Formal Guarantees in Data-Driven Model Identification and Control Synthesis

Sadra Sadraddini
Boston University
73o Commonwealth Ave.
Boston, MA
sadra@bu.edu

Calin Belta
Boston University
73o Commonwealth Ave.
Boston, MA
cbelta@bu.edu

ABSTRACT

For many performance-critical control systems, an accurate (simple) model is not available in practice. Thus, designing controllers with formal performance guarantees is challenging. In this paper, we develop a framework to use input-output data from an unknown system to synthesize controllers from signal temporal logic (STL) specifications. First, by imposing mild assumptions on system continuity, we find a set-valued piecewise affine (PWA) model that contains all the possible behaviors of the concrete system. Next, we introduce a novel method for STL control of PWA systems with additive disturbances. By taking advantage of STL quantitative semantics, we provide lower-bound certificates on the degree of STL satisfaction of the closed-loop concrete system. Illustrative examples are presented.

CCS CONCEPTS

• **Theory of computation → Timed and hybrid models**; *Mixed discrete-continuous optimization*; Modal and temporal logics; • **Applied computing →** *Engineering*;

KEYWORDS

Model Identification, Control Synthesis, Signal Temporal Logic

ACM Reference format:
Sadra Sadraddini and Calin Belta. 2018. Formal Guarantees in Data-Driven Model Identification and Control Synthesis. In *Proceedings of 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week), Porto, Portugal, April 11–13, 2018 (HSCC '18),* 10 pages.
DOI: 10.1145/3178126.3178145

1 INTRODUCTION

Many systems are performance-critical in the sense that all of their possible executions must meet certain properties. Most of such properties are expressed as hard constraints on the trajectories of the system. Examples include collision avoidance for mobile agents [32], safety thresholds in clinical applications [33], rules for traffic management [12], and temporal logic requirements for synthetic gene networks [6].

HSCC '18, Porto, Portugal

DOI: 10.1145/3178126.3178145

The goal of formal synthesis is to design provably-correct controllers for performance-critical systems. A wide range of interesting specifications can be described using temporal logics [5]. Several methods have been proposed to synthesize temporal logic controllers [7, 30]. Any guarantee in formal synthesis is valid as long as the model is valid. However, in many engineering applications perfect models are not available, or are too complex to use for synthesis purposes.

One way to approach a model-free synthesis problem is to adopt learning-based control methods. For example, [1, 27] studied reinforcement learning from temporal logic specifications. However, a completely model-free approach does not provide any formal guarantee since it is always possible to observe a new behavior from the system that may cause the specification to be violated. To overcome this issue, some works proposed considering a (highly non-deterministic) model that contains all the possible behaviors of yet-to-be-learned system. Thus, the state-space is safely explored - in the sense that a temporal logic specification is respected - while a possibly more accurate model and better performance may be obtained during learning [2, 4, 29]. In many applications, availability of a prior accurate model is asking for too much information. For example, the work in [4] assumes the prior system to be linear with all unknown non-linearities contained in a known polytope, which acts as a set-valued additive disturbance. In this setting, the values representing both the linear model and the disturbance polytope have to be known beforehand.

In this paper, we consider an unknown discrete-time system from which a finite set of input-output data is given. We also assume known values for bounds describing the system continuity - in a Lipschitz sense that is clarified in the paper. Continuity is essential to characterize the range of possible system behaviors. The goal is to design controllers from signal temporal logic (STL) [17] specifications over predicates on state. If STL satisfaction is not possible, we are still interested in finding the least-violating controllers. Our main results and contributions are as follows:

- We fit a piecewise affine (PWA) model to data and continuity constants. PWA models can capture arbitrarily high degrees of nonlinearity by tuning the number of modes. Unlike existing works on hybrid model identification [3, 8, 21], the nondeterministic set-valued model we compute is guaranteed to contain all the behaviors of the concrete system. All the non-determinism is captured by polytopic additive disturbances. Since such PWA models are not unique, we find ones that have the smallest non-determinism - in a sense that is clarified in the paper. The model identification technique in this paper is based on solving a series of non-convex

optimization problems, which are handled using mixed-integer linear programming (MILP).

- Once we have obtained a PWA model with additive polytopic disturbances, we design controllers from STL specifications. Unlike the existing works on STL control [13, 25, 28], our method is able to provide formal guarantees on the degree of satisfaction. The key idea is adopting the Tube model predictive control (Tube MPC), a well known technique in robust predictive control of linear systems [18], for STL control of hybrid systems. To serve this purpose, we introduce a method to exploit STL quantitative semantics for controller design. A novel method is introduced to compute the "best" STL tubes. Once the controller is designed, a lower-bound for the degree of satisfaction is obtained. Unlike finite-abstraction-based methods [7, 31, 34] which can only handle Boolean satisfaction, our approach takes advantage of the STL quantitative semantics and is optimization-based.

This paper is organized as follows. The notation and the necessary background on STL are provided in Sec. 2. The problem and the underlying assumptions are stated in Sec. 3. Technical details on model identification and control synthesis are provided in Sec. 4 and Sec. 5, respectively. An illustrative example is demonstrated in Sec. 6.

2 PRELIMINARIES

2.1 Notation

The set of natural, positive natural, real, and non-negative real numbers are denoted by $\mathbb{N}$, $\mathbb{N}_+$, $\mathbb{R}$, and $\mathbb{R}_+$, respectively. The empty set is denoted by $\emptyset$. The set of all finite sequences and infinite sequences that can be generated from an alphabet A are denoted by A^* and A^ω, respectively. A discrete-time real signal - simply referred to as *signal* in the rest of the paper - is an infinite sequence $s[0]s[1]s[2]\cdots$, where $s \in (\mathbb{R}^n)^\omega$, $s[k] \in \mathbb{R}^n, k \in \mathbb{N}$. All time intervals in this paper are interpreted in discrete-time: $[a,b] = \{a, a+1, \cdots, b\}, a, b \in \mathbb{N}, a < b$. Given sets $X, Y \subset \mathbb{R}^n$, their Minkowski sum is denoted by $X \oplus Y = \left\{x + y \middle| x \in X, y \in Y\right\}$. The Pontryagin's difference is defined as $X \ominus Y := \left\{x \middle| \{x\} \oplus Y \subset X\right\}$. The relation $\leq$ between two matrices of the same size is interpreted element-wise. The transpose of matrix M is denoted by M^T. The unit-vector in i^{th} direction and the vector of all ones in $\mathbb{R}^n$ are denoted by $e_{[i]}$ and 1_n, respectively. The absolute value of $x \in \mathbb{R}$ is shown by $|x|$, and the p-norm of $x \in \mathbb{R}^n$ is denoted by $\|x\|_p$. The unit p-norm ball is $\mathcal{B}_p := \left\{x \in \mathbb{R}^n \middle| \|x\|_p \leq 1\right\}$. The convex hull of $S \subset \mathbb{R}^n$ is denoted by $\text{Convh}(S)$. The indicator function is denoted by $\mathcal{I}$, where $\mathcal{I}(p)$ returns 1 if p = True, and 0 otherwise.

2.2 Signal Temporal Logic

In this paper, a subset of STL formulas is used. The original STL was introduced in [17] for monitoring bounded continuous-time real signals, but its principles still apply to discrete-time setting and unbounded signals.

Definition 2.1. The set of all *bounded* STL formulas is denoted by Φ^b, which is recursively defined as:

- $\pi \in \Phi^b$, where π is a predicate in the form $\pi := (f(s) \geq 0)$, $f : \mathbb{R}^n \to \mathbb{R}$;
- $\varphi_1, \varphi_2 \in \Phi^b \to \varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2 \in \Phi^b$, where $\wedge$ and $\vee$ are conjunction and disjunction connectives, respectively;
- $\varphi \in \Phi^b \to \mathbf{X}_{\{a\}}\varphi \in \Phi^b$, where $\mathbf{X}$ is the temporal "next" operator, and $a \in \mathbb{N}$.

Given a bounded interval $[a,b]$, other useful temporal operators are constructed:

- $\mathbf{F}_{[a,b]}\varphi := \bigvee_{k\in[a,b]} \mathbf{X}_{\{k\}}\varphi$,
- $\mathbf{G}_{[a,b]}\varphi := \bigwedge_{k\in[a,b]} \mathbf{X}_{\{k\}}\varphi$,
- $\varphi_1 \mathbf{U}_{[a,b]}\varphi_2 := \bigvee_{k\in[a,b]}(\mathbf{X}_{\{k\}}\varphi_2 \wedge \mathbf{G}_{[a,k]\varphi_1})$,

where $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{U}$ are temporal "eventually", "always", and "until" operators, respectively. We omit negation connective ($\neg$) from our definition as any temporal logic formula can be written in negation normal form [20]. We also do not allow predicates involving strict inequalities - our synthesis approach is optimization based and can handle only non-strict inequalities.

A *global* temporal logic formula is in the form $\mathbf{G}_{[a,\infty)}\varphi$, where $\varphi \in \Phi^b$. The set of *bounded-global* STL formulas - denoted by Φ - is defined as the set of all formulas that can be written in the form:

$$\phi = \bigvee_{i=1}^{n_\phi} \varphi_1^i \wedge \mathbf{G}_{[a,\infty)}\varphi_2^i, \tag{1}$$

where $\varphi_1^i, \varphi_2^i \in \Phi^b$, $i = 1, \cdots, n_\phi$, and $a \in \mathbb{N}$. It is straightforward to show that Φ is closed under disjunction, conjunction, and temporal "next" operator. For the remainder of this paper, we refer to bounded-global STL simply as STL.

Remark 1. *For any STL formula in negation normal form, unbounded intervals in "eventually" and "until"" operators can be safely under-approximated by bounded intervals [20]. This is not the case for unbounded "always". In addition, the form in* (1) *is not closed under bounded "always". However, nesting multiple unbounded "always" in a single STL formula - such that it can not be simplified - is rarely useful in applications.*

Definition 2.2. Given predicates on $\mathbb{R}^n$, the STL score function $\rho : \mathbb{R}^n \times \Phi \times \mathbb{N} \to \mathbb{R}$ is recursively defined as:

- $\rho(s, f(s) \geq 0, k) = f(s[k])$;
- $\rho(s, \varphi_1 \wedge \varphi_2, k) = \min(\rho(s, \varphi_1, k), \rho(s, \varphi_2, k)$;
- $\rho(s, \varphi_1 \vee \varphi_2, k) = \max(\rho(s, \varphi_1, k), \rho(s, \varphi_2, k)$;
- $\rho(s, \mathbf{X}_{\{a\}}\varphi, k) = \rho(s, \varphi, k + a)$;
- $\rho(s, \mathbf{G}_{[a,\infty)}\varphi, k) = \inf_{k' \in \mathbb{N}} \rho(s, \varphi, k + a + k')$.

The STL score function has the following distance property [17]:

$$\left|\rho(s, \varphi, k) - \rho(s', \varphi, k)\right| \leq C \sup_{k'>k} \|s'[k'] - s[k']\|_\infty, \tag{2}$$

where C is a constant determined by the functions appearing in the predicates. It measures how sensitive the predicates are to changes in signal values - see [17] for further discussion.

Definition 2.3. Given $\varphi \in \Phi$ with predicates on $\mathbb{R}^n$, and $\epsilon \in \mathbb{R}$, the ϵ-language of φ is defined as the following set:

$$\mathcal{L}(\varphi, \epsilon) := \left\{s \in (\mathbb{R}^n)^\omega \middle| \rho(s, \varphi, 0) \geq \epsilon\right\}. \tag{3}$$

If $\epsilon_1 \geq \epsilon_2$, then $\mathcal{L}(\varphi, \epsilon_1) \subseteq \mathcal{L}(\varphi, \epsilon_2)$. In a Boolean sense, s satisfies φ if and only if $s \in \mathcal{L}(\varphi, 0)$.

Remark 2. *In our discrete-time setting, any STL formula can be translated into a linear temporal logic (LTL) [5] formula by appropriately using the "next" operator. However, LTL representation of bounded-interval temporal operators may be very inefficient. Since we deal with real-valued systems and quantitative semantics, the formalism of STL is preferred.*

3 PROBLEM STATEMENT

Informally, the goal is to use data gathered from an unknown system to design a control policy such that an STL formula over predicates on state is satisfied by all closed-loop trajectories originating from a designated set of initial conditions. If STL satisfaction is not possible, we still want to compute the control policy resulting in the least worst-case STL violation. We formalize this problem in this section.

3.1 System and Assumptions

Definition 3.1. Given a workspace $X \subset \mathbb{R}^n$ and an admissible set of inputs $U \subset \mathbb{R}^m$, a control system $\mathcal{F}$ is defined as a *triadic relation*

$$\mathcal{F} \subset X \times U \times \mathbb{R}^n, \tag{4}$$

which is *left-total* in the sense that $\forall x \in X, \forall u \in U, \exists x^+ \in \mathbb{R}^n$ such that $(x, u, x^+) \in \mathcal{F}$.

Sets X and U are assumed to be compact, bounded, and locally connected in their respective domain. Note that we have not assumed that X is invariant for all controls. It may be possible that $\exists (x, u, x^+) \in \mathcal{F}$ such that $x \in X, u \in U$, but $x^+ \notin X$. Keeping the state within the workspace is a non-trivial task that is an implicit objective of our problem. A control system $\mathcal{F}$ is *deterministic* if for all $(x_1, u_1, x_1^+), (x_2, u_2, x_2^+) \in \mathcal{F}$, $x_1 = x_2$ and $u_1 = u_2$ implies $x_1^+ = x_2^+$.

Definition 3.2. A control policy $\mu : X^* \to U$ is a function that determines the control input at time t as a feedback of the history of the system:

$$u[t] = \mu(x[0]x[1] \cdots x[t]). \tag{5}$$

Definition 3.3. Given a control system $\mathcal{F}$, a control policy μ, a set of initial conditions $X_0 \subseteq X$, we define the *closed-loop language* as $\mathcal{L} = \mathcal{L}(\mathcal{F}, X_0, \mu) \subset X^\omega$ such that $x[0]x[1] \cdots \in \mathcal{L}$ if and only if $x[0] \in X_0$ and

$$(x[k], \mu(x[0]x[1] \cdots x[k]), x[k+1]) \in \mathcal{F}, \forall k \in \mathbb{N}.$$

Assumption 1. *There exists a control policy μ such that for some $X_0 \subseteq X$, we have $\mathcal{L}(\mathcal{F}, X_0, \mu) \neq \emptyset$.*

Assumption 1 is obviously essential for our purpose. Otherwise, it is not possible to keep the system in the workspace. Assumption 1 can be relaxed for applications in which the objective can be accomplished in finite time and the state is allowed to exit the workspace afterwards. In this paper, our emphasis is on infinite-time properties and finite-time specifications are treated as a special case.

We assume no knowledge of $\mathcal{F}$, which we refer to as the *concrete* control system, except the following assumptions.

Assumption 2. *(Data Points) We are given a set of N data points* $\mathcal{D} := \left\{(x_i, u_i, x_i^+) \in \mathcal{F}\right\}_{i=1,\cdots,N}$.

Assumption 2 is not restrictive as long as perfect state knowledge is available. We treat $\mathcal{F}$ as an input-output black-box. Assumption 2 may also prove useful when some analytical form of $\mathcal{F}$ is available, but is too complex to use for control synthesis purposes. In this case, we may sample data points from $\mathcal{F}$ rather than using its analytical form. In this paper, we are given data points as a priori. An immediate extension to our framework is gathering data points while controlling the system - discussed in Sec. 7.

Assumption 3. *(Continuity bounds) We are given non-negative constants $\kappa_0, \kappa_x, \kappa_u$, referred to as continuity constants, such that for all $(x_1, u_1, x_1^+), (x_2, u_2, x_2^+) \in \mathcal{F}$, the following relation holds:*

$$\left\|x_2^+ - x_1^+\right\|_p \leq \kappa_0 + \kappa_x \left\|x_2 - x_1\right\|_p + \kappa_u \left\|u_2 - u_1\right\|_p, \tag{6}$$

where $p \geq 1$ is a choice of norm.

Constant κ_0 characterizes the degree of non-determinism in $\mathcal{F}$ and κ_x, κ_u characterize how continuous (in a Lipschitz sense) $\mathcal{F}$ is on X and U. If we know $\mathcal{F}$ is deterministic, we let $\kappa_0 = 0$. The assumption that the evolution of a physical system is continuous in state and controls is reasonable. Even many hybrid systems demonstrate continuity in the Lipschitz sense. Therefore, there always exists constants $\kappa_0, \kappa_x, \kappa_u$ such that (6) holds. The stronger assumption that we make in Assumption 3 is that we *know* the values of continuity constants. Note that $\kappa_0, \kappa_x, \kappa_u$ do not need to be the *best constants*. In other words, the inequality (6) does not need to be tight. Any upper-bound for the best values of $\kappa_0, \kappa_x, \kappa_u$ is sufficient for the soundness of the results in this paper, but very large values obviously lead to conservativeness .

Any guarantee is provided against the values of continuity constants. Estimating the continuity constants using $\mathcal{D}$ is not a sound approach since it is always possible to observe new data points that falsify the validity of the estimated constants. However, in practice, there might not be any other option than using $\mathcal{D}$ for estimating the continuity constants. One way to approach this issue is multiplying the tightest estimates for continuity constants by some safety factor, depending on the application. There exists several methods for estimating Lipschitz constants from data [11, 19].

3.2 Problem Formulation and Approach

We are given a STL formula φ in the form (1). The predicates of φ are considered to be linear over state:

$$\mathcal{P}_i := (\pi_i^T x \leq \zeta_i), \tag{7}$$

where $\pi_i \in \mathbb{R}^n$, $i = 1, \cdots, n_\mathcal{P}$, - $n_\mathcal{P}$ is the number of predicates, and $\zeta_i \in \mathbb{R}$. We also define

$$\Pi := \left(\pi_1^T \cdots \pi_{n_\mathcal{P}}^T\right)^T. \tag{8}$$

Matrix $\Pi \in \mathbb{R}^{n_\mathcal{P} \times n}$ characterizes the sensitivity of the predicates to changes in the state. We do not formulate predicates over controls but the state may be augmented to include controls (see, e.g., [26]).

Problem 1. *Given $\mathcal{D}$ from a control system $\mathcal{F}$ as in* (4)*, constants $\kappa_0, \kappa_x, \kappa_u$ corresponding to Assumption 3, an STL formula φ over predicates as in* (7)*, find the optimal control policy μ^* and a set of*

initial conditions $X_0^* \subseteq X$ such that:

$$(X_0^*, \mu^*) = \begin{array}{ll} \underset{X_0, \mu}{argmax} & \epsilon \\ \textit{subject to} & \mathcal{L}(\mathcal{F}, X_0, \mu) \subseteq \mathcal{L}(\varphi, \epsilon), \\ & \mathcal{L}(\mathcal{F}, X_0, \mu) \neq \emptyset \end{array} \tag{9}$$

Our framework is able to accommodate slight variations of Problem 1. For instance, X_0 may be fixed by a user-specified set or point. Alternatively, given a certain ϵ, the (largest) corresponding set of admissible initial conditions may be the asked to be computed. We may also consider some weighted cost functions added to ϵ, such as penalizing the controls or distance from a reference trajectory.

Our solution to Problem 1 has two main steps. First, we construct a model from $\mathcal{D}$ and continuity constants. Our model has to serve two purposes: i) it has to contain all the behaviors of $\mathcal{F}$, as formalized shortly, and ii) has to be simple enough for control synthesis. We choose PWA models with user-specified number of modes - formally defined in Sec. 4.2. PWA models are able to capture arbitrarily high nonlinearities by increasing the number of modes in exchange for higher computational complexity - both in identifying the model and also synthesis based on the model. We focus on a a particular class of control strategies that are computationally tractable to compute. Therefore, completeness may be lost and we may obtain suboptimal solutions for Problem 1. However, we do not trade off correctness. Once our method returns some ϵ^* for Problem 1, we have the guarantee that all closed-loop trajectories of $\mathcal{F}$ are in ϵ^*-language of φ.

4 FORMAL MODEL IDENTIFICATION

In this section, we introduce a method to identify a set-valued PWA model using the data and continuity bounds. First, we provide the necessary background in Sec. 4.1 to formalize the model identification subproblem in Sec. 4.2. The solution has two stages, which are discussed in Sec. 4.3 and Sec. 4.4.

4.1 Simulation Relation

Definition 4.1. Given two systems $\mathcal{F} \subset X \times U \times \mathbb{R}^n$ and $\mathcal{G} \subset X \times U \times \mathbb{R}^n$, we say $\mathcal{G}$ *simulates* $\mathcal{F}$ if and only if $\mathcal{F} \subseteq \mathcal{G}$.

Definition 4.1 is reminiscent of the simulation relation in concurrent systems [7, 30]. Here we do not define simulation relation with respect to a particular equivalence class. Every state is equivalent only to itself - no abstraction is used. Simulation is a partial order relation since the following properties hold: $\mathcal{F}$ simulates $\mathcal{F}$; If $\mathcal{G}$ simulates $\mathcal{F}$ and $\mathcal{H}$ simulates $\mathcal{G}$, then $\mathcal{H}$ simulates $\mathcal{F}$; If $\mathcal{G}$ simulates $\mathcal{F}$ and vice-versa, then $\mathcal{F} = \mathcal{G}$. The following result holds from language inclusion properties of simulation relation [7].

Lemma 4.2. *If $\mathcal{G}$ simulates $\mathcal{F}$, then for all μ and X_0 we have $\mathcal{L}(\mathcal{F}, X_0, \mu) \subseteq \mathcal{L}(\mathcal{G}, X_0, \mu)$.*

Definition 4.3. Given data points $\mathcal{D} = \{(x_i, u_i, x_i^+)\}_{i=1,\cdots,N}$ from an unknown control system $\mathcal{F} \subset X \times U \times \mathbb{R}^n$, constants $\kappa_0, \kappa_x, \kappa_u$ corresponding to Assumption 3, the *tightest simulating* control system $\overline{\mathcal{F}} \subset X \times U \times \mathbb{R}^n$ is defined such that for all $(x, u, x^+) \in \overline{\mathcal{F}}$, the following holds:

$$x^+ \in \bigcap_{i=1}^{N} \left\{ \{x_i^+\} \oplus k_x \|x - x_i\|_p \mathcal{B}_p \oplus k_u \|u - u_i\|_p \mathcal{B}_p \oplus k_0 \mathcal{B}_p \right\}.$$

Lemma 4.4. *The following properties hold: 1) $\mathcal{F} \subseteq \overline{\mathcal{F}}$; 2) For any $\mathcal{G}$ that simulates $\mathcal{F}$, we have $\overline{\mathcal{F}} \subseteq \mathcal{G}$.*

Proof. 1) The proof follows from two facts that establish $\mathcal{F} \subseteq \overline{\mathcal{F}}$. First, we require that $(x_i, u_i, x_i^+) \in \overline{\mathcal{F}}, i = 1, \cdots, N$. Second, any point that is included in $\overline{\mathcal{F}}$ is sufficiently close to other data points in the sense of (6). 2) We prove by contradiction. If there exists $\mathcal{G}$ simulating $\mathcal{F}$ such that $\overline{\mathcal{F}} \not\subset \mathcal{G}$, then there exists some $(x_s, u_s, x_s^+) \in \overline{\mathcal{F}}$ that is allowed to be in $\mathcal{F}$ by Assumption 3. Thus, $\mathcal{F} \subseteq \mathcal{G}$ does not necessarily hold.

We may use $\overline{\mathcal{F}}$ for control synthesis, but its representation is data-size dependent and is often too complex. We need simpler forms of systems that simulates $\overline{\mathcal{F}}$ (and hence $\mathcal{F}$).

4.2 Piecewise Affine Systems

Definition 4.5. A control system $\mathcal{G} \subset X \times U \times \mathbb{R}^n$ is PWA if $\forall x \in X, \forall u \in U$, we have $(x, u, x^+) \in \mathcal{G}$ if and only if

$$x^+ \in \begin{cases} \{A_1 x + B_1 u + c_1\} \oplus W_1, & x \in X_1, \\ \vdots & \vdots \\ \{A_{\mathcal{M}} x + B_{\mathcal{M}} u + c_{\mathcal{M}}\} \oplus W_{\mathcal{M}}, & x \in X_{\mathcal{M}}, \end{cases} \tag{10}$$

where $X_i, i = 1, \cdots, \mathcal{M}$, are polyhedral sets with disjoint interiors, $\bigcup_{i=1}^{\mathcal{M}} X_i = X$, $\mathcal{M}$ is the number of modes, and $W_i \subset \mathbb{R}^n, i = 1, \cdots, \mathcal{M}$, are polytopic sets of additive disturbances. Each mode is an affine system with constants $A_i \in \mathbb{R}^{n \times n}, B_i \in \mathbb{R}^{n \times m}$ and $c_i \in \mathbb{R}^n$. In the rest of this section, we propose a method to solve the following subproblem:

Subproblem 1. *Given data points $\mathcal{D}$ from control system $\mathcal{F} \subset X \times U \times \mathbb{R}^n$, constants $\kappa_0, \kappa_x, \kappa_u$ corresponding to Assumption 3, find a PWA control system $\mathcal{G}$ in the form of* (10)*, where an upper-bound for $\mathcal{M}$ is given, such that $\mathcal{F} \subseteq \mathcal{G}$ and $\alpha(W_1, \cdots, W_{\mathcal{M}})$ is minimized, where $\alpha : (2^{\mathbb{R}^n})^{\mathcal{M}} \to \mathbb{R}$ is a cost function that promotes smaller disturbance sets.*

The reason that we add a cost criteria to Subproblem 1 is that a PWA $\mathcal{G}$ that simulates $\mathcal{F}$ is not unique. In fact, by making the disturbance sets sufficiently large, $\mathcal{G}$ can simulate any system. Having large disturbance sets is undesirable for control synthesis. For computational purposes, we focus on simple forms of disturbance sets and α. For example, we let $W_i, i = 1, \cdots, \mathcal{M}$, to be axis-aligned hyper-rectangles and α is a norm of side lengths.

4.3 Piecewise Affine Fitting

Here we simultaneously find values representing the sets X_i, and matrices A_i, B_i, c_i in (10) by solving an optimization problem. We find sets W_i afterwards. Consider $K \in \mathbb{N}_+$ hyperplanes - which we refer to as *guards*:

$$h_i^T x + 1 = 0, \tag{11}$$

where $h_i \in \mathbb{R}^n, i = i = 1, \cdots, K$. The guards partition X into at most 2^K polyhedral sets with disjoint interiors:

$$X_k = \left\{ x \in X \middle| h_i^T x + 1 \overset{(k,i)}{\sim} 0, i = 1, \cdots, K \right\}, k = 1, \cdots, 2^K, \tag{12}$$

where $\sim: \mathbb{N}_+ \times \mathbb{N}_+ \to \{\leq, \geq\}$ is defined in the following way: $\overset{(k,i)}{\sim}$ is $\geq$ if the i^{th} digit from the right of k written in binary numeral

system is one, and $\leq$ otherwise. For example, we have $5 = (101)_2$. Hence $\overset{(5,1)}{\sim}=\geq$, $\overset{(5,2)}{\sim}=\leq$ and $\overset{(5,3)}{\sim}=\geq$. We interpret further digits on the left as zero: $\overset{(5,i)}{\sim}=\leq$, $i > 3$. The set of decision parameters are

$$\Theta := \left\{\{A_k, B_k, c_k,\}_{i=1,\cdots,2^K}, \{h_i,\}_{i=1,\cdots,K}\right\}$$

The best values for Θ are found using the following optimization problem:

$$\begin{aligned}\Theta^* = \underset{\Theta}{\arg\min} \quad & \delta \\ \text{subject to} \quad & \left|x_j^+ - (A_i x_j + B_i u_j + c_i)\right| z_j^k \leq \delta 1_n, \\ & z_j^k = 1 \Leftrightarrow x_j \in X_k, z_j^k = 0 \Leftrightarrow x_j \notin X_i, \\ & X_i, i = 1, \cdots, 2^K, \text{ given by (12), (11),} \\ & j = 1, \cdots, N, i = 1, \cdots, K,\end{aligned} \tag{13}$$

where δ is error. Eq. (13) finds the best PWA fit (additive disturbances are not yet considered) such that all data points in $\mathcal{D}$ are within $\delta\mathcal{B}_\infty$ of their respective PWA predications.

Since all the values and sets are bounded, we show that (13) can be cast as a MILP problem using the *big-M method*. First, the expression $\left|x_j^+ - (A_i x_j + B_i u_j + c_i)\right| z_j^k \leq \delta 1_n$ is equivalent to the following set of constraints:

$$\begin{cases} -M(1 - z_j^k) \leq x_j^+ - A_i x_j - B_i u_j - c_i + \delta_j^k \leq M(1 - z_j^k), \\ -\delta 1_n \leq \delta_j^k \leq \delta 1_n, \end{cases}$$

where $\delta_j^k \in \mathbb{R}^n$ are auxilary variables and M is a large positive number. Second, we need to capture (12). We define K binaries for each data point - denoted by $b_j^i, i = 1, \cdots, K$, introducing a total of NK binary variables. We have

$$b_j^i = \mathcal{I}(h_i^T x_j + 1 \geq 0) \Leftrightarrow \begin{cases} h_i^T x_j + 1 \leq M b_j^i, \\ h_i^T x_j + 1 \geq -M(1 - b_j^i). \end{cases}$$

The relation $z_j^k = \mathcal{I}(x_j^k \in X_k)$ can be converted to:

$$z_j^k = \bigwedge_{i=1}^{K} \text{Sgn}(k, i, b_j^i),$$

where $\text{Sgn}(k, i, b_j^i) = b_j^i$ if $\overset{k,i}{\sim}=\geq$, and is $\neg b_j^i$ otherwise. The conjunction and negation applying to integers is interpreted in a Boolean sense - e.g., $1 \wedge 0 = 0, 1 \wedge 1 = 0, \neg 1 = 0$, etc. The variables $z_j^k \in [0, 1], j = 1, \cdots, N, k = 1, \cdots, 2^K$, are declared as continuous variables, but always take binary variables. Encoding Boolean operations as mixed-integer constraints is a standard procedure (see, e.g., [9]) and further details are not presented here.

4.4 Adjusting Disturbances

Now we find disturbance in (10) such that $\mathcal{G}$ simulates $\mathcal{F}$. We let every disturbance set to be a hyper-rectangle that is symmetric around the origin. The length of the side in q^{th} cartesian direction of W_i, $q = 1, \cdots, n$, is found using the following optimization problem:

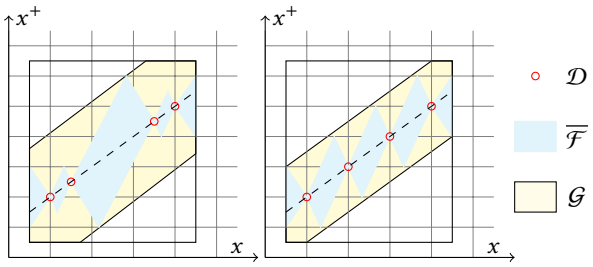

Figure 1: Example 4.7: The degree of non-determinism may depend on the distribution of the data.

$$\begin{aligned}\eta_{i[q]}^* = \underset{x,u,\eta}{\text{argmax}} \quad & \left|e_{[q]}^T \eta_i\right| \\ \text{subject to} \quad & x^+ = x_j^+ + k_0\beta_j^0 + k_x\beta_j^x + k_u\beta_j^u \\ & \beta_j^x \in \|x - x_j\|_p \mathcal{B}_p, \\ & \beta_j^u \in \|u - u_j\|_p \mathcal{B}_p, \|\beta_j^0\|_\infty \in \mathcal{B}_\infty \\ & x^+ = A_i x + B_i u + c_i + \eta_i \\ & x \in X_i, u \in U, j = 1, \cdots, N.\end{aligned} \tag{14}$$

We let:

$$W_i = \left\{w \in \mathbb{R}^n \mid -\eta_i^* \leq w \leq \eta_i^*\right\} \tag{15}$$

Theorem 4.6. *The PWA system constructed from solutions of* (13), *and* (14), (15), $i = 1, \cdots, 2^K$, *simulates* $\mathcal{F}$.

Proof. Eq. (14) gives the farthest point (in the q^{th} direction) in $\overline{\mathcal{F}}$ from the model given by (13). Since the worst-case distances are considered in each direction, the sets in (15) added to the solution in (13) establish $\overline{\mathcal{F}} \subseteq \mathcal{G}$. □

The following simple example shows that the distribution of data affects model identification, even if their PWA fits - models without disturbances - are identical.

Example 4.7. Consider fitting a line (a single mode PWA model) to a one-dimensional data set of $N = 4$ points, shown by red circles in Fig. 1. There are no control inputs. The box represents X^2. The relation $\overline{\mathcal{F}}$ and the learned affine $\mathcal{G}$ are shown by cyan and yellow regions, respectively. In the figure to the left, points are closer to the edges of X, creating a vacuum of data in the center of X, which leads to large non-determinism in $\mathcal{G}$. On the other hand, data points on the right are more evenly-spaced, leading to less non-deterministic $\mathcal{G}$.

Remark 3. *Overfitting arises when few data is used to decide about a large number of variables. Since our model identification is set-valued and takes into account all "other" possible data, overfitting is never an issue. Even with one single data point, Definition* (4.3) *defines a set-valued model* $\overline{\mathcal{F}}$ *with very large non-determinism - virtually useless for control synthesis.*

4.5 Computational Aspects

Eq. (13) is a combinatorial optimization problem with NK binary variables and $O\left(K \max(n^2 + nm + Nn)\right)$ continuous variables. The

worst-case complexity of MILPs scales exponentially with the number of its binary variables and polynomially with the number of its continuous variables and constraints. MILP solvers find the global optimum. We may terminate a large MILP early to obtain a suboptimal solution - after a feasible integer solution is found. Eq. (14) is a non-convex optimization problem that can also be cast as a MILP when $p = 1, \infty$. The number of its binary variables scales by $O(N(n+m))$ and the number of continuous variables and constraints scale similarly to (13). Validity of the continuity constants is implicit in the feasibility of (14) - if k_0, k_x, k_u are under-estimated then (14) may become infeasible.

In practice, exact solutions for (14) may be unachievable. Heuristics may be used to over-approximate the sets in (15). The value of δ found in (13) provides a lower bound for the sides of disturbance sets. As (14) is sensitive to the number of data points, increasing the density of data decreases uncertainties in a linear fashion - as illustrated in Fig. 1. Thus, a good heuristic may be to solve (14) for small data sets and accordingly adjust the values for larger data sets.

5 FORMAL CONTROL SYNTHESIS

Once we have a PWA model with user-specified complexity, we can use it for control synthesis. In this section, we propose a method - called tube STL - for solving the following subproblem.

Subproblem 2. *Given a PWA system in the form* (10) *and STL formula φ with predicates in the form in* (7)*, find an optimal control policy μ^* and a set of initial conditions $X_0^* \subseteq X$ subject to* (9).

We first explain the key results of tube STL in Sec. 5.1. The technical details on designing the tube and nominal trajectories are discussed in Sec. 5.2 and Sec. 5.3, respectively.

5.1 Tube STL Control

Definition 5.1. Given a PWA system as in Definition 4.5, the *nominal* PWA system is defined as function $g : X \times U \to \mathbb{R}^n$, where:

$$g(x,u) = A_i x + B_i u + c_i, x \in X_i. \tag{16}$$

Note that $\mathcal{G}^{\text{nom}} := \{(x,u,g(x,u)) | x \in X, u \in U\}$ is a deterministic system. Thus, controlling it from STL specifications can be accomplished by planning the controls in an open-loop fashion - which is also a complete method for finite time specifications [16]. Any PWA system can be transformed into a mixed-logical dynamical (MLD) system [9]. Temporal logic control of deterministic MLDs - with linear predicates and PWA cost functions - maps to solving MILPs [16, 24, 25]. However, when disturbances are present, the problem becomes very hard as predicting the switching behavior is intractable. The method in [25] proposed reactive STL receding horizon planning, but this approach does not provide any guarantee on STL satisfaction and maintaining MILP feasibility. Conservative solutions for stabilization of PWA are proposed using fixed feedback gains (e.g., see [15]). However, STL requirements are often more complicated than stabilization and desirable trajectories may need to traverse the polyhedral regions $X_i, i = 1, \cdots, \mathcal{M}$, in a complex manner.

Definition 5.2. Given a PWA system as in Definition 4.5, the *switching-disturbance system* is defined as

$$\mathcal{G}_{\text{swd}} := \bigcup_{\sigma=1}^{\mathcal{M}} \Big\{ \quad (x,u,x^+) \Big| x \in \mathbb{R}^n, u \in \mathbb{R}^m, \\ x^+ \in \{A_\sigma x + B_\sigma u\} \oplus W_\sigma \Big\}. \tag{17}$$

The switching-disturbance system is an aggregation of linear systems with polytopic disturbances. Its switching is modeled as fully non-deterministic - this way of modeling is conservative but useful as explained shortly. We propose control policies of the following form:

$$\mu(x[0]x[1]\cdots x[t]) = \mu^{\text{nom}}(x[0],t) + \mu^{\text{fb}}(x[t]), \tag{18}$$

where $\mu^{nom} : X \times \mathbb{N} \to U$ is an open-loop control policy and $\mu^{fb} : \mathbb{R}^n \to U$ is a state-feedback control policy. Informally, μ^{nom} is a precomputed plan of controls for the nominal system, while μ^{fb} takes responsibility of attenuating disturbances - the deviations from the nominal trajectory. We design μ^{fb} for all possible switchings of the system. This approach is conservative, but decouples the design of μ^{fb} and μ^{nom}. Note that $\mu^{fb}(x[t])$ has access to the knowledge of mode at time t. Thus, the mode of $\mathcal{G}_{\text{swd}}$ is observable but its future is not predictable. We define $\Sigma : X \to \{1, \cdots, \mathcal{M}\}$ such that $\sigma = \Sigma(x)$ if $x \in X_\sigma$, $\sigma = \{1, \cdots, \mathcal{M}\}$.

Definition 5.3. A *tube* corresponding to control policy $\mu^{\text{tube}} : \mathbb{R}^n \times \{1, \cdots, \mathcal{M}\} \to \mathbb{R}^m$ is a set $T \subset \mathbb{R}^n$ such that $0 \in T$ and the following relation holds:

$$\mathcal{L}(\mathcal{G}_{\text{swd}}, T, \mu^{\text{tube}}) \subseteq T^\omega \tag{19}$$

Moreover, the set $T_u \subset \mathbb{R}^m$ is defined as:

$$T_u := \left\{ \mu^{\text{tube}}(x,\sigma) \Big| x \in T, \sigma \in \{1, \cdots, \mathcal{M}\} \right\}. \tag{20}$$

A tube is a robust forward invariant set [10] for $\mathcal{G}_{\text{swd}}$. The *invariance-inducing* control policy μ^{tube} can be rewritten as $\mu^{\text{tube}} : T \times \{1, \cdots, \mathcal{M}\} \to T_u$. The following theorem is the key result of this section.

Theorem 5.4. *Consider a PWA system $\mathcal{G}$, its nominal and switching-disturbance versions $\mathcal{G}_{\text{nom}}$ and $\mathcal{G}_{\text{swd}}$, initial condition $x_0 \in X$, and an STL formula φ with predicates in the form of* (7)*. Let T and $\mu^{\text{tube}} : T \times \{1, \cdots, \mathcal{M}\} \to T_u$ be a tube and its invariance controller, respectively. Assume $x_0 \in X \ominus T$. Let*

$$\mu^{\text{nom}} : \bigcup_{i=1}^{\mathcal{M}} (X_i \ominus T) \times \mathbb{N} \to (U \ominus T_U)$$

be an open-loop control policy such that

$$\mathcal{L}(\mathcal{G}_{\text{nom}}, x_0, \mu^{\text{nom}}) \subseteq \mathcal{L}(\varphi, \epsilon),$$

where $\mathcal{L}(\mathcal{G}_{\text{nom}}, x_0, \mu^{\text{nom}}) = \{x_{\text{nom}}[0]x_{\text{nom}}[1]\cdots, x_{\text{nom}}[0] = x_0\}$ - it consists of only one signal. Then, by replacing the following policy in (18)

$$\mu^{\text{fb}}(x[t]) = \mu^{\text{tube}}\Big(x[t] - x_{\text{nom}}[t], \Sigma(x[t])\Big),$$

the following guarantee holds:

$$\mathcal{L}(\mathcal{G}, \{x_0\} \oplus T, \mu) \subseteq \mathcal{L}(\varphi, \epsilon - \max_{x \in T} \|\Pi x\|_\infty), \tag{21}$$

where Π is the predicates matrix defined in (8).

Proof. The proof is constructive. The modes are switched according to the actual trajectory, but the tube invariance holds for arbitrary switching. First, we show that the closed-loop trajectories remain within T-vicinity of the nominal trajectory. The proof is by induction. Base: $x[0] \in \{x_{\text{nom}}[0]\} \oplus T$ since $x[0] = x_{\text{nom}}[0]$ and T contains the origin. Inductive step: $x[t] \in \{x_{\text{nom}}[t]\} \oplus T$ implies $x[t+1] \in \{x_{\text{nom}}[t+1]\} \oplus T$, which is immediately verified by the fact that T is robust forward invariant. Also note that each nominal point is at the center of a tube that is fully contained within a region corresponding to a certain mode. It follows (2) that any trajectory within T-vicinity of the nominal trajectory can decrease the STL score by at most $\max_{x\in T}\|\Pi x\|_\infty$, which is the largest possible change in the value of a predicate. The rest of the proof holds by (2). □

Our solution to Problem 1 is established from Theorem 5.4 and Lemma 4.2, which is stated as follows:

Corollary 5.5. *Let $\mathcal{G}$ be a PWA system that simulates $\mathcal{F}$. If a nominal policy μ^{nom}, initial condition x_0, and a tube T and its control policy μ^{tube} exists such that* (21) *holds, then we have*

$$\mathcal{L}(\mathcal{F}, \{x_0\} \oplus T, \mu) \subseteq \mathcal{L}(\varphi, \epsilon - \max_{x\in T}\|\Pi x\|_\infty). \tag{22}$$

5.2 Tube Design

A tube can be viewed as a robust control invariant (RCI) set [10]. We desire the tube with the smallest $\max_{x\in T}\|\Pi x\|_\infty$. It is also desirable to keep T_u small so the nominal trajectory can take benefit of a larger set of admissible controls. Given a bounded set $\mathcal{S} \subset \mathbb{R}^n$, finding the maximal RCI set - a fixed-point - inside $\mathcal{S}$ is a well-known undecidable problem [10]. Moreover, performing the fixed-point algorithm for PWA systems with additive disturbances is computationally challenging [23]. We desire finding the smallest RCI set by solving an optimization problem. The authors in [22] formulated the problem of computing RCI sets for linear systems with polytopic disturbances and constraints as convex optimization problems. The cost function can be designed to promote small RCI sets. However, the method in [22] does not apply to switching systems. Here we propose a new method to compute optimized RCI sets for switching systems with additive disturbances.

5.2.1 Set-Parameterization. We parameterize the RCI sets by a user-specified number of hyper-rectangles. For each mode, there should exist a control input for each vertex of each hyper-rectangle such that for all allowable disturbances the vertex finds a successor in at least one of the hyper-rectangles. Thus, we enforce invariance by design. We show that the convex-hull of the hyper-rectangles is a RCI set. We define Γ axis-aligned hyper-rectangles as

$$\mathcal{R}(p^\gamma, a^\gamma) := \{x \in \mathbb{R}^n | a^\gamma \le x \le p^\gamma + a^\gamma\},$$

where $p^\gamma \in \mathbb{R}^n, a^\gamma \in \mathbb{R}^n_+$. $\gamma = 1, \cdots, \Gamma$, represent the lower-left corners and sides, respectively. The vertices of hyper-rectangles are denoted by $q^\gamma_k, k = 1, \cdots, 2^n, \gamma = 1, \cdots, \Gamma$. Note that

$$\mathcal{R}(p^\gamma, a^\gamma) = \text{Convh}\left(\left\{q^\gamma_k\right\}_{k=1,\cdots,2^n}\right).$$

Define $\theta := \{p^\gamma, a^\gamma\}_{\gamma=1,\cdots,\Gamma}$ as the set of $2n\Gamma$ parameters. The vertex representation of disturbance set W_σ is $\text{Convh}(w_{\sigma,1}, \cdots, w_{\sigma,d_\sigma})$, $\sigma \in \{1, \cdots, \mathcal{M}\}$, where $d_\sigma \in \mathbb{N}_+$ is the number of vertices of W_σ. We define

$$\mathcal{T}(\theta) := \text{Convh}\left(\left\{q^\gamma_k\right\}_{\gamma=1,\cdots,\Gamma, k=1,\cdots,2^n}\right). \tag{23}$$

Theorem 5.6. *If there exists $u^\gamma_{k,\sigma} \in \mathbb{R}^m, \gamma = 1, \cdots, \Gamma, k = 1, \cdots, 2^n, \sigma \in \{1, \cdots, \mathcal{M}\}$, such that*

$$y^\gamma_{k,\sigma,j} \in \bigcup_{\gamma=1}^{\Gamma} \mathcal{R}(p^\gamma, a^\gamma), \tag{24}$$

where $y^\gamma_{k,\sigma,j} := A_\sigma p^\gamma_k + B_\sigma u^\gamma_{k,\sigma} + w_{\sigma,j}$, then $\mathcal{T}(\theta)$ is a tube with

$$T_u(\theta) = \text{Convh}\{u^\gamma_{k,\sigma}\}_{k=1,\cdots,2^n, \gamma=1,\cdots,\Gamma, \sigma\in\{1,\cdots,\mathcal{M}\}}.$$

Proof. Consider any point $x \in \mathcal{T}(\theta)$ and $\sigma \in \{1, \cdots, \mathcal{M}$. There exists $\lambda^\gamma_k \ge 0, \sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k = 1$, such that $x = \sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k q^\gamma_k$. Let

$$v_\sigma := \sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k u^\gamma_{k,\sigma}.$$

Note that $v_\sigma \in T_u$ since it is a convex combination of $2^n\Gamma$ points in T_u. We have the following deductions:

$$\begin{aligned}
y = \;& A_\sigma x + B_\sigma v_\sigma + w_{\sigma,j} \\
= \;& A_\sigma \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k q^\gamma_k + B_\sigma \sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k u^\gamma_{k,\sigma} + w_{\sigma,j} \\
= \;& \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k(A_\sigma q^\gamma_k + B_\sigma u^\gamma_{k,\sigma} + w_{\sigma,j}) = \\
= \;& \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k y^\gamma_{k,\sigma,j} \\
\Rightarrow \;& (A_\sigma x + B_\sigma v_\sigma) \oplus \{w_{\sigma,j}\}_{j=1,\cdots,d_\sigma} \subseteq \mathcal{T}(\theta).
\end{aligned}$$

By taking the convex hulls of both sides, we verify $\{A_\sigma x + B_\sigma v_\sigma\} \oplus W_\sigma \subseteq \mathcal{T}(\theta)$, and the proof is complete.

5.2.2 Optimization. The conditions in Theorem 5.6 are formulated as a set of constraints. Eq. (24) is equivalent to the following Boolean logic formula being true for $\gamma = 1, \cdots, \Gamma, k = 1, \cdots, 2^n, \sigma \in \{1, \cdots, \mathcal{M}\}, j = 1, \cdots, d_\sigma$:

$$\bigvee_{\beta=1}^{\Gamma}\left((p^\beta \le y^\gamma_{k,\sigma,j}) \wedge (y^\gamma_{k,\sigma,j} \le p^\beta + a^\beta)\right). \tag{25}$$

We encode (25) using binary decision variables and big-M method - technically similar to Sec. 4.3. Using basic convexity notions, we formulate the following MILP:

$$\begin{aligned}
\theta^* = \arg\min_{\theta} \quad & \xi \\
s.t. \quad & \|\Pi q^\gamma_k\|_\infty \le \xi, k = 1, \cdots, 2^n, \\
& (23), (24), (25), \gamma = 1, \cdots, \Gamma.
\end{aligned} \tag{26}$$

The solution of (26) yields the smallest STL tube. A corresponding invariance-inducing control policy μ^{tube} can be designed from the proof of Theorem 5.6:

$$\begin{aligned}
\mu^{\text{tube}}(x,\sigma) = \quad & \arg\min_{u} \quad J(u) \\
& \text{subject to} \quad x = \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k q^\gamma_k \\
& \qquad\qquad u = \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k u^\gamma_{k,\sigma}, \\
& \qquad\qquad \lambda^\gamma_k \ge 0, \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{2^n}\lambda^\gamma_k = 1.
\end{aligned} \tag{27}$$

where $J(u)$ is a user-defined convex linear/quadratic cost function. Eq. (27) is a linear/quadratic program with $2^n\Gamma$ decision variables.

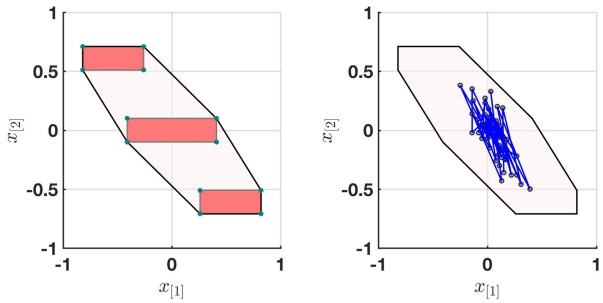

Figure 2: Example 5.7: tube constructed from convex hull of 3 hyper-rectangles, and a sample trajectory.

Thus, μ^{tube} takes a PWA form. A proper choice for $J(u)$ is $\|A_\sigma x + B_\sigma u\|_\infty$, which penalizes the distance from the center of the tube.

Example 5.7. Consider a double integrator disturbance-switching system in $\mathbb{R}^2$ with two modes, where

$$A_1 = \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{3}{2} \end{pmatrix}, A_2 = \begin{pmatrix} \frac{3}{2} & 1 \\ -\frac{1}{2} & 1 \end{pmatrix}, B_1 = B_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The disturbance set are $W_1 = W_2 = \frac{1}{10}\mathcal{B}_\infty$, and $U = [-1, 1]$. Note that both matrices are unstable. We found a tube for $\Gamma = 3$ in 95 seconds using Gurobi MILP solver on a 3.0 GHz dual core MacBook Pro. The hyper-rectangles, the RCI set and a sample trajectory inside the tube are shown in Fig. 2. The switches and disturbances were generated randomly with uniform distributions.

5.3 Nominal Trajectory Design

As mentioned earlier, STL satisfaction of a deterministic PWA system can be mapped into a MILP problem. The details are not included here as they are well documented in [16, 24]. The objective is maximizing the STL robustness score function. In order to deal with unbounded "always" operator in (1), lasso trajectories are used (similar to [24]):

$$x_{\text{nom}}[0]x_{\text{nom}}[1]\cdots = \zeta_0(\zeta_p)^\omega, \tag{28}$$

where $\zeta_0 = x_{\text{nom}}[0]x_{\text{nom}}[1]\cdots x_{\text{nom}}[\tau_0]$ is the prefix, and $\zeta_p = x_{\text{nom}}[\tau_0+1]x_{\text{nom}}[\tau_0+2]\cdots x_{\text{nom}}[\tau_0+\tau_p]$ is the periodic suffix. The values for τ_0 and τ_p are designated by the user. Typically, larger values result in more flexibility and better performance, in exchange of larger computation cost. The control sequence corresponding to the open-loop control policy is computed from the following optimization problem:

$$\begin{aligned} \{\mu^{ol}, \epsilon^*\} = \quad & \underset{u[\tau]}{\text{argmax}} \quad \epsilon \\ & \text{subject to} \quad x_{\text{nom}}[\tau+1] = g(x_{\text{nom}}[\tau], u_{\text{nom}}[\tau]), \\ & \exists i \in \{1, \cdots, \mathcal{M}\} \text{ s.t. } x_{\text{nom}}[\tau] \in X_i \ominus T, \\ & u_{\text{nom}}[\tau] \in U \ominus T_u, \\ & x_{\text{nom}}[0] = x_0, \tau = 0, \cdots, \tau_{in} + \tau_p \\ & \epsilon = \rho(x_{\text{nom}}[0]x_{\text{nom}}[1]\cdots, \varphi, 0). \end{aligned} \tag{29}$$

An immediate extension to our STL control framework is to adopt an online MPC algorithm that replans the nominal trajectory at each time to achieve a better performance. We leave this extension for our future work.

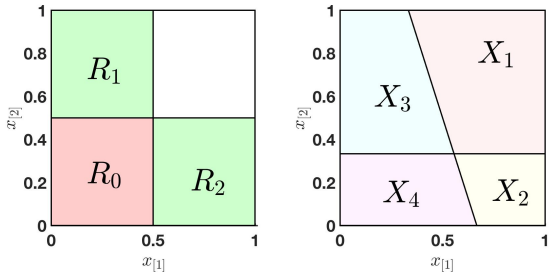

Figure 3: Case Study: (Left): Regions of interest. (Right): Computed polyhedral partition from (13).

5.4 Computational Aspects

The tube is given by the MILP in (26), which has $2^n\Gamma^2\sum_{\sigma=1}^{\mathcal{M}} d_\sigma$ binary variables - resulting in a computational bottleneck. The design of tube is offline. Similarly, computing the nominal trajectory is also a MILP solved in an offline fashion. The number of binary variables scales linearly with the trajectory length, the number of modes, and the number of predicates. Encoding robustness into the optimization cost function can be accomplished without introducing additional binary variables [28]. The only online part of our control policy is (27), which is a convex program and it is solved efficiently.

6 CASE STUDY

We adopt a controlled version of the genetic toggle switch model in [14]. The control system $\mathcal{F}$ is constructed from:

$$\begin{aligned} x^+_{[1]} &\in \left\{\tfrac{4}{5}x_{[1]} + \tfrac{1}{2(1+x^3_{[2]})} + e^{-\frac{1}{5}x_{[1]}[t]}u_{[1]}\right\} \oplus \tfrac{1}{20}[-1,1], \\ x^+_{[2]} &\in \left\{\tfrac{4}{5}x_{[2]} + \tfrac{1}{2(1+x^2_{[1]})} + e^{-\frac{1}{5}x_{[2]}[t]}u_{[2]}\right\} \oplus \tfrac{1}{20}[-1,1], \end{aligned} \tag{30}$$

where $X = [0,1]^2$ and $U = [-1,1]^2$. The state components represent gene repressor concentrations. The goal is to oscillate the concentration levels. The STL specification is:

$$\varphi = \mathbf{G}_{[0,\infty]}\neg R_0 \wedge \mathbf{F}_{[0,10]}R_1 \wedge \mathbf{G}_{[10,\infty]}(\mathbf{F}_{[0,10]}R_1 \wedge \mathbf{F}_{[0,10]}R_2), \tag{31}$$

where R_0, R_1, R_2 are conjunctions of predicates that characterize the regions illustrated in Fig. 3 (left). Specification (31) states that "within 10 time units, R_1 has to be visited. Afterwards, R_1 and R_2 must be visited infinitely often while the time between two consecutive visits is never greater than 10. Also, always avoid R_0." All rows of Π are unit vectors.

Note that (30) is unknown to the controller. We sampled 400 evenly-distributed data points from $X \times U$. The number of guards is set to $K = 2$. We used (13) - the computation time was about 5 minutes using Gurobi MILP solver - and obtained the following guards:

$$h_1 = (-1.5, -0.5)^T, h_2 = (0, -3.0)^T, \delta = 0.07,$$

and affine dynamics:

$$A_1 = \begin{pmatrix} 0.75 & -0.38 \\ -0.3 & 0.78 \end{pmatrix}, B_1 = \begin{pmatrix} 0.85 & 0.0 \\ -0.0 & 0.85 \end{pmatrix}, c_1 = \begin{pmatrix} 0.66 \\ 0.56 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 0.79 & -0.04 \\ -0.35 & 0.73 \end{pmatrix}, B_2 = \begin{pmatrix} 0.85 & -0.01 \\ 0.01 & 0.97 \end{pmatrix}, c_2 = \begin{pmatrix} 0.5 \\ 0.6 \end{pmatrix},$$

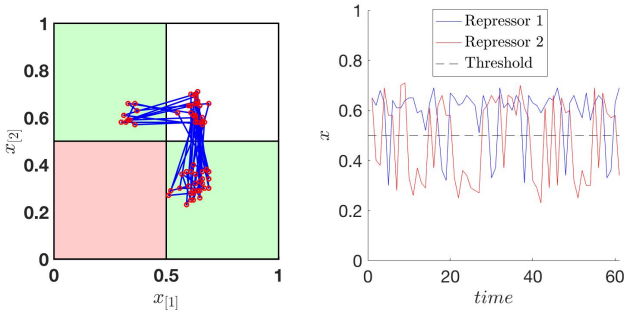

Figure 4: Case Study: a closed-loop trajectory of (30).

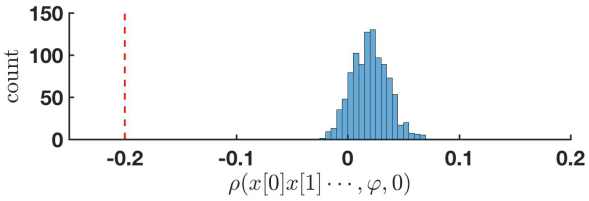

Figure 5: Case Study: histogram of STL scores.

$$A_3 = \begin{pmatrix} 0.76 & -0.4 \\ -0.17 & 0.83 \end{pmatrix}, B_3 = \begin{pmatrix} 0.98 & 0.01 \\ -0.01 & 0.86 \end{pmatrix}, c_3 = \begin{pmatrix} 0.66 \\ 0.48 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} 0.87 & -0.01 \\ -0.15 & 0.81 \end{pmatrix}, B_4 = \begin{pmatrix} 0.96 & -0.0 \\ 0.01 & 0.97 \end{pmatrix}, c_4 = \begin{pmatrix} 0.49 \\ 0.5 \end{pmatrix}.$$

Note that some non-diagonal elements of matrices B are non-zero as disturbances influence data, making it impossible to fully distinguish the effect of control inputs from disturbances. The polyhedral partitions are shown in Fig. 3 (right). We use the infinity norm everywhere and set $k_0 = 0.05, k_x = 1.5, k_u = 1$, which is verified both against (30) and data. Using the procedure outlined in Sec. 4.4, we solved (14) 8 times (2 for each mode) - the computation times were about 15 seconds for each case - and obtained:

$W_1 = [-0.25, 0.25] \times [-0.33, 0.33]$, $W_2 = [-0.13, 0.13] \times [-0.28, 0.28]$,
$W_3 = [-0.28, 0.28] \times [-0.21, 0.21]$, $W_4 = [-0.17, 0.17] \times [-0.23, 0.23]$.

We found $T = [-0.28, 0.28] \times [-0.33, 0.33]$ and $T_u = 0.39\mathbb{B}_\infty$. Thus, $\max_{x \in T} \|\Pi x\|_\infty = 0.33$ for $\Gamma = 1$. The nominal trajectory was computed for $x_0 = (0.65, 0.65)^T$, $\tau_0 = 10$, $\tau_p = 40$, took 4 seconds to solve, and resulted in $\epsilon^* = 0.13$. *Thus, we obtain the guarantee that all the closed-loop trajectories of* (30) *are in the* -0.2*-language of* φ *as* $0.13 - 0.33 = -0.2$. A sample trajectory of (30) is shown in Fig. 4. In simulations, we observed that STL scores were always greater than the guarantee (-0.2). A histogram of STL scores for 1000 simulations of 60 time steps is shown in Fig. 5. The difference between the worst-case theoretical STL score (denoted by the red line) and STL score measured from simulations highlight the conservativeness of the methods in this paper. All simulations were performed by sampling disturbances in (30) uniformly from their respective domains.

7 CONCLUSION AND FUTURE WORK

We developed a framework to use data for provably correct model identification and control synthesis. The methods that we introduced in this paper are sound, but come at theoretically very high computational complexity. Moreover, the methods are conservative. We highlighted the tradeoffs between conservativeness and computational complexity.

Future work will focus on improving the methods introduced in this paper. In particular, we will look for useful heuristics to speed up model identification. Furthermore, we shall extend the framework in this paper to the case when identification and synthesis are performed simultaneously - which is particularly useful in engineering applications.

8 ACKNOWLEDGMENTS

This work was partially supported by NSF awards IIS-1723995, CPS-1446151, and CMMI-1400167. We thank the anonymous reviewers for their helpful comments and suggestions. We also acknowledge Austin Jones of MIT Lincoln Laboratory for early discussions on the ideas presented in this paper.

REFERENCES

[1] Derya Aksaray, Austin Jones, Zhaodan Kong, Mac Schwager, and Calin Belta. 2016. Q-learning for robust satisfaction of signal temporal logic specifications. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 6565–6570.

[2] Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. 2017. Safe Reinforcement Learning via Shielding. *arXiv preprint arXiv:1708.08611* (2017).

[3] Raieev Alur and Nimit Singhania. 2014. Precise piecewise affine models from input-output data. In *Embedded Software (EMSOFT), 2014 International Conference on*. IEEE, 1–10.

[4] Anil Aswani, Humberto Gonzalez, S Shankar Sastry, and Claire Tomlin. 2013. Provably safe and robust learning-based model predictive control. *Automatica* 49, 5 (2013), 1216–1226.

[5] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. Vol. 26202649. MIT press Cambridge.

[6] Grégory Batt, Boyan Yordanov, Ron Weiss, and Calin Belta. 2007. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics* 23, 18 (2007), 2415–2422.

[7] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. 2016. *Formal Methods for Discrete-Time Dynamical Systems*. Springer.

[8] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. 2005. A bounded-error approach to piecewise affine system identification. *IEEE Trans. Automat. Control* 50, 10 (Oct 2005), 1567–1580. https://doi.org/10.1109/TAC.2005.856667

[9] Alberto Bemporad and Manfred Morari. 1999. Control of systems integrating logic, dynamics, and constraints. *Automatica* 35, 3 (1999), 407–427. https://doi.org/10.1016/S0005-1098(98)00178-2

[10] F. Blanchini. 1999. Set invariance in control–a survey. *Automatica* 35, 11 (1999), 1747–1767.

[11] J. P. Calliess. 2017. Lipschitz optimisation for Lipschitz Interpolation. In *2017 American Control Conference (ACC)*. 3141–3146. https://doi.org/10.23919/ACC.2017.7963430

[12] S. Coogan, E. A. Gol, M. Arcak, and C. Belta. 2016. Traffic Network Control From Temporal Logic Specifications. *IEEE Transactions on Control of Network Systems* 3, 2 (June 2016), 162–172. https://doi.org/10.1109/TCNS.2015.2428471

[13] Samira S Farahani, Vasumathi Raman, and Richard M Murray. 2015. Robust Model Predictive Control for Signal Temporal Logic Synthesis. *IFAC-PapersOnLine* 48, 27 (2015), 323–328.

[14] Timothy S Gardner, Charles R Cantor, and James J Collins. 2000. Construction of a genetic toggle switch in Escherichia coli. *Nature* 403, 6767 (2000), 339.

[15] Mohammad S Ghasemi and Ali A Afzalian. 2017. Robust tube-based MPC of constrained piecewise affine systems with bounded additive disturbances. *Nonlinear Analysis: Hybrid Systems* 26 (2017), 86–100.

[16] Sertac Karaman, Ricardo G. Sanfelice, and Emilio Frazzoli. 2008. Optimal control of Mixed Logical Dynamical systems with Linear Temporal Logic specifications. In *2008 47th IEEE Conference on Decision and Control*. IEEE, 2117–2122. https://doi.org/10.1109/CDC.2008.4739370

[17] Oded Maler and Dejan Nickovic. 2004. Monitoring Temporal Properties of Continuous Signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 152 – 166. https://doi.org/10.1007/978-3-540-30206-3_12

[18] David Q Mayne, María M Seron, and SV Raković. 2005. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41, 2 (2005), 219–224.

[19] Mario Milanese and Carlo Novara. 2004. Set membership identification of nonlinear systems. *Automatica* 40, 6 (2004), 957–975.

[20] Joël Ouaknine and James Worrell. 2006. Safety metric temporal logic is fully decidable. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 411–425.

[21] Simone Paoletti, Aleksandar Lj Juloski, Giancarlo Ferrari-Trecate, and René Vidal. 2007. Identification of hybrid systems a tutorial. *European journal of control* 13, 2-3 (2007), 242–260.

[22] SV Raković, Eric C Kerrigan, David Q Mayne, and Konstantinos I Kouramas. 2007. Optimized robust control invariance for linear discrete-time systems: Theoretical foundations. *Automatica* 43, 5 (2007), 831–841.

[23] Sasa V Raković, P Grieder, M Kvasnica, D Q Mayne, and M Morari. 2004. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, Vol. 2. IEEE, 1418–1423.

[24] Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia. 2014. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*. IEEE, 81–87.

[25] Vasumathi Raman, Alexandre Donzé, Dorsa Sadigh, Richard M Murray, and Sanjit A Seshia. 2015. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 239–248.

[26] Matthias Rungger, Manuel Mazo, and Paulo Tabuada. 2013. Controller synthesis for linear systems and safe linear-time temporal logic. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*. ACM, 333–342.

[27] Dorsa Sadigh, Eric S Kim, Samuel Coogan, S Shankar Sastry, and Sanjit A Seshia. 2014. A learning based approach to control synthesis of markov decision processes for linear temporal logic specifications. In *53rd IEEE Conference on Decision and Control*. IEEE, 1091–1096.

[28] S. Sadraddini and C. Belta. 2015. Robust temporal logic model predictive control. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 772–779. https://doi.org/10.1109/ALLERTON.2015.7447084

[29] Sadra Sadraddini and Calin Belta. 2017. Formal Methods for Adaptive Control of Dynamical Systems. In *56th IEEE Conference on Decision and Control (CDC)*. IEEE, 1782–1787.

[30] P Tabuada. 2008. *Verification and Control of Hybrid Systems* . Springer Science & Business Media.

[31] Paulo Tabuada and George J. Pappas. 2006. Linear time logic control of discrete-time linear systems. *IEEE Trans. Automat. Control* 51, 12 (2006), 1862–1877. https://doi.org/10.1109/TAC.2006.886494

[32] C Tomlin, G J Pappas, and S S Sastry. 1998. Conflict resolution for air traffic management: {A} study in multiagent hybrid systems. *IEEE Trans. Automat. Control* 43, 4 (1998), 509–521. https://doi.org/10.1109/9.664154

[33] Mahdi Yousefi, Klaske van Heusden, Guy A Dumont, and J Mark Ansermino. 2015. Safety-preserving closed-loop control of anesthesia. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 454–457.

[34] Majid Zamani, Giordano Pola, Manuel Mazo, and Paulo Tabuada. 2012. Symbolic models for nonlinear control systems without stability assumptions. *Automatic Control, IEEE Transactions on* 57, 7 (2012), 1804–1809.