U})$. Then, the one-step BRS of (1) is over-approximated by a union of CZs given as:

$$\overline{\mathcal{P}}(\mathcal{T}) = \bigcup_{\substack{1 \le i \le n_\pi \\ 1 \le j \le n_f}} \overline{\mathcal{H}}_{p_{i,j}} \triangleq \bigcup_{\substack{1 \le i \le n_\pi \\ 1 \le j \le n_f}} \langle \overline{\boldsymbol{G}}_{p_{i,j}}, \overline{\boldsymbol{c}}_{p_{i,j}}, \overline{\boldsymbol{A}}_{p_{i,j}}, \overline{\boldsymbol{b}}_{p_{i,j}} \rangle$$

where $\overline{\boldsymbol{G}}_{p_{i,j}} = [\boldsymbol{G}_x, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}]$, $\overline{\boldsymbol{c}}_{p_{i,j}} = \boldsymbol{c}_x$, $\overline{\boldsymbol{b}}_{p_{i,j}} = [\boldsymbol{b}_{\boldsymbol{\pi}_i}; \boldsymbol{b}_{\boldsymbol{f}_j}[n_{c_x} + 1, :]; \boldsymbol{b}_\tau; \boldsymbol{c}_\tau - \boldsymbol{c}_{\boldsymbol{f}_j}[-n :, :] - \boldsymbol{B}_\pi \boldsymbol{c}_{\boldsymbol{\pi}_i}[-m :, :]]$, and $\overline{\boldsymbol{A}}_{p_{i,j}} = [[\boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1 : n_{g_x}], \boldsymbol{A}_{\boldsymbol{\pi}_i}[:, n_{g_x} + 1 :], \boldsymbol{0}, \boldsymbol{0}]; [\boldsymbol{A}_{\boldsymbol{f}_j}[n_{c_x} + 1 :, 1 : n_{g_x}], \boldsymbol{0}, \boldsymbol{A}_{\boldsymbol{f}_j}[n_{c_x} + 1 :, n_{g_x} + 1 :], \boldsymbol{0}]; [\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{A}_\tau]; [\boldsymbol{G}_{\boldsymbol{f}_j}[-n :, 1 : n_{g_x}] + \boldsymbol{B}_\pi \boldsymbol{G}_{\boldsymbol{\pi}_i}[-m :, 1 : n_{g_x}], \boldsymbol{B}_\pi \boldsymbol{G}_{\boldsymbol{\pi}_i}[-m :, n_{g_x} + 1 :], \boldsymbol{G}_{\boldsymbol{f}_j}[-n :, n_{g_x} + 1 :], -\boldsymbol{G}_\tau]]$.

Proof: For any $\boldsymbol{x} \in \mathcal{P}(\mathcal{T}) \subset \mathcal{X}$, there exist $i \in \{1, \cdots, n_\pi\}$ and $j \in \{1, \cdots, n_f\}$ such that $[\boldsymbol{x}; \boldsymbol{\pi}(\boldsymbol{x})] \in \mathcal{H}_{\boldsymbol{\pi}_i}$ and $[\boldsymbol{x}; \boldsymbol{f}(\boldsymbol{x})] \in \mathcal{H}_{\boldsymbol{f}_j}$. Hence, there exists $\boldsymbol{\xi}_{\boldsymbol{\pi}_i} \in \mathcal{B}_\infty(\boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{b}_{\boldsymbol{\pi}_i})$ and $\boldsymbol{\xi}_{\boldsymbol{f}_j} \in \mathcal{B}_\infty(\boldsymbol{A}_{\boldsymbol{f}_j}, \boldsymbol{b}_{\boldsymbol{f}_j})$ such that $[\boldsymbol{x}; \boldsymbol{\pi}(\boldsymbol{x})] = \boldsymbol{G}_{\boldsymbol{\pi}_i}\boldsymbol{\xi}_{\boldsymbol{\pi}_i} + \boldsymbol{c}_{\boldsymbol{\pi}_i}$ and $[\boldsymbol{x}; \boldsymbol{f}(\boldsymbol{x})] = \boldsymbol{G}_{\boldsymbol{f}_j}\boldsymbol{\xi}_{\boldsymbol{f}_j} + \boldsymbol{c}_{\boldsymbol{f}_j}$. From Assumption 1 and (i) in Proposition 1, we have $\boldsymbol{A}_{\boldsymbol{f}_j}[1 : n_{c_x}, :$

$] = \boldsymbol{A}_{\boldsymbol{\pi}_i}[1:n_{c_x},:] = [\boldsymbol{A}_x, \boldsymbol{0}]$ and $\boldsymbol{G}_{\boldsymbol{f}_j}[1:n,:] = \boldsymbol{G}_{\boldsymbol{\pi}_i}[1:n,:] = [\boldsymbol{G}_x, \boldsymbol{0}]$. Therefore, partitioning $\boldsymbol{\xi}_{\boldsymbol{\pi}_i}$ and $\boldsymbol{\xi}_{\boldsymbol{f}_j}$ into $\boldsymbol{\xi}_{\boldsymbol{\pi}_i} = [\boldsymbol{\xi}_{\boldsymbol{\pi}_i,1}; \boldsymbol{\xi}_{\boldsymbol{\pi}_i,2}]$ and $\boldsymbol{\xi}_{\boldsymbol{f}_j} = [\boldsymbol{\xi}_{\boldsymbol{f}_j,1}; \boldsymbol{\xi}_{\boldsymbol{f}_j,2}]$ with $\boldsymbol{\xi}_{\boldsymbol{\pi}_i,1} \in \mathbb{R}^{n_{g_x}}$ and $\boldsymbol{\xi}_{\boldsymbol{f}_j,1} \in \mathbb{R}^{n_{g_x}}$ respectively yields $\boldsymbol{x} = \boldsymbol{G}_x\boldsymbol{\xi}_{\boldsymbol{\pi}_i,1} + \boldsymbol{c}_x = \boldsymbol{G}_x\boldsymbol{\xi}_{\boldsymbol{f}_j,1} + \boldsymbol{c}_x$ and $\boldsymbol{A}_x\boldsymbol{\xi}_{\boldsymbol{\pi}_i,1} = \boldsymbol{A}_x\boldsymbol{\xi}_{\boldsymbol{f}_j,1} = \boldsymbol{b}_x$. Therefore, $\boldsymbol{\xi}_{\boldsymbol{\pi}_i,1} = \boldsymbol{\xi}_{\boldsymbol{f}_j,1} + \boldsymbol{q}$ where $\boldsymbol{q} \in \ker(\boldsymbol{G}_x) \cap \ker(\boldsymbol{A}_x)$. Since $\ker(\boldsymbol{G}_x) \cap \ker(\boldsymbol{A}_x) \subseteq \ker(\boldsymbol{G}_{\boldsymbol{\pi}_i}[:, 1:n_{g_x}]) \cap \ker(\boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1:n_{g_x}])$ from Proposition 1, we have $\boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1:n_{g_x}]\boldsymbol{q} = \boldsymbol{0}$ and $\boldsymbol{G}_{\boldsymbol{\pi}_i}[:, 1:n_{g_x}]\boldsymbol{q} = \boldsymbol{0}$.

Since $\boldsymbol{\xi}_{\boldsymbol{\pi}_i} \in \mathcal{B}_\infty(\boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{b}_{\boldsymbol{\pi}_i})$ and $\boldsymbol{\xi}_{\boldsymbol{f}_j} \in \mathcal{B}_\infty(\boldsymbol{A}_{\boldsymbol{f}_j}, \boldsymbol{b}_{\boldsymbol{f}_j})$, it gives that $\boldsymbol{A}_{\boldsymbol{\pi}_i}\boldsymbol{\xi}_{\boldsymbol{\pi}_i} = \boldsymbol{A}_{\boldsymbol{\pi}_i}[\boldsymbol{\xi}_{\boldsymbol{f}_j,1}; \boldsymbol{\xi}_{\boldsymbol{\pi}_i,2}] + \boldsymbol{A}_{\boldsymbol{\pi}_i}[:, 1:n_{g_x}]\boldsymbol{q} = \boldsymbol{A}_{\boldsymbol{\pi}_i}[\boldsymbol{\xi}_{\boldsymbol{f}_j,1}; \boldsymbol{\xi}_{\boldsymbol{\pi}_i,2}] = \boldsymbol{b}_{\boldsymbol{\pi}_i}$ and $\boldsymbol{A}_{\boldsymbol{f}_j}\boldsymbol{\xi}_{\boldsymbol{f}_j} = [[\boldsymbol{A}_x, \boldsymbol{0}]\boldsymbol{\xi}_{\boldsymbol{f}_j}; \boldsymbol{A}_{\boldsymbol{f}_j}[n_{c_x}+1:, 1:n_{g_x}]\boldsymbol{\xi}_{\boldsymbol{f}_j,1}, \boldsymbol{A}_{\boldsymbol{f}_j}[n_{c_x}+1:, n_{g_x}+1:]\boldsymbol{\xi}_{\boldsymbol{f}_j,2}] = [\boldsymbol{b}_x; \boldsymbol{b}_{\boldsymbol{f}_j}[n_{c_x}+1:,:]]$. In addition, by Definition 2, $\mathcal{P}(\mathcal{T}) = \{\boldsymbol{x} \in \mathcal{X} \mid \exists \boldsymbol{u} \in \mathcal{U} : \boldsymbol{f}_{\text{cl}}(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{T}\} = \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{\pi}(\boldsymbol{x}) \in \mathcal{T} \oplus -\boldsymbol{B}\mathcal{U}\}$. Since $\boldsymbol{x} \in \mathcal{P}(\mathcal{T})$, there exists $\boldsymbol{\xi}_\tau \in \mathcal{B}_\infty(\boldsymbol{A}_\tau, \boldsymbol{b}_\tau)$ such that $\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{\pi}(\boldsymbol{x}) = \boldsymbol{G}_{\boldsymbol{f}_j}[-n:,:]\boldsymbol{\xi}_{\boldsymbol{f}_j} + \boldsymbol{c}_{\boldsymbol{f}_j}[-n,:] + \boldsymbol{B}_{\boldsymbol{\pi}}(\boldsymbol{G}_{\boldsymbol{\pi}_i}[-m:,:]\boldsymbol{\xi}_{\boldsymbol{\pi}_i} + \boldsymbol{c}_{\boldsymbol{\pi}_i}[-m,:]) = \boldsymbol{G}_\tau\boldsymbol{\xi}_\tau + \boldsymbol{c}_\tau$. Since $\boldsymbol{\xi}_{\boldsymbol{\pi}_i,1} = \boldsymbol{\xi}_{\boldsymbol{f}_j,1} + \boldsymbol{q}$ and $\boldsymbol{G}_{\boldsymbol{\pi}_i}[:, 1:n_{g_x}]\boldsymbol{q} = \boldsymbol{0}$, simplifying the equation above yields $\boldsymbol{G}_{\boldsymbol{f}_j}[-n:,:]\boldsymbol{\xi}_{\boldsymbol{f}_j} + \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{G}_{\boldsymbol{\pi}_i}[-m:, 1:n_{g_x}]\boldsymbol{\xi}_{\boldsymbol{f}_j,1} + \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{G}_{\boldsymbol{\pi}_i}[-m:, n_{g_x}+1:]\boldsymbol{\xi}_{\boldsymbol{\pi}_i,2} - \boldsymbol{G}_\tau\boldsymbol{\xi}_\tau = \boldsymbol{c}_\tau - \boldsymbol{c}_{\boldsymbol{f}_j}[-n:,:] - \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{c}_{\boldsymbol{\pi}_i}[-m:,:]$. Let $\boldsymbol{\xi}_{p_{i,j}} = [\boldsymbol{\xi}_{\boldsymbol{f}_j,1}; \boldsymbol{\xi}_{\boldsymbol{\pi}_i,2}; \boldsymbol{\xi}_{\boldsymbol{f}_j,2}; \boldsymbol{\xi}_\tau]$, combining all the equality constraints above yields that $\boldsymbol{\xi}_{p_{i,j}} \in \mathcal{B}_\infty(\overline{\boldsymbol{A}}_{p_{i,j}}, \overline{\boldsymbol{b}}_{p_{i,j}})$. In addition, $\boldsymbol{x} = \boldsymbol{G}_x\boldsymbol{\xi}_{\boldsymbol{f}_j,1} + \boldsymbol{c}_x = \overline{\boldsymbol{G}}_{p_{i,j}}\boldsymbol{\xi}_{p_{i,j}} + \overline{\boldsymbol{c}}_{p_{i,j}}$, which implies $\boldsymbol{x} \in \overline{\mathcal{H}}_{p_{i,j}}$. Hence, $\mathcal{P}(\mathcal{T}) \subseteq \overline{\mathcal{P}}(\mathcal{T})$. ∎

Based on Theorem 2, the K-step over-approximated BRS of NNCS (1) can be computed recursively as follows:

$$\overline{\mathcal{P}}_0(\mathcal{T}) = \mathcal{T}, \quad \overline{\mathcal{P}}_t(\mathcal{T}) = \overline{\mathcal{P}}(\overline{\mathcal{P}}_{t-1}(\mathcal{T})),\ t = 1, \ldots, K. \quad (3)$$

The set $\overline{\mathcal{P}}_t(\mathcal{T})$ in (3) is, by construction, a finite union of CZs. Since $\mathcal{G}(\boldsymbol{\pi}, \mathcal{X})$ is defined over the entire state domain $\mathcal{X}$, which is typically larger than the BRS, some CZ subsets in $\overline{\mathcal{P}}_i(\mathcal{T})$ may be empty. These empty subsets are identified via LPs and removed before propagation to the next time step to reduce computational overhead.

For the special case where $\boldsymbol{f}(\boldsymbol{x})$ in (1) is linear, the exact BRS of (1) can be obtained by the following corollary, whose proof follows directly as an extension of Theorem 2.

Corollary 1: Consider the NNCS (1) with $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$. Given $\mathcal{X} = \langle \boldsymbol{G}_x, \boldsymbol{c}_x, \boldsymbol{A}_x, \boldsymbol{b}_x \rangle$, let $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \cup_{i=1}^{n_\pi} \mathcal{H}_{\boldsymbol{\pi}_i} \triangleq \cup_{i=1}^{n_\pi} \langle \boldsymbol{G}_{\boldsymbol{\pi}_i}, \boldsymbol{c}_{\boldsymbol{\pi}_i}, \boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{b}_{\boldsymbol{\pi}_i} \rangle$ be the output of Algorithm 1. For a CZ-represented target set, $\mathcal{T} \subset \mathbb{R}^n$, define $\langle \boldsymbol{G}_\tau, \boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{b}_\tau \rangle = \mathcal{T} \oplus (-\boldsymbol{B}\mathcal{U})$. Then, the *exact* one-step BRS of (1) is given by a union of CZs: $\mathcal{P}(\mathcal{T}) = \cup_{i=1}^{n_\pi} \mathcal{H}_{p_i}$ where $\mathcal{H}_{p_i} \triangleq \langle \boldsymbol{G}_{p_i}, \boldsymbol{c}_{p_i}, \boldsymbol{A}_{p_i}, \boldsymbol{b}_{p_i} \rangle$, $\boldsymbol{G}_{p_i} = [\boldsymbol{G}_{\boldsymbol{\pi}_i}[1:n,:], \boldsymbol{0}]$, $\boldsymbol{c}_{p_i} = \boldsymbol{c}_{\boldsymbol{\pi}_i}[1:n,:]$, $\boldsymbol{A}_{p_i} = [\text{diag}(\boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{A}_\tau); [\boldsymbol{D}\boldsymbol{G}_{\boldsymbol{\pi}_i}, -\boldsymbol{G}_\tau]]$, $\boldsymbol{b}_{p_i} = [\boldsymbol{b}_{\boldsymbol{\pi}_i}; \boldsymbol{b}_\tau; \boldsymbol{c}_\tau - \boldsymbol{D}\boldsymbol{c}_{\boldsymbol{\pi}_i}]$, and $\boldsymbol{D} = [\boldsymbol{A}, \boldsymbol{B}_{\boldsymbol{\pi}}]$.

B. BRS Under-Approximation

In this subsection, we compute $\underline{\mathcal{P}}_K(\mathcal{T})$ for the NNCS (1) by utilizing the sequential linearization technique.

For a given linearization point $\boldsymbol{x}^*$, the function $\boldsymbol{f}$ is linearized so that the NNCS (1) is locally approximated as:

$$\boldsymbol{x}(t+1) = \tilde{\boldsymbol{f}}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{A}_t\boldsymbol{x}^* + \boldsymbol{L}_t \quad (4)$$

where $\tilde{\boldsymbol{f}}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \triangleq \boldsymbol{A}_t\boldsymbol{x}(t) + \boldsymbol{B}_\pi\boldsymbol{\pi}(\boldsymbol{x}(t)) + \boldsymbol{B}\boldsymbol{u}(t)$ with $\boldsymbol{A}_t = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{x}^*}$. The term $\boldsymbol{L}_t \in \mathbb{R}^n$ in (4) represents the Lagrange remainder error, where the i-th component is given by $L_{t,i} = \frac{1}{2}(\boldsymbol{x}(t) - \boldsymbol{x}^*)^\top \frac{\partial^2 f_i}{\partial \boldsymbol{x}^2}(\xi_i)(\boldsymbol{x}(t) - \boldsymbol{x}^*)$, with $\xi_i \in \{\lambda_i\boldsymbol{x}^* + (1-\lambda_i)\boldsymbol{x}(t) \mid \lambda_i \in [0,1]\}$. Within a convex input domain $\mathcal{X}_{\boldsymbol{L}_t} \subset \mathbb{R}^n$, the linearization error $\boldsymbol{L}_t$ can be over-approximated by a zonotope $\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t}) = \langle \boldsymbol{0}, |\hat{\boldsymbol{L}}_t| \rangle$ where $|\hat{\boldsymbol{L}}_t| \in \mathbb{R}^n_+$ can be obtained by interval analysis [19]. Then, over domain $\mathcal{X}_{\boldsymbol{L}_t}$, $\boldsymbol{x}(t+1)$ can be enclosed as follows:

$$\boldsymbol{x}(t+1) \in \tilde{\boldsymbol{f}}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{A}_t\boldsymbol{x}^* + \mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t}).$$

The following theorem computes an under-approximation of the one-step BRS for the NNCS (1). The key idea is to enclose the linearization error $\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t})$ within a zonotope, enabling the computation of $\underline{\mathcal{P}}(\mathcal{T})$ as the BRS of the model $\boldsymbol{x}(t+1) = \tilde{\boldsymbol{f}}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ with respect to a suitably shrunken target set.

Theorem 3: Given a CZ-represented $\mathcal{X} \subset \mathbb{R}^n$, let $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) = \cup_{i=1}^{n_\pi} \mathcal{H}_{\boldsymbol{\pi}_i} \triangleq \cup_{i=1}^{n_\pi} \langle \boldsymbol{G}_{\boldsymbol{\pi}_i}, \boldsymbol{c}_{\boldsymbol{\pi}_i}, \boldsymbol{A}_{\boldsymbol{\pi}_i}, \boldsymbol{b}_{\boldsymbol{\pi}_i} \rangle$ be output of Algorithm 1. Given a CZ-represented target set $\mathcal{T} = \langle \boldsymbol{G}_\tau, \boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{b}_\tau \rangle \subset \mathbb{R}^n$, a linearization point $\boldsymbol{x}^*$, and a zonotopic bound of the linearization error $\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t})$ over a prior set $\mathcal{X}_{\boldsymbol{L}_t} \supseteq \mathcal{P}(\mathcal{T})$, a one-step under-approximated BRS of (1) for $\mathcal{T}$ is given as $\underline{\mathcal{P}}(\mathcal{T}) \triangleq \{\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{L}_t} \mid \exists \boldsymbol{u} \in \mathcal{U} : \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u}) \in \tilde{\mathcal{T}}\}$ where $\tilde{\mathcal{T}} \triangleq \mathcal{T} \ominus (\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t}) + (\boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{A}_t\boldsymbol{x}^*))$.

Proof: From Def. 2, $\mathcal{P}(\mathcal{T}) = \{\boldsymbol{x} \in \mathcal{X} \mid \exists \boldsymbol{u} \in \mathcal{U} : \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}_\pi\boldsymbol{\pi}(\boldsymbol{x}) + \boldsymbol{B}\boldsymbol{u} \in \mathcal{T}\} = \{\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{L}_t} \mid \exists \boldsymbol{u} \in \mathcal{U} : \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{L}_t + \boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{A}_t\boldsymbol{x}^* \in \mathcal{T}\} \supseteq \{\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{L}_t} \mid \exists \boldsymbol{u} \in \mathcal{U} : \forall \boldsymbol{L}_t \in \mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t}), \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{A}_t\boldsymbol{x}^* + \boldsymbol{L}_t \in \mathcal{T}\} = \{\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{L}_t} \mid \exists \boldsymbol{u} \in \mathcal{U} : \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{T} \ominus \tilde{\mathcal{T}}\} = \underline{\mathcal{P}}(\mathcal{T})$. ∎

Remark 2: The conservatism of this under-approximation depends on the tightness of the error bound $\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t})$, which is determined by the choice of the linearization point $\boldsymbol{x}^*$ and the prior set $\mathcal{X}_{\boldsymbol{L}_t} \supseteq \mathcal{P}(\mathcal{T})$. To reduce conservatism of $\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t})$, $\boldsymbol{x}^*$ should be chosen near the geometric center of the true BRS $\mathcal{P}(\mathcal{T})$, and $\mathcal{X}_{\boldsymbol{L}_t}$ should approximate $\mathcal{P}(\mathcal{T})$ as tightly as possible. In practice, we use the BRS over-approximation from Section IV-A, setting $\mathcal{X}_{\boldsymbol{L}_t} = \square\overline{\mathcal{P}}(\mathcal{T})$ as the prior set and its center as the linearization point, where $\square$ is the interval hull operator. Moreover, increasing the number of CZs used to bound the nonlinearity $\boldsymbol{f}$ can reduce the conservatism of the BRS over-approximation, thereby tightening the error bound $\mathcal{L}(\mathcal{X}_{\boldsymbol{L}_t})$, although this comes at the expense of increased computational complexity.

The multi-step under-approximated BRS computation for the NNCS (1) is outlined in Algorithm 2. The procedure starts by constructing the NN graph set $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X})$ over the state domain $\mathcal{X}$ via Algorithm 1 (Line 1). For each CZ subset $\mathcal{T}_{t,i}$ of the target set $\mathcal{T}_t$, the function $\text{Pre}_{\text{over}}()$ computes an over-approximated BRS $\overline{\mathcal{P}}(\mathcal{T}_{t,i})$ of the NNCS (1) based on Theorem 2 (Line 5). A prior set $\mathcal{X}_{\boldsymbol{L}_t}$ is then obtained as the interval hull of $\overline{\mathcal{P}}(\mathcal{T}_{t,i})$, and its geometric center $\boldsymbol{x}^*$ is computed via $\text{center}()$ (Line 6). The one-step BRS of $\mathcal{T}_{t,i}$, $\underline{\mathcal{P}}(\mathcal{T}_{t,i})$, is computed using Theorem 3 and Corollary 1 (Line 7). Subsequently, $\underline{\mathcal{P}}_{t+1}(\mathcal{T})$ is computed as the union of $\underline{\mathcal{P}}(\mathcal{T}_{t,i})$ for $i \in \{1, 2, \ldots, r_t\}$ (Line 8), expressed as a finite union of CZ subsets for the next iteration (Line 9).

Remark 3: In practice, the number of CZ subsets in $\underline{\mathcal{P}}_t(\mathcal{T})$ may grow rapidly with t. This can be mitigated using a sam-

Algorithm 2: Multi-step Under-approximated BRSs

Input: domain $\mathcal{X}$ as a CZ, control bound $\mathcal{U}$ as a CZ, target set $\mathcal{T} \triangleq \cup_{i=1}^{r_0} \mathcal{T}_{0,i}$ as a finite union of CZs, FNN $\boldsymbol{\pi}$, a large scalar α, time step K

Output: under-approximated BRSs $\underline{\mathcal{P}}_1(\mathcal{T}), \cdots, \underline{\mathcal{P}}_K(\mathcal{T})$

1 $\mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}) \leftarrow \text{GetNNGraphSet}(\mathcal{X}, \boldsymbol{\pi}, \alpha)$
2 **for** $t = 0, 1, \ldots, K-1$ **do**
3 $\underline{\mathcal{P}}_{t+1}(\mathcal{T}) \leftarrow \emptyset$;
4 **for** $i \in \{1, \cdots, r_t\}$ **do**
5 $\overline{\mathcal{P}}(\mathcal{T}_{t,i}) \leftarrow \text{Pre}_{\text{over}}(\mathcal{T}_{t,i}, \boldsymbol{f}_{\text{cl}}, \mathcal{H}_{\boldsymbol{\pi}}(\mathcal{X}), \mathcal{U})$
6 $\mathcal{X}_{\boldsymbol{L}_t} \leftarrow \square(\overline{\mathcal{P}}(\mathcal{T}_{t,i}))$; $\boldsymbol{x}^* = \text{center}(\mathcal{X}_{\boldsymbol{L}_t})$;
7 compute $\underline{\mathcal{P}}(\mathcal{T}_{t,i})$ using Theorem 3 & Corollary 1;
8 $\underline{\mathcal{P}}_{t+1}(\mathcal{T}) = \underline{\mathcal{P}}_{t+1}(\mathcal{T}) \cup \underline{\mathcal{P}}(\mathcal{T}_{t,i})$;
9 $\cup_{i=1}^{r_{t+1}} \mathcal{T}_{t+1,i} \leftarrow \underline{\mathcal{P}}_{t+1}(\mathcal{T})$;
10 **return** $\underline{\mathcal{P}}_1(\mathcal{T}), \cdots, \underline{\mathcal{P}}_K(\mathcal{T})$

pling strategy, such as the farthest-point sampling algorithm [20] applied to the geometric centers of the CZ interval hulls to ensure uniform coverage.

C. Goal-Reaching Control Synthesis

We now address the control synthesis part of Problem 1.

From Algorithm 2, the t-step under-approximated BRS is a finite union of CZs, $\underline{\mathcal{P}}_t(\mathcal{T}) = \cup_{i=1}^{r_t} \mathcal{T}_{t,i}$, where each $\mathcal{T}_{t,i}$ is a CZ. If $\boldsymbol{x}(0) \in \underline{\mathcal{P}}_K(\mathcal{T})$, then there exists $i_0 \in \{1, 2, \ldots, r_K\}$ such that $\boldsymbol{x}(0) \in \mathcal{T}_{K,i_0}$. By the construction in Algorithm 2, a control sequence $\boldsymbol{u}(0), \boldsymbol{u}(1), \cdots, \boldsymbol{u}(K-1)$ that drives the NNCS to $\boldsymbol{x}(K) \in \mathcal{T}$ is *guaranteed to exist*.

For any CZ subset $\mathcal{T}_{t,i} \in \underline{\mathcal{P}}_t(\mathcal{T})$, where $t \in \{1, 2, \ldots, K\}$ and $i \in \{1, 2, \ldots, r_t\}$, we define its *parent* as the CZ subset $\mathcal{T}_{t-1,j} \in \underline{\mathcal{P}}_{t-1}(\mathcal{T})$ such that $\mathcal{T}_{t,i} \in \underline{\mathcal{P}}(\mathcal{T}_{t-1,j})$, as described in Lines 7-8 of Algorithm 2. The parent of a CZ is denoted by $\text{Pa}()$, e.g., $\text{Pa}(\mathcal{T}_{t,i}) = \mathcal{T}_{t-1,j}$. Using this notation, we define sets $\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_K$ for the initial condition $\boldsymbol{x}(0) \in \mathcal{T}_{K,i_0}$ iteratively as follows (see Fig. 1 for an illustration):

$$\mathcal{R}_0 = \mathcal{T}_{K,i_0}, \ \mathcal{R}_t = \text{Pa}(\mathcal{R}_{t-1}), \ t = 1, 2, \cdots, K. \tag{5}$$

Note that $\mathcal{R}_0$ depends on the choice of $i_0 \in \{1, 2, \ldots, r_K\}$, so the sets $\mathcal{R}_0, \ldots, \mathcal{R}_K$ vary with i_0. For instance, in Fig. 1, $i_0 \in \{2, 3\}$ if $\boldsymbol{x}(0) \in \mathcal{T}_{2,2} \cap \mathcal{T}_{2,3}$. The control admissible set for any $\boldsymbol{x} \in \mathcal{R}_t$ associated with i_0 is defined as

$$\mathcal{U}_t^{i_0}(\boldsymbol{x}) \triangleq \{\boldsymbol{u} \in \mathcal{U} \mid \boldsymbol{f}_{\text{cl}}(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{R}_{t+1}\}. \tag{6}$$

It is clear that $\mathcal{U}_t^{i_0}(\boldsymbol{x}) \neq \emptyset$ whenever Algorithm 2 returns non-empty BRSs $\underline{\mathcal{P}}_1(\mathcal{T}), \cdots, \underline{\mathcal{P}}_K(\mathcal{T})$, which guarantees the feasibility of the goal-reaching task. For an initial state $\boldsymbol{x}(0) \in \mathcal{R}_0$, we can select *any* $\boldsymbol{u}(t) \in \mathcal{U}_t^{i_0}(\boldsymbol{x}(t))$ such that $\boldsymbol{x}(t+1) \in \mathcal{R}_{t+1}$ for $t = 0, 1, \ldots, K-1$. Since $\mathcal{T} = \mathcal{R}_K$, it follows that $\boldsymbol{x}(K) \in \mathcal{T}$, the target set.

The minimum-norm control sequence $\{\boldsymbol{u}(t)\}_{t=0}^{K-1}$ can be synthesized iteratively by solving the following convex optimization problem at each time step t where $0 \leq t \leq K-1$:

$$\begin{aligned} &\min_{\boldsymbol{u}(t) \in \mathcal{U}} \ \|\boldsymbol{u}(t)\|^2 \\ &\text{s.t.} \ \ \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{B}_\pi \boldsymbol{\pi}(\boldsymbol{x}(t)) + \boldsymbol{B}\boldsymbol{u}(t) \in \mathcal{R}_{t+1}. \end{aligned} \tag{7}$$

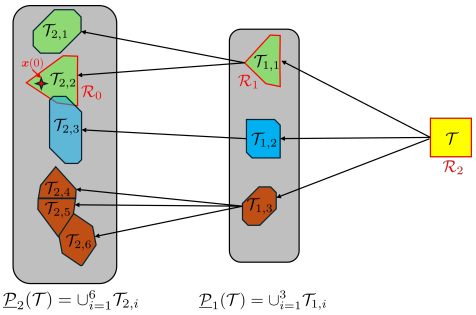

Fig. 1. Given an initial state $\boldsymbol{x(0)} \in \boldsymbol{\mathcal{T}_{2,2} \backslash \mathcal{T}_{2,3}}$ (red star) and a target set $\boldsymbol{\mathcal{T}}$ (yellow region), Algorithm 2 yields $\boldsymbol{\underline{\mathcal{P}}_1(\mathcal{T})}$ and $\boldsymbol{\underline{\mathcal{P}}_2(\mathcal{T})}$ with 3 and 6 CZ subsets, respectively. Since $\boldsymbol{i_0 = 2}$, the sets are $\boldsymbol{\mathcal{R}_0 = \mathcal{T}_{2,2}}$, $\boldsymbol{\mathcal{R}_1 = \mathcal{T}_{1,1}}$, and $\boldsymbol{\mathcal{R}_2 = \mathcal{T}}$ from (5).

Since $\mathcal{R}_{t+1}$ is a CZ, the constraint above is linear, making (7) a convex program that can be solved efficiently. Alternatively, the control sequence $\{\boldsymbol{u}(t)\}_{t=0}^{K-1}$ can be synthesized by solving a single optimization problem at $t = 0$:

$$\begin{aligned} &\min_{\substack{\{\boldsymbol{u}(t)\}_{t=0}^{K-1}, \{\boldsymbol{x}(t)\}_{t=1}^{K}, \\ 1 \leq i_0 \leq r_K}} \ \sum_{t=0}^{K-1} \|\boldsymbol{u}(t)\|^2 \\ &\text{s.t.} \ \ \boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{B}_\pi \boldsymbol{\pi}(\boldsymbol{x}(t)) + \boldsymbol{B}\boldsymbol{u}(t), \\ &\quad \boldsymbol{u}(t) \in \mathcal{U}_t^{i_0}(\boldsymbol{x}(t)), \quad t = 0, \ldots, K-1. \end{aligned} \tag{8}$$

Although (8) may be computationally demanding, it becomes convex when $\boldsymbol{f}(\boldsymbol{x})$ is linear. Note that $\boldsymbol{\pi}(\boldsymbol{x}(t))$ can be expressed as an affine function of $\boldsymbol{x}$, since each CZ subset $\mathcal{R}_t$ corresponds to a fixed activation pattern.

V. Simulation Results

We present an academic Dubins car example to demonstrate the effectiveness of the proposed methods. The example is implemented in MATLAB R2022a and run on a desktop equipped with an Intel Core i9-12900K CPU and 32 GB of RAM. The code is available at `https://github.com/wisc-arclab/goal-reaching-control-of-nncs`.

Consider the discrete-time Dubins car model:

$$\begin{aligned} p_x(t+1) &= p_x(t) + 0.1\cos(\theta(t)), \\ p_y(t+1) &= p_y(t) + 0.1\sin(\theta(t)), \\ \theta(t+1) &= \theta(t) + 0.1\boldsymbol{\pi}(\boldsymbol{x}(t)) + 0.1u(t), \end{aligned}$$

where $\boldsymbol{x} = [p_x, p_y, \theta]^\top$ denotes the state vector consisting of the horizontal coordinate, vertical coordinate, and heading angle. The nominal controller $\boldsymbol{\pi}$ is a ReLU-activated FNN with hidden layers of size $[5, 5]$, trained to approximate a nonlinear MPC controller, and u is the corrective control input to be synthesized. We choose the state domain as $\mathcal{X} = [-2, 2] \times [-2, 2] \times [-\pi, \pi]$, the input domain as $\mathcal{U} = [-0.5, 0.5]$, and the target set as $\mathcal{T} = [0.95, 1.05] \times [0.95, 1.05] \times [\pi/4 - 0.1, \pi/4 + 0.1]$. We use the functional decomposition technique in [21] to construct $\mathcal{H}_{\boldsymbol{f}}(\mathcal{X})$, which ensures Assumption 1 holds. For $K = 3$, we compute the over-approximated BRSs $\overline{\mathcal{P}}_1(\mathcal{T}), \overline{\mathcal{P}}_2(\mathcal{T}), \overline{\mathcal{P}}_3(\mathcal{T})$ via Theorem 2, and the under-approximated BRSs $\underline{\mathcal{P}}_1(\mathcal{T}), \underline{\mathcal{P}}_2(\mathcal{T}), \underline{\mathcal{P}}_3(\mathcal{T})$ via Algorithm 2. With the cost $L = \sum_{t=0}^{2} |u(t)|^2$, we synthesize the control sequence $u(0), u(1), u(2)$ by solving (8). The left plot in Fig. 2 shows the sets $\overline{\mathcal{P}}_1(\mathcal{T}), \overline{\mathcal{P}}_2(\mathcal{T}), \overline{\mathcal{P}}_3(\mathcal{T})$ and $\underline{\mathcal{P}}_1(\mathcal{T}), \underline{\mathcal{P}}_2(\mathcal{T}), \underline{\mathcal{P}}_3(\mathcal{T})$. Initial states are sampled from $\underline{\mathcal{P}}_3(\mathcal{T})$,

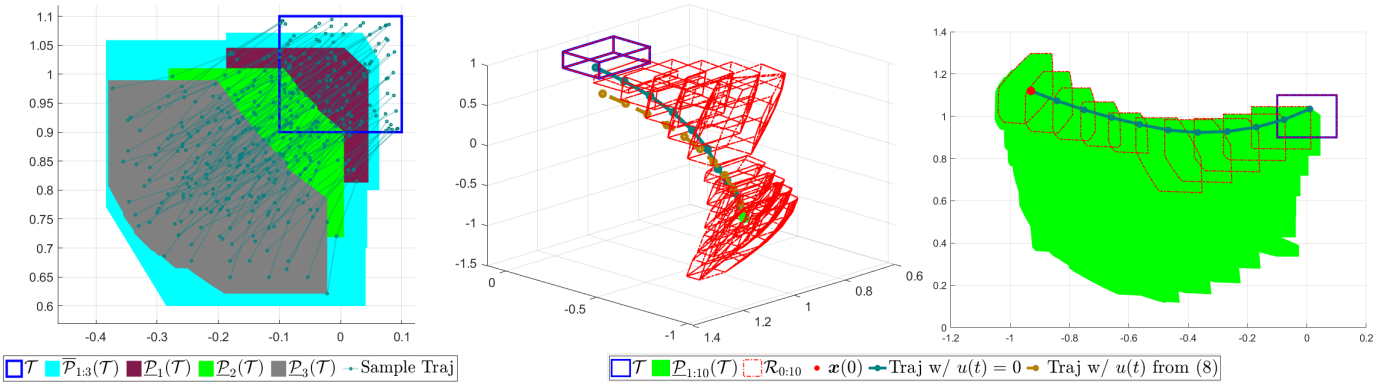

Fig. 2. (left) The $\boldsymbol{p_x}$-$\boldsymbol{p_y}$ plane projection of the target set $\boldsymbol{\mathcal{T}}$, over-approximated BRSs $\boldsymbol{\overline{\mathcal{P}}_1(\mathcal{T}), \overline{\mathcal{P}}_2(\mathcal{T}), \overline{\mathcal{P}}_3(\mathcal{T})}$, under-approximated BRSs $\boldsymbol{\underline{\mathcal{P}}_1(\mathcal{T}), \underline{\mathcal{P}}_2(\mathcal{T}), \underline{\mathcal{P}}_3(\mathcal{T})}$, and state trajectories with initial condition sampled from $\boldsymbol{\underline{\mathcal{P}}_3(\mathcal{T})}$. (middle) Under-approximated BRSs and state trajectories with and without the control sequence $\boldsymbol{u(t)}$ synthesized by (8) in 3D space. The state trajectory with $\boldsymbol{u(t) = 0}$ fails to reach the target set $\boldsymbol{\mathcal{T}}$, whereas the trajectory with $\boldsymbol{u(t)}$ synthesized from (8) successfully reaches $\boldsymbol{\mathcal{T}}$. (right) Projection onto the $\boldsymbol{p_x}$-$\boldsymbol{p_y}$ plane showing under-approximated BRSs $\boldsymbol{\underline{\mathcal{P}}_1(\mathcal{T}), \ldots, \underline{\mathcal{P}}_{10}(\mathcal{T})}$, parent CZs $\boldsymbol{\mathcal{R}_0, \ldots, \mathcal{R}_{10}}$, and the state trajectory with $\boldsymbol{u(t)}$ synthesized from (8).

and the corresponding state trajectories under the synthesized control sequence are simulated. All trajectories reach the target set $\mathcal{T}$, thereby validating the correctness of the goal-reaching control sequence.

For the initial state $\boldsymbol{x}(0) = [-0.95, 1.1, -0.5]^\top$ and time horizon $K = 10$, Algorithm 2 produces the sets $\underline{\mathcal{P}}_1(\mathcal{T}), \ldots, \underline{\mathcal{P}}_{10}(\mathcal{T})$, with 3 CZ subsets of $\underline{\mathcal{P}}_{10}(\mathcal{T})$ containing $\boldsymbol{x}(0)$. The control sequence is synthesized by solving (8) with $L = \sum_{t=0}^{9} |u(t)|^2$. The middle plot in Fig. 2 illustrates the under-approximated BRSs and the state trajectories with and without the controller synthesized by (8) in 3D space. It is evident that the state trajectory with $u(t) = 0$ fails to reach the target set $\mathcal{T}$ since $\boldsymbol{x}(10) \notin \mathcal{T}$, whereas the trajectory with $u(t)$ synthesized from (8) reaches the target set $\mathcal{T}$ since $\boldsymbol{x}(10) \in \mathcal{T}$. The right plot in Fig. 2 shows the under-approximated BRSs, parent CZs, and the state trajectory with $u(t)$ from (8), all projected onto the (p_x, p_y) plane.

VI. Conclusion

We proposed a CZ-based backward reachability analysis framework to address the goal-reaching control synthesis problem for NNCSs. The method constructs both over- and under-approximations of the BRS, enabling the synthesis of control sequences with formal feasibility guarantees. Simulation results demonstrated that the synthesized controls successfully drive the system to the target set, which validates the effectiveness of the proposed approach.

References

[1] R. Schwan, C. N. Jones, and D. Kuhn, "Stability verification of neural network controllers using mixed-integer programming," *IEEE Trans. on Autom. Control*, vol. 68, no. 12, pp. 7514–7529, 2023.

[2] F. Fabiani and P. J. Goulart, "Reliably-stabilizing piecewise-affine neural network controllers," *IEEE Trans. on Autom. Control*, vol. 68, no. 9, pp. 5201–5215, 2022.

[3] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural lander: Stable drone landing control using learned dynamics," in *International Conference on Robotics and Automation*. IEEE, 2019, pp. 9784–9790.

[4] S. Yu, C. Shen, J. Dallas, B. I. Epureanu, P. Jayakumar, and T. Ersal, "A real-time terrain-adaptive local trajectory planner for high-speed autonomous off-road navigation on deformable terrains," *IEEE Trans. on Intell. Transp. Syst.*, vol. 26, no. 3, pp. 3324–3340, 2025.

[5] T. Salzmann, E. Kaufmann, J. Arrizabalaga, M. Pavone, D. Scaramuzza, and M. Ryll, "Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2397–2404, 2023.

[6] 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.

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

[8] M. Wetzlinger and M. Althoff, "Backward reachability analysis of perturbed continuous-time linear systems using set propagation," *IEEE Trans. on Autom. Control*, 2025.

[9] N. Rober, S. M. Katz, C. Sidrane, E. Yel, M. Everett, M. J. Kochenderfer, and J. P. How, "Backward reachability analysis of neural feedback loops: Techniques for linear and nonlinear systems," *IEEE Open Journal of Control Systems*, 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 *American Control Conference*, 2023, pp. 3523–3528.

[11] Y. Zhang, H. Zhang, and X. Xu, "Backward reachability analysis of neural feedback systems using hybrid zonotopes," *IEEE Control Systems Letters*, vol. 7, pp. 2779–2784, 2023.

[12] H. Zhang, Y. Zhang, and X. Xu, "Hybrid zonotope-based backward reachability analysis for neural feedback systems with nonlinear plant models," in *American Control Conference*, 2024, pp. 4155–4161.

[13] Y. Zhang, H. Zhang, and X. Xu, "Reachability analysis of neural network control systems with tunable accuracy and efficiency," *IEEE Control Systems Letters*, vol. 8, pp. 1697–1702, 2024.

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

[15] B. Hanin and D. Rolnick, "Deep ReLU networks have surprisingly few activation patterns," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *NeurIPS*, vol. 31, 2018.

[17] E. Beale and J. Tomlin, "Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables," *Operational Research*, vol. 69, pp. 447–454, 1969.

[18] 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, pp. 1–45, 2022.

[19] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *IEEE Conference on Decision and Control*, 2008, pp. 4042–4048.

[20] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[21] J. A. Siefert, T. J. Bird, A. F. Thompson, J. J. Glunt, J. P. Koeln, N. Jain, and H. C. Pangborn, "Reachability analysis using hybrid zonotopes and functional decomposition," *IEEE Trans. on Autom. Control*, vol. 70, no. 7, pp. 4671–4686, 2025.