2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Data-Driven Approximate Abstraction for Black-Box Piecewise Affine Systems

Gang Chen and Zhaodan Kong

Abstract— **How to effectively and reliably guarantee the correct functioning of safety-critical cyber-physical systems in uncertain conditions is a challenging problem. This paper presents a data-driven algorithm to derive approximate abstractions for piecewise affine systems with unknown dynamics. It advocates a significant shift from the current paradigm of abstraction, which starts from a model with known dynamics. Given a black-box system with unknown dynamics and a linear temporal logic specification, the proposed algorithm is able to obtain an abstraction of the system with an arbitrarily small error and a bounded probability. The algorithm consists of three components, system identification, system abstraction, and active sampling. The effectiveness of the algorithm is demonstrated by a case study with a soft robot.**

I. Introduction

The proliferation of cyber-physical systems (CPSs) brings how to effectively and reliably guarantee their correct behaviors to the forefront of problems we as control engineers need to address. One natural choice to attain correct functioning is to consider formal methods techniques, e.g., model checking [1]. One crucial component of formal methods is a precise and potentially concise mathematical model of the system under investigation. However, in reality we rarely have full knowledge of complex CPSs during their design and even testing phases. Thus how to attain formal guarantee for systems with partially or fully unknown dynamics becomes a problem of practical significance.

In this paper, we aim to address this problem in the context of *abstraction*. Given a system model $\mathcal{T}$ and a formal specification ϕ written, for instance, in linear temporal logic (LTL), an *abstract* model of $\mathcal{T}$ is a simpler model $\mathcal{T}'$; checking whether the simpler model $\mathcal{T}'$ satisfies ϕ suffices to decide whether $\mathcal{T}$ satisfies ϕ. For systems that can be described by discrete state models, abstraction can be achieved by using the concepts of simulation and bi-simulation [2], [3]. In control community, recently there have been many successful efforts pertaining to the abstraction of systems of more realistic dynamics, such as those that are piecewise affine [4], [5], [6], [7]. All these studies, as far as we know, assume models of known dynamics, which significantly impedes the application of abstraction in the analysis and design of systems with inherent uncertainties, e.g., those needing to interact with a variety of human users.

One principle way of mitigating uncertainties is to utilize machine learning. Actually, the integration of formal methods and machine learning has shown great potential in the formal design, verification, and validation of CPS recently [8], [9], [10], [11]. In this paper, we will focus on how to combine machine learning techniques, particularly system identification [12] and active learning [18], and formal methods techniques [13] to generate approximate abstractions for systems with black-box (unknown), piecewise affine (PWA) dynamics.

The major contribution of this paper is that it addresses many theoretical and algorithmic issues pertaining to the integration of existing approximate abstraction techniques [13] and system identification techniques [12]. Given a system with unknown PWA dynamics, the paper shows that it is possible to extract an abstract model with an arbitrarily small error and a bounded probability (under certain mild assumptions). Even though the paper focuses on PWA systems, the preliminary results obtained in it can potentially pave the way for future developments for systems with more complex dynamics.

II. Preliminaries and Notation

A N dimensional *polytope* $\mathcal{X}$ is defined as the convex hull of at least $N+1$ affinely independent vectors in $\mathbb{R}^N$. A complete partition of $\mathcal{X}$ is a set of open polytopes $\mathcal{X}_i, i \in I$ (I is a finite index set) in $\mathbb{R}^N$ such that $\mathcal{X}_{i_1} \cap \mathcal{X}_{i_2} = \emptyset$ for all $i_1, i_2 \in I, i_1 \neq i_2$ and $cl(\mathcal{X}) = \cup_{i\in I} cl(\mathcal{X}_i)$, where $cl(\mathcal{X}_i)$ denotes the closure set of $\mathcal{X}_i$. According to the H-representation, each $\mathcal{X}_i, i \in I$ can be represented as $\mathcal{X}_i = \{x \in \mathbb{R}^N : H_i x \prec K_i\}$, where $\prec$ denotes componentwise inequality.

A *piecewise affine (PWA) system* [14] can be written as follows:

$$\begin{aligned} x_{k+1} &= f(x_k) + e \\ f(x) &= \begin{cases} A_1 x + b_1 \text{ if } x \in \mathcal{X}_1 \\ \vdots \\ A_s x + b_s \text{ if } x \in \mathcal{X}_s \end{cases} \end{aligned} \tag{1}$$

where x_k is the state of the system at step k; $f : \mathcal{X} \to \mathcal{R}^N$ is a PWA map; $e \in \mathcal{N}(0, \sigma_e^2)$ is an independently, identically distributed and zero mean Gaussian noise with standard deviation σ_e; s is the number of modes; A_i, b_i are the parameters of the i-th mode ($A_i, i = 1, \cdots, s$ is assumed to be nonsingular in this paper); and all s modes together constitute a complete partition of $\mathcal{X}$.

A *transition system* is a tuple $\mathcal{T} = (Q, \delta, O, o)$, where Q is the state space; $\delta : Q \to 2^Q$ (2^Q is the powerset of Q) is a transition map assigning a state $q \in Q$ to its next state $q' \in Q$; O is the set of observations; and $o : Q \to O$ is

Authors are with the Department of Mechanical and Aerospace Engineering, University of California, Davis. Z. Kong is the corresponding author (email: zdkong@ucdavis.edu). This work was partially supported by the Hyundai Motor Company.

an observation map assigning each $q \in Q$ an observation $o(q) \in O$ [1]. We denote a region of the state space as $P \subset Q$. The *embedding transition system* of a PWA system $\mathcal{S}$ described by Eqn. (1) is a tuple $\mathcal{T}_e = (Q_e, \delta_e, O_e, o_e)$, where $Q_e = \cup_{i \in I} \mathcal{X}_i$; $\delta_e : x \to x'$ iff there exists $i \in I$ such that the transition from x to x' satisfies Eqn. (1); $O_e = I$; and $o_e(x) = i$ iff $x \in \mathcal{X}_i$ [13].

The *reachability metric* over two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$ is defined as [15]:

$$d(\mathcal{T}_1, \mathcal{T}_2) = h(Reach(\mathcal{T}_1), Reach(\mathcal{T}_2)),$$

where h is the Hausdorff distance and $Reach(\mathcal{T})$ is the set of all reachable states of $\mathcal{T}$. Given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$ with the same observation set O and a reachability metric d defined over them, a relation $\mathcal{S}_\sigma \subseteq Q_1 \times Q_2$ is called a *$\sigma-$approximate simulation relation* of $\mathcal{T}_1$ by $\mathcal{T}_2$ [15] if for all $(q_1, q_2) \in \mathcal{S}_\sigma$:

- $d(o(q_1), o(q_2)) \leq \sigma$,
- $\forall q_1' = Post(q_1)$, there exists $q_2' = Post(q_2)$, such that $(q_1', q_2') \in \mathcal{S}_\sigma$, where $Post(P) = \{q \in Q \mid \exists p \in P \text{ with } p \to q\}$.

Moreover, $\mathcal{T}_1$ is said to be *$\sigma-$approximately simulated* by $\mathcal{T}_2$, denoted $\mathcal{T}_1 \prec_\sigma \mathcal{T}_2$.

III. Problem Statement

Formally, in this paper, we wish to solve the following problem:

Problem 1: Given a PWA system $\mathcal{S}$ with unknown dynamics, an LTL specification ϕ, and a bound $\sigma > 0$, find a finite transition system $\hat{\mathcal{T}}$ such that $p(\hat{\mathcal{T}} \prec_\sigma \mathcal{T}) > 1 - \delta$, where $p(.)$ stands for probability, $\mathcal{T}$ is the true abstract transition system of $\mathcal{S}$, and δ is bounded.

Remark 1: By unknown dynamics, we mean that the following system characteristics are unknown: (i) the number of modes, s, (ii) the parameters related to the dynamics of each mode, $\{A_i, b_i, i = 1, \cdots, s\}$, (iii) the parameters related to the partitions (regions) of the state space, $\{H_i, K_i, i = 1, \cdots, s\}$, and (iv) the standard deviation of the Gaussian noise, σ_e. But we assume that our algorithm, which will be presented in the next section, can use the system as a black-box simulator to generate samples. This is a reasonable assumption since during system design and testing phases, engineers can always get access to, e.g., a computer simulation of the system [16].

Remark 2: Notice that the requirement $p(\hat{\mathcal{T}} \prec_\sigma \mathcal{T}) > 1 - \delta$ is inspired by the concept of probably approximately correct (PAC) models in machine learning [17]. It simply says that the probability that the transition system $\hat{\mathcal{T}}$ (obtained by using our algorithm) is a σ-approximate simulation by the PWA system $\mathcal{S}$ is higher than $1 - \delta$. In other words, given a PWA system with unknown dynamics, we intend to find out its approximate abstract transition system $\hat{\mathcal{T}}$ with a high enough confidence.

IV. Data-Driven Abstraction Algorithm

Fig. 1 illustrates the basic architecture of our data-driven abstraction algorithm to solve Problem 1. The inputs of the

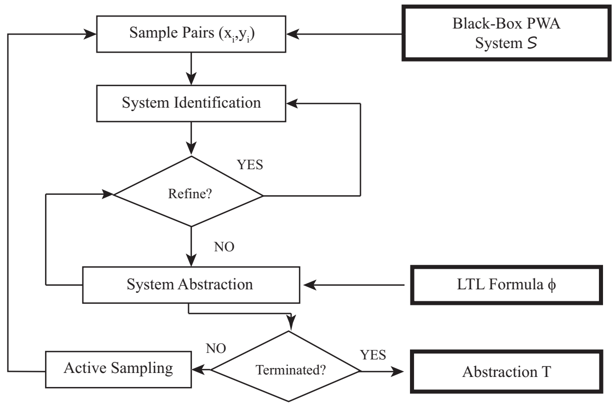


Fig. 1: Architecture of our data-driven abstraction algorithm.

algorithm are a black-box PWA system $\mathcal{S}$ with unknown dynamics and an LTL specification ϕ; the output of the system is a transition system $\mathcal{T}$. The algorithm can be roughly divided into three components: system identification, system abstraction, and active sampling. The goal of the *system identification* component is to derive an estimated PWA model $\hat{\mathcal{S}}$ based on the data sampled from the black-box system $\mathcal{S}$ (serving as a simulator). The goal of the *system abstraction* component is to derive a transition system $\mathcal{T}$ given the identified PWA model $\hat{\mathcal{S}}$ and the specification ϕ. One important procedure of the system identification component is refinement, which refines the estimated model $\hat{\mathcal{S}}$, until no significant improvement can be achieved, based on the currently available data and the current abstraction $\mathcal{T}$. Finally, if no satisfactory abstraction $\mathcal{T}$ can be found after the refinement, the *active sampling* component will be implemented to draw new data points with the help of the black-box simulator $\mathcal{S}$.

A. System Identification

Given a black-box PWA system $\mathcal{S}$, or subsequently a set of K samples $\mathcal{D} := \{y_k, x_k\}, k = 1, \cdots, K$, the system identification component identifies a PWA model, specified by the number of modes s as well as the mode parameters $\hat{A}_i, \hat{b}_i, \hat{H}_i$ and $\hat{K}_i$ with $i = 1, \cdots, s$. The problem itself is a well-studied problem [12]. Specifically, we need to find (i) a minimum positive integer, s, (ii) a set of parameter matrices, $\{\hat{A}_i\}_{i=1}^s$ and $\{\hat{H}_i\}_{i=1}^s$, and (iii) a set of parameter vector $\{\hat{b}_i\}_{i=1}^s$ and $\{\hat{K}_i\}_{i=1}^s$ (notice that $\{\hat{H}_i\}_{i=1}^s$ and $\{\hat{b}_i\}_{i=1}^s$ together constitute a complete partition $\{\mathcal{X}_i\}_i^s$ of the PWA system's state space $\mathcal{S}$), such that the estimated parameters are the solution of the following minimization problem:

$$(\hat{A}_i, \hat{b}_i, \hat{H}_i, \hat{K}_i) = \underset{(x_k, y_k) \in (\mathcal{D} \cap \mathcal{X}_i)}{\operatorname{argmin}} \frac{1}{K} \sum_{k=1}^{K} c(y_k - \hat{f}(x_k)) \quad (2)$$

where $\hat{f}(.)$ is specified by $\hat{A}_i, \hat{b}_i, \hat{H}_i$ and $\hat{K}_i, i = 1, \cdots, s$ and c is a given penalty function, which is chosen to be $c(\cdot) = ||\cdot||_2$ in this paper. Notice that solving the identification problem involves the simultaneous solving of two subproblems, data classification and parameter estimation. Once the data points have been classified into clusters $\{\mathcal{D}_i\}_{i=1}^s$

such that $(y_k, x_k) \in \mathcal{D}_i$, i.e., (y_k, x_k) is attributed to the i-th mode, mode parameters $\hat{A}_i, \hat{b}_i, \hat{H}_i$ and $\hat{K}_i$ can be easily estimated by solving Eqn. (2).

Our system identification component is modified from the method proposed in [14]. It consists of two main procedures: initialization and refinement. One major difference between our method and the one in [14] is that we utilize the current abstract transition system to guide the refinement.

Algorithm 1: System Identification Initialization

```
Input : A bound $\hat{\sigma}$, a set of samples
        $\mathcal{D} := \{(y_k, x_k)\}, k \in \{1, \cdots, K\}$, a ratio
        $0 < r < 1$, and a large number $J$
Output: $\hat{A}_i, \hat{b}_i, \mathcal{D}_i, \hat{\mathcal{H}}_i, \hat{\mathcal{K}}_i, i = 1, \cdots, s$
1: Set $l = 0$ and $I_0 = \{1, \cdots, K\}$;
2: for $|I_{l+1}| \geq rK$ do
3:     Randomly generate a set of parameters
       $\{A_j, b_j\}, j \in \{1, \cdots, J\}$;
4:     Construct sets
       $\Sigma_j = \{\| y_k - (A_j x_k + b_j) \| \leq \hat{\sigma}, k \in I_l\}$ for each
       $j \in \{1, \cdots, J\}$;
5:     Set $l = l + 1$ and $\Sigma_{max} = \text{argmax}_{j=1}^{J} |\Sigma_j|$;
6:     With $\Sigma_{max}$, estimate $\hat{A}_l$ and $\hat{b}_l$ by solving
           $(\hat{A}_l, \hat{b}_l) = \underset{A,b}{\text{argmin}} \sum_{k=1}^{|\Sigma_{max}|} c(y_k - (Ax_k + b))$;
7:     Set $\mathcal{D}_l = \{k \in I_l : \| y_k - (\hat{A}_l x_k + \hat{b}_l) \| \leq \hat{\sigma}\}$ and
       $I_{l+1} = I_l \setminus \mathcal{D}_l$;
8: Find boundaries specified by $\{\hat{H}\}_{i=1}^{l}$ and $\{\hat{K}\}_{i=1}^{l}$
   between sets $\{\mathcal{D}_i\}_{i=1}^{l}$; set $s = l$.
```

1) Initialization: The pseudo code of the initialization procedure is shown in Alg. 1. The steps are rather self-explanatory. Here we would like to provide a few simple comments for clarification. We need to randomly generate a matrix $\hat{A}_j$ and a vector $\hat{b}_j$ (Line 3) in each loop, which is quite inefficient; thus the termination condition $|I_{l+1}| \geq rK$ (Line 2) can be set loosely, i.e., with a rather large r. Such a practice is reasonable, given the fact that the initialization procedure is only meant to generate some good enough partitions (modes), which will be further refined in the refinement procedure. As for the boundaries specified by $\{\hat{H}\}_{i=1}^{l}$ and $\{\hat{K}\}_{i=1}^{l}$ between sets $\{\mathcal{D}_i\}_{i=1}^{l}$ (Line 8), standard support vector machines (SVM) regression methods can be used to compute them.

2) Refinement: Given the random nature of the way $\hat{A}_i, \hat{b}_i, i = 1, \cdots, s$ are generated in Alg. 1, it is quite unlikely that we are able to identify all the correct modes with the initialization procedure. There are two main issues: data points that belong to more than one mode (called "undecidable data points" in [14]) and data points that don't belong to any mode (called "unfeasible data points" in [14]). We refine the identified system model by eliminating these two types of data points as shown in Alg. 2.

Algorithm 2: System Identification Refinement

```
Input : Current abstract transition system $\mathcal{T}$,
        parameters $\hat{A}_i, \hat{b}_i, \mathcal{D}_i, \hat{H}_i, \hat{K}_i$, $i = 1, \cdots, s$
        obtained in Alg. 1, and a bound $\hat{\sigma}$
Output: $\hat{A}_i, \hat{b}_i, \hat{H}_i, \hat{K}_i, i = 1, \cdots, s$
  Set $\beta \geq 0, \mu \geq 0, \kappa \geq 0, 0 < \theta < 1$, $l = 1$;
while Not terminated do
    1: Compute $(i^*, j^*) = \text{argmin}_{1 \leq i < j \leq s} \beta_{i,j}$ with
       $\beta_{i,j} = \| \hat{A}_i - \hat{A}_j \|$;
    2: if $\beta_{i^*,j^*} \leq \theta^t \beta$ then
    3:     Merge modes $i^*$ and $j^*$; Set $s = s - 1$;
    4:     Recompute $\mathcal{D}_{i^*} = \{k \in \{1, \cdots, K\} :$
           $\| y_k - (\hat{A}_{i^*} x_k + \hat{b}_{i^*}) \| \leq \hat{\sigma}\}$;
    5: Use $CR()$ and rule $Dis()$ to reassign points;
    6: Compute $i^* = \text{argmin}_{i=1,\cdots,s} |\mathcal{D}_i|/|\mathcal{D}|$;
    7: if $|\mathcal{D}_i|/|\mathcal{D}| \leq \theta^t \mu$ then
    8:     Discard mode $i^*$; let $s = s - 1$;
    9:     Go to Step 4;
    10: Store $\{\hat{A}_i\}_{i=1}^{s}$ as $\{\hat{A}_i^{old}\}_{i=1}^{s}$;
    11: Update $\hat{A}_i, \hat{b}_i, \hat{H}_i, \hat{K}_i$ with new $\{\mathcal{D}_i\}_{i=1}^{s}$;
    12: if $\| \hat{A}_i - \hat{A}_i^{old} \| \leq \kappa$ then
    13:     Terminated.
    14: else
    15:     $l = l + 1$;
end
```

B. System Abstraction

Given an estimated PWA model $\hat{\mathcal{S}}$ (or $\hat{f}$), parameterized by $\hat{A}_i, \hat{b}_i, \hat{H}_i$ and $\hat{K}_i$ with $i = 1, \cdots, s$, and an LTL formula ϕ, the goal of the system abstraction component is to generate a good enough abstraction $\mathcal{T}$. We roughly follow the approximate abstraction procedures described in [13] to design and implement the system abstraction component. Here we are just going to provide a rough outline of the abstraction algorithm. Interested readers can refer to [13] for more details. First, a deterministic Buchi automaton $\mathcal{B}_\phi$ is constructed from the formula ϕ. Second, the corresponding embedding transition system $\mathcal{T}_e$ is constructed for $\hat{\mathcal{S}}$ by simply using the definition of embedding transition system. Third, an observation map o_e is created by partitioning the state space of the system $\mathcal{T}_e$ into uniform grids. Fourth, given the system $\mathcal{T}_e$, the observation map o_e, and the LTL formula ϕ (or its corresponding Buchi automaton $\mathcal{B}_\phi$), an initial transition system $\mathcal{T}_0$ is constructed by following standard abstraction procedures, such as those prescribed in [1]. Fifth, a product automaton $\mathcal{P}$ is constructed as $\mathcal{P} = \mathcal{T}_0 \times \mathcal{B}_\phi$, which concerns both the initial transition system $\mathcal{T}_0$ and the specification ϕ. The product automaton is a tuple $\mathcal{P} = (S_p, S_{p0}, \delta_p, F_p)$, where S_p is the set of states, S_{p0} is the set of initial states, δ_p is the transition map, and F_p is the acceptance condition. Finally, refinement is conducted by solving a deterministic Rabin game. The pseudo code of the abstraction refinement procedure is shown in Alg. 3. In the algorithm, $S_\top$ is the set of states from which all traces are accepted by $\mathcal{P}$, $S_\bot$ is the set of states from which no trace is accepted by $\mathcal{P}$, and S_u: the set of states from which some but not all traces are accepted by $\mathcal{P}$.

Algorithm 3: System Abstraction Refinement

```
Input : Current abstract transition system $\mathcal{T}$, current
        product automaton $\mathcal{P}$, an initial state $q$ of $\mathcal{T}$,
        and a ratio $0 < \eta < 1$
Output: Refined abstract transition system $\hat{\mathcal{T}}$ and
        refined product automaton $\hat{\mathcal{P}}$
  Initialize $\hat{\mathcal{T}} = \mathcal{T}, \hat{\mathcal{P}} = \mathcal{P}$, and $S_u = \emptyset$;
while $|S_u| \geq \eta|Q|$ do
  1: Compute $S_\top$ and $S_\bot$ for $\hat{\mathcal{P}}$;
  2: Set $S_u := \mathcal{P}_p \setminus (S_\top \cup S_\bot)$;
  3: for all $(q, g) \in S_u$ do
  4:   if $q \in S_u$ then
  5:     Set $\mathbb{q} := q$;
  6:     for all $(\exists q_r \in \mathbb{q}, q' \in Post(q_r)$ and
         $(q_r \cap Pre(q')) \neq \emptyset)$ do
  7:       Construct states $q_1, q_2$ such that
           $q_1 := q_r \cap Pre(q')$,
           $q_2 := q_r \setminus Pre(q')$;
  8:       $\mathbb{q} := (\mathbb{q} \setminus q_r) \cup \{q_1, q_2\}$;
  9:       $q := \mathbb{q}$;
 10:       Update $\delta$ and $o$ for $\hat{\mathcal{T}}$;
 11: Update $\hat{\mathcal{P}}$ and $\hat{\mathcal{T}}$.
end
```

C. Active Sampling

The system identification component and the system abstraction component described in the last two sub-sections are based on a fixed data set $\mathcal{D} := \{(y_k, x_k)\}_{k=1}^K$. It is quite obvious that the quality of the system identification and the system abstraction depends on the quality of the data set. To improve the quality of the system identification (in other words, to decrease the number of data points needed for the system identification), we use an active learning algorithm developed by our group [18] to sample high quality (or "informative") data points for the system identification component after the initial round (see Fig. 1).

The strategy to find the next data point to sample for round $t+1$ has two steps. In the first step, the best candidate for each mode is identified as follows:

$$x_i := \underset{x \in \hat{\mathcal{X}}_i}{\operatorname{argmax}}(\Psi_{i,t}(x) + \lambda_t^{1/2} \varrho_{i,t}(x)) \tag{3}$$

where $\Psi_{i,t}(x)$ is the Gaussian process regression mean of the prediction error defined over the data points in $\mathcal{D}_i$, $\varrho_{i,t}(x)$ is the Gaussian process regression variance of the prediction error defined over the data points in $\mathcal{D}_i$, and λ_t is a regularization factor. In the second step, the active learning algorithm chooses mode $i^* = \operatorname{argmin}_{i=1,\cdots,s}(\max_{x \in \hat{\mathcal{X}}_i} \Psi_{i,t}(x))$ and the corresponding best candidate x_{i^*} to sample.

D. Main Theorem

Assumption 1: Assume the prediction error of the system identification component described in Section IV-A can be characterized by a zero mean Gaussian with a bounded variance, i.e., $f(x) - \hat{f}(x) \sim \mathcal{N}(0, \sigma_p(x)^2)$ and $\sigma_p(x) \leq C$, where $f(x)$ is the real PWA dynamics and $\hat{f}(x)$ is the estimated PWA dynamics.

Given this assumption, we can now proceed to the main theorem of this paper (interested readers can refer to our report [19] for the proof.)

Theorem 1: Given a PWA system $\mathcal{S}$ with unknown dynamics, for any $\sigma > \varepsilon > 0$, the algorithm described in this paper (with the assumption that the standard deviation of the prediction error for the system identification component is bounded by C) can obtain an approximate abstract transition system $\hat{\mathcal{T}}$ that is σ-approximately simulated by the true abstract transition system $\mathcal{T}$ of $\mathcal{S}$ with a bounded probability that is greater than $1 - \delta$, where $\delta = 1 - \frac{1}{\sqrt{2\pi(\sigma_e + C)}} \int_{-\sigma+\varepsilon}^{\sigma-\varepsilon} exp(-\upsilon^2/(2(\sigma_e + C)))d\upsilon$.

V. Case Study

In this section, we will use a soft robot driven by series pneumatic artificial muscles as an example to demonstrate our algorithm.

A. Model and LTL Specification

In this case study, we will focus on a particular type of soft robots, those that are driven by series pneumatic artificial muscles (sPAM). Such a robot can have two polyethylene tubing sPAMs, each of which is controlled by a corresponding actuator with pressurized air. Even though the dynamics of such a robot is nonlinear, in [20], the authors have shown that its closed-loop behavior can be approximated by piecewise affine dynamics. Here let's assume that the robot under investigation has dynamics as follows:

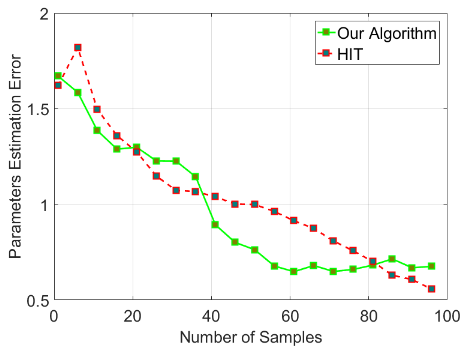
$$x_{k+1} = f(x_k) + e \tag{4}$$

and

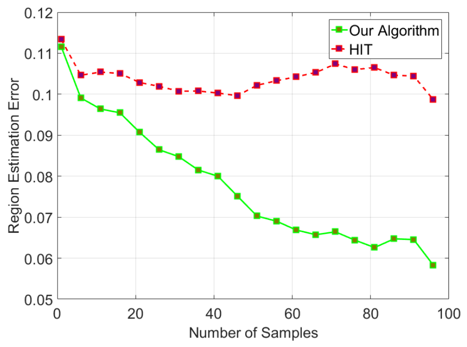
$$f(x) = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 0.98 \end{bmatrix} x \text{ if } 0 \leq x_k(1) \leq 0.3 \\ \begin{bmatrix} 0.83 & 0.12 \\ 0.12 & 0.81 \end{bmatrix} x + \begin{bmatrix} 0.01 \\ 0.03 \end{bmatrix} \quad \text{if } x_k(1) \geq 0.3 \end{cases}$$

where $x = (x(1), x(2))^T$ is the coordinate of the moving platform and e is a Gaussian noise with a zero mean and a standard deviation of 0.1. Please keep in mind that the model is unknown to our algorithm but the model itself can be used by our algorithm as a simulator to generate samples.

The specification that needs to be verified is an LTL formula $\phi := \Box(\pi_1 \wedge \Diamond\pi_2)$, where "$\pi_1 := x(1)$ is below 0.3", "$\pi_2 := x(2)$ is above 0.6", and $\Box$ is the temporal operator "Always", $\Diamond$ is the temporal operator "Eventually". Put together, ϕ specifies that "it should always be true that $x(1)$ is below 0.3 and eventually $x(2)$ is above 0.6".



(a)



(b)

Fig. 2: Comparison results of the system identification component used in this paper (green) and the HIT algorithm (red): (a) Average parameter estimation error with respect to the number of samples; (b) Average region (partition) estimation error with respect to the number of samples.

B. Implementation Results

The algorithm proposed in this paper is implemented as a Matlab tool. The tool takes an LTL formula and a black-box PWA system as inputs and it outputs an abstract transition system. The active learning used in the tool is the Gaussian Process Adaptive Confidence Bound (GP-ACB) algorithm proposed and implemented by our group in [18]. Moreover, a Gaussian kernel function is used in GP-ACB. In order to demonstrate the effectiveness of our algorithm, here we will provide two sets of implementation results, one regarding the system identification component and the other one regarding the entire algorithm.

1) Performance of the System Identification Component: Here we compare the performance of our system identification component with an existing state-of-the-art black-box system identification tool called the Hybrid Identification Tool (HIT) [21]. One thing we would like to point out here is that the evaluation of our system identification component is conducted in conjunction with other components, e.g., abstraction and active sampling, in the loop. It should be expected that, at worst, our component should have the same performance as the HIT. However, it may also be expected that factors such as active-learning based sampling and abstraction-guided refinement can potentially improve the identification performance, which turns out to be the case, at least for this particular case study.

We use two metrics to quantify system identification errors: *parameter estimation error* and *region estimation error*. Given the parameters of a real PWA model (unknown to the investigated algorithms), the *parameter estimation error* is the sum of the Euclidean distances between the real parameters and the estimated parameters. The *region estimation error* is defined the sum of the following Hausdorff distance:

$$e(\hat{\mathcal{X}}_i, \mathcal{X}_i) = \max\{\sup_{x\in\mathcal{X}_i}\inf_{y\in\hat{\mathcal{X}}_i} d(x,y), \sup_{y\in\hat{\mathcal{X}}_i}\inf_{x\in\mathcal{X}_i} d(x,y)\},$$

where $e(\hat{\mathcal{X}}_i, \mathcal{X}_i)$ the region estimation error related to the ith mode with $\mathcal{X}_i$ as the real region or partition and $\hat{\mathcal{X}}_i$ as the estimated one, $d(x,y)$ is the Euclidean distance between x and y, and $\sup$ and $\inf$ stand for supremum and infimum, respectively. To calculate the Hausdorff distance, we randomly generate 100 samples inside the state space $\hat{\mathcal{X}}_i$ and the state space $\mathcal{X}_i$.

The comparison results based on 5 trials are shown in Fig. 2. It shows that, averagely speaking, the system identification component proposed in this paper has a comparable or better performance than HIT. Particularly, Fig. 2b shows that our algorithm has a faster convergent rate regarding the region estimation error. This is probably due to the fact that, generally speaking, active learning, which is used in our algorithm, out-performs its randomly sampling counterpart, which is used in HIT.

2) Performance of the Entire Algorithm: As the true abstract transition system of the system, described by Eqn. (4), is unknown, here we use the abstract transition system obtained by using the algorithm proposed in [13] as a benchmark. We will call this abstract transition system as $\mathcal{T}^*$. In [13], the authors have shown that even though their algorithm cannot attain the true abstract transition system, it still can get an abstract transition system that is arbitrarily close to the real one. Of course, we should point out that, in order to get this abstract transition system $\mathcal{T}^*$, the algorithm in [13] should have access to the system model, i.e., $\mathcal{T}^*$ is generated with a known model. In parallel, we run our algorithm to extract an abstract transition system $\hat{\mathcal{T}}$ without access to the dynamics of the model, i.e., $\hat{\mathcal{T}}$ is generated with a black-box system with unknown dynamics. Then in order to demonstrate the effectiveness of our algorithm, we need to show that $\hat{\mathcal{T}}$ should approach $\mathcal{T}^*$. This turns out to be the case, at least for this particular case study.

Based on the system dynamics, Eqn. (4), and the LTL specification ϕ, we implement the abstraction algorithm proposed in [13] and obtain an abstract transition system $\mathcal{T}^*$, which will be used as a benchmark. Then the algorithm proposed in this paper is applied to the same system, but with unknown dynamics. The output of the algorithm is another abstract transition system $\hat{\mathcal{T}}$. Here we use a metric that is inspired by the concept of approximate simulation to quantify the difference between the two transition systems. To be more specific, we set the abstraction error to σ if $\mathcal{T}^*$ is $\sigma-$approximately simulated by $\hat{\mathcal{T}}$. Moreover, the metric

σ is normalized to $\bar{\sigma}$ by the volume of the state space, i.e., $\bar{\sigma} := \sigma/|\mathcal{X}|$.

TABLE I: Comparison result with respect to different number of samples and a fixed number of refinement steps, which is set to 20.

	Number of samples		
	20	40	60
$\bar{\sigma}$	0.046	0.042	0.035

TABLE II: Comparison result with respect to different number of refinement steps and a fixed number of samples in active sampling component, which is set to 10.

	Refinement Steps		
	5	10	20
$\bar{\sigma}$	0.077	0.068	0.050

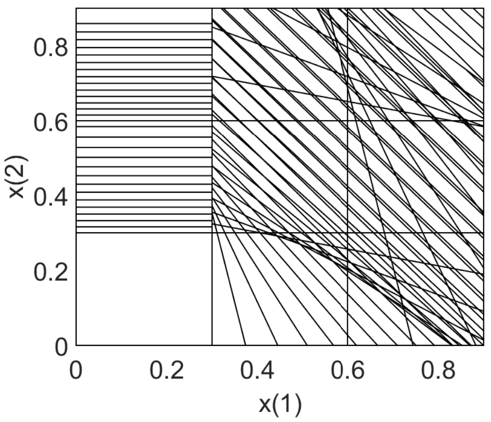


Fig. 3: State space partitions of the abstract transition system obtained by using our algorithm for the case study. The number of refinement steps and the number of samples in the active sampling component are both set at 20.

The comparison results are shown in Table I and Table II. Table I shows the comparison results with respect to different number of samples and a fixed number of refinement steps in the system abstraction component. Table II shows the comparison results with respect to different number of refinement steps and a fixed number of samples in the active sampling component. The tables show the average normalized errors $\bar{\sigma}$ based on 5 trials. The results show that the larger the number of refinement steps, the smaller the abstraction error; and the larger the number of samples, the smaller the abstraction error. The results also show that, even with a black-box system, our algorithm can attain an approximate abstract transition system $\hat{\mathcal{T}}$ that is close to the benchmark abstract transition system $\mathcal{T}^*$.

VI. Conclusions

In this paper, we proposed a data-driven approximate abstraction algorithm for piecewise affine systems with unknown dynamics. We demonstrated both theoretically and empirically that given a black-box PWA system and an LTL specification, we were able to derive an abstract transition system that is approximately simulated by the true abstract transition system. We demonstrated the effectiveness of our proposed algorithm with a soft robot as a case study.

References

[1] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of model checking*. MIT press, 2008.

[2] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 116–126, 2010.

[3] G. Pola, P. Pepe, and M. D. Di Benedetto, "Symbolic models for time-varying time-delay systems via alternating approximate bisimulation," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 14, pp. 2328–2347, 2015.

[4] B. Yordanov, J. Tumova, I. Cerna, J. Barnat, and C. Belta, "Temporal logic control of discrete-time piecewise affine systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1491–1504, 2012.

[5] E. A. Gol, X. Ding, M. Lazar, and C. Belta, "Finite bisimulations for switched linear systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3122–3134, 2014.

[6] J. A. DeCastro, V. Raman, and H. Kress-Gazit, "Dynamics-driven adaptive abstraction for reactive high-level mission and motion planning," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 369–376.

[7] G. Pola and M. D. Di Benedetto, "Symbolic models and control of discrete-time piecewise affine systems: An approximate simulation approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 175–180, 2014.

[8] T. Dreossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components," *arXiv preprint arXiv:1703.00978*, 2017.

[9] S. A. Seshia, S. Hu, W. Li, and Q. Zhu, "Design automation of cyber-physical systems: Challenges, advances, and opportunities," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016.

[10] A. Kozarev, J. Quindlen, J. How, and U. Topcu, "Case studies in data-driven verification of dynamical systems," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM, 2016, pp. 81–86.

[11] Z. Kong, A. Jones, and C. Belta, "Temporal logics for learning and detection of anomalous behavior," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1210–1222, 2017.

[12] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 344–355, 2012.

[13] B. Yordanov, J. Tumova, I. Cerna, J. Barnat, and C. Belta, "Formal analysis of piecewise affine systems through formula-guided refinement," *Automatica*, vol. 49, no. 1, pp. 261–266, 2013.

[14] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.

[15] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 782–798, 2007.

[16] X. Jin, A. Donzé, J. V. Deshmukh, and S. A. Seshia, "Mining requirements from closed-loop control models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1704–1717, 2015.

[17] L. Valiant, *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books (AZ), 2013.

[18] G. Chen, Z. Sabato, and Z. Kong, "Active learning based requirement mining for cyber-physical systems," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 4586–4593.

[19] G. Chen and Z. Kong, "Data-driven approximate abstraction for black-box piecewise affine systems," *arXiv preprint arXiv:1801.09289*, 2018.

[20] G. Andrikopoulos, G. Nikolakopoulos, I. Arvanitakis, and S. Manesis, "Piecewise affine modeling and constrained optimal control for a pneumatic artificial muscle," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 904–916, 2014.

[21] G. Ferrari-Trecate, "Hybrid identification Toolbox (HIT)," 2005.