

Transfer of Safety Controllers Through Learning Deep Inverse Dynamics Model

Alireza Nadali, Ashutosh Trivedi, and Majid Zamani

University of Colorado Boulder, Boulder, CO 80301, USA

(e-mail: {a_nadali, ashutosh.trivedi, majid.zamani}@colorado.edu).



Abstract:

Control barrier certificates have proven effective in formally guaranteeing the safety of the control systems. However, designing a control barrier certificate is a time-consuming and computationally expensive endeavor that requires expert input in the form of domain knowledge and mathematical maturity. Additionally, when a system undergoes slight changes, the new controller and its correctness certificate need to be recomputed, incurring similar computational challenges as those faced during the design of the original controller. Prior approaches have utilized transfer learning to transfer safety guarantees in the form of a barrier certificate while maintaining the control invariant. Unfortunately, in practical settings, the source and the target environments often deviate substantially in their control inputs, rendering the aforementioned approach impractical. To address this challenge, we propose integrating *inverse dynamics*—a neural network that suggests required action given a desired successor state—of the target system with the barrier certificate of the source system to provide formal proof of safety. In addition, we propose a validity condition that, when met, guarantees correctness of the controller. We demonstrate the efficacy of our method with an inverted pendulum case study.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Controller synthesis, Neural networks, Control barrier certificates.

1. INTRODUCTION

The ever-increasing presence of autonomy in our safety-critical infrastructure—such as self-driving cars, robotics, implantable medical devices, and power grids—has underscored the importance of guaranteed safety in cyber-physical systems. While formal verification of cyber-physical systems against safety requirements is an undecidable problem (Alur et al., 1996), deductive verification approaches have shown considerable promise. Control barrier certificates (CBCs) (Prajna and Jadbabaie, 2004) are leading deductive approach for effectively synthesizing safe controllers. CBCs, real-valued functions of the state space, act as a *barrier* between over-approximation of reachable set by a system and unsafe set, guaranteeing safety. On the other hand, computing a CBC, if one exists, requires a search in the space of templates of certain form, using optimization methods such as sum-of-squares (SOS) (Parrilo, 2003) optimizations or satisfiability modulo theory (SMT) solvers (De Moura and Bjørner, 2011). Such search is computationally expensive, and requires a nontrivial understanding of the system and optimization approaches. Moreover, systems undergoing even minor changes may render CBCs unusable and necessitate a fresh synthesis of CBCs. This paper proposes a *transfer learning approach* to transfer a control barrier certificate between systems “close” to one-another.

The need for transferring control. Existing controllers may require modification for a variety of reasons including: 1) mechanical wear and tear; 2) upgrade in sensors or actuators; 3) different operating conditions such as ambient temperature or pressure; and 4) mismatch between simulation and reality. Notice that such changes, while causing a slight discrepancy in control and in guarantees, do not alter the behavior of a system drastically. Transfer learning provides a learning-based paradigm to adapt to changes precisely in such settings.

Transfer Learning for Control. Transfer learning approaches (Weiss et al., 2016) are concerned with utilizing a previously learned “knowledge” in the *source* domain in order to apply it to the *target* domain. Classical research in transfer learning was concerned with leveraging learned weights of a neural network from a source domain to speed up training in a related target domain (Bozinovski, 2020; Torrey and Shavlik, 2010). Christiano et al. (2016) proposed an approach to transfer controller from the source domain to the target domain by learning the *inverse dynamics* (Lane and Stengel, 1988) of the target system as a neural network. While showing practical success, this approach did not focus on transferring safety guarantees from one domain to another.

Transfer Learning for Safety Guarantees. In a recent work, Nadali et al. (2023) advocated the use of transfer learning to transfer the proof of safety in the form of barrier certificates while using the same controller. The application of their approach, while effective in lifting guarantees, is limited to the settings where the source and the target domains are close enough to permit similar con-

* This work was supported in part by the NSF through the CAREER awards CCF-2146563 and CNS-2145184, and grants CNS-2039062 and CNS-2111688.

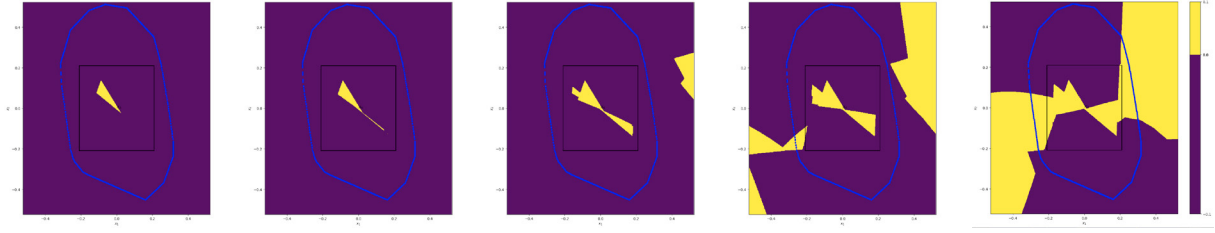


Fig. 1. This figure depicts the dynamics for a series of inverted pendula (angular position on x axis and angular velocity on y) differing in their weights and length. The leftmost figure is for the source system for which a CBC has already been computed. We show the gradual failure of the source CBC with changes in the dynamics of an inverted pendulum (details in Section 4). The blue enclosed region and black square region indicate the zero level set of the CBC and the initial set, respectively. Moreover, the yellow regions show the violation in a key condition (i.e., decrement in barrier value along transitions) of CBC, while the purple regions show its satisfaction.

trol. We posit that in several practical settings (depicted in Figure 1), two systems may be far enough so as to preclude the use of same control, still may share the “logical control structure” to merit a transfer learning of safety guarantees through learning a controller.

Transfer Learning for Control and Safety Guarantees. We utilize the idea of learning inverse dynamics (Christiano et al., 2016) with the idea of transferring guarantees (Nadali et al., 2023), to provide an approach to transfer controller while lifting formal guarantees. Naïvely, one can learn inverse dynamics first, and then use the approach of Nadali et al. (2023) to transfer the barrier certificate. Unfortunately, if for a given inverse dynamics, a barrier certificate cannot be transferred, it provides no further information to modify the inverse dynamics. To overcome this challenge, we propose barrier certificate guided inverse dynamics algorithm, which utilizes a previously learned CBC for the source system to learn a controller with safety guarantees. In particular, we transform the conditions of CBCs into gradients to train an inverse dynamics controller that provides safety guarantees.

Contribution. We propose a data-driven approach to synthesize provably correct controllers for target systems while taking advantage of previously learned CBCs for source systems. Our approach relies on a learned neural network known as an *inverse dynamics model* to act as a controller for a target system. Moreover, we implement a validity condition within the training of neural networks which ensures correctness of the transferred controller, without the need of post-facto verification. This so-called validity condition is based on Lipschitz continuity of the source and the target dynamics, and the CBC of the source system. Our method provides a formal proof of safety for a transferred controller, i.e., the closed-loop trajectories of system do not enter the unsafe set. We illustrate the effectiveness of our algorithm with an inverted pendulum case study.

Related Work. Results in (Prajna et al., 2007; Huang et al., 2017) expand on CBCs for the safety verification of stochastic systems. Control barrier certificates were then proposed for controller synthesis of deterministic systems (Wieland and Allgöwer, 2007; Ames et al., 2019) as well as stochastic ones (Jagtap et al., 2020; Clark, 2021).

There are some drawbacks to the aforementioned methods, as they require mathematical model of a given system. A

model of a system is not always available, due to security concerns (one is protecting its intellectual property) or the complexity of the system. Moreover, these methods are computationally expensive, since one requires to fix the template of barrier certificates and its controller beforehand (typically in the form of polynomial functions of a certain degree), and then search for its parameters (coefficients of those functions). In most cases, this search relies on optimization methods such as sum-of-squares (SOS) (Parrilo, 2003) optimizations or satisfiability modulo theory (SMT) solvers (De Moura and Bjørner, 2011). One often fails to find a barrier certificate with its corresponding controller due to the fix template or computational complexity. In order to guarantee safety of systems with unknown models, one should rely on data-driven methods. The results in (Nejati et al., 2023) propose a method based on scenario convex program for safety verification of unknown continuous systems, whereas results in (Nejati et al., 2022) address the controller synthesis.

Neural network-based safety guarantees have gained considerable attention in recent years (Dawson et al., 2022; Zhou et al., 2022). Neural networks are universal approximators (Hornik et al., 1989) and can represent any Borel-measurable function, therefore they do not suffer from the limitations of a fixed template. Moreover, neural networks are trained via finitely many data points and their training is completely data-driven, thus it does not need a precise mathematical model.

2. PROBLEM DEFINITION

We denote the set of real and non-negative reals by $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$, respectively. We denote the cardinality of the set A by $|A|$; and denote the set difference and Cartesian product of sets A and B by $A \setminus B$ and $A \times B$, respectively. We consider n -dimensional Euclidean space $\mathbb{R}^n$ equipped with infinity norm, defined as $\|x - y\| = \max_{1 \leq i \leq n} |x_i - y_i|$ for $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n$. Similarly, we denote Euclidean norm as $\|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$.

2.1 Discrete-time Control Systems

In this paper, we focus on the safety problem for discrete-time control systems (dtCS).

Definition 1 A discrete-time control system (dtCS) is a tuple $\mathfrak{S} = (\mathcal{X}, U, f)$, where $\mathcal{X} \subset \mathbb{R}^n$ represents the state

set, $U \subset \mathbb{R}^m$ is the set of inputs, and $f : \mathcal{X} \times U \rightarrow \mathcal{X}$ is the state transition function. The evolution of the system under input sequence $u = \langle u(1), u(2), \dots \rangle$ is described by:

$$\mathfrak{S} : x(t+1) = f(x(t), u(t)), \quad \text{for all } t \in \mathbb{N}. \quad (1)$$

We assume the state and input sets, $\mathcal{X}, U$, respectively, to be compact, and the map f is unknown but can be simulated via a black-box representation. Moreover, we assume that f is Lipschitz continuous as in the following assumption:

Assumption 2 (Lipschitz Continuity) Consider a discrete time control system $\mathfrak{S} = (\mathcal{X}, U, f)$. The map f is Lipschitz continuous such that for all $x, x' \in \mathcal{X}$, and $u, u' \in U$ one gets:

$$\|f(x, u) - f(x', u')\| \leq \mathcal{L}_x \|x - x'\| + \mathcal{L}_u \|u - u'\|, \quad (2)$$

for some positive constants $\mathcal{L}_u$ and $\mathcal{L}_x$.

2.2 Safety and Control Barrier Certificate

A dtCS $\mathfrak{S} = (\mathcal{X}, U, f)$ is safe with respect to initial set of states $\mathcal{X}_0 \subseteq \mathcal{X}$ and unsafe set $\mathcal{X}_u \subseteq \mathcal{X}$ if there exists a feedback controller $k : \mathcal{X} \rightarrow U$ such that for every trace $\langle x(0), x(1), \dots \rangle$, where $x(t+1) = f(x(t), k(x(t)))$, we have that $x(t) \notin \mathcal{X}_u$ for all $t \in \mathbb{N}$. We employ the following notion of control barrier certificates (CBCs) (Prajna and Jadbabaie, 2004) which provides sufficient conditions for ensuring safety.

Definition 3 Consider a system $\mathfrak{S} = (\mathcal{X}, U, f)$. A function $B : \mathcal{X} \rightarrow \mathbb{R}$ is called a control barrier certificate (CBC) for $\mathfrak{S}$ with respect to initial set of states $\mathcal{X}_0 \subseteq \mathcal{X}$ and unsafe set $\mathcal{X}_u \subseteq \mathcal{X}$ if there exists a controller $k : \mathcal{X} \rightarrow U$ such that, for some $\eta \in \mathbb{R}_{\geq 0}$, we have:

$$B(x) \leq -\eta, \quad \forall x \in \mathcal{X}_0; \quad (3)$$

$$B(x) > \eta, \quad \forall x \in \mathcal{X}_u; \text{ and} \quad (4)$$

$$B(f(x, k(x))) - B(x) \leq -\eta, \quad \forall x \in \mathcal{X}. \quad (5)$$

The existence of a barrier certificate for $\mathfrak{S}$ implies that every state sequences starting from $\mathcal{X}_0$ under inputs provided by k , will never reach $\mathcal{X}_u$ (Prajna and Jadbabaie, 2004).

2.3 Neural Networks

Neural networks are universal approximators (Hornik et al., 1989). Therefore, these networks can learn any Borel-measurable function based on input-output data. Consider a network with k fully-connected layers where each layer i is characterized with a weight matrix W_i and a bias vector b_i of appropriate size and is followed by an activation function. One can train neural networks on finitely many data points.

A neural network with $k \in \mathbb{N}$ layers can be viewed as a function $F : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_o}$. Given an input $y_0 \in \mathbb{R}^{n_i}$, a neural network will compute an output $y_k \in \mathbb{R}^{n_o}$ as follows:

$$\begin{aligned} y_1 &= \sigma(W_1 y_0 + b_1), \\ y_2 &= \sigma(W_2 y_1 + b_2), \\ &\vdots \\ y_k &= \sigma(W_k y_{k-1} + b_k). \end{aligned}$$

We call y_{i-1} and y_i for $i \in \{1, \dots, k\}$ the input and output of the i -th layer, respectively. One observes that neural networks with ReLU ($\sigma(x) = \max(0, x)$) activations describe Lipschitz continuous functions. Moreover, one can obtain the trivial upper bound of Lipschitz constant of a neural network with ReLU activations by multiplying the largest weight of each layer. Neural networks are trained on a appropriately defined loss function using a gradient based optimization method (Goodfellow et al., 2016).

2.4 Problem Definition

We seek to train a neural network to act as a controller for the target system, while utilizing the CBC and its controller for the source system. The main problem we aim to solve in this paper is formalized below.

Controller Transfer. Consider two systems: the *source* $\mathfrak{S} = (\mathcal{X}, U, f)$ and the *target* $\mathcal{T} = (\mathcal{X}, \hat{U}, \hat{f})$. Assume that a control barrier certificate $B : \mathcal{X} \rightarrow \mathbb{R}$ and its corresponding controller $k : \mathcal{X} \rightarrow U$ with respect to initial and unsafe sets, $\mathcal{X}_0, \mathcal{X}_u$, respectively, are available for the source system. Furthermore, both CBC and its controller are Lipschitz continuous. The *controller transfer problem* is to synthesize a controller $\hat{k} : \mathcal{X} \rightarrow \hat{U}$ for $\mathcal{T}$ to ensure safety of $\mathcal{T}$ under $\hat{k}$, with respect to initial and unsafe sets, $\mathcal{X}_0, \mathcal{X}_u$, respectively

3. CONTROLLER SYNTHESIS USING INVERSE DYNAMICS

In this section, we propose an algorithm to leverage previously learned CBCs and their corresponding controllers for the source systems, to synthesize a provably correct controller for the target systems which ensures safety.

Since both source and target systems share the same state set, it can be readily observed that conditions (3) and (4) are satisfied for the target system, with the same CBC as in the source system.

Instead of directly applying the controller of the source system to the target one, we aim to transfer high-level properties of the source system's controller and learn the low-level properties using a deep inverse dynamics model. At each time step, our approach computes the expected actions of the source system's controller. Instead of implementing these actions on the target system, we simulate the expected next state of the source system. We then rely on a deep inverse dynamics model to determine a suitable input for the target system. This appropriate action steers the target system toward the expected state of the source system. Consider the current state of the source system to be $x(t)$ with its corresponding control input $k(x(t))$. Using the black-box representation of the source system, one can obtain the next state of the source system as $x(t+1) = f(x(t), k(x(t)))$. Given the CBC for the source system, $x(t+1)$ satisfies condition (5). Our objective is to learn a controller for the target system such that the next state of the target system matches that of the source system. In other words, we aim to learn a controller (a.k.a inverse dynamics controller) $\hat{k}(x) : \mathcal{X} \rightarrow \hat{U}$ so that the target system follows the trajectory of the source system at each step, assuming both start from the same initial

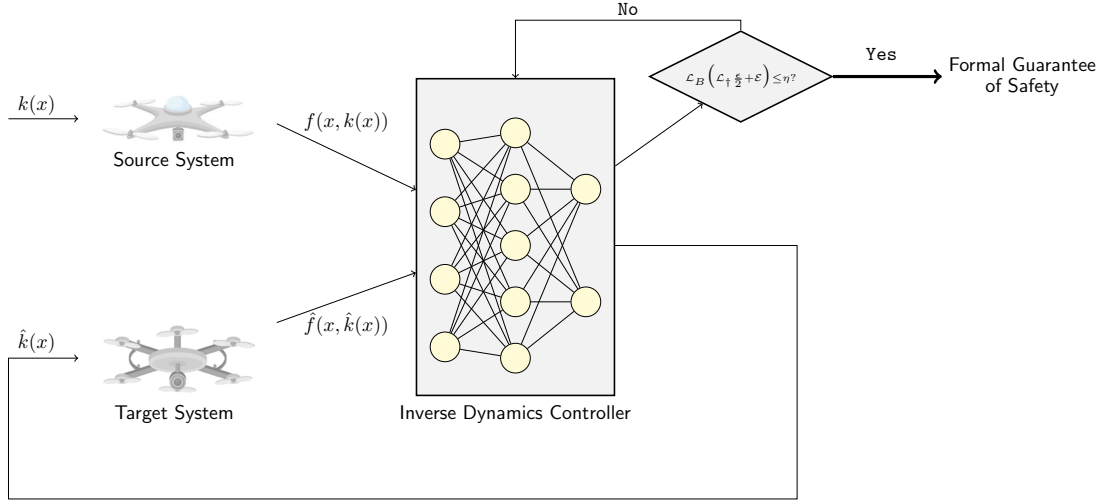


Fig. 2. Transfer of Safety Controllers Through Learning Deep Inverse Dynamics Model.

state. Since the source system is safe, the target system will never go into unsafe set as well.

In order to generate finitely many training data points, we partition the state set $\mathcal{X}$ into finitely many cells $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_M$, by picking a discretization parameter $\epsilon > 0$. We then pick sample points $x_i \in \mathcal{X}_i$ from each of these cells such that:

$$\|x - x_i\| \leq \frac{\epsilon}{2}, \text{ for all } x \in \mathcal{X}_i. \quad (6)$$

Let us denote the set of all those sampled points by $\mathcal{X}_d$. One way of partitioning the state set into such cells, is to partition it into hyperrectangles. We then pick the centers of these hyperrectangles as representative points. Finally, we employ the mean squared error (MSE) loss to train the neural network:

$$L = \frac{1}{2|\mathcal{X}_d|} \sum_{x_i \in \mathcal{X}_d} (\|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\|_2^2). \quad (7)$$

Here, $\hat{k}(x_i)$ represents the output of the neural network. Figure 2 depicts an overview of our method.

3.1 Validity Condition

Neural networks are trained on finitely many data points. Thus, we do not have guarantee for unseen data. In order to transfer formal guarantee, we propose a validity condition based on the mismatch between f and $\hat{f}$ over the finitely many data points:

$$\mathcal{E} := \max_{x_i \in \mathcal{X}_d} \|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\|. \quad (8)$$

Now, we state the main theoretical result of the paper that provides a validity condition, under which the trained controller is formally correct.

Theorem 4 Consider a source system $\mathfrak{S} = (\mathcal{X}, U, f)$ with a control barrier certificate $B : \mathcal{X} \rightarrow \mathbb{R}$ with the corresponding parameter η in (3)-(5), corresponding controller $k : \mathcal{X} \rightarrow U$, and a target system $\hat{\mathfrak{S}} = (\mathcal{X}, \hat{U}, \hat{f})$. Both k and B are assumed to be Lipschitz continuous with $\mathcal{L}_k$ and $\mathcal{L}_B$ as their Lipschitz constant, respectively. Moreover, we

assume that f and $\hat{f}$ satisfy Assumption 2 with Lipschitz constants $\mathcal{L}_x, \mathcal{L}_u$ for the source system and $\mathcal{L}_{\hat{x}}, \mathcal{L}_{\hat{u}}$ for the target one, respectively. Let $\mathcal{X}_d$ be a finite set of sampled data points according to (6) with discretization parameter ϵ , and $\mathcal{X}_0, \mathcal{X}_u \subseteq \mathcal{X}$ to be the corresponding set of initial and unsafe states, respectively. The target system with an inverse dynamics controller $\hat{k} : \mathcal{X} \rightarrow \hat{U}$ synthesized according to Algorithm 1, is guaranteed to be safe.

Proof. As mentioned in (6), for every $x \in \mathcal{X}$, there exists $x_i \in \mathcal{X}_d$ such that $\|x - x_i\| \leq \frac{\epsilon}{2}$. Thus, one obtains:

$$\begin{aligned} & \|f(x, k(x)) - \hat{f}(x, \hat{k}(x))\| \\ & \leq \|f(x, k(x)) - f(x_i, k(x_i)) + f(x_i, k(x_i)) \\ & \quad - \hat{f}(x, \hat{k}(x)) + \hat{f}(x_i, \hat{k}(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\| \end{aligned} \quad (9)$$

$$\begin{aligned} & \leq \|f(x, k(x)) - f(x_i, k(x_i))\| + \|\hat{f}(x_i, \hat{k}(x_i)) - \hat{f}(x, \hat{k}(x))\| \\ & \quad + \|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\| \end{aligned} \quad (10)$$

$$\begin{aligned} & \leq \mathcal{L}_x \|x - x_i\| + \mathcal{L}_{\hat{x}} \|x - x_i\| + \mathcal{L}_u \|k(x) - k(x_i)\| \\ & \quad + \mathcal{L}_{\hat{u}} \|\hat{k}(x) - \hat{k}(x_i)\| + \|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\|, \end{aligned} \quad (11)$$

for all $x \in \mathcal{X}$, $x_i \in \mathcal{X}_d$. Here, the inequality (10) follows from the triangular inequality and the inequality (11) follows from Lipschitz continuity. Since both $k(x)$ and $\hat{k}(x)$ are Lipschitz continuous with some constants $\mathcal{L}_k$ and $\mathcal{L}_{\hat{k}}$, respectively, we have:

$$\begin{aligned} & \mathcal{L}_u \|k(x) - k(x_i)\| + \mathcal{L}_{\hat{u}} \|\hat{k}(x) - \hat{k}(x_i)\| \leq \\ & \mathcal{L}_u \mathcal{L}_k \|x - x_i\| + \mathcal{L}_{\hat{u}} \mathcal{L}_{\hat{k}} \|x - x_i\|, \quad \text{for all } x \in \mathcal{X}, x_i \in \mathcal{X}_d. \end{aligned}$$

Thus, according to (11), one obtains:

$$\|f(x, k(x)) - \hat{f}(x, \hat{k}(x))\| \leq (\mathcal{L}_x + \mathcal{L}_u \mathcal{L}_k + \mathcal{L}_{\hat{x}} + \mathcal{L}_{\hat{u}} \mathcal{L}_{\hat{k}}) \frac{\epsilon}{2} + \mathcal{E},$$

for all $x \in \mathcal{X}$, $x_i \in \mathcal{X}_d$, $\|x - x_i\| \leq \frac{\epsilon}{2}$, and $\mathcal{E}$ is defined as (8).

We employ the CBC of the source system for the target one. It can be readily verified that conditions (3) and (4) hold for the target system. Let us focus our attention on condition (5):

$$\begin{aligned}
& B(\hat{f}(x, \hat{k}(x))) - B(x) \\
& \leq B(\hat{f}(x, \hat{k}(x))) - B(f(x, k(x))) + B(f(x, k(x))) - B(x) \\
& \leq B(\hat{f}(x, \hat{k}(x))) - B(f(x, k(x))) - \eta \\
& \leq \mathcal{L}_B \|f(x, k(x)) - \hat{f}(x, \hat{k}(x))\| - \eta \\
& \leq \mathcal{L}_B \left((\mathcal{L}_x + \mathcal{L}_u \mathcal{L}_k + \mathcal{L}_{\hat{x}} + \mathcal{L}_{\hat{u}} \mathcal{L}_{\hat{k}}) \frac{\epsilon}{2} + \mathcal{E} \right) - \eta, \quad (12)
\end{aligned}$$

for all $x \in \mathcal{X}$, where $\mathcal{L}_B$ is the Lipschitz constant of the CBC, and η is its corresponding parameter in (3)-(5). In order to satisfy condition (5) for the target system, inequality (12) must be less or equal to 0. Thus, one needs:

$$\mathcal{L}_B \left(\mathcal{L}_{\dagger} \frac{\epsilon}{2} + \mathcal{E} \right) - \eta \leq 0, \quad (13)$$

where $\mathcal{L}_{\dagger} = \mathcal{L}_x + \mathcal{L}_u \mathcal{L}_k + \mathcal{L}_{\hat{x}} + \mathcal{L}_{\hat{u}} \mathcal{L}_{\hat{k}}$, which is satisfied based on Algorithm 1. Therefore, B together with controller $\hat{k}$ is a CBC for the target system. $\square$

The pseudo-code of our proposed method is introduced in Algorithm 1. Note that the guarantees provided by our approach are subject to the termination of this algorithm.

Algorithm 1 Learning the inverse dynamics controller

Input: $\mathcal{X}_0, \mathcal{X}_u, \mathcal{X}, \epsilon, B, \eta, k, \mathcal{L}_x, \mathcal{L}_{\hat{x}}, \mathcal{L}_u, \mathcal{L}_{\hat{u}}, \mathcal{L}_k, \mathcal{L}_B$, Neural network architecture

Output: Inverse dynamics controller $\hat{k}$

Construct the training set $\mathcal{X}_d$.

Initialize neural network's ($\hat{k}$) weights and biases.

$\mathcal{E} \leftarrow \max_{x_i \in \mathcal{X}_d} \|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\|$

$\mathcal{L}_{\hat{k}} \leftarrow$ Lipschitz constant of $\hat{k}$

$\mathcal{L}_{\dagger} \leftarrow (\mathcal{L}_x + \mathcal{L}_u \mathcal{L}_k) + (\mathcal{L}_{\hat{x}} + \mathcal{L}_{\hat{u}} \mathcal{L}_{\hat{k}})$

while $\mathcal{L}_B \left(\mathcal{L}_{\dagger} \frac{\epsilon}{2} + \mathcal{E} \right) > \eta$ **do**

$L \leftarrow \frac{1}{2|\mathcal{X}_d|} \sum_{x_i \in \mathcal{X}_d} \|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\|_2^2$

Train the neural network $\hat{k}$ based on L .

$\mathcal{E} \leftarrow \max_{x_i \in \mathcal{X}_d} \|f(x_i, k(x_i)) - \hat{f}(x_i, \hat{k}(x_i))\|$

$\mathcal{L}_{\hat{k}} \leftarrow$ Lipschitz constant of $\hat{k}$

$\mathcal{L}_{\dagger} \leftarrow (\mathcal{L}_x + \mathcal{L}_u \mathcal{L}_k) + (\mathcal{L}_{\hat{x}} + \mathcal{L}_{\hat{u}} \mathcal{L}_{\hat{k}})$

end while

Return $\hat{k}$

4. EXPERIMENT

In this section, we illustrate the effectiveness of our algorithm with an inverted pendulum case study. The experiment was conducted on a Nvidia RTX 4090 GPU.

We assume that Lipschitz constants of the CBC, f , and $\hat{f}$ are known. If the Lipschitz constants are unknown, one can leverage sampling methods such as (Strongin et al., 2019; Calliess et al., 2020) to estimate those constants.

We adopted the results in (Anand and Zamani, 2023) to obtain a CBC and its corresponding controller for the source system.

It is important to emphasize that in our case study, when referring to the model of the source system, we are treating it solely as a black-box representation for the simulation purposes. We did not incorporate the model to encode the conditions of control barrier certificates.

4.1 Inverted Pendulum

We consider the source system $\mathfrak{S} = (\mathcal{X}, U, f)$ to be an inverted pendulum where $\mathcal{X} = [-\frac{\pi}{4}, \frac{\pi}{4}] \times [-\frac{\pi}{4}, \frac{\pi}{4}]$, $\mathcal{X}_0 = [-\frac{\pi}{15}, \frac{\pi}{15}] \times [-\frac{\pi}{15}, \frac{\pi}{15}]$, and $\mathcal{X}_u = \mathcal{X} \setminus [-\frac{\pi}{6}, \frac{\pi}{6}] \times [-\frac{\pi}{6}, \frac{\pi}{6}]$. The transition function is given by:

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

where x_1 and x_2 are the angular position and velocity, respectively. Moreover, $g = 9.8$ is the gravitational acceleration, and $l = 1$ and $m = 1$ are the length and mass of the pendulum, respectively. Constant $\tau = 0.01$ is the sampling rate, and constants $\mathcal{L}_x = 1.1$, $\mathcal{L}_B = 2$, and $\eta = 0.07637$ are Lipschitz constants according to (13), Lipschitz constant of B , and its corresponding parameter, respectively. For the target system, we choose $l = 1.5$ and $m = 1.5$. After training the neural network, we get $\mathcal{L}_{\dagger} = 2.2$. For both systems, the discretization parameter and input set are $\epsilon = 9 \times 10^{-4}$, and $U = \hat{U} = [-10, 10]$, respectively.

In this experiment, if one uses the same controller, the CBC does not provide a guarantee of safety, and based on simulations, the target system enters the unsafe set. Our training method converged with 10000 iterations in 3 minutes, with $\mathcal{E} = 2 \times 10^{-4}$. Note that computing a control barrier certificate with its corresponding controller from scratch, takes roughly about 1.5 to 2 hours. The CBC values for both controllers are depicted in Figure 3.

5. CONCLUSION

We proposed a data-driven approach to synthesize provably correct controllers for target systems while leveraging previously learned control barrier certificates for source systems. Our approach relies on a trained neural network, referred to as an "inverse dynamics model," to serve as a controller for a target system. Additionally, we have incorporated a validity condition into the neural network training process, ensuring the correctness of the transferred controller without the necessity for post-facto verification. We demonstrated the effectiveness of our algorithm through an inverted pendulum case study.

REFERENCES

- Alur, R., Henzinger, T.A., and Sontag, E.D. (1996). *Hybrid systems III: Verification and control*, volume 3. Springer Science & Business Media.
- Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, 3420–3431. IEEE.
- Anand, M. and Zamani, M. (2023). Formally verified neural network control barrier certificates for unknown systems. In *Proceedings of the 22nd World Congress of the International Federation of Automatic Control*, 2742–2747.
- Bozinovski, S. (2020). Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3).
- Calliess, J.P., Roberts, S.J., Rasmussen, C.E., and Maciejowski, J. (2020). Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control. *Automatica*, 122, 109216.

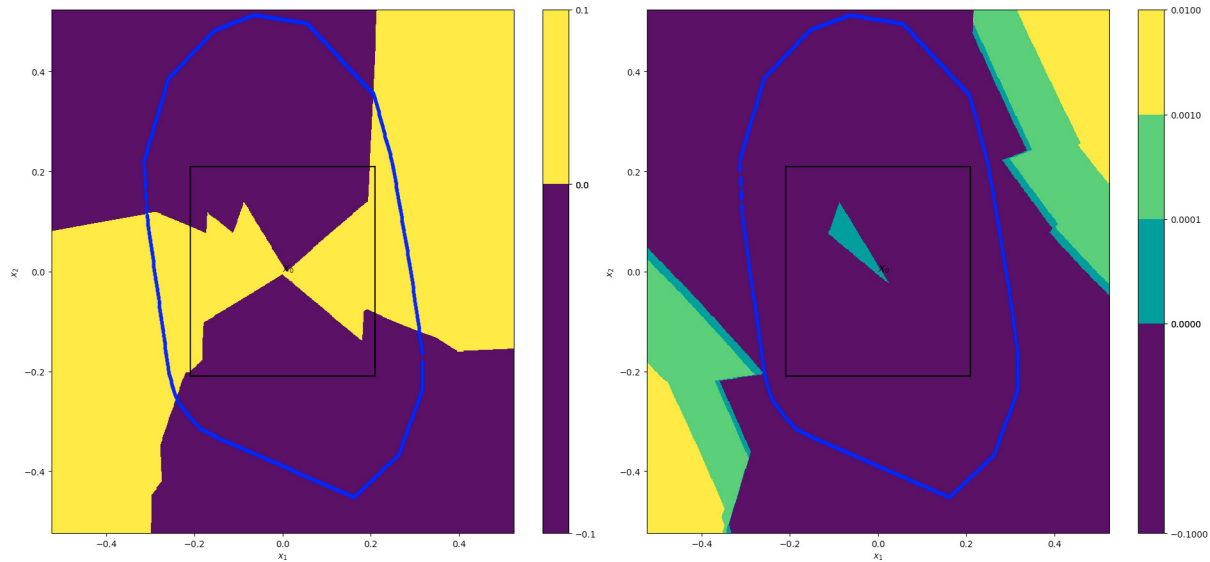


Fig. 3. Barrier certificate for the target system. Figure on the left indicates the value of the CBC when using the control policy of the source system, whereas Figure on the right depicts the same CBC but with the learned inverse dynamics controller.

- Christiano, P., Shah, Z., Mordatch, I., Schneider, J., Blackwell, T., Tobin, J., Abbeel, P., and Zaremba, W. (2016). Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*.
- Clark, A. (2021). Control barrier functions for stochastic systems. *Automatica*, 130, 109688.
- Dawson, C., Gao, S., and Fan, C. (2022). Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39, 1749–1767.
- De Moura, L. and Bjørner, N. (2011). Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9), 69–77.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.
- Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.
- Huang, C., Chen, X., Lin, W., Yang, Z., and Li, X. (2017). Probabilistic safety verification of stochastic hybrid systems using barrier certificates. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s), 1–19.
- Jagtap, P., Soudjani, S., and Zamani, M. (2020). Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control*, 66(7), 3097–3110.
- Lane, S.H. and Stengel, R.F. (1988). Flight control design using non-linear inverse dynamics. *Automatica*, 24(4), 471–483.
- Nadali, A., Trivedi, A., and Zamani, M. (2023). Transfer Learning for Barrier Certificates. In *62nd IEEE Conference on Decision and Control*.
- Nejati, A., Lavaei, A., Jagtap, P., Soudjani, S., and Zamani, M. (2023). Formal verification of unknown discrete- and continuous-time systems: A data-driven approach. *IEEE Transactions on Automatic Control*, 68(5), 3011–3024.
- Nejati, A., Zhong, B., Caccamo, M., and Zamani, M. (2022). Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates. In *Learning for Dynamics and Control Conference*, 763–776. PMLR.
- Parrilo, P.A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96, 293–320.
- Prajna, S. and Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, 477–492. Springer Berlin Heidelberg, Berlin, Heidelberg.
- 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.
- Strongin, R., Barkalov, K., and Bevizuk, S. (2019). Acceleration of global search by implementing dual estimates for Lipschitz constant. In *International Conference on Numerical Computations: Theory and Algorithms*, 478–486. Springer.
- Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, 242–264. IGI global.
- Weiss, K., Khoshgoftaar, T.M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1), 1–40.
- Wieland, P. and Allgöwer, F. (2007). Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12), 462–467.
- Zhou, R., Quartz, T., De Sterck, H., and Liu, J. (2022). Neural Lyapunov control of unknown nonlinear systems with stability guarantees. *Advances in Neural Information Processing Systems*, 35, 29113–29125.