Data-driven heuristic symbolic models and application to limit-cycle detection

Julien Calbert Raphaël M. Jungers

***Abstract*— Symbolic control allows to provide formal guarantees for generic optimal control problems on nonlinear systems. It relies on the construction of a finite abstraction of the system which requires the discretization of the state space. Therefore, these methods suffer from the curse of dimensionality and a critical step is the choice of the state space partition. In this paper, we propose a data-driven heuristic abstraction approach relying on a probabilistic interpretation of the discretization error. Our approach can be used to automatically compare different partitions of the state space and to infer complex properties about the original system. As a proof of concept, we use our approach for the detection of limit cycles.**

I. Introduction

In Symbolic Control [1], [2], [3], one aims at translating a control theoretic problem into a formulation in terms of a finite automaton (or any other object on which a decision problem can be solved in finite time). Thus, a critical step resides in the construction of a finite abstraction, i.e. a dynamical system composed of a finite number of states (called *cells*) whose behavior *simulates* that of the original system. A critical step in these techniques is the choice of the state space partition from which the abstract system will be constructed.

In this work, we propose a heuristic approach to tame the curse of dimensionality resulting from discretization. Our approach is based on a probabilistic reinterpretation of the non-determinism inherent to the discretization procedure. Thanks to this reinterpretation, we obtain a probabilistic abstraction in the form of a Markov chain. Our idea is to use this Markov chain and its properties to heuristically adapt the partition of the state space for the problem of interest. These probabilities have no real meaning with respect to the underlying dynamics but we will show that they can be used as a proxy for the non-determinism incurred by discretization, in order to compare different partitions and infer complex properties about the original system.

The classical abstraction method [4] relies on a uniform discretization of the complete state space and thus suffers greatly from the curse of dimensionality. In order to solve this problem, one potential avenue is to compute a state partition in an ad-hoc and optimized way, depending on both the original system dynamics and the desired goal.

JC is a FRIA Research Fellow. RJ is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 864017 - L2C. RJ is also supported by the Innoviris Foundation and the FNRS (Chist-Era Druid-net).

J. Calbert and R. Jungers are with the ICTEAM, UCLouvain, 4 Av. G. Lemaître, 1348 Louvain-la-Neuve, Belgium. {julien.calbert, raphael.jungers}@uclouvain.be

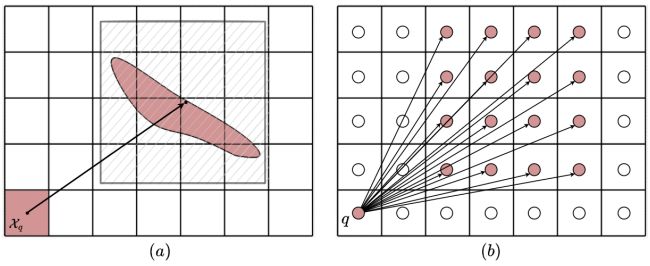

Fig. 1. Classical abstraction approach. Illustrations of transitions defined for a given state $q \in \mathcal{Q}$ and an input $u \in \mathcal{U}$ for an alternating simulation abstraction constructed using a growth-bound function. Left: The reachable set of the continuous state space $\mathcal{X}_q$ and its over-approximation. Right: The symbolic model. As one can see, the global upper bound on the Lipschitz constant being quite large, the over-approximation of the set of reachable cells leads to a high degree of conservatism. A similar image can be found in [4, Fig. 5].

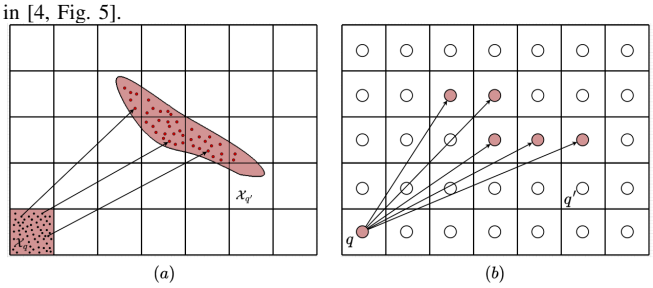

Fig. 2. Data-driven approach. We draw N samples from cell $\mathcal{X}_q$ and compute their continuous successors. Left: The reachable set of the continuous state space $\mathcal{X}_q$ and its approximation by sampling. Right: The symbolic model. As can be seen, some possible transitions may have been missed like the one from cell q to cell q'. A similar image can be found in [5, Fig. 1].

Some works go in this direction, such as [6], where the authors propose to adapt the size of the abstraction gradually. In a previous work [7], we proposed to circumvent this curse of dimensionality by introducing a hierarchical abstraction approach and by co-designing the abstraction and the controller guided by the optimal control problem. Another approach is to use learning-based techniques. In [5], the authors propose a data-driven approach based on a black-box model to construct an abstraction relying on the notion of *PAC (probably approximately correct) approximate alternating simulation relation*. Transitions in the abstraction are computed by simulating trajectories starting in a cell. In the same vein, the authors of [8] also propose a method to provide a PAC bound on a reachability control problem by computing probabilistic bounds on transitions based on scenario-based approach methods. In [9], Gaussian process (GP) regression and abstraction are used to compute a strategy that maximizes the probability of satisfying an LTL formula [10] of a partially-known stochastic system from

a given data set. Some data-driven techniques have also been proposed [11] to build an alternating simulation for the particular class of monotone systems with bounded disturbances. In [12], the authors propose a procedure to compute bounds on the probability of satisfying a specification for the particular class of mixed monotone stochastic systems.

The classical abstractions are based on the notion of *alternating simulation relation* [1, Definition 4.19] which provides formal guarantees. Indeed, thanks to this relation between the original system and the abstraction, we can from a controller for the abstraction derive a controller for the original system. However, these abstractions can be conservative, making the abstract problem impossible to solve. The computation of these abstractions is essentially reduced to the over-approximation of the reachable sets of the original system, i.e. the computation of a bounding set, encompassing the attainable states, from each particular cell. A large number of over-approximation methods have been proposed, e.g. [13], [14], [15]. However, there is a trade-off between computing a high precision over-approximation of the reachable set, leading to a less conservative abstraction, and the computation time. A method proposed in [4] is based on the computation of a *growth-bound function*, which is a bound on the discretization error based on continuity arguments. While this function has the advantage of being a local estimate and of depending on the input used, it can also be very conservative. It requires narrow bounds on the partial derivatives on an a priori enclosure of the trajectory. Moreover, it provides an over-approximation componentwise, which can in some situations lead to a bad approximation as shown in Fig. 1. For this reason, we use a data-driven technique that relies on a finite set of simulated trajectories of the original system. This method do not suffer from the conservatism resulting from the over-approximation of the reachable set (see Fig. 2), in the sense that it does not add impossible transitions in the abstraction. Our primary goal is not to provide firm guarantees of correctness for the original system. Indeed, some trajectories of the original system may not have been simulated, and therefore will not appear in the abstraction. Rather, our goal is to heuristically, and efficiently, emit high-level hypotheses on the qualitative behavior of the system (here, the existence of a limit-cycle), hoping that such well-defined hypothesis will be proved more easily in a second step.

Unlike the previously cited works [5], [8], [9], [11], [12], we do not provide robust guarantees such as PAC. Instead, we provide a heuristic methodology, based on concepts borrowed from information theory, to design the partition of the state space in order to infer more involved system properties. In this work, we show as a proof of concept that our technique allows to achieve limit cycle detection.

The notation used in this paper is as follows. The nonnegative real set is indicated by $\mathbb{R}^+$. Given a metric d on $\mathcal{X}$, the diameter of a set $\mathcal{A} \subset \mathcal{X}$, denoted $\mathrm{diam}(\mathcal{A})$, is the supremum of distances between its points $\mathrm{diam}(\mathcal{A}) = \sup_{x,y\in\mathcal{A}} d(x,y)$. Additional notation will be introduced as required in the text.

II. Markov chain abstraction

In this work, we restrict our attention to autonomous systems. We start by a formal definition of the considered systems. Given a set $\mathcal{X}$, with $\mathcal{P}(\mathcal{X})$ we denote its power set.

Definition 1 (Continuous-time system)**.** *An* (autonomous) continuous-time system *is a couple $S = (\mathcal{X}, f)$ where $\mathcal{X}$ denotes the set of states and $f : \mathcal{X} \to \mathcal{X}$ is the transition single-valued map. The evolution of the state is defined by the following differential equation*

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t)),\ t \in \mathbb{R}^+ \tag{1}$$

and we denote by $\phi(t, x)$ the solution of the system 1 with the initial condition x at time $t = 0$.

Definition 2 (Discrete-time system)**.** *An* (autonomous) discrete-time system *is a couple $S = (\mathcal{X}, F)$ where $\mathcal{X}$ denotes the set of states and $F : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ is the transition set-valued map. The successor state $\boldsymbol{x}_{k+1}$ of the state $\boldsymbol{x}_k$ must follow the following inclusion $\boldsymbol{x}_{k+1} \in F(\boldsymbol{x}_k)$.*

The abstraction will be of the following type.

Definition 3 (Discrete-time Markov chain)**.** *A* discrete-time Markov chain *is a stochastic process $\{X_k\}_{k=0}^{\infty}$ over a finite state space $\mathcal{Q}$, where the probability of moving to the next state depends only on the present state. That is, for states $s_i \in \mathcal{Q}$, the model assumes that $P(X_{k+1} = s_{k+1} \mid X_k = s_k, \ldots, X_1 = s_1) = P(X_{k+1} = s_{k+1} \mid X_k = s_k)$. Therefore a discrete-time Markov chain is characterized by a tuple $\mathcal{M} = (\mathcal{Q}, \boldsymbol{\Phi})$ where $\boldsymbol{\Phi}$ is the matrix containing the conditional probabilities $\Phi_{ij} = P(X_{k+1} = j \mid X_k = i)$ for $i, j \in \mathcal{Q}$. Given an initial distribution vector $\boldsymbol{p}_k$ defined as $(p_k)_i = P(X_k = i)$, its evolution is determined by the matrix-vector product $\boldsymbol{p}_{k+1} = \boldsymbol{\Phi}^\top \boldsymbol{p}_k$.*

We divide in tree steps our procedure to abstract a continuous-time dynamical system $S = (\mathcal{X}, f)$. First, we discretize the continuous state space of the original system by partitioning it into connected cells. This partition of $\mathcal{X}$ is denoted by $\mathcal{P} = (\mathcal{X}_q)_{q\in\mathcal{Q}}$ and satisfies the following properties: $\cup_{q\in\mathcal{Q}}\mathcal{X}_q = \mathcal{X}$, $\forall q, q' \in \mathcal{Q}, q \neq q' : \mathcal{X}_q \cap \mathcal{X}_{q'} = \emptyset$ and $\forall q \in \mathcal{Q} :\ \mathcal{X}_q$ is a connected subset of $\mathcal{X}$. We say that two cells $\mathcal{X}_{q_1}$ and $\mathcal{X}_{q_2}$ are *adjacent* if they share a contact point. We define the associated quantization function $\mathcal{R} : \mathcal{X} \to \mathcal{Q}$ such that

$$\forall \boldsymbol{x} \in \mathcal{X} :\ \mathcal{R}(\boldsymbol{x}) = q \Leftrightarrow \boldsymbol{x} \in \mathcal{X}_q. \tag{2}$$

Second, for each cell, we uniformly randomly sample a finite number of points. Third, we construct a Markov chain $\mathcal{M}$ as follows: the state space of the Markov chain is given by $\mathcal{Q}$ and its transition matrix $\boldsymbol{\Phi}$ is obtained by simulating the trajectories of the samples of each cell for a given time step. Precisely, the transition probability between a cell q_1 and q_2 is determined by the relative number of simulations starting in $\mathcal{X}_{q_1}$ and ending up in $\mathcal{X}_{q_2}$. The procedure is illustrated in Fig. 3. The resulting Markov chain is not complete in the sense that it may miss a feasible transition from cell q_1 to q_2 due to the finiteness of sampling. In fact, in our setting, probabilities are used as a proxy for representing the nondeterminism due to the finiteness of the abstraction.

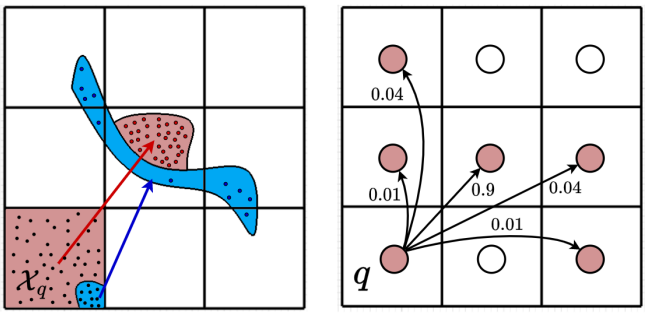

Fig. 3. Data-driven abstraction by a Markov chain. Left: Reachable set of the cell $\mathcal{X}_q$ with in blue, the subset of the cell mapped in the blue part of the reachable set. Right: The Markov chain abstraction. Each edge is labeled with a probability, which is the relative number of simulations ending up in certain cells of the state space.

Observe that there is some conservatism with the idea of using probabilities as a proxy for the non-determinism of abstractions. Indeed, the Markov chain abstraction makes the implicit assumption of the so-called *Markov Property*, namely that the probability only depends on the current state i. While the transitions of the original system through the quantization function may be correlated with the previous state sequence, and therefore do not in general satisfy the Markov property.

To obtain more precise results with horizon greater than one time step, we can extend the discrete state space by introducing a notion of memory over the previous time steps.

Definition 4 (Discrete-time Markov chain with memory)**.** *A* Discrete-time Markov chain with memory m *is a stochastic process* $\{X_k\}_{k=0}^{\infty}$ *over a finite state space* $\mathcal{Q}$ *satisfying:* $P(X_{k+1} = s_{k+1} \mid X_k = s_k, \ldots, X_1 = s_1) = P(X_{k+1} = s_{k+1} \mid X_k = s_k, \ldots, X_{k-m+1} = s_{k-m+1})$ *for states* $s_i \in \mathcal{Q}$ *and* $k \geq m$*. By defining the augmented state* $\tilde{X}_k = (X_k, X_{k-1}, \ldots, X_{k-m+1})$*, the process* $\{\tilde{X}_k\}_{k=0}^{\infty}$ *satisfies the Markov property. A discrete-time Markov chain with memory* m *is characterized by a Markov chain* $\mathcal{M}_m = (\mathcal{Q}^m, \mathbf{\Phi}_m)$ *where* $\mathcal{Q}^m$ *is the* m*-th power set of* $\mathcal{Q}$ *and* $\mathbf{\Phi}_m \in \mathbb{R}^{n^m \times n^m}$ *is the matrix containing the conditional probabilities* $(\Phi_m)_{ij} = P(\tilde{X}_{k+1} = j \mid \tilde{X}_k = i)$ *for* $i, j \in \mathcal{Q}^m$*.*

For other works (in a non-data-driven framework) making use of memory in an iterative way, in order to refine an abstraction of a dynamical system, we refer the reader to [16], [17]. See also [18] for a more recent contribution in the data-driven framework.

A probability distribution $\boldsymbol{\pi}$ on the state space $\mathcal{Q}$ is called a *stationary distribution* with respect to a Markov chain $\mathcal{M} = (\mathcal{Q}, \mathbf{\Phi})$ if $\boldsymbol{\pi}^\top \mathbf{\Phi} = \boldsymbol{\pi}^\top$. An *attractive stationary distribution* is a stationary distribution that has the property that any initial distribution of the Markov Chain converges to it.

Definition 5 (Memory-m Markov chain abstraction)**.** *Let* $S = (\mathcal{X}, f)$ *a continuous-time dynamical system,* $(\mathcal{X}_q)_{q \in \mathcal{Q}}$ *a partition of* $\mathcal{X}$*,* $\tau > 0$ *a given time-step. A memory-*m *Markov chain abstraction of system* S *is a Markov chain with memory* m *defined as* $\mathcal{M}_m = (\mathcal{Q}^m, \mathbf{\Phi}_m)$ *as obtained by Algorithm 1.*

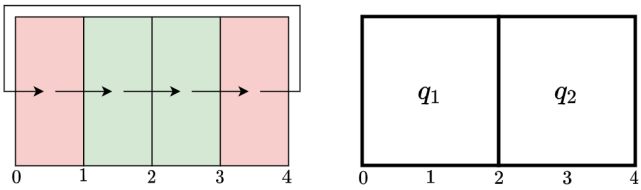

Fig. 4. Left: Original dynamical system (3). Right: State space discretization. The red region is the pre-image of cell q_1, and the green region is the pre-image of cell q_2.

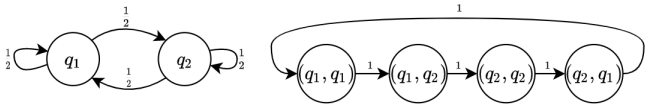

Fig. 5. Left: Memory-1 Markov chain abstraction of system (3). Right: Memory-2 Markov chain abstraction of system (3).

An example is given for the following 2-dimensional dynamical system

$$\begin{cases} x_{k+1} = (x_k + 1) \bmod 4 \\ y_{k+1} = y_k \end{cases} \tag{3}$$

whose state space and its discretization are given in Fig. 4. As shown in Fig. 5, constructing an abstraction with memory 1 can lead to erroneous conclusions about the original system: for example, one could wrongly conclude that there is a non-zero probability of reaching cell q_1 in two steps from cell q_1, which never happens for the original system. However, with a memory-2 abstraction, these spurious behaviors disappear.

In what follows, we introduce a relaxed definition of attractive limit cycle [19, Definition 1, Section 3.3] to deal with the discretization of the state space and time.

Definition 6 (ϵ-τ-attractive limit cycle)**.** *Let* $S = (\mathcal{X}, f)$ *a continuous-time system. Let us fix a metric* d *on* $\mathcal{X}$*, a time step* $\tau > 0$ *and a precision* $\epsilon > 0$*. We say that a closed curve* $\mathcal{C} \subseteq \mathcal{X}$ *is an* ϵ-τ*-attractive limit cycle of* S *if for all initial points* $x_0 \in \mathcal{X}$*, there exists a representation*[1] $\xi : \mathbb{R}_+ \to \mathcal{C}$ *of* $\mathcal{C}$ *and* $k_0 \in \mathbb{N}$ *such that*

$$\forall k \geq k_0 : d(\phi(k\tau, x_0), \xi(k\tau)) \leq \epsilon.$$

We will now establish conditions on the abstraction to guarantee the existence of an ϵ-τ-attractive limit cycle for the original system.

In the following assumption, we require that a sufficient number of trajectories have been simulated to guarantee that all trajectories of duration $m\tau$ of the original system through the quantization function have been simulated.

Assumption 1. *Consider a continuous-time system* S *as introduced in Definition 1 and its memory-*m *Markov chain abstraction* $\mathcal{M}_m = (\mathcal{Q}^m, \mathbf{\Phi}_m)$*. For* $\tilde{q}_1, \tilde{q}_2 \in \mathcal{Q}^m$*:*

$$(\Phi_m)_{\tilde{q}_1, \tilde{q}_2} \neq 0$$
$$\Leftrightarrow \exists x_0 \in \mathcal{X}_{q_0} : \ \forall k = 0, \ldots, m : \ \phi(k\tau, x_0) \in \mathcal{X}_{q_k}$$

[1]We say that $\xi : \mathbb{R}_+ \to \mathcal{C}$ is a representation of a closed curve $\mathcal{C}$ if ξ is continuous, differentiable and $\exists c > 0 : \ \forall t \in \mathbb{R}_+ : \ \|\xi'(t)\| > c$.

Algorithm 1: Algorithm for computing a memory-m Markov chain abstraction of a continuous-time system based on a finite number of trajectory simulations.

Input: A continuous-time dynamical system $S = (\mathcal{X}, f)$, a partition $(\mathcal{X}_q)_{q\in\mathcal{Q}}$ of $\mathcal{X}$ and its associated quantization function $\mathcal{R}$, a time step $\tau > 0$, a number of simulations per cell $N \in \mathbb{N}$ and a memory $m \in \mathbb{N}$.

Output: A memory-m Markov chain abstraction.

$N_{\tilde{q}_1,\tilde{q}_2} \leftarrow 0,\ \forall \tilde{q}_1, \tilde{q}_2 \in \mathcal{Q}^m$;
$(\Phi_m)_{\tilde{q}_1,\tilde{q}_2} \leftarrow 0,\ \forall \tilde{q}_1, \tilde{q}_2 \in \mathcal{Q}^m$;
for $q \in \mathcal{Q}$ **do**
  for $k = 1, \dots, N$ **do**
    $\boldsymbol{x}_0 \leftarrow \text{sample}(\mathcal{X}_q)$;
    for $i = 1, \dots, m$ **do**
      $\boldsymbol{x}_i \leftarrow \phi(\tau, \boldsymbol{x}_{i-1})$;
      $q_i \leftarrow \mathcal{R}(\boldsymbol{x}_i)$;
    end
    $\tilde{q}_1 \leftarrow (q, q_1, \dots, q_{m-2}, q_{m-1})$;
    $\tilde{q}_2 \leftarrow (q_1, q_2, \dots, q_{m-1}, q_m)$;
    $N_{\tilde{q}_1,\tilde{q}_2} \leftarrow N_{\tilde{q}_1,\tilde{q}_2} + 1$;
  end
end
for $\tilde{q}_1 \in \mathcal{Q}^m$ **do**
  $M_{\tilde{q}_1} \leftarrow \sum_{\tilde{q}\in\mathcal{Q}^m} N_{\tilde{q}_1,\tilde{q}}$;
  for $\tilde{q}_2 \in \mathcal{Q}^m$ **do**
    $(\Phi_m)_{\tilde{q}_1,\tilde{q}_2} \leftarrow \frac{N_{\tilde{q}_1,\tilde{q}_2}}{M_{\tilde{q}_1}}$;
  end
end
return $\mathcal{M}_m = (\mathcal{Q}^m, \boldsymbol{\Phi}_m)$;

where $\tilde{q}_1 = (q_0, q_1, \dots, q_{m-2}, q_{m-1})$, $\tilde{q}_2 = (q_1, q_2, \dots, q_{m-1}, q_m)$ *and* $q_0, \dots, q_m \in \mathcal{Q}$.

We introduce the following definition to characterize that a set of elements of the partition forms an adjacent cycle.

Definition 7. *Consider a partition $\mathcal{P} = (\mathcal{X}_q)_{q\in\mathcal{Q}}$ of a set $\mathcal{X}$. Let us define the graph $G = (\mathcal{Q}, E)$ with $\mathcal{Q}$ the set of cells in $\mathcal{P}$ and $(q_1, q_2) \in E$ if and only if $\mathcal{X}_{q_1}$ and $\mathcal{X}_{q_2}$ are adjacent in the partition. A subset $\mathcal{A} \subset \mathcal{Q}$ is said to be an* adjacent cycle *if there exists a cycle in G that contains each node in $\mathcal{A}$, and no other node.*

The subsequent theorem presents criteria for determining the set of cells that contains the ϵ-τ-attractive limit cycle.

Theorem 1. *Let $S = (\mathcal{X}, f)$ be a continuous-time dynamical system, and let $\mathcal{P} = (\mathcal{X}_q)_{q\in\mathcal{Q}}$ be a partition of $\mathcal{X}$. Let $\tau > 0$ be a given time-step, and let $\mathcal{M}_m = (\mathcal{Q}^m, \Phi_m)$ be a memory-m Markov chain abstraction of S, as defined in Definition 5, and assume it satisfies Assumption 1. Let us assume that $\mathcal{M}_m$ has an attractive stationary distribution $\boldsymbol{\pi}$. We define $\tilde{\mathcal{V}} = \{\tilde{q} \in \mathcal{Q}^m \mid \pi_{\tilde{q}} > 0\}$, $\mathcal{V} = \{\tilde{q}_i, i = 1, \dots, m \mid \tilde{q} \in \tilde{\mathcal{V}}\}$. Suppose that $\mathcal{V}$ is an adjacent cycle as defined in Definition 7. Then, the system S admits an ϵ-τ-attractive limit cycle $\mathcal{C}$ with $\epsilon = \max_{q\in\mathcal{V}} \operatorname{diam}(\mathcal{X}_q)$, $\mathcal{C} \subset \cup_{q\in\mathcal{V}} \mathcal{X}_q$.*

Proof. We extend the quantization function $\mathcal{R}$ associated to the partition $\mathcal{P} = (\mathcal{X}_q)_{q\in\mathcal{Q}}$ of a vector $\boldsymbol{x} \in \mathbb{R}^n$ to a k-uple of vectors $\mathcal{R}_k : \mathcal{X}^k \to \mathcal{Q}^k$ such that $\mathcal{R}_k(\boldsymbol{x}_1, \dots, \boldsymbol{x}_k) = (\mathcal{R}(\boldsymbol{x}_1), \dots, \mathcal{R}(\boldsymbol{x}_k))$. Given an initial point $\boldsymbol{x}_0 \in \mathcal{X}$, we define $\tilde{\boldsymbol{x}}_k = (\phi(k\tau, \boldsymbol{x}_0), \dots, \phi((k+m)\tau, \boldsymbol{x}_0))$. By Assumption 1, for all $k \geq 0$, there is a transition in the abstraction from the cell $\tilde{q}_k = \mathcal{R}_m(\tilde{\boldsymbol{x}}_k)$ to the cell $\tilde{q}_{k+1} = \mathcal{R}_m(\tilde{\boldsymbol{x}}_{k+1})$. Hence, for any initial point $\boldsymbol{x}_0 \in \mathcal{X}$, there exists $k_0 \in \mathbb{N}$ such that

$$\forall k \geq k_0 :\ \mathcal{R}_m(\tilde{\boldsymbol{x}}_k) \in \tilde{\mathcal{V}}.$$

This is a consequence of the fact that in a Markov Chain, if there exists a unique attractive stationary distribution, then there is also a unique strongly connected component, and all the paths in the corresponding graph terminate in this connected component. As a result, we have

$$\forall k \geq k_0\ \exists q \in \mathcal{V} : \phi(k\tau, \boldsymbol{x}_0) \in \mathcal{X}_q. \tag{4}$$

Now, let $\mathcal{V} = \{q_0, q_1, \dots, q_{l-1}\}$ be an adjacent cycle. Since $\mathcal{V}$ is an adjacent cycle, there exists an ordering of the elements in $\mathcal{V}$, say $(q_0, q_1, \dots, q_{l-1})$, such that for $k = 0, \dots, l-1$, the subsets $\mathcal{X}_{q_k}$ and $\mathcal{X}_{q_{(k+1) \bmod l}}$ have a contact point. Furthermore, since the subsets $\mathcal{X}_q$ are connected, we can deduce that there is a closed curve $\mathcal{C} \subset \cup_{q\in\mathcal{V}} \mathcal{X}_q$ that sequentially passes through the subsets $\mathcal{X}_{q_0}, \mathcal{X}_{q_1}, \dots, \mathcal{X}_{q_{l-1}}$. Equation (4) can be expressed as the existence of a representation $\xi : \mathbb{R}_+ \to \mathcal{C}$ of $\mathcal{C}$ designed to fulfill the following condition

$$\forall k \geq k_0\ \forall q \in \mathcal{V} :\ \phi(k\tau, \boldsymbol{x}_0) \in \mathcal{X}_q \Leftrightarrow \xi(k\tau) \in \mathcal{X}_q.$$

Consequently, we have

$$\forall k \geq k_0 :\ d(\phi(k\tau, \boldsymbol{x}_0), \xi(k\tau)) \leq \epsilon$$

for $\epsilon = \max_{q\in\mathcal{V}} \operatorname{diam}(\mathcal{X}_q)$, thereby concluding the proof. □

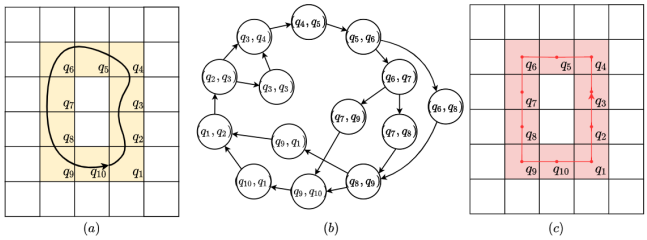

Fig. 6. An ϵ-τ-attractive limit cycle. Left: Attractive limit cycle of a 2-dimensional continuous-time system and its state space partition $\mathcal{P} = (\mathcal{X}_q)_{q\in\mathcal{Q}}$. Middle: Stationary sub-graph of the memory-2 Markov chain abstraction $\mathcal{M}_2$, i.e. the sub-graph of the Markov chain that contains all the states in the stationary distribution and all the transitions between them. Right: In red, the ϵ-τ-attractive limit cycle $\mathcal{C} \subset \cup_{i=1,\dots,10} \mathcal{X}_{q_i}$.

Theorem 1 is illustrated in Fig. 6 where it has been assumed that its applicability conditions are satisfied for a 2-memory Markov chain abstraction.

III. Information Theory

In this section, we propose a technique in order to automatically evaluate the quality of a given abstraction. Here again, we mainly leverage Markov Chain theory. Even if the original system is deterministic, Algorithm 1 may result in a non-deterministic Markov chain abstraction because of the quantization step.

Given a particular Markov chain, it is well-known [20, Theorem 7.2] that (under some hypotheses, namely irreducibility and aperiodicity) the left eigenvector corresponding to the maximal eigenvalue 1 (a.k.a. Perron vector) of

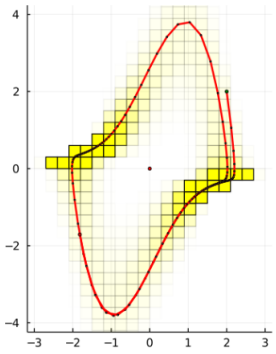

Fig. 7. In yellow, the stationary distribution of the Markov chain abstraction of the system Eq. (6) with $\mu = 2$. The color level is proportional to the stationary probability of the corresponding cell in the abstraction. In red, a simulation converging to the limit cycle.

the adjacency matrix provides the asymptotic stationary distribution. If the abstraction leads to a Markov chain such that the stationary distribution is confined in a small number of cells, it should in principle imply that we have a high accuracy on the state in the long term run. In order to numerically evaluate this accuracy, we propose to use the *entropy of the stationary distribution*, defined as follows

$$C(\boldsymbol{\pi}) = -\sum_{q \in \mathcal{Q}} \pi_q \log_2(\pi_q) \tag{5}$$

where $\boldsymbol{\pi}$ is the stationary distribution.

As we will show in the next section, this value can be used to design a partition suitable for the limit cycle detection problem.

IV. Numerical experiment

In this section, we use the heuristic tools developed in the previous sections for limit cycle detection. The strategy is illustrated on the two-dimensional form of the Van der Pol oscillator

$$\begin{cases} \dot{x} = y \\ \dot{y} = \mu(1 - x^2)y - x \end{cases} \tag{6}$$

where $\mu > 0$ is a parameter indicating the non linearity and for which we know that a limit cycle exists thanks to the Poincare-Bendixson Criterion [21, Section 2.6] and [22, Theorem 1, Section 3.8].

A. Limit cycle detection by analysis of the stationary distribution

First, a memory-1 Markov chain abstraction of the system is constructed based on Monte Carlo simulation (Definition 5). Then, we calculate the stationary distribution of the Markov chain. The resulting stationary distribution is illustrated on Fig. 7. As one can see, the stationary distribution is concentrated on the limit cycle.

To identify the subset, called the candidate region, of the state space that potentially contains a limit cycle, we use a probabilistic interpretation by examining cells with stationary probability above a given threshold. This is illustrated in our approach as shown in Fig. 8. Specifically, Fig. 8.a depicts the set of cells in the abstraction with a non-zero stationary probability, representing the reachability set obtained using the classical, formal abstraction approach, and one cannot conclude much, because few cells can be ruled out with the formal approach.

We note that, once we have obtained an approximation of the limit cycle like Fig. 8.c, we can refine the candidate region of the state space by simulating new trajectories starting in this region with a longer time step. If a trajectory does not end in this region, the cell associated to its initial position is removed from the candidate region. We can continue until we reach a fixed point. We leave this strategy for later work.

Note that under Assumption 1, the part of the state space associated with the support of the stationary distribution vector is an attractive and forward-invariant set. That is, the trajectories of the system converge to this region and remain there. But this does not imply an oscillatory behavior, such as that associated with a limit cycle.

In order to go further and prove the existence of an ϵ-τ limit cycle, we can increase the memory of the Markov chain abstraction until the stationary distribution vector satisfies the applicability conditions of Theorem 1.

B. Using entropy as a proxy in order to optimize the change of coordinates

As mentioned in the previous section, the Shannon entropy of the stationary distribution allows to empirically estimate the quality of the limit cycle approximation for different partitions of the state space. The results for rotated grid partitions are presented in Fig. 9. In Fig. 10, the stationary distribution is shown for the best and worst rotated partition. One can see that the left figure is better aligned with the limit cycle. This was automatically detected by our heuristic, in that the entropy of the Markov chain abstraction is smaller. Then, we can construct the Markov chain abstraction with increasing memory and observe that the support of the stationary distribution vector, and thus the accuracy around the limit cycle, decreases faster for the best rotated partition of angle $\theta = 0.785$ than the other angles appearing in Fig. 9 as observed in Fig. 10.

Note, to reduce the number of simulations, one can avoid computing the abstraction with memory m over the entire state space, but can restrict to the region corresponding to the support of the stationary distribution vector of the memory-1 Markov chain abstraction.

V. Conclusion

In this paper, we have proposed a data-driven heuristic abstraction approach that relies on a probabilistic interpretation of discretization error and on information theory tools. Our goal is to automatically compare different partitions of the state space in order to select the best one in order to build a *smart abstraction*. For now, as a proof of concept, we have limited our analysis to partition comparison for the limit cycle detection problem. As future work we plan to extend the methods to improve the partition by iterative refinement for more general properties. The philosophy of our approach is to trade off formal guarantees for scalability. However, we believe that one may sometimes recover formal

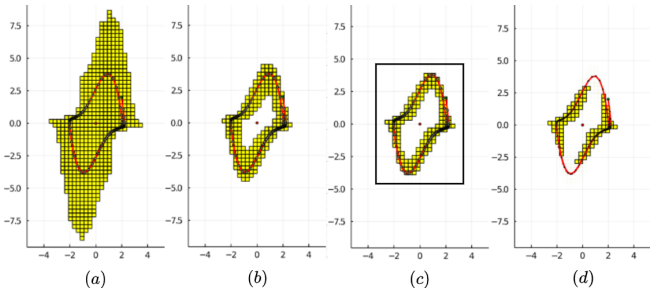

Fig. 8. Thresholding procedure to detect the limit cycle. We consider the Markov chain abstraction of the system Eq. (6) with $\mu = 2$. In yellow, the cells with stationary probabilities above a given threshold probability. From left to right, the threshold probabilities are respectively $0, 10^{-4}, 10^{-3}, 1.5 \cdot 10^{-3}$. The best approximation (Fig. 8.c) is emphasized. Fig. 8.d has a too high threshold, and misses some cells that are part of the limit cycle. In red, a simulation converging to the limit cycle.

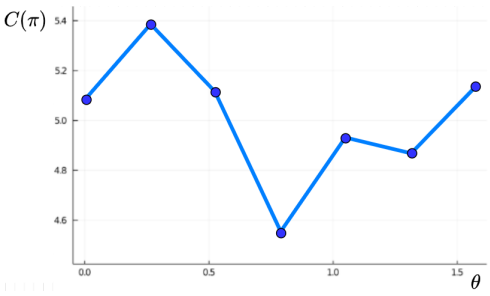

Fig. 9. Comparison of rotated partitions. Shannon entropy of the stationary distribution for several abstractions constructed from a rotated grid of angle θ.

guarantees in a second step by applying formal methods on the (heuristically obtained) abstraction.

A potential direction to improve Theorem 1 is to apply the heuristic test proposed in [18] to estimate whether the level of memory is adequate for achieving the required accuracy. Moreover, obtaining a PAC version of Theorem 1 may render Assumption 1 unnecessary, as demonstrated in recent works such as [23], [24]. In future research, we aim to establish a more robust definition of the approximate limit cycle that is not limited to regular time intervals (multiples of τ) but is also valid at any arbitrary time $t \in \mathbb{R}$. To achieve this, instead of quantizing at fixed time intervals, we plan to quantize when the trajectory exits a cell during simulation.

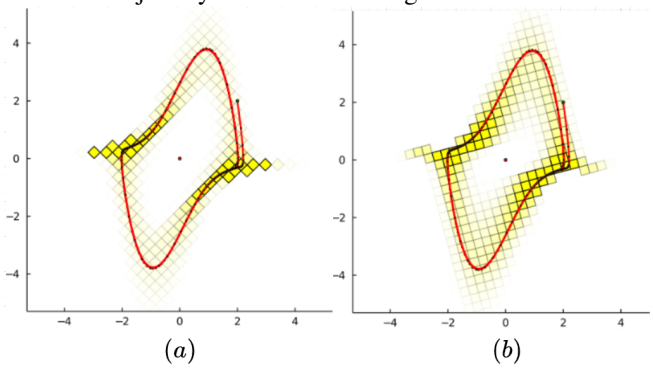

Fig. 10. Comparison of rotated partitions. Left: Partition with a rotated grid of angle $\theta = 0.785$ corresponding to the minimal value in Fig. 9. Right: Partition with a rotated grid of angle $\theta = 0.26$ corresponding to the maximal value in Fig. 9.

REFERENCES

[1] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[2] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 15.

[3] D. Lind and B. Marcus, *An introduction to symbolic dynamics and coding*. Cambridge university press, 2021.

[4] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1781–1796, 2016.

[5] A. Devonport, A. Saoud, and M. Arcak, "Symbolic abstractions from data: A pac learning approach," *arXiv preprint arXiv:2104.13901*, 2021.

[6] K. Hsu, R. Majumdar, K. Mallik, and A.-K. Schmuck, "Multi-layered abstraction-based controller synthesis for continuous-time systems," in *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (Part of CPS Week)*, ser. HSCC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 120–129.

[7] J. Calbert, B. Legat, L. N. Egidio, and R. Jungers, "Alternating simulation on hierarchical abstractions," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 593–598.

[8] T. S. Badings, A. Abate, N. Jansen, D. Parker, H. A. Poonawala, and M. Stoelinga, "Sampling-based robust control of autonomous systems with non-gaussian noise," *arXiv preprint arXiv:2110.12662*, 2021.

[9] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Strategy synthesis for partially-known switched stochastic systems," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*. ACM, may 2021.

[10] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. Association for Computing Machinery, 2013, pp. 854–860.

[11] 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.

[12] M. Dutreix and S. Coogan, "Specification-guided verification and abstraction refinement of mixed monotone stochastic systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 2975–2990, 2020.

[13] G. Reissig, "Computing abstractions of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2583–2598, 2011.

[14] A. Weber and G. Reissig, "Classical and strong convexity of sublevel sets and application to attainable sets of nonlinear systems," *SIAM J. Control. Optim.*, vol. 52, pp. 2857–2876, 2014.

[15] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[16] R. Majumdar, N. Ozay, and A.-K. Schmuck, "On abstraction-based controller design with output feedback," in *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, 2020, pp. 1–11.

[17] A.-K. Schmuck and J. Raisch, "Asynchronous l-complete approximations," *Systems & Control Letters*, vol. 73, pp. 67–75, 2014.

[18] A. Banse, L. Romao, A. Abate, and R. M. Jungers, "Data-driven memory-dependent abstractions of dynamical systems," *arXiv preprint arXiv:2212.01926*, 2022.

[19] L. Perko, *Differential equations and dynamical systems*. Springer Science & Business Media, 2013, vol. 7.

[20] N. Privault, *Understanding Markov chains*. Springer, 2013, vol. 357.

[21] H. K. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[22] L. Perko, *Differential Equations and Dynamical Systems*, ser. Texts in Applied Mathematics. Springer-Verlag New York, 1991.

[23] R. Coppola, A. Peruffo, and M. Mazo Jr, "Data-driven abstractions for verification of deterministic systems," *arXiv preprint arXiv:2211.01793*, 2022.

[24] M. Kazemi, R. Majumdar, M. Salamati, S. Soudjani, and B. Wooding, "Data-driven abstraction-based control synthesis," *arXiv preprint arXiv:2206.08069*, 2022.