Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 2431–2436

Formally Verified Neural Network Control Barrier Certificates for Unknown Systems

Mahathi Anand * **Majid Zamani** *,**

* *LMU Munich, Germany 80539 Munich, Germany (e-mail: mahathi.anand@lmu.de).*

** *University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: majid.zamani@colorado.edu)*

Abstract: This paper is concerned with the controller synthesis problem for discrete-time unknown systems against safety specifications via control barrier certificates. Typically, control barrier certificates provide sufficient conditions for the satisfaction of safety specifications by separating the safe and unsafe regions of the system. By synthesizing these certificates in conjunction with control policies, one is able to keep the system safe. In our work, we parameterize the control barrier certificates and corresponding control policies as neural networks and learn them simultaneously by utilizing finitely many data samples obtained from the unknown system. We derive a so-called validity condition to formally verify the obtained certificates and integrate this condition within the training framework to achieve provably correct guarantees at the end of training time. In particular, we exploit Lipschitz continuity properties of the neural networks and utilize robust training techniques to ensure that the trained networks not only satisfy the required control barrier certificate conditions across the finitely many training data samples but over the entire state set. We then demonstrate the effectiveness of our approach with the help of a case study.

Keywords: Control barrier certificates, Data-driven approaches, Neural networks, Formal controller synthesis, Safety specifications

1. INTRODUCTION

Formal verification and synthesis of complex systems against safety specifications is of utmost importance in the field of control theory, especially due to the emergence of safety-critical applications such as autonomous vehicles, drones, robots, medical devices, etc. Methods utilizing control barrier certificates (CBCs) (Prajna and Jadbabaie, 2004; Ames et al., 2019) provide an effective mechanism to formally synthesize controllers enforcing the satisfaction of safety specifications. In particular, control barrier certificates are real-valued functions defined over the state set of the system. They act as a *barrier* between the safe and unsafe regions, providing sufficient conditions such that the barrier is never crossed by the closed-loop system trajectories. Therefore, by successfully synthesizing a CBC in conjunction with a suitable controller, one can provide formal guarantees for safety satisfaction. However, a major caveat of the aforementioned approach is that it requires mathematical knowledge about the system in the form of a model. Since precise mathematical models are not always available due to large complexity of the systems, one needs to consider alternative model-free approaches. Another drawback of the approach is the difficulty in searching for suitable CBCs. In general, one requires to fix the template of the CBC and controller beforehand (e.g. polynomial functions of a specific degree) and compute the parameters (e.g. coefficients of the polynomials) by utilizing numerical techniques such as sum-of-squares (SOS) optimization (Parrilo, 2003) or satisfiability modulo theory (SMT) solvers (De Moura and Bjørner, 2011). Unfortunately, in many cases, one fails to synthesize functions with such fixed templates due to the complexity of system dynamics (e.g. non-polynomial systems) or computational complexity.

Neural network-based safety certificate synthesis has gained considerable attention recently (Dawson et al., 2023) since it has great potential to alleviate the issues concerning model-based one. First, neural network training is completely data-driven. Therefore, by parameterizing CBCs and control policies as neural networks, one avoids the requirement of a mathematical model. Second, neural networks are capable of approximating any continuous function (Barron, 1994). As a result, they evade the limitations caused by utilizing certificates of fixed templates as in the case of model-based approaches. However, certificates based on neural networks lack formal guarantees and cannot be applied to safety-critical systems without further verification. This is due to the fact that neural networks are trained on a finite number of samples from the state set, and as a result, the trained networks are not guaranteed to satisfy the required CBC conditions over the entire state set (*i.e.* unseen samples from the state set). Therefore, one is required to formally verify the trained certificates post facto to ensure safety.

⋆ This work was supported in part by the National Science Foundation (NSF) under grant CNS-2145184 and in part by the German Research Foundation (DFG) through Research Training Group 2428.

10.1016/j.ifacol.2023.10.1219

In this context, our work proposes a training framework to synthesize provably correct CBCs and control policies parameterized as neural networks for unknown discrete-time systems without any need for post facto verification. To do this, we first derive a so-called validity condition by formulating a scenario convex problem (SCP) under Lipschitz continuity assumptions on the system dynamics as well as neural networks to verify the correctness of data-driven CBCs and corresponding control policies obtained from some training process. Then, we incorporate the obtained validity condition within the training framework by enforcing smaller Lipschitz bounds on the neural networks. This ensures robust training of the networks such that they not only satisfy the required CBC conditions over the finitely many training samples obtained from the state set, but also for all the unseen points over the state set. This way, the synthesized neural network-based CBCs and control policies are formally guaranteed to ensure that the closed-loop trajectories of the system are safe, *i.e.*, they do not cross the *barrier* and hence, visit unsafe regions. Finally, we demonstrate the applicability of our approach with the help of suitable case studies.

Related Work. Barrier certificates were first introduced by (Prajna and Jadbabaie, 2004; Prajna, 2006) in the context of safety verification for nonlinear and hybrid dynamical systems. It was later extended to safety verification for stochastic (hybrid) systems by (Prajna et al., 2007; Wisniewski and Bujorianu, 2018). Control barrier certificates were then proposed for synthesizing controllers for safety specifications of control affine systems (Ames et al., 2019; Wieland and Allgöwer, 2007) as well as stochastic control systems (Jagtap et al., 2020; Clark, 2021). All the aforementioned works require knowledge of the mathematical model in order to construct the certificates and control policies. More recently, data-driven approaches to synthesize (control) barrier certificates have gained significant attentions. In this context, safety verification via scenario convex program for unknown continuous-time nonlinear systems was proposed by (Lavaei et al., 2021). Controller synthesis under rank conditions via control barrier certificates for unknown continuous time nonlinear polynomial systems were proposed by (Nejati et al., 2022). While these works are applicable when the system models are unknown, the certificates are limited by their templates (function space), and hence, hard to find.

On the other hand, neural network-based certificate synthesis was developed in the context of safety verification for continuous-time nonlinear systems by (Zhao et al., 2020; Peruffo et al., 2021), for hybrid systems by (Zhao et al., 2021b), and for stochastic systems by (Mathiesen et al., 2023). It was later extended to address the controller synthesis problem by (Jin et al., 2020; Zhao et al., 2021a; Dawson et al., 2023). Most of the aforementioned works utilize posteriori verification techniques to formally guarantee whether the certificates obtained are valid. For example, the results in (Zhao et al., 2020, 2021a) encode the CBC constraints into an SMT problem to verify the validity of trained CBCs. The results in (Zhao et al., 2021b) transform the CBC conditions to mixed-integer linear programming (MILP) to verify its correctness. The results in (Mathiesen et al., 2023) check the validity of the trained CBCs by approximating them as linear functions by leveraging bound propagation techniques. However, these approaches consider verifying the trained CBCs a posteriori which is costly. Moreover, they also require the knowledge of the system model.

The results in (Jin et al., 2020) train neural network barrier certificates for continuous-time systems and verify their correctness a posteriori by utilizing Lipschitz continuity of the trained certificates as well as the system dynamics. In contrast, our work aims to integrate the training and verification process to synthesize provably correct CBCs at one go. Note also that the results in (Peruffo et al., 2021) synthesize formally verified neural barrier certificates via an SMT based counter-example guided synthesis (CEGIS) approach. However, they work in the continuous-time setting and require the knowledge of the system dynamics. Moreover, they only consider the safety verification problem, *i.e.*, they do not synthesize controllers for closed-loop control systems that enforce safety specifications. On the other hand, our work is capable of handling systems with unknown dynamics under some Lipschitz continuity assumptions. Moreover, we simultaneously synthesize control policies along with CBCs to provide provable guarantees over safety specifications.

2. PRELIMINARIES

2.1 Notations

We denote the set of real, positive and non-negative numbers by $\mathbb{R}, \mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$, respectively, while $\mathbb{R}^n$ denotes a real space of dimension n. Notations $\mathbb{N}$ and $\mathbb{N}_{>0}$ are utilized to denote the set of non-negative and positive integers, respectively. Given N vectors $x_i \in \mathbb{R}^{n_i}$, the corresponding column vector of dimension $\sum_i n_i$ is denoted by $x = [x_1; \ldots; x_N]$. For a vector $x \in \mathbb{R}^n$, Euclidean norm of x is denoted by $\|x\|$. For a set S, we define the indicator function of S, denoted by $\mathbb{1}_S(x)$, by $\mathbb{1}_S(x) = 1$ when $x \in S$ and 0 otherwise. The complement of a set S within a set $\mathcal{S}$ is denoted by $\mathcal{S} \setminus S$. For a matrix $A \in \mathbb{R}^{m \times n}$, the inequality $A \geq 0$ is element-wise, whereas the inequality $A \succeq 0$ means that A is positive semi-definite. Moreover, A^T denotes its transpose. Finally, the set of diagonal matrices of dimension n is denoted by $\mathcal{D}^n$, and $\mathcal{D}^n_{\geq 0}$ is the set of diagonal matrices with non-negative elements.

2.2 Problem Definition

Definition 1. (System). A discrete-time control system (dt-CS) is a tuple $\mathfrak{S} = (X, U, f)$, where $X \subseteq \mathbb{R}^n$ is the state set of the system, $U \subseteq \mathbb{R}^m$ is the input set of the system, and $f : X \times U \to X$ describes the state evolution of the system via the following difference equation:

$$x(t+1) = f(x(t), u(t)), \quad \forall t \in \mathbb{N}, \tag{1}$$

where $x(t) \in X$ and $u(t) \in U$, $\forall t \in \mathbb{N}$, denote the state and input of the system, respectively.

We consider a feedback controller $g : X \to U$ for the dt-CS $\mathfrak{S}$, such that at any time instant $t \in \mathbb{N}$, the control input is given as $u(t) = g(x(t))$. For a given initial state $x(0) = x_0$ and a controller g, we denote by $\mathbf{x}_{x_0} = (x_0, x(1), x(2), \ldots)$ the infinite state sequence generated by applying inputs $g(x(t))$ at each time step t. The focus of this paper is on unknown systems, *i.e.*, the function f in (1) unknown. However, we assume to have access to a black-box model of the system, *i.e.*, given finitely many samples $\{x_1, \ldots, x_N\}$ from the state set X and $\{u_1, \ldots, u_N\}$ from the input set U, we have $\{f(x_1, u_1), \ldots, f(x_N, u_N)\}$. Note that in the

remainder of the paper, we refer to an unknown dt-CS as simply dt-CS for ease of presentation. The aim of this paper is to synthesize a suitable controller g such that the dt-CS $\mathfrak{S}$ satisfies some safety specification by preventing $\mathfrak{S}$ from visiting some unsafe regions of the state set.

Problem 2. Given a dt-CS $\mathfrak{S}$ as in Definition 1, a set of initial states X_0 and a set of unsafe states X_u, compute a feedback controller $g : X \to U$ such that the state sequences $\mathbf{x}_{x_0}$ of $\mathfrak{S}$ starting from $x_0 \in X_0$ under the policy g do not visit any states in X_u for all time $t \in \mathbb{N}$, by using a finite number of samples collected from the system.

2.3 Control Barrier Certificates

In this section, we introduce the notion of control barrier certificates (CBC) (Prajna and Jadbabaie, 2004), which provide sufficient conditions together with controllers for the satisfaction of safety specifications. Control barrier certificates are formally defined as follows:

Definition 3. We say that a function $\mathrm{B} : X \to \mathbb{R}$ is a control barrier certificate (CBC) for a dt-CS $\mathfrak{S}$ with respect to the initial set $X_0 \subseteq X$ and unsafe set $X_u \subseteq X$ if there exists a controller $g : X \to U$ such that the following conditions hold:

$$\mathrm{B}(x) \leq 0, \qquad \forall x \in X_0, \tag{2}$$
$$\mathrm{B}(x) > 0, \qquad \forall x \in X_u, \tag{3}$$
$$\mathrm{B}(f(x, g(x))) - \mathrm{B}(x) \leq 0, \qquad \forall x \in X. \tag{4}$$

The following lemma allows us to synthesize controllers for dt-CS $\mathfrak{S}$ ensuring the satisfaction of safety properties.

Lemma 4. For a dt-CS $\mathfrak{S}$, initial set $X_0 \subseteq X$, and unsafe set $X_u \subseteq X$, the existence of a CBC B under a controller g implies that the state sequences $\mathbf{x}_{x_0}$ of $\mathfrak{S}$ starting from $x_0 \in X_0$ under g do not reach any unsafe states in X_u.

It is straightforward to see how the existence of CBC implies safety. The zero-level set of the CBC $\mathrm{B}(x) = 0$ separates the unsafe regions from the safe ones. When the system starts from an initial state x_0, $\mathrm{B}(x_0) \leq 0$ due to condition (2). Due to condition (4), which requires $\mathrm{B}(x)$ to remain non-increasing for all x, the level set cannot be crossed, and as a result, the unsafe regions cannot be reached. In order to ensure that a system is safe, it therefore suffices to compute suitable control barrier certificates with the corresponding control policies. Usually, for a known dt-CS $\mathfrak{S}$, one may compute CBC and corresponding controller by first selecting their respective templates (e.g. polynomial functions) and utilizing appropriate search techniques such as sum-of-squares optimization (Parrilo, 2003) and satisfiability modulo theories (De Moura and Bjørner, 2011).

However, there are two main challenges that prevent us from utilizing the aforementioned approaches in our setting. First, since we work with dt-CS $\mathfrak{S}$ where f is unknown, it is not possible to compute CBC and the corresponding controller via condition (4). Second, fixing the template of CBCs and control policies may be restrictive, and as a result, it may not always be possible to find a suitable CBC ensuring safety of the system. In this paper, we find a solution to overcome both challenges by parameterizing CBCs and control policies as neural networks and training the neural networks in a data-driven fashion with no requirement on the knowledge of the system or the templates. However, since data-driven CBCs and control policies computed in this way are not formally verified to satisfy conditions (2)-(4) over the entire state set, we derive a validity condition for verification in Section 3 and employ this condition within the training process in Section 4 to generate formally verified CBCs with their corresponding control policies.

3. CORRECTNESS OF CONTROL BARRIER CERTIFICATES

In this work, we represent the CBC and controller as neural networks and train them via finitely many data samples whilst providing formal guarantees on their correctness over the entire state set with the help of a so-called validity condition. This section focuses on the derivation of this validity condition that is utilized later in our proposed training framework (cf. Section 4). However, to do so, we first need to assume that the neural network candidates corresponding to CBC and controller are already given to us. In particular, we consider CBC B and controller g introduced in Definition 3 to be neural networks, denoted by $\mathrm{B}_{\theta,b}$ and $g_{\bar{\theta},\bar{b}}$, respectively, that are parameterized by their weights $\theta, \bar{\theta}$ and biases $b, \bar{b}$, respectively (refer to Section 4 for more details on the neural network parameterization). Having this, we then utilize scenario-based verification techniques (Salamati and Zamani, 2022) to obtain the validity condition that determines whether the CBC and controller satisfy conditions (2)-(4). To do this, we first reformulate the aforementioned conditions as a robust convex problem (RCP):

$$\text{RCP:} \begin{cases} \min\limits_{\eta} & \eta \\ s.t. & \max(q_k(x)) \leq 0,\ k \in \{1,2,3\}, \\ & \forall x \in X, \eta \in \mathbb{R}, \end{cases} \tag{5}$$

where

$$q_1(x) = (\mathrm{B}_{\theta,b}(x) - \eta)\mathbb{1}_{X_0}, \tag{6}$$
$$q_2(x) = (-\mathrm{B}_{\theta,b}(x) + \varepsilon - \eta)\mathbb{1}_{X_u}, \tag{7}$$
$$q_3(x) = \mathrm{B}_{\theta,b}(f(x, g_{\bar{\theta},\bar{b}}(x))) - \mathrm{B}_{\theta,b}(x) - \eta, \tag{8}$$

where ε is a small positive value to ensure the strict inequality in (3). Let the optimal solution of the RCP be η^*_{RCP}. Then, if $\eta^*_{\mathrm{RCP}} \leq 0$, then the satisfaction of conditions (6)-(8) implies the satisfaction of conditions (2)-(4), and the neural network-based CBC $\mathrm{B}_{\theta,b}$ and controller $g_{\bar{\theta},\bar{b}}$ are valid for the dt-CS $\mathfrak{S}$. However, finding a solution to the RCP is not possible due to function f in (8) being unknown. To circumvent this, one can take a sampling-based approach by appropriately selecting N data samples from the state set X, and then reformulating the RCP in (5) into a scenario convex problem (SCP). We obtain the data as follows. First, cover sets are constructed for X, X_0 and X_u, respectively. Cover of a set X (resp. X_0, X_u) is a set consisting of subsets of X (resp. X_0, X_u) whose union equals X (resp. X_0, X_u). The cover sets are constructed such that they consist of hyper-rectangles

$$H_i(x, \epsilon_i) := \{x \in X \mid -\epsilon_i \leq x - x_i \leq \epsilon_i\}, \tag{9}$$

centered at grid points $x_i \in X$, $i \in \{1, \ldots, N\}$, with $\epsilon_i \in \mathbb{R}^n$. Now, consider $\hat{\epsilon} = \max_i \|\epsilon_i\|$. Then, for any $x \in X$ (resp. X_0, X_u), there exists $x_i, i \in \{1, \ldots, N\}$, such that $\|x - x_i\| \leq \hat{\epsilon}$. The data sets $\mathcal{I}, \mathcal{U}$ and $\mathcal{E}$ corresponding to the initial set X_0, unsafe set X_u, and state set X, respectively, are then obtained by considering the representative points x_i, for all $i \in \{1, \ldots, N\}$, as

$$\mathcal{I} = \{x_i \mid x_i \in X_0,\ \forall i \in \{1,\dots,N\}\}, \tag{10}$$
$$\mathcal{U} = \{x_i \mid x_i \in X_u,\ \forall i \in \{1,\dots,N\}\},\ \text{and} \tag{11}$$
$$\mathcal{E} = \{(x_i, f(x_i, g_{\bar\theta,\bar b}(x_i))),\ \forall i \in \{1,\dots,N\}\}. \tag{12}$$

Utilizing these sets, we construct a scenario convex problem (SCP), defined as follows:

$$\text{SCP:} \begin{cases} \min\limits_{\eta} & \eta \\ s.t. & q_1(x_i) \le 0,\ \forall x_i \in \mathcal{I}, \\ & q_2(x_i) \le 0,\ \forall x_i \in \mathcal{U}, \\ & q_3(x_i) \le 0,\ \forall (x_i, f(x_i, g_{\bar\theta,\bar b}(x_i))) \in \mathcal{E}, \\ & \eta \in \mathbb{R}, i \in \{1,\dots,N\}, \end{cases} \tag{13}$$

where $q_k(x), k \in \{1,2,3\}$ are defined as in (6)-(8). Since there are finitely many data samples x_i, and SCP is a linear program with respect to the decision variable η, finding a solution to the SCP is tractable. Let the optimal solution of the SCP be η^*. Now we want to prove that η^* is also a feasible solution to the original RCP (5). To do so, one must utilize a Lipschitz continuity assumption on the system as well as conditions (6)-(8).

Assumption 5. The function f in (1) is Lipschitz continuous in variables x and u over the state set X and input set U with Lipschitz constants $\mathcal{L}_x$, and $\mathcal{L}_u$, respectively. Moreover, functions q_k, $k \in \{1,2,3\}$, are also Lipschitz continuous in x with Lipschitz constants $\mathcal{L}_{q_k}$, respectively. The maximum of constants $\mathcal{L}_{q_k}$ is denoted by $\mathcal{L}_{\max}$.

Remark 6. Note that Lipschitz continuity of conditions q_k, $k \in \{1,2,3\}$, can be guaranteed by Lipschitz continuity of the neural network CBC $\mathrm{B}_{\theta,b}$, corresponding controller $g_{\bar\theta,\bar b}$ and map f (cf. Section 4.2).

We now state the following theorem that connects the solution of the SCP to that of the RCP based on the data samples collected from the system and Assumption 5.

Theorem 7. Consider a dt-CS $\mathfrak{S}$, and initial and unsafe sets $X_0 \subseteq X$ and $X_u \subseteq X$, respectively. Moreover, let $\mathrm{B}_{\theta,b}$ and $g_{\bar\theta,\bar b}$ be the neural network-based CBC and controller obtained for $\mathfrak{S}$, respectively. For the SCP (13) constructed by utilizing N samples as given in equations (10)-(12), let η^* be the optimal value. Then under Assumption 5, if the following condition holds:

$$\mathcal{L}_{\max}\hat\epsilon + \eta^* \le 0, \tag{14}$$

then $\mathrm{B}_{\theta,b}$ and $g_{\bar\theta,\bar b}$ are valid for $\mathfrak{S}$, *i.e.*, they satisfy conditions (2)-(4).

4. TRAINING PROVABLY CORRECT NEURAL NETWORK CONTROL BARRIER CERTIFICATES

In Section 3, we derived the validity condition (14) that guarantees the correctness of neural network-based control barrier certificates and corresponding controllers. In this section, we utilize the obtained validity condition and propose a training framework to synthesize provably correct CBCs and corresponding controllers parameterized as neural networks. In particular, we train the CBC and controller simultaneously to achieve formal guarantees on their validity without requiring post facto verification by constructing suitable loss functions incorporating the satisfaction of conditions (6)-(8) as well as condition (14).

4.1 Neural Network Structure

Given a dt-CS $\mathfrak{S}$, consider $\mathrm{B}_{\theta,b}$ and $g_{\bar\theta,\bar b}$ as neural networks representing the control barrier certificate and controller, respectively. $\mathrm{B}_{\theta,b}$, the neural network parameterized by weights θ and biases b, consists of an input layer with n (*i.e.*, system dimension) neurons, and an output layer with one neuron (due to the scalar value of the CBC). Moreover, the number of hidden layers (depth) is arbitrary and is denoted by l_b. Similarly, for every hidden layer $1 \le i \le l_b$, the number of neurons in that layer is arbitrary and is denoted by h_b^i. The activation function in all the layers except the output layer is chosen to be ReLU function (*i.e.*, $\mathrm{ReLU}(s) = \max(0,s),\ s \in \mathbb{R}$). The activation function of the output layer is considered to be identity, and the resulting neural network function is obtained by recursively applying the activation functions at every layer, *i.e.*,

$$\begin{cases} x^0 = x \in \mathbb{R}^n, \\ x^{i+1} = \mathrm{ReLU}(\theta^i x^i + b^i) \quad \text{for } i \in \{0,\dots,l_b-1\}, \\ \mathrm{B}_{\theta,b}(x) = \theta^{l_b} x^{l_b} + b^{l_b}. \end{cases} \tag{15}$$

where $\mathrm{ReLU}(\cdot)$ is applied element-wise. Notations for the controller $g_{\bar\theta,\bar b}$ follow similarly. In this case, the input layer and output layer are of dimension n and m respectively, and the depth and width of the neural network are l_g and h_g^i, $1 \le i \le l_g$, respectively. While the activation function of the hidden layers is ReLU function and is formulated similar to (15), the activation function of the output layer is considered to be HardTanh function to accommodate the boundedness of the input set U (Zhao et al., 2021a). In particular, an inner-approximation of U is obtained to construct the hyper-rectangle $\hat U = \{u \in U \mid u_{\min} \le u \le u_{\max}\}$. Then, HardTanh function is applied to the output layer as

$$g_{\bar\theta,\bar b}(x) = \begin{cases} u_{\min}, & \bar\theta^{l_g} x^{l_g} + \bar b^{l_g} \le u_{\min}, \\ u_{\max}, & \bar\theta^{l_g} x^{l_g} + \bar b^{l_g} \ge u_{\max}, \\ \bar\theta^{l_g} x^{l_g} + \bar b^{l_g}, & \text{otherwise.} \end{cases} \tag{16}$$

4.2 Training with Formal Guarantees

In this section, we discuss the procedure to train the CBC $\mathrm{B}_{\theta,b}$ and corresponding controller $g_{\bar\theta,\bar b}$ for the dt-CS $\mathfrak{S}$ while ensuring the satisfaction of the validity condition (14) such that the trained CBC and controller satisfy conditions (2)-(4) over the entire state set X. To do this, one requires Lipschitz continuity of $\mathrm{B}_{\theta,b}$ and $g_{\bar\theta,\bar b}$ due to Assumption 5. Note that $\mathrm{B}_{\theta,b}$ and $g_{\bar\theta,\bar b}$ already satisfy this assumption since they consist of ReLU activation layers. First, observe that $\mathrm{B}_{\theta,b}$ is Lipschitz continuous with a bound $\mathcal{L}_b$ if:

$$\|\mathrm{B}_{\theta,b}(x) - \mathrm{B}_{\theta,b}(y)\| \le \mathcal{L}_b\|x-y\|, \quad \forall x,y \in \mathbb{R}^n.$$

The definition for the controller $g_{\bar\theta,\bar b}$ follows accordingly and the Lipschitz bound in this case is denoted by $\mathcal{L}_g$. Then, the Lipschitz constant of $\mathrm{B}_{\theta,b}$ is bounded by $\mathcal{L}_b$ if the following matrix inequality holds (Pauli et al., 2022a):

$$\underbrace{\begin{bmatrix} \mathcal{L}_b^2 I_n & -{\theta^0}^T\Lambda_1 & 0 & \cdots & 0 \\ -\Lambda_1\theta^0 & 2\Lambda_1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & -{\theta^{l_b-1}}^T\Lambda_{l_b} & 0 \\ \vdots & \ddots & -\Lambda_{l_b}\theta^{l_b-1} & 2\Lambda_{l_b} & -W_l^T \\ 0 & \cdots & 0 & -W_l & I_{{h_b}^{l_b}} \end{bmatrix}}_{:=M(\theta,\Lambda)} \succeq 0, \tag{17}$$

where $\Lambda = (\Lambda_1,\dots,\Lambda_{l_b})$, $\Lambda_i \in \mathcal{D}_{\ge 0}^{h_b^i}$, $i \in \{1,\dots,l_b\}$. We refer the readers to (Fazlyab et al., 2019; Pauli et al.,

2022b) for more details on obtaining inequality (17). Lipschitz bound of $\mathcal{L}_g$ for $g_{\bar{\theta},\bar{b}}(x)$ is ensured similarly by checking for equivalent inequality denoted by $M(\bar{\theta},\bar{\Lambda}) \succeq 0$.

We now describe the construction of suitable loss functions for the training of $\mathrm{B}_{\theta,b}$ and $g_{\bar{\theta},\bar{b}}$ such that its minimization leads to the satisfaction of conditions required by the SCP according to (13) over the training data sets (10)-(12), while also allowing the satisfaction of the validity condition (14). Consider the following sub-loss functions characterizing conditions (6)-(8):

$$L_0(\kappa,\eta) = \sum_{x\in\mathcal{I}} \mathrm{ReLU}(\mathrm{B}_{\theta,b}(x) - \eta), \tag{18}$$

$$L_1(\kappa,\eta) = \sum_{x\in\mathcal{U}} \mathrm{ReLU}(-\mathrm{B}_{\theta,b}(x) + \varepsilon - \eta), \tag{19}$$

$$L_2(\kappa,\eta) = \sum_{(x,f(x,g_{\bar{\theta},\bar{b}}(x)))\in\mathcal{E}} \mathrm{ReLU}(\mathrm{B}_{\theta,b}(f(x),g_{\bar{\theta},\bar{b}}(x)) - \mathrm{B}_{\theta,b}(x) - \eta), \tag{20}$$

where $\kappa = [\theta, b, \bar{\theta}, \bar{b}]$ and η are trainable parameters. Now, consider the loss function as a weighted sum of sub-loss functions L_0, L_1 and L_2, respectively, as

$$L(x) = c_0 L_0(\kappa,\eta) + c_1 L_1(\kappa,\eta) + c_2 L_2(\kappa,\eta), \tag{21}$$

where c_0, c_1 and c_2 are positive coefficients denoting the weights of the sub-loss functions. Furthermore, consider two additional loss functions characterizing the satisfaction of condition (14) as

$$L_M(\kappa,\Lambda,\bar{\Lambda}) = -c_{l_1}\log\det(M(\theta,\Lambda)) - c_{l_2}\log\det(M(\bar{\theta},\bar{\Lambda})), \tag{22}$$

$$L_v(\eta) = \mathrm{ReLU}(\mathcal{L}_{\max}\hat{\epsilon} + \eta), \tag{23}$$

where Λ and $\bar{\Lambda}$ are trainable parameters and $\mathcal{L}_b$ and $\mathcal{L}_g$ that appear in $M(\theta,\Lambda)$ and $M(\bar{\theta},\bar{\Lambda})$, respectively, are used to compute $\mathcal{L}_{\max}$ required in (23) (see Assumption 5). These parameters, along with $\hat{\epsilon}$, are hyper-parameters that are chosen a priori. Moreover, c_{l_1} and c_{l_2} are positive weight coefficients for the sub-loss functions in (22).

We now present the following theorem that provides formal guarantees of safety for the dt-CS $\mathfrak{S}$ by utilizing the trained neural networks corresponding to control barrier certificate $\mathrm{B}_{\theta,b}$ and controller $g_{\bar{\theta},\bar{b}}$, respectively.

Theorem 8. Consider a dt-CS $\mathfrak{S}$ with $X_0, X_u \subseteq X$ as initial and unsafe sets, respectively. Suppose $\mathcal{B}_{\theta,b}$ and $g_{\bar{\theta},\bar{b}}$ are trained neural networks representing the CBC and the corresponding feedback controller, respectively, such that the loss functions $L, L_v = 0$, and $L_M \leq 0$ over the training data sets $\mathcal{I}$,$\mathcal{U}$ and $\mathcal{E}$ as obtained in (10)-(12). Then, $\mathfrak{S}$ is guaranteed to be safe under controller $g_{\bar{\theta},\bar{b}}$, *i.e.*, the state sequences $\mathbf{x}_{x_0}$ starting from $x_0 \in X_0$ under $g_{\bar{\theta},\bar{b}}$ do not reach the unsafe states in X_u.

The training process can be explained as follows. We first fix all the hyper-parameters required for the training, including $\hat{\epsilon}, \mathcal{L}_b, \mathcal{L}_c$, $c = [c_0, c_u, c_b, c_{l_1}, c_{l_2}]$, and the maximum number of epochs considered n_{ep}. Note that at every epoch, we perform the training over several batches denoted by n_{bat}, which is also fixed a priori. This means that the training data sets $\mathcal{I}$,$\mathcal{U}$ and $\mathcal{E}$ are randomly shuffled into several batches, and the loss is calculated for each batch at a time. Then the trainable parameters $\kappa, \Lambda, \bar{\Lambda}$ and η maybe updated by utilizing Adam or stochastic gradient descent (SGD) optimization algorithm with a specified learning rate lr (Ruder, 2016). Note that different learning rates may be chosen for different training parameters. When the cumulative loss (*i.e.* epoch loss) across all the batches is minimized according to Theorem 8, the training process concludes with success and we obtain the CBC $\mathrm{B}_{\theta,b}$ and corresponding controller $g_{\bar{\theta},\bar{b}}$. If the algorithm does not converge, one cannot judge the safety of dt-CS $\mathfrak{S}$ with the specified hyper-parameters and considered initial parameters of κ, η, Λ and $\bar{\Lambda}$. It must be noted that the initial feasibility of inequality (17) is essential for training convergence due to the way that the loss L_M is formulated in (22). This can be ensured by choosing sufficiently small initial weights and biases for $\mathrm{B}_{\theta,b}$ and $g_{\bar{\theta},\bar{b}}$ (Pauli et al., 2022a). In practice, one may also utilize the pre-training and fine-tuning approach by (Zhao et al., 2021a) for better convergence of the algorithm.

Remark 9. In some cases, especially when the training data set contains an equilibrium point of the closed-loop system, it might be impossible for one to achieve the satisfaction of condition (8) with $\eta < 0$, and as a result, the training algorithm may never converge. In order to circumvent this issue, without any loss of generality one can remove a small neighborhood of the equilibrium point from the training data set.

5. CASE STUDY

For our case study, we consider the discrete-time dynamics of an inverted pendulum given by

$$\mathfrak{S}: \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) + \tau(x_2(k)) \\ x_2(k) + \tau(\frac{g}{l}(sin(x_1(k)) + \frac{1}{ml^2}u) \end{bmatrix},$$

where x_1, x_2 are the angular position and velocity of the pendulum, respectively, and the constants $m = 1$ and $l = 1$ are the mass and length of the pendulum, respectively. Moreover, $g = 9.8$ is the gravitational acceleration, and $\tau = 0.01$ is the sampling time. The domain is given as $X = [-\frac{\pi}{4}, \frac{\pi}{4}]^2$, the initial set is $X_0 = [-\frac{\pi}{15}, \frac{\pi}{15}]^2$, the safe set is $X_s = [-\frac{\pi}{6}, \frac{\pi}{6}]^2$, and the unsafe set X_u is derived by taking the complement of the safe set as $X_u = X \setminus X_s$. Moreover, the input is considered to be bounded within set $U = [-10, 10]$. We assume that the model is *unknown*. However, we assume that Lipschitz constants of $\mathfrak{S}$ are known to us as $L_x = 1.1$ and $L_u = 0.01$. Note that if L_f and L_u are not available, one can estimate their values by generating data from $\mathfrak{S}$ and utilizing reverse Weibull distribution (Wood and Zhang, 1996).

The goal is to synthesize a feedback controller $g_{\bar{\theta},\bar{b}}$ as a neural network in order to keep the pendulum within the safe regions by utilizing the control barrier certificate $\mathrm{B}_{\theta,b}$, also parameterized as a neural network. To do this, we first fix the training hyper-parameters $\hat{\epsilon} = 0.00016$, $\mathcal{L}_b = 2$, and $\mathcal{L}_g = 22$. Then, we compute $\mathcal{L}_{\max} = \max(\mathcal{L}_b, \mathcal{L}_b(\mathcal{L}_x + \mathcal{L}_g\mathcal{L}_u + 1)) = 4.2$. Then, we fix the structure of $\mathrm{B}_{\theta,b}$ as $l_b = 1$ and $h_b = 20$, whereas that of $g_{\bar{\theta},\bar{b}}$ is fixed as $l_g = 1$ and $h_g = 5$. By considering the training data obtained according to (10)-(12), we perform the training to simultaneously minimize the loss functions L, L_M, and L_v. The training algorithm then converges to obtain the CBC $\mathrm{B}_{\theta,b}$ and $g_{\bar{\theta},\bar{b}}$ along with $\eta = -0.0007637$. By utilizing Theorem 8, we can then confirm that the obtained CBC $\mathrm{B}_{\theta,b}$ and controller $g_{\bar{\theta},\bar{b}}$ are valid and safety is ensured. The obtained $\mathrm{B}_{\theta,b}$ and its level set $\mathrm{B}_{\theta,b} = 0$ is illustrated in Figures 1 and 2, respectively. The successful runs of the algorithm have an average convergence time of 15 minutes,

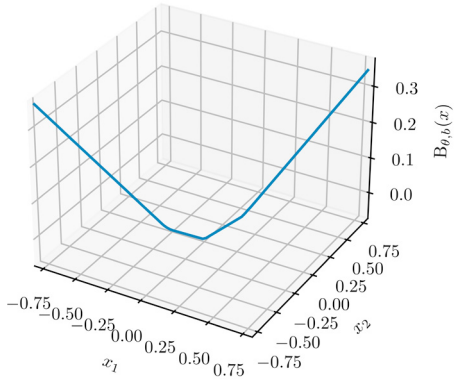

Fig. 1. Plot of $B_{\theta,b}$ over X.

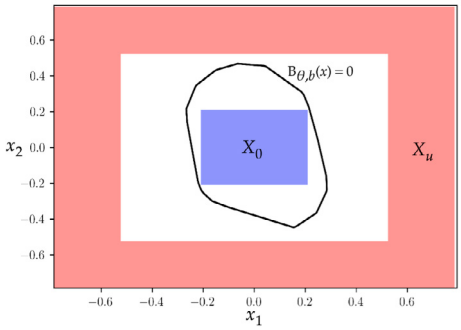

Fig. 2. The level set $B_{\theta,b} = 0$ dividing the unsafe regions X_u (in red) from the safe ones.

and the training data generation takes an additional time of 7 seconds. The computations were performed using PyTorch in Python 3.9 by modifying the **nncontroller** tool developed by (Zhao et al., 2020) on a machine with Linux Ubuntu (Intel i7-8665U CPU, with 32 GB of RAM).

REFERENCES

Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *European Control Conference*, 3420–3431.

Barron, A.R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1), 115–133.

Clark, A. (2021). Control barrier functions for stochastic systems. *Automatica*, 130, 109688.

Dawson, C., Gao, S., and Fan, C. (2023). Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 1–19.

De Moura, L. and Bjørner, N. (2011). Satisfiability modulo theories: Introduction and applications. *Commun. ACM*, 54(9), 69–77.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G.J. (2019). Efficient and accurate estimation of lipschitz constants for deep neural networks. In *International Conference on Neural Information Processing Systems*.

Jagtap, P., Soudjani, S., and Zamani, M. (2020). Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control*, 1–1.

Jin, W., Wang, Z., Yang, Z., and Mou, S. (2020). Neural certificates for safe control policies. *arXiv: 2006.08465*.

Lavaei, A., Nejati, A., Jagtap, P., and Zamani, M. (2021). Formal safety verification of unknown continuous-time systems: A data-driven approach. In *International Conference on Hybrid Systems: Computation and Control*.

Mathiesen, F.B., Calvert, S.C., and Laurenti, L. (2023). Safety certification for stochastic systems via neural barrier functions. *IEEE Control Systems Letters*, 7, 973–978.

Nejati, A., Zhong, B., Caccamo, M., and Zamani, M. (2022). Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates. In *Annual Learning for Dynamics and Control Conference*, volume 168, 763–776.

Parrilo, P.A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2), 293–320.

Pauli, P., Funcke, N., Gramlich, D., Msalmi, M.A., and Allgöwer, F. (2022a). Neural network training under semidefinite constraints. *arXiv: 2201.00632*.

Pauli, P., Koch, A., Berberich, J., Kohler, P., and Allgöwer, F. (2022b). Training robust neural networks using lipschitz bounds. *IEEE Control Systems Letters*, 6, 121–126.

Peruffo, A., Ahmed, D., and Abate, A. (2021). Automated and formal synthesis of neural barrier certificates for dynamical models. In *Tools and Algorithms for the Construction and Analysis of Systems*, 370–388.

Prajna, S. (2006). Barrier certificates for nonlinear model validation. *Automatica*, 42(1), 117–126.

Prajna, S. and Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, 477–492.

Prajna, S., Jadbabaie, A., and Pappas, G.J. (2007). A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8), 1415–1428.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv:1609.04747*.

Salamati, A. and Zamani, M. (2022). Data-driven safety verification of stochastic system via barrier certificates: A wait-and-judge approach. In *Annual Learning for Dynamics and Control Conference*, 441–452.

Wieland, P. and Allgöwer, F. (2007). Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12), 462–467.

Wisniewski, R. and Bujorianu, M.L. (2018). Stochastic safety analysis of stochastic hybrid systems. In *IEEE Conference on Decision and Control*, 2390–2395.

Wood, G.R. and Zhang, B.P. (1996). Estimation of the Lipschitz constant of a function. *Journal of Global Optimization*, 8(1), 91–103.

Zhao, H., Zeng, X., Chen, T., and Liu, Z. (2020). Synthesizing barrier certificates using neural networks. In *International Conference on Hybrid Systems: Computation and Control*.

Zhao, H., Zeng, X., Chen, T., Liu, Z., and Woodcock, J. (2021a). Learning safe neural network controllers with barrier certificates. *Formal Aspects of Computing*, 33(3), 437–455.

Zhao, Q., Chen, X., Zhang, Y., Sha, M., Yang, Z., Lin, W., Tang, E., Chen, Q., and Li, X. (2021b). Synthesizing relu neural networks with two hidden layers as barrier certificates for hybrid systems. In *International Conference on Hybrid Systems: Computation and Control*.