Data-Driven Controller Synthesis via Finite Abstractions With Formal Guarantees

Daniel Ajeleye, Abolfazl Lavaei, *Senior Member, IEEE*, and Majid Zamani, *Senior Member, IEEE*

***Abstract*—Construction of finite-state abstractions (*a.k.a.* symbolic abstractions) is a promising approach for formal verification and controller synthesis of complex systems. Finite-state abstractions provide simpler models that can replicate the behaviors of original complex systems. These abstractions are usually constructed by leveraging precise knowledge of systems' dynamics, which is often unknown in real-life applications. In this letter, we develop a data-driven technique for constructing finite abstractions for continuous-time control systems with unknown dynamics. In our data-driven context, we collect samples from trajectories of unknown systems to construct finite abstractions with a guarantee of correctness. We propose a data-based gridding method to efficiently determine state-set discretization parameters while minimizing the expected number of transitions in the abstraction construction, thus reducing computational efforts. By establishing a feedback refinement relationship between an unknown system and its data-driven finite abstraction, one can design a controller over the data-driven finite abstraction. The controller can then be refined back to the original unknown system to meet a desired property of interest. We illustrate our proposed data-driven approach using a vehicle motion planning benchmark.**

***Index Terms*—Data driven control, optimal control, sampled-data control.**

I. Introduction

OVER the past two decades, formal methods have garnered significant attention within the autonomous and hybrid systems community. They have proven invaluable in addressing the inherent complexities of systems' dynamics and offering formal guarantees. These complex systems encompass a wide range, including biological systems, (air) traffic management systems, chemical processes, autonomous vehicles, and more. Given that these systems possess uncountable (infinite) state and input sets, synthesizing controllers to enforce intricate specifications presents formidable challenges. On the other hand, considerable progress has been made in the last two decades concerning the creation of finite abstractions, which serve as approximate representations of the concrete systems (*e.g.,* [1], [2], [3]). To employ the constructed finite abstraction in the controller synthesis process, it is essential to establish a systematic relationship between the concrete system and its finite abstraction (*e.g.,* [4], [5], [6]).

Regrettably, a significant portion of the pertinent literature on finite abstraction construction (*e.g.,* [1], [2], [3], [4], [5], [6], [7]) demands the availability of concrete system models, which are often absent in many real-world applications. To address this challenge, researchers have explored various *indirect data-driven* strategies to derive models for unknown dynamical systems through identification techniques (*e.g.,* [8] and references therein). Nevertheless, obtaining an accurate model can be arduous, time-intensive, and computationally expensive, particularly when dealing with intricate dynamics, as is often the case in real-world scenarios. This inherent complexity has spurred our pursuit of a *direct data-driven* approach for constructing finite abstractions. Our method involves directly collecting data from trajectories of unknown concrete systems, bypassing the need for a system identification phase.

Contributions: The contribution of this letter lies in the development of a data-driven technique for constructing low-complexity finite abstractions for continuous-time control systems with unknown dynamics, all while ensuring a correctness guarantee. Our data-driven approach entails the collection of data from trajectories of unknown systems. We introduce a data-driven gridding method focused on optimizing the selection of state-set discretization parameters, which efficiently reduces the expected number of transitions in the abstraction computations, thus reducing computational efforts. By establishing a feedback refinement relation, as discussed in [4], we forge a vital link between an unknown system and its data-driven finite abstraction. This relation enables us to design a controller based on the abstract model and subsequently refine it back to the original system satisfying desired properties.

Related Work: There exists a limited work dedicated to the construction of finite abstractions using data. In the work by [9], a data-driven approach is introduced for constructing finite abstractions tailored to unknown monotone systems. Notably, this approach demonstrates minimal conservatism when applied to unperturbed systems. In contrast, our data-driven approach extends its applicability to perturbed systems

Manuscript received 15 September 2023; accepted 19 October 2023. Date of publication 9 November 2023; date of current version 29 November 2023. This work was supported by NSF under Grant CNS-2145184. Recommended by Senior Editor A. P. Aguiar. *(Corresponding author: Daniel Ajeleye.)*

Daniel Ajeleye and Majid Zamani are with the Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: daniel.ajeleye@colorado.edu; majid.zamani@colorado.edu).

Abolfazl Lavaei is with the School of Computing, Newcastle University, NE4 5TG Newcastle upon Tyne, U.K. (e-mail: abolfazl.lavaei@newcastle.ac.uk).

Digital Object Identifier 10.1109/LCSYS.2023.3331385

characterized by nonlinear dynamics and is not confined to monotone systems. Another data-driven approach, as outlined in [10], strives to construct finite abstractions for the synthesis of controllers to enforce finite-horizon specifications. Our results broaden the scope compared to those presented in [10] in two aspects. Firstly, the results proposed in [10] exclusively address finite-horizon specifications, whereas our approach can be employed for both finite and infinite horizon properties. Secondly, our data-driven approach constructs a finite abstraction with guaranteed confidence, while the abstractions developed in [10] are valid with a probability less than 1.

More recently, [11] and [12] have introduced data-driven techniques for constructing finite abstractions in the context of incrementally input-to-state stable control systems and continuous-time perturbed systems, respectively. Nevertheless, the outcomes presented in [11] and [12] are accompanied by probabilistic confidence levels, necessitating an increasing volume of data as the confidence approaches one. In stark contrast, our data-driven approach yields a finite abstraction with a confidence level of 1, all without imposing any stability requirements on the underlying systems. Additionally, we introduce a data-driven gridding method designed to optimize the selection of state-set discretization parameters, thereby efficiently reducing the expected number of transitions and computational efforts during abstraction computations.

II. Preliminaries and Definitions

A. Notation

Symbols $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$, respectively, represent sets of real, positive, and non-negative real numbers. Notation $\cup$ and $\cap$ indicate, respectively, set union and intersection. The symbol $\mathbb{N}$ denotes the set of natural numbers and $\mathbb{N}_{\geq 0}$ is $\mathbb{N} \cup \{0\}$. For any non-empty set Q and $n \in \mathbb{N}$, Q^n indicates the Cartesian product of n duplicates of Q. Given N vectors $x_l \in \mathbb{R}^{n_l}$, $n_l \in \mathbb{N}$, and $l \in \{1, \ldots, N\}$, we use $x = [x_1; \cdots; x_N]$ to denote the corresponding column vector of dimension $\sum_l n_l$. The vector $\mathbf{1} \in \mathbb{R}^n$ is defined as $[1; 1; \cdots; 1] \in \mathbb{R}^n$, and $\mathcal{I}_n \in \mathbb{R}^{n \times n}$ represents the $n \times n$ identity matrix. For any $\bar{p}, \bar{q} \in \mathbb{R}^n$ and relational operator $\simeq \in \{\leq, <, =, >, \geq\}$, where $\bar{p} = [p_1; \cdots; p_n]$ and $\bar{q} = [q_1; \cdots; q_n]$, $\bar{p} \simeq \bar{q}$ is interpreted as $p_l \simeq q_l$ for every $l \in \{1, 2, \ldots, n\}$, *i.e.,* component-wise comparison. Similarly, $e^{\bar{p}} := [e^{p_1}; \cdots; e^{p_n}] \in \mathbb{R}^n$. Assuming $\bar{p} < \bar{q}$, then the *compact hyper-interval* $[\bar{p}, \bar{q}]$ is given as $[p_1, q_1] \times \cdots \times [p_n, q_n]$. Furthermore, given $c = [c_1; \cdots; c_n] \in \mathbb{R}^n$, we define the sum $\oplus$ as $c \oplus [\bar{p}, \bar{q}] := [p_1 + c_1, c_1 + q_1] \times \cdots \times [p_n + c_n, c_n + q_n]$. Notation $|c|$ means the entry-wise absolute value of $c \in \mathbb{R}^n$, *i.e.,* $[|c_1|; \cdots; |c_n|]$, while $\|c\|$ means the infinity norm of c. For any $\bar{r} \in \mathbb{R}^n_{>0}$ and $c_0 \in \mathbb{R}^n$, notation $\Phi_{\bar{r}}(c_0)$ is interpreted as $c_0 \oplus [-\bar{r}, \bar{r}]$. We define the norm of a function $\lambda : [0, \tau] \to \mathbb{R}^n$ as $\|\lambda\|_{[0,\tau]} := \sup_{t \in [0,\tau]} \|\lambda(t)\|$. For any $\vartheta \in \mathbb{R}^{n \times n}$, $\|\vartheta\|$ denotes the infinity norm of ϑ.

B. Continuous-Time Control Systems

Here, we first present the notions of simple systems and feedback refinement relations, which are borrowed from [4].

Definition 1: A simple system S is a tuple (X, U, F), where $X \neq \emptyset$, and $U \neq \emptyset$ are, respectively, state and input sets. The transition function $F : X \times U \rightrightarrows X$ is a set-valued map, whereby for a given input $u(k) \in U, k \in \mathbb{N}$, the state evolves as $x(k+1) \in F(x(k), u(k))$.

Given a simple system S, we define the set of *admissible inputs* for state $x \in X$ as $U_S(x) = \{u \in U \mid F(x, u) \neq \emptyset\}$. Later in this section, we use the notion of simple systems to define a sampled version of continuous-time control systems. Next, we present the notion of feedback refinement relation [4].

Definition 2: Consider two simple systems $S_i = (X_i, U_i, F_i)$, where $i \in \{1, 2\}$, such that $U_2 \subseteq U_1$. A feedback refinement relation from S_1 to S_2 is a nonempty relation $\mathcal{E} \subseteq X_1 \times X_2$ such that for all $(x_1, x_2) \in \mathcal{E}$, the following two conditions hold:

- $U_{S_2}(x_2) \subseteq U_{S_1}(x_1)$ and;
- if $u \in U_{S_2}(x_2)$, then $\mathcal{E}(F_1(x_1, u)) \subseteq F_2(x_2, u)$.

If there exists a feedback refinement relation $\mathcal{E}$ from S_1 to S_2, we denote it by $S_1 \propto_{\mathcal{E}} S_2$. Given such a relation, one can synthesize a controller over the finite abstraction and then refine it back to the original system, while satisfying the same property of interest. Here, we study continuous-time control systems, affected by bounded disturbances, as defined next.

Definition 3: A continuous-time control system (ct-CS) is represented via a tuple

$$\Xi = (X, U, f, \Lambda), \tag{1}$$

where:

- $X \subseteq \mathbb{R}^n$ is the state set and $U \subseteq \mathbb{R}^m$ is the input set;
- $f : X \times U \to X$ is the vector field, which is assumed to be locally Lipschitz continuous [4] for all input $u \in U$;
- $\Lambda \subset \mathbb{R}^n$ is the disturbance set which is assumed to be of the form $\Lambda = [-\bar{\lambda}, \bar{\lambda}]$, where $\bar{\lambda} \in \mathbb{R}^n_{\geq 0}$.

The state evolution of ct-CS Ξ in (1) is described by

$$\dot{x}(t) = f(x(t), \nu(t)) + \lambda(t), \tag{2}$$

with $\lambda(t) \in \Lambda$ being additive disturbances for all $t \in \mathbb{R}_{\geq 0}$.

Consider the sampling time $\tau \in \mathbb{R}_{>0}$. We consider input signals $\nu : \mathbb{R}_{\geq 0} \to U$ to be piecewise constant, i.e., for any $n \in \mathbb{N}_{\geq 0}$ and $t' \in [n\tau, (n+1)\tau)$, one has $\nu(t') = \nu(n\tau)$. In addition, for an initial state $x \in X$, a trajectory of the system over $[0, \tau]$ is defined as an absolutely continuous function $\xi_{x,u,\lambda} : [0, \tau] \to \mathbb{R}^n$ which satisfies (2) for every $t \in [0, \tau]$ given $\nu : [0, \tau] \to \{u\}$ and $\lambda : [0, \tau] \to \Lambda$. A collection of such trajectories over the interval $[0, \tau]$, starting from $x \in X$ under a given constant input with value $u \in U$, is denoted by $\Upsilon(x, u) := \{\xi_{x,u,\lambda} \mid \nu : [0, \tau] \to \{u\}$ and $\lambda : [0, \tau] \to \Lambda\}$. With a slight abuse of notation, we also use $\xi(x, u, \lambda) := \xi_{x,u,\lambda}(\tau)$ throughout this letter. We now describe a sampled version of a ct-CS as a simple system as in Definition 1.

Definition 4: Consider a ct-CS Ξ as in (1) with a sampling time $\tau \in \mathbb{R}_{>0}$. We call the simple system $\Xi_\tau = (X_\tau, U_\tau, F)$ a sampled version of ct-CS Ξ, if $X_\tau = X \subseteq \mathbb{R}^n$, $U_\tau = U \subseteq \mathbb{R}^m$, and the following holds: for any $x_1, x_2 \in X$ and $u \in U$, $x_2 \in F(x_1, u)$ if and only if there is a trajectory $\xi_{x_1,u,\lambda}$ of (2) over $[0, \tau]$ that satisfies $\xi_{x_1,u,\lambda}(0) = x_1$ and $\xi_{x_1,u,\lambda}(\tau) = x_2$.

In the next subsection, we aim at approximating the system in (2) (a.k.a. concrete system) by a finite abstraction (a.k.a. symbolic abstraction). In the constructed finite abstraction, each discrete state and input associate to a collection of continuous states and inputs of the system, respectively.

C. Symbolic Abstractions

In this section, we construct finite abstractions of ct-CS in (1) with sampling time $\tau \in \mathbb{R}_{>0}$. For a given compact state set $X \subset \mathbb{R}^n$ and discretization parameter vector $\eta_x \in \mathbb{R}^n_{\geq 0}$, we create a partition of X into cells $\Phi_{\eta_x}(x)$ such that $X \subseteq \bigcup_{x\in[X]_{\eta_x}} \Phi_{\eta_x}(x)$, where $[X]_{\eta_x}$ represents a finite set of representative points selected from those partition sets. A similar procedure is applied to the input set $U \subset \mathbb{R}^m$ using the discretization vector $\eta_u \in \mathbb{R}^m_{>0}$. Hence, we generate symbolic state set $\widehat{X} := [X]_{\eta_x}$ and input set $\widehat{U} := [U]_{\eta_u}$. Accordingly, the exact reachable set of states from an abstract state $\hat{x} \in \widehat{X}$ under a given input $\hat{u} \in \widehat{U}$ is defined as $\mathcal{R}(\hat{x}, \hat{u}) := \{\xi_{x,\hat{u},\lambda}(\tau) \mid \xi_{x,\hat{u},\lambda} \in \Upsilon(x, \hat{u}) \text{ and } x \in \Phi_{\eta_x}(\hat{x})\}$. However, we aim at constructing an over-approximation of $\mathcal{R}(\hat{x}, \hat{u})$ for any pair $(\hat{x}, \hat{u}) \in \widehat{X} \times \widehat{U}$, using a function called *growth bound*, which is formally defined next.

Definition 5: Given a ct-CS Ξ as in (1), and $\widehat{X}$, $\widehat{U}$ as symbolic states and input sets of Ξ, a function $\chi : \mathbb{R}^n_{\geq 0} \times \widehat{X} \times \widehat{U} \to \mathbb{R}^n_{\geq 0}$ satisfying

$$|\xi_{x',\hat{u},\lambda_1}(\tau) - \xi_{\hat{x},\hat{u},\lambda_2}(\tau)| \leq \chi(|x' - \hat{x}|, \hat{x}, \hat{u}), \tag{3}$$

for any $\hat{x} \in \widehat{X}$, $\hat{u} \in \widehat{U}$, $x' \in \Phi_{\eta_x}(\hat{x})$, $\lambda_i : [0, \tau] \to \Lambda, i \in \{1, 2\}$, is called the *growth bound* of ct-CS Ξ.

We now formally define a finite abstraction of ct-CS.

Definition 6: Given a ct-CS Ξ and a growth bound χ, let Ξ_τ be the sampled system associated with Ξ. Then $\widehat{\Xi} = (\widehat{X}, \widehat{U}, \hat{f})$ is a finite abstraction of Ξ_τ, with the transition map $\hat{f} : \widehat{X} \times \widehat{U} \rightrightarrows \widehat{X}$ if:

- the set $\bigcup_{\hat{x}\in\widehat{X}} \Phi_{\eta_x}(\hat{x})$ forms a non-empty cover for X;
- for any $\hat{x}, \hat{x}' \in \widehat{X}$ and $\hat{u} \in \widehat{U}$, $(\xi_{\hat{x},\hat{u},\lambda}(\tau) \oplus [-p', p']) \cap \Phi_{\eta_x}(\hat{x}') \neq \emptyset \implies \hat{x}' \in \hat{f}(\hat{x}, \hat{u})$ where $p' = \chi(\eta_x, \hat{x}, \hat{u})$ and $\lambda : [0, \tau] \to \Lambda$.

Next theorem, borrowed from [4], shows the usefulness of finite abstractions in Definition 6 by establishing a feedback refinement relation between the sampled system associated with a concrete ct-CS and its finite abstraction.

Theorem 1: Consider a ct-CS Ξ and its sampled version Ξ_τ. Let $\widehat{\Xi} = (\widehat{X}, \widehat{U}, \hat{f})$ be the finite abstraction of Ξ_τ according to Definition 6. Then $\Xi_\tau \propto_{\mathcal{E}} \widehat{\Xi}$, where the feedback refinement relation $\mathcal{E}$ is defined as $(x, \hat{x}) \in \mathcal{E}$ if $x \in \Phi_{\eta_x}(\hat{x})$.

D. Problem Formulation

In this letter, we assume that the vector field f in (1) is *unknown* and the main goal is to synthesize controllers for the unknown system. Although the underlying dynamics of ct-CS are considered to be unknown, its trajectories are accessible and could be sampled over $[0, \tau]$. We collect these sampled data points in a set $\mathcal{D}_N := \{(x_l, u_l, x'_l) \mid x'_l = \xi_{x_l,u_l,\lambda}(\tau), \text{ for some } \xi_{x_l,u_l,\lambda} \in \Upsilon(x_l, u_l), x_l \in X, u_l \in U, l = 1, 2, \ldots, N\}$. Now, we formalize the main problem that we aim to solve in this letter.

Problem 1: Consider a ct-CS Ξ with an unknown vector field f, affected by some bounded disturbances. Develop a data-driven approach based on the set of data $\mathcal{D}_N$ for constructing a finite abstraction $\widehat{\Xi}$, such that $\Xi_\tau \propto_{\mathcal{E}} \widehat{\Xi}$ with a set membership relation $\mathcal{E}$.

III. Data-Driven Construction of Finite Abstractions

In this section, given the growth bound in (3), we first present the required procedures for computing an over-approximation of a reachable set for ct-CS. We then propose our data-driven approach to compute the growth bound of unknown ct-CS based on data points in $\mathcal{D}_N$.

A. Reachable Set Computation via Growth Bound

Consider a ct-CS Ξ as in (1). Using Definitions 5 and 6 for every abstract state-input pair, the reachable set $\mathcal{R}(\hat{x}, \hat{u})$ is over-approximated using an hyper-interval $(\xi_{\hat{x},\hat{u},\lambda}(\tau) \oplus [-\chi(\eta_x, \hat{x}, \hat{u}), \chi(\eta_x, \hat{x}, \hat{u})])$. From the model-based analysis in [4], the growth bound is expressed as

$$\chi(\bar{s}, \hat{x}, \hat{u}) = e^{M(\hat{u})\tau}\bar{s} + \int_0^\tau e^{M(\hat{u})r}\bar{\lambda}dr, \tag{4}$$

for any $\bar{s} \in \mathbb{R}^n_{\geq 0}$, $\hat{x} \in \widehat{X}, \hat{u} \in \widehat{U}$, where function $M : \widehat{U} \to \mathbb{R}^{n\times n}$ deduces matrix $M(\hat{u})$ whose entries satisfy the following inequality:

$$M_{l,m}(\hat{u}) \geq \begin{cases} |\partial_m f_l(x, \hat{u})| & l \neq m, \\ \partial_m f_l(x, \hat{u}) & l = m, \end{cases} \tag{5}$$

for any $x \in X$ and $l, m \in \{1, 2, \ldots, n\}$. The *l-th* element of $f(x, \hat{u})$ is denoted by $f_l(x, \hat{u})$, while $\partial_m f_l(x, \hat{u})$ is its partial derivative with respect to *m-th* component of x.

If the underlying model is unknown, the matrix $M(\hat{u})$ in (5) and, hence, the growth bound in (4) cannot be computed. Here, we introduce a candidate growth bound, inspired by the analysis in [12], as follows:

$$\chi_\vartheta(\bar{s}, \hat{x}, \hat{u}) := \vartheta_1(\hat{x}, \hat{u})\big(\bar{s} + \bar{\lambda}\,\tau\big), \tag{6}$$

for any $\bar{s} \in \mathbb{R}^n_{\geq 0}$, $\hat{x} \in \widehat{X}, \hat{u} \in \widehat{U}$, where $\vartheta_1 \in \mathbb{R}^{n\times n}_{\geq 0}$, and $\vartheta \in \mathbb{R}^{n^2}_{\geq 0}$ is a column vector by stacking those of ϑ_1. It should be noted that for every abstract state, the parameters of ϑ_1 are locally defined.

Remark 1: Note that the presence of ϑ_1 in (6) is a consequence of our parameterization of the proposed growth bound. This parameter will serve as the decision variable in optimization problems (12) and (13), which are formulated as robust and scenario convex programs, respectively. For dimensional consistency, ϑ_1 is represented as a square matrix, while ϑ is the vector formed by stacking the columns of ϑ_1. This distinction is made to account for the need for the subscript "1" to differentiate between them.

In the next subsection, we present a data-driven approach utilizing dataset $\mathcal{D}_N$ to compute the candidate growth bound outlined in (6). Our approach also offers a formal correctness guarantee for (6) implying that it is a growth bound for ct-CS in (2) (cf. Theorem 2).

B. Lipschitz Continuity of System's Trajectories

The vector field f is assumed to be locally Lipschitz continuous. Hence, for any $x, x' \in X$, $\lambda_i : [0, \tau] \to \Lambda$, $i \in \{1, 2\}$, and given $\hat{u} \in \widehat{U}$, there exist Lipschitz constants $\mathcal{L}_x(\hat{u}), \mathcal{L}_\Lambda(\hat{u}) \in \mathbb{R}_{>0}$, such that:

$$\begin{cases} \|\xi(x, \hat{u}, \lambda_1) - \xi(x', \hat{u}, \lambda_1)\| \leq \mathcal{L}_x(\hat{u})\|x - x'\|, \\ \|\xi(x, \hat{u}, \lambda_1) - \xi(x, \hat{u}, \lambda_2)\| \leq \mathcal{L}_\Lambda(\hat{u})\|\lambda_1 - \lambda_2\|_{[0,\tau]}. \end{cases} \tag{7}$$

The two inequalities in (7) can be combined to get

$$\|\xi(x,\hat{u},\lambda_1)-\xi(x',\hat{u},\lambda_2)\| \leq \mathcal{L}_x(\hat{u})\|x-x'\| + \mathcal{L}_\Lambda(\hat{u})\|\lambda_1-\lambda_2\|_{[0,\tau]}. \quad (8)$$

Remark 2: Note that the results proposed in [13], specifically [14, Algorithm 2], can be leveraged to estimate the Lipschitz constants $\mathcal{L}_x$ and $\mathcal{L}_\Lambda$ for unknown system dynamics. This estimation is achieved using a finite dataset collected from the unknown system. However, for the purposes of this letter, we assume the availability of correct upper bound for the Lipschitz constants. Therefore, we do not account for any associated confidence levels in the estimation of these constants in our results.

Next lemma shows that the Lipschitz constant $\mathcal{L}_x(\hat{u})$ is an upper bound for $\|\vartheta_1(\hat{x},\hat{u})\|$. This lemma will be leveraged to show the results of Theorem 2.

Lemma 1: For a given input $\hat{u}\in\widehat{U}$, the Lipschitz constant $\mathcal{L}_x(\hat{u})$ is an upper bound for $\|\vartheta_1(\hat{x},\hat{u})\|$, *i.e.,*

$$\|\vartheta_1(\hat{x},\hat{u})\| \leq \mathcal{L}_x(\hat{u}), \quad \forall \hat{x}\in\widehat{X}. \quad (9)$$

Proof: We show the proof by contradiction. Suppose (9) does not hold, *i.e.,* $\|\vartheta_1(\hat{x},\hat{u})\| > \mathcal{L}_x(\hat{u})$. Since $\mathcal{L}_x(\hat{u})$ is a Lipschitz constant, (7) implies that $\forall x'\in X$, $\forall\hat{x}\in\widehat{X}$, for any $\lambda:[0,\tau]\to\Lambda$ and given $\hat{u}\in\widehat{U}$,

$$\mathcal{L}_x(\hat{u}) > \frac{\|\xi(x',\hat{u},\lambda)-\xi(\hat{x},\hat{u},\lambda)\|}{\|x'-\hat{x}\|+\|\bar{\lambda}\|}.$$

The aforementioned inequality, in conjunction with the contrary assumption, leads to

$$\|\xi(x',\hat{u},\lambda)-\xi(\hat{x},\hat{u},\lambda)\| < \|\vartheta_1\|\,\|x'-\hat{x}\|+\|\vartheta_1\|\,\|\bar{\lambda}\|. \quad (10)$$

By employing (6), (3), and then applying the infinity norm on the resulting inequality, one has

$$\|\xi(x',\hat{u},\lambda)-\xi(\hat{x},\hat{u},\lambda)\| \leq \|\vartheta_1\|\big(\|x'-\hat{x}\|+\|\bar{\lambda}\|\,\tau\big). \quad (11)$$

By subtracting (10) from (11), one has $\|\vartheta_1\|\leq\tau$, which does not hold $\forall\vartheta_1\in\mathbb{R}^{n\times n}_{\geq 0}$ and $\forall\tau>0$, completing the proof. ∎

Next, we present our data-driven approach for computing the growth bound through sampled trajectories.

C. Data-Driven Computation of Growth Bound

The main goal is to search for a less conservative growth bound, in terms of over-approximating the reachable sets. In our proposed setting, we first cast the candidate growth bound in (6) as the following robust convex program (RCP):

$$\begin{cases} \min_{\vartheta} \quad \mathbf{1}^\top\vartheta \\ \textbf{s.t.}\ \ \vartheta\in[0,\bar{\vartheta}],\ \forall x_1,x_2\in\Phi_{\eta_x}(\hat{x}),\ \forall\lambda_1,\lambda_2:[0,\tau]\to\Lambda, \\ \quad |\xi(x_1,\hat{u},\lambda_1)-\xi(x_2,\hat{u},\lambda_2)|-\vartheta_1(\hat{x},\hat{u})\big(|x_1-x_2|+\bar{\lambda}\,\tau\big) \\ \quad \leq 0, \end{cases} \quad (12)$$

where $\mathbf{1}\in\mathbb{R}^{n^2}$ and $\bar{\vartheta}\in\mathbb{R}^{n^2}_{>0}$ is a sufficiently large vector component-wise.

One can readily verify that a feasible solution of the RCP in (12) provides a growth bound as in (3) for ct-CS in (2). Unfortunately, a precise knowledge of the dynamic is required for solving the problem. To resolve these issues, we collect data from trajectories of unknown ct-CS and propose a scenario convex program (SCP) corresponding to the original RCP. To do so, consider a set of N sampled data points $\mathcal{D}_N$ collected from cells $\Phi_{\hat{\eta}_x}(\tilde{x})$, where $\tilde{x}\in[\Phi_{\eta_x}(\hat{x})]_{\hat{\eta}_x}$, which are *sub-grids* within a cell $\Phi_{\eta_x}(\hat{x})$, where $\hat{\eta}_x:=\frac{1}{\sqrt[n]{N}}\eta_x$ (cf. Fig. 1). The size of the sub-grid cells is determined by the extraction of N data points from the primary cell $\Phi_{\eta_x}(\hat{x})$, which has a dimension n. In fact, the process of sampling N data points from a sub-grid inside a cell with length η necessitates that the sub-grid cell's size remains upper bounded by $\frac{1}{\sqrt[n]{N}}\eta_x$. Using $\mathcal{D}_N$, for any $\hat{x}\in\widehat{X}$ and $\hat{u}\in\widehat{U}$, we propose the SCP associated to the RCP (12) for a cell $\Phi_{\eta_x}(\hat{x})$ as

$$\text{SCP}:\begin{cases} \min_{\vartheta} \quad \mathbf{1}^\top\vartheta \\ \textbf{s.t.}\ \ \vartheta\in[0,\bar{\vartheta}],\ \forall\, l,\bar{l}\in\{1,\dots,N\}, \\ \quad |x'_l-x'_{\bar{l}}|-\vartheta_1(\hat{x},\hat{u})\big(|x_l-x_{\bar{l}}|+\bar{\lambda}\,\tau\big)+\varrho\leq 0, \end{cases} \quad (13)$$

where $\varrho\in\mathbb{R}^n_{\geq 0}$ is a bias term and computed as follows.

Theorem 2: Consider a ct-CS Ξ as in (1) with sampling time $\tau\in\mathbb{R}_{>0}$. For any $\hat{x}\in[X]_{\eta_x}$ and $\hat{u}\in[U]_{\eta_u}$, suppose for a cell $\Phi_{\eta_x}(\hat{x})$, $[\Phi_{\eta_x}(\hat{x})]_{\hat{\eta}_x}$ is constructed where $\hat{\eta}_x:=\frac{1}{\sqrt[n]{N}}\eta_x$. Then, the solution of (13) provides a growth bound as in (3) corresponding to $(\hat{x},\hat{u})$ where

$$\varrho := 4\big(\mathcal{L}_x(\hat{u})\hat{\eta}_x+\mathcal{L}_\Lambda(\hat{u})\bar{\lambda}\big), \quad (14)$$

$\mathcal{L}_x(\hat{u})$ and $\mathcal{L}_\Lambda(\hat{u})$ are the Lipschitz constants as in (7).

Proof: One can readily verify that the optimization problem (13) always admits a feasible solution. For any fixed pair $(\hat{x},\hat{u})\in\widehat{X}\times\widehat{U}$, let

$$\beta(\vartheta,x_1,x_2,\lambda_1,\lambda_2) := |\xi(x_1,\hat{u},\lambda_1)-\xi(x_2,\hat{u},\lambda_2)| - \vartheta_1(\hat{x},\hat{u})\big(|x_1-x_2|+\bar{\lambda}\,\tau\big),$$

for all $x_1,x_2\in\Phi_{\eta_x}(\hat{x})$ and $\lambda_i,\lambda'_i:[0,\tau]\to\Lambda$ with $i\in\{1,2\}$. Furthermore, let ϑ^* be the optimal solution of SCP (13). By considering $x_1,x_2\in\Phi_{\eta_x}(\hat{x})$ and picking samples $x_l,x_{\bar{l}}$ from cells $\Phi_{\hat{\eta}_x}(x_l)$, $\Phi_{\hat{\eta}_x}(x_{\bar{l}})\subset\Phi_{\eta_x}(\hat{x})$, one gets

$$\begin{aligned} &|\beta(\vartheta^*,x_1,x_2,\lambda_1,\lambda_2)-\beta(\vartheta^*,x_l,x_{\bar{l}},\lambda'_1,\lambda'_2)| \\ &\leq |\xi(x_1,\hat{u},\lambda_1)-\xi(x_l,\hat{u},\lambda_2)|+\|\vartheta_1(\hat{x},\hat{u})\|\,|x_1-x_l| \\ &\quad +|\xi(x_2,\hat{u},\lambda'_1)-\xi(x_{\bar{l}},\hat{u},\lambda'_2)|+\|\vartheta_1(\hat{x},\hat{u})\|\,|x_2-x_{\bar{l}}| \\ &\leq 2\mathcal{L}_x(\hat{u})\big(|x_1-x_l|+|x_2-x_{\bar{l}}|\big) \\ &\quad +\mathcal{L}_\Lambda(\hat{u})\big(\|\lambda_1-\lambda_2\|_{[0,\tau]}+\|\lambda'_1-\lambda'_2\|_{[0,\tau]}\big)\mathbf{1} \\ &\leq 4\big(\mathcal{L}_x(\hat{u})\hat{\eta}_x+\mathcal{L}_\Lambda(\hat{u})\bar{\lambda}\big)=:\varrho. \end{aligned}$$

The above inequality implies that for any $x_1,x_2\in\Phi_{\eta_x}(\hat{x})$ and any disturbance within Λ,

$$\beta(\vartheta^*,x_1,x_2,\lambda_1,\lambda_2)\leq\beta(\vartheta^*,x_l,x_{\bar{l}},\lambda'_1,\lambda'_2)+\varrho. \quad (15)$$

Therefore, within any cell $\Phi_{\eta_x}(\hat{x})$, (15) implies that any optimal solution of SCP (13) is always feasible for RCP (12). In particular, any feasible solution of (13) results in a growth bound χ_ϑ of the form (6) that satisfies inequality (3), which concludes the proof. ∎

Remark 3: By applying Theorem 2 to all partition sets, it is possible to establish a set membership relation $\propto_{\mathcal{E}}$ (cf. Theorem 1) between a concrete ct-CS Ξ and its data-driven finite abstraction $\widehat{\Xi}$ by solving the SCP (13) over the grid cells.

D. Abstraction by Optimized State Set Partitioning

Here, we expand upon the model-based approach introduced in [15], which addressed the problem of choosing the state discretization parameter $\eta_x = [\eta_x^1; \cdots; \eta_x^n] \in \mathbb{R}^n_{>0}$. This parameter choice defines a predetermined volume $\eta_x^1\eta_x^2, \ldots \eta_x^n$ for each grid cell. The primary aim here is to reduce memory and time consumption during abstraction construction by minimizing the number of transitions within the abstraction. We build upon the work in [15] and develop a data-driven strategy that optimizes the shape of state space grids, effectively reducing computational efforts.

Given a finite input set $\widehat{U}$, we first choose a grid parameter η_x arbitrarily, to construct the symbolic set $[X]_{\eta_x}$. We proceed as follows to compute an optimized grid parameter while preserving the volume of each cell $\Phi_{\eta_x}(\hat{x})$. Let $J \in \mathbb{N}$ be the cardinality of $[X]_{\eta_x} \times \widehat{U}$. Constructing an abstraction $\widehat{\Xi}$ of ct-CS Ξ in (1) necessitates computing a sequence of $(\vartheta_1^j)_{j=1,\ldots,J}$, obtained by solving (13) for each $(\hat{x}, \hat{u}) \in [X]_{\eta_x} \times \widehat{U}$. Hereafter, we present an optimization technique aimed at reducing the size of abstractions (in terms of the number of transitions) that contain grid cells with the volume $\eta_x^1\eta_x^2, \ldots \eta_x^n$:

$$\min_{\alpha} \sum_{j=1}^{J} \mathbb{E}_j(e^{\alpha}) \text{ subjected to } \sum_{l=1}^{n} \alpha_l = \varsigma, \tag{16}$$

where $\varsigma = \ln(\eta_x^1\eta_x^2, \ldots \eta_x^n)$, $\alpha = [\alpha_1; \cdots; \alpha_n] \in \mathbb{R}^n$ and $\mathbb{E}_j : \mathbb{R}^n_{>0} \to \mathbb{R}_{>0}$, such that $\mathbb{E}_j(e^{\alpha}) = \prod_{i=1}^{n} [2\vartheta_1^j \bar{\lambda}\, \tau + \sum_{l=1}^{n} (\mathcal{I}_n + \vartheta_1^j)_{i,l} e^{\alpha_l}] e^{-\alpha_i}$ and $\sum_{j=1}^{J} \mathbb{E}_j(e^{\alpha})$ is the expected number of transitions in $\widehat{\Xi}$. One can readily verify that $\mathbb{E}_j(e^{\alpha})$ is a convex function with respect to $\alpha \in \mathbb{R}^n$. Hence, the problem in (16) can be solved numerically. In addition, we present the following theorem, which gives sufficient condition for (16) to have a unique optimizer.

Theorem 3: Given $\varsigma \in \mathbb{R}$, $J \in \mathbb{N}$, $\tau > 0$, let $\vartheta_1^j \in \mathbb{R}^{n \times n}_{\geq 0}$ be irreducible for some $j = 1, \ldots, J$. Then, the optimization problem in (16) admits a unique optimizer.

The proof is similar to that of [15, Th. V.4] and is omitted here due to lack of space. Following this approach, we obtain a grid parameter $\eta_x^* := e^{\alpha^*}$, where α^* is the optimizer of (16). We refer the interested reader to [15] for deeper insights on the optimization problem in (16).

We propose Algorithm 1 to describe the required procedure in Theorem 2 for the data-driven construction of finite abstractions. The solution obtained from (16) is fed into Algorithm 1, yielding a data-driven finite abstraction with a minimized number of transitions. We visualize the transition function $\hat{f}(\hat{x}, \hat{u})$ of a finite abstraction in Fig. 1.

Algorithm 1 Data Driven Construction of Finite Abstractions

Inputs: X, U, Λ, η_u and η_x^*

1: Construct $\widehat{X} = [X]_{\eta_x^*}$ and $\widehat{U} = [U]_{\eta_u}$
2: **for each** $(\hat{x}, \hat{u}) \in \widehat{X} \times \widehat{U}$ **do**
3: Initiate $\hat{f}(\hat{x}, \hat{u}) = \emptyset$, $\rho = \mathbf{0} \in \mathbb{R}^n$ and $c = \xi_{\hat{x},\hat{u},\lambda}(\tau) \in \Upsilon(\hat{x}, \hat{u})$ where $\lambda : [0, \tau] \to \Lambda$
4: Compute $\varrho \in \mathbb{R}^n_{\geq 0}$ using (14)
5: As outlined in Theorem 2, generate $[\Phi_{\eta_x^*}(\hat{x})]_{\hat{\eta}_x}$ and select N sampled data points $(x_l, \hat{u}, x_l')$ from it.
6: Obtain the optimal value $\vartheta^*(\hat{x}, \hat{u})$ for SCP (13)
7: Update: $\rho = \chi_{\vartheta^*}(\eta_x^*, \hat{x}, \hat{u})$
8: $\hat{f}(\hat{x}, \hat{u}) = \{\hat{x}' \in \widehat{X} \mid \Phi_{\eta_x^*}(\hat{x}') \cap \Phi_{\rho}(c) \neq \emptyset\} \cup \hat{f}(\hat{x}, \hat{u})$
9: **end for**

Output: $\widehat{\Xi} = (\widehat{X}, \widehat{U}, \hat{f})$

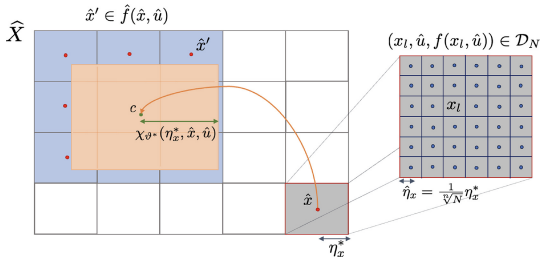

Fig. 1. A 2-dimensional depiction of a finite abstraction, constructed using Algorithm 1.

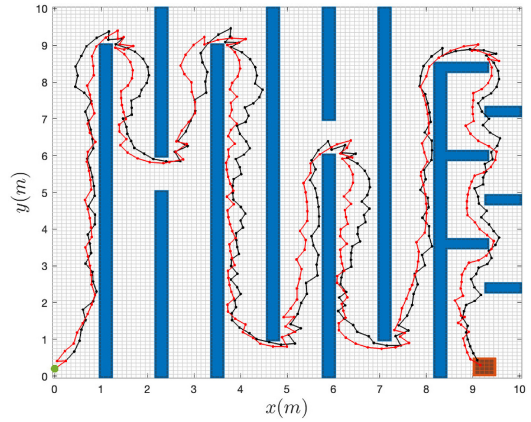

Fig. 2. Closed-loop trajectories by our data-driven abstraction-based technique (in black) and by the model-based one (in red) in [4] with perturbation set $\Lambda = [-\bar{\lambda}, \bar{\lambda}]$, where $\bar{\lambda} = [0.15; 0.15; 0.015]$ and initial state $x_0 = [0; 0.2; 0]$ (green spot). Target and obstacle sets are depicted by red and blue boxes, respectively.

IV. Case Study

Here, we show the efficacy of our data-driven approaches by applying them to a vehicle motion planning benchmark with unknown dynamics. The main goal is to design a controller via the proposed abstraction-based approach to navigate the vehicle reaching to a target set, while avoiding potential obstacles. We also provide a detailed comparison between our proposed results and those presented in [12] (cf. Table I and Fig. 3). The model of the vehicle is borrowed from [4] as

$$\dot{x}(t) = \begin{bmatrix} u_1 \cos(q + x_3)/\cos(q) \\ u_1 \sin(q + x_3)/\cos(q) \\ u_1 \tan(u_2) \end{bmatrix} + \lambda(t), \tag{17}$$

where $q := \arctan(\tan(u_2)/2)$, $\lambda : [0, \tau] \to \Lambda = [-\bar{\lambda}, \bar{\lambda}] \subset \mathbb{R}^3$ with $\bar{\lambda} = [0.15; 0.15; 0.015]$, $x(t) = [x_1; x_2; x_3] \in [0, 10]^2 \times [-\pi - 0.3798, \pi + 0.3798]$ and $u(t) = [u_1; u_2] \in [-1, 1]^2$. States $[x_1; x_2]$ are the two-dimensional coordinate of the vehicle's position, while x_3 is its orientation. We assume the model in (17) is unknown to us. However, the model is used to sample the system trajectories with the sampling time $\tau = 0.25s$. The main goal is to find a controller that steers the

TABLE I
COMPARISON RESULTS FOR THE MOTION PLANNING EXPERIMENT

Methods	Abstraction Construction			
	number of transitions	memory (RAM GB)	computation time (mins)	Samples N
Algorithm 1 naive η_x	3.5×10^8	0.86	103.75	1331
Algorithm 1 η_x^* from (16)	1.6×10^8	0.397	42.35	1331
Work [12] η_x^* from (16)	1.7×10^{10}	1.887	583	4410

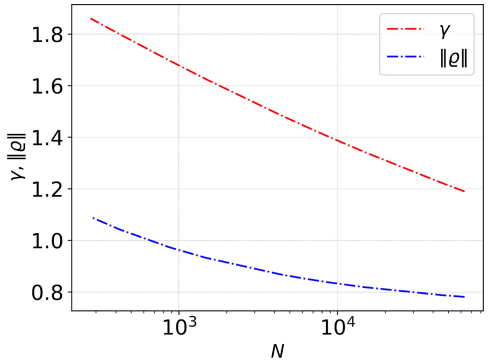

Fig. 3. Comparing bias term ϱ in (14) with γ in [12] across the number of sampled trajectories and $\bar{\lambda} = [0.15; 0.15; 0.015]$, while assuming a confidence of 99.9% in the analysis of [12].

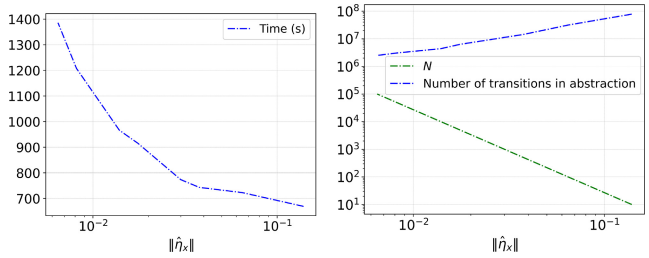

Fig. 4. The relationship between the time spent by Algorithm 1, the number of samples N within the sub-grid, and the number of abstraction transitions as a function of $\|\hat{\eta}_x\|$, while maintaining the primary cell size at a fixed value of $\eta_x = [0.3; 0.3; 0.3]$.

vehicle to the target set $[9, 9.51] \times [0, 0.51]$ while avoiding the areas as depicted in blue in Fig. 2.

We construct $[U]_{\eta_u}$ using $\eta_u = [0.3; 0.3]$ and with an initial choice $\eta_x = [0.9; 0.9; 0.001]$, solving (16) yields $\eta_x^* \approx [0.0669; 0.0669; 0.1899]$, which was input into Algorithm 1. We utilized the estimated Lipschitz constants, specifically $\mathcal{L}_x = 1.1697$ and $\mathcal{L}_\Lambda = 1.21$, as detailed in Remark 2. All implementations for the construction of the data-driven finite abstraction have been done in SCOTS [16] with some modifications on a 64-bit MacBook Pro with 64GB RAM (3.2 GHz). We then leverage SCOTS for synthesizing a controller enforcing the reach-avoid property. A closed-loop state trajectory of the unknown vehicle starting from $x_0 = [0; 0.2; 0]$ based on our data-driven abstraction-based technique is depicted in Fig. 2 (see black trajectory). The implementation results are reported in Table I.

As evident from the results, the controller effectively ensures that the vehicle avoids collisions with obstacles, as illustrated by the blue boxes in Fig. 2. The controller's domain comprises 927, 916 states. Table I records a notable reduction of 54.3% in transitions, 59.2% in computational time, and 53.8% in memory consumption, compared to the use of a naive choice of discretization vector. It is also evident that our approach consistently surpasses the method presented in [12] in all of these aspects. Additionally, Fig. 3 shows that the bias term ϱ consistently maintains lower conservatism than γ in [12, eq. (19)] with an increase in the number of sampled trajectories. Hence, our approach demonstrates reduced conservatism in contrast to the one in [12]. Note that the sub-grid size significantly impacts the number of sample points. Finer sub-grid size leads to a larger number of samples, thereby reducing the value of ϱ. This results in the SCP algorithm producing a less conservative growth bound in comparison to when fewer samples are used. An illustrative comparison of this effect is provided in Fig. 4.

ACKNOWLEDGMENT

The authors would like to thank Ben Wooding for providing the initial codes of data-based growth bound computation.

REFERENCES

[1] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. New York, NY, USA: Springer, 2009.
[2] M. Zamani, G. Pola, and P. Tabuada, "Symbolic models for unstable nonlinear control systems," in *Proc. Amer. Control Conf.*, 2010, pp. 1021–1026.
[3] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros, "Symbolic control of stochastic systems via approximately bisimilar finite abstractions," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3135–3150, Dec. 2014.
[4] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1781–1796, Apr. 2017.
[5] K. Mallik, A.-K. Schmuck, S. Soudjani, and R. Majumdar, "Compositional synthesis of finite-state abstractions," *IEEE Trans. Autom. Control*, vol. 64, no. 6, pp. 2629–2636, Jun. 2019.
[6] A. Girard and G. J. Pappas, "Hierarchical control system design using approximate simulation," *Automatica*, vol. 45, no. 2, pp. 566–571, 2009.
[7] G. Pola, P. Pepe, and M. D. Di Benedetto, "Symbolic models for networks of control systems," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3663–3668, Nov. 2016.
[8] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Inf. Sci.*, vol. 235, pp. 3–35, Jun. 2013.
[9] A. Makdesi, A. Girard, and L. Fribourg, "Efficient data-driven abstraction of monotone systems with disturbances," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 49–54, 2021.
[10] A. Devonport, A. Saoud, and M. Arcak, "Symbolic abstractions from data: A PAC learning approach," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 599–604.
[11] A. Lavaei and E. Frazzoli, "Data-driven synthesis of symbolic abstractions with guaranteed confidence," *IEEE Control Syst. Lett.*, vol. 7, pp. 253–258, 2022.
[12] M. Kazemi, R. Majumdar, M. Salamati, S. Soudjani, and B. Wooding, "Data-driven abstraction-based control synthesis," 2022, *arXiv:2206.08069*.
[13] G. Wood and B. Zhang, "Estimation of the Lipschitz constant of a function," *J. Global Optim.*, vol. 8, pp. 91–103, Jan. 1996.
[14] A. Nejati, A. Lavaei, P. Jagtap, S. Soudjani, and M. Zamani, "Formal verification of unknown discrete- and continuous-time systems: A data-driven approach," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 3011–3024, May 2023.
[15] A. Weber, M. Rungger, and G. Reissig, "Optimized state space grids for abstractions," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5816–5821, Nov. 2017.
[16] M. Rungger and M. Zamani, "SCOTS: A tool for the synthesis of symbolic controllers," in *Proc. 19th Int. Conf. Hybrid Syst., Comput. Control*, 2016, pp. 99–104.