Controller Synthesis for Unknown-Mode Linear Systems with an Epistemic variant of LTL

Kwesi Rutledge Yuhang Mei Necmiye Ozay

***Abstract*— Linear temporal logic (LTL) with the knowledge operator, denoted as KLTL, is a variant of LTL that incorporates what an agent knows or learns at run-time into its specification. Therefore it is an appropriate logical formalism to specify tasks for systems with unknown components that are learned or estimated at run-time. In this paper, we consider a linear system whose system matrices are unknown but come from an a priori known finite set. We introduce a form of KLTL that can be interpreted over the trajectories of such systems. Finally, we show how controllers that guarantee satisfaction of specifications given in fragments of this form of KLTL can be synthesized using optimization techniques. Our results are demonstrated in simulation and on hardware in a drone scenario where the task of the drone is conditioned on its health status, which is unknown a priori and discovered at run-time.**

I. Introduction

The ability to adapt one's goals when they learn something new about the world is important for human survival. As an example, Sully Sullenberger and his copilot Jeffrey Skiles experienced complete engine failure on their passenger airplane shortly after takeoff. Without power, they were unable to reach their destination or the airfield that they departed from and so they changed their goal to be safely guiding their aircraft into the Hudson River [1]. Similarly, a motorcyclist that notices one of their brakes has become stuck needs to quickly determine how to pull over instead of reaching their final destination. If they do not quickly decide, then a vehicle behind them can accidentally collide with them or their brake pads can become overheated and fail. Enabling robots or other cyber-physical systems (CPSs) to have knowledge- or learning-based goals is thus critical as these types of goals can help guarantee safety when actuators or other parts of a system can fail.

Controller synthesis as a topic in formal methods aims to design tools that can guaranteeably drive robots to achieve tasks or goals [2]. Controller synthesis has been used to define robot policies that correctly achieve obstacle avoidance [3], [4], surveillance [5], bipedal walking [6], and much more. Each of these successes has relied on a correct model of the system being known a priori to make their guarantees. This paper seeks to make guarantees when the exact model is not known a priori, but a set of potential models is.

This work was supported in part by NSF Grant CNS # 1931982 and ONR grant N00014-21-1-2431 (CLEVR-AI). KR is with the Massachusetts Institute of Technology, Cambridge, MA, email: kwesir@mit.edu. YM and NO are with the University of Michigan, Ann Arbor, MI, emails: {yuhangm,necmiye}@umich.edu.

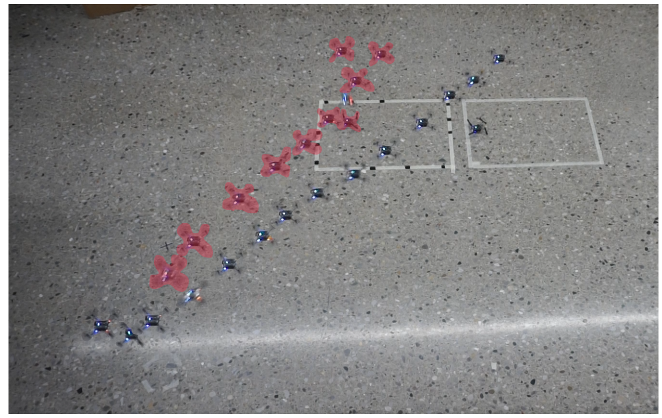

Fig. 1. The controller synthesis approach defined in this paper finds adaptive controllers for unknown linear systems using only a definition of the system class and a learning task (written in KLTL). In the above task, the drone learns whether or not its dynamics are faulty (faulty drone shown in red, normal drone is not colored) and uses that information to sorts itself into two different regions of the workspace.

A. Related Work

A typical approach when there is parametric uncertainty in the model is to use robust control where the controller guarantees correctness against all possible parameters in the uncertainty set [7]. Similar approaches are developed to robustly satisfy rich temporal logic properties [8]. However, this can be conservative, especially in scenarios where it is possible to learn the unknown parameters at run-time. Adaptive control is proposed to take advantage of such online estimated quantities. With the idea of narrowing the set of possible models in mind, adaptive controllers use an online model estimator to narrow down the set of potential models by incorporating an online system identification module [9]. Recent results in adaptive control aim to handle safety [10] or temporal logic constraints [11] for different classes of systems. Among approaches that combine temporal logics and adaptation, [11], [5] use discrete state spaces that can be obtained, for instance, using state abstraction, however, these methods suffer from the curse of dimensionality and thus are known to scale poorly when compared to methods that avoid discretization [12]. A discretization-free method for fault-tolerance has been proposed in [13] where open-loop trajectories, which are tracked online with closed-loop controllers, are designed [4], [14]. However, [13] makes assumptions on fault detectability, which might be nontrivial nor can always be enforced since detectability is indeed a function of closed-loop trajectories.

Following up on our work in [12], in this paper we propose a discretization-free method for designing adaptive controllers that incorporate information that is learned online to select appropriate controller gains. Our controllers incorporate a mode-estimation module, which not only enables adaptation of the control gains as in [12], hence not as conservative as robust controllers, but also allows us to reason about what is "known" at a given time by the system. We capture specifications that depend on online estimated quantities using a form of epistemic logic KLTL that can express hyperproperties, such as how to react to new knowledge gained, which cannot be expressed by traditional temporal logics such as LTL. We show how controller synthesis problems for fragments of KLTL can be reduced to bilinear optimization problems. Finally, we demonstrate our results both in simulation and with hardware experiments with a drone whose task depends on the knowledge of its fault conditions.

B. Mathematical Preliminaries

Throughout this work, we will use the symbols $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{N}$ to represent the set of real, non-negative real, and natural numbers, respectively. All other sets will be referred to with calligraphic symbols (e.g., $\mathcal{X}$). We will frequently refer to a signal, or sequence, $x : \mathbb{N} \to \mathcal{S}$ at specific ranges of times. The symbol $x_t \in \mathcal{S}$ represents the signal x at time t. The symbol $x_{i:j}$ represents the sequence of values that the signal takes from time i to time j.

Lemma 1 (Polytope Containment [15])**.** *Consider the following two polytopes $\mathcal{X} = \{x \in \mathbb{R}^n \mid H_{\mathcal{X}} x \leq h_{\mathcal{X}}\}$ and $\mathcal{Y} = \{y \in \mathbb{R}^n \mid H_{\mathcal{Y}} y \leq h_{\mathcal{Y}}\}$ where $H_{\mathcal{X}} \in \mathbb{R}^{q_x \times n}$ and $H_{\mathcal{Y}} \in \mathbb{R}^{q_y \times n}$. Polytope $\mathcal{Y}$ contains $\mathcal{X}$ (i.e. $\mathcal{X} \subseteq \mathcal{Y}$) if and only if there exists a matrix $\Lambda \in \mathbb{R}_+^{q_x \times q_y}$ such that:*

$$\Lambda H_{\mathcal{X}} = H_{\mathcal{Y}} \tag{1a}$$
$$\Lambda h_{\mathcal{X}} \leq h_{\mathcal{Y}}. \tag{1b}$$

II. Background

In this section, we review some of the relevant work on linear systems and adaptive control. First, we introduce a linear system which can be used to model many discrete-time processes. Then, we discuss a set-membership estimator for this system and how it has been used in previous work [12].

A. Linear System with Unknown Mode

In this paper, we consider a system of the form:

$$x_{t+1} \quad = A^{(\theta)} x_t + B^{(\theta)} u_t + w_t, \quad w_t \in \mathcal{W}^{(\theta)} \tag{2}$$

where $\theta \in \Theta$ is the mode of the system taking values in a finite set Θ of models (i.e. $\theta \in \Theta$), $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state of the system taking values in the state space $\mathcal{X}$, $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the input at time t that must satisfy input constraints represented by polytope $\mathcal{U}$, $w_t \in \mathcal{W}_\theta \subset \mathbb{R}^n$ is the unmeasured disturbance to the system that lies in the polytope $\mathcal{W}_\theta$. The system starts with a fixed, yet unknown, mode θ. It is assumed that the set Θ (including the matrices $A^{(\theta)}, B^{(\theta)}$ and polytopes $\mathcal{W}^{(\theta)}$ for each mode) and the polytope $\mathcal{U}$ are known.

In this paper, we will assume that the initial state of the system is from a polytope $\mathcal{X}_0 \subseteq \mathcal{X}$. To simplify some of our exposition, this paper uses a set $\mathcal{X}_0$ that is a singleton but the results in this paper hold even without this simplification.

This system produces trajectories that we can exactly represent with a convex set:

Definition 1. *The reachable behavior set of mode θ at time t is*

$$\mathcal{R}(\theta, t) = \left\{ (x_{0:t}, u_{0:t-1}) \;\middle|\; \begin{array}{l} x_0 \in \mathcal{X}_0, u_{0:t-1} \in \mathcal{U}^t, \\ \forall k \in [0, t-1], \\ \quad x_{k+1} - A^{(\theta)} x_k - B^{(\theta)} u_k \in \mathcal{W}^{(\theta)} \end{array} \right\} \tag{3}$$

As a slight abuse of notation, we introduce the reachable behavior set for a set *of modes $\Theta' \subseteq \Theta$, $\mathcal{R}(\Theta', t) = \cap_{\theta \in \Theta'} \mathcal{R}(\theta, t)$. Additionally, the reachable behavior set for a sequence of values $\mathbf{m} \in (2^\Theta)^T$ is*

$$\mathcal{R}(\mathbf{m}) \triangleq \left\{ (x_{0:t}, u_{0:t-1}) \;\middle|\; \begin{array}{l} \forall \tau \in [1, T-1] : \\ \quad (x_{0:\tau}, u_{0:\tau-1}) \in \mathcal{R}(\mathbf{m}_\tau, \tau) \end{array} \right\}. \tag{4}$$

The reachable behavior set is the set of all state-input trajectories that can be produced by the mode θ. The set is a polytope as is proven in Lemma 3 of [12].

B. Set Membership Estimation

For such a system, we use the following estimator to identify the set of modes $\mu(x_{0:t}, u_{0:t-1}) \subseteq \Theta$ that is consistent with the observed data $(x_{0:t}, u_{0:t-1})$:

$$\mu(x_{0:t}, u_{0:t-1}) \triangleq \{\theta \in \Theta \mid (x_{0:t}, u_{0:t-1}) \in \mathcal{R}(\theta, t)\}.$$

We will occasionally simplify the notation used to refer to the estimator output as $\mu_t \triangleq \mu(x_{0:t}, u_{0:t-1})$. This makes it clearer that μ is a signal as well. The set of state-input trajectories that produce the same estimator output is the *consistency set*.

Definition 2 (Consistency Set)**.** *Consider the sequence of estimates $\mathbf{m} \in (2^\Theta)^{t+1}$. The consistency set $\mathcal{C}(\mathbf{m})$ is the set of all state-input trajectories $(x_{0:t}, u_{0:t-1})$ that lead to the estimate $\mathbf{m}$. That is:*

$$\mathcal{C}(\mathbf{m}) = \{(x_{0:t}, u_{0:t-1}) \mid \mathbf{m} = \mu_{0:t}\}.$$

Crucially, consistency sets are generally not convex.

C. Adaptive Controller Synthesis for Reachability Task

The set membership estimator discussed above is employed in an adaptive, switched linear controller. The adaptive controller's structure γ was proposed in [12] to control the system (2). In this paper, the controller will be written using the following symbols:

$$\gamma(x_{0:t}, u_{0:t-1}) = \gamma|_{\mathbf{m}^*}(x_{0:t}, u_{0:t-1}) \tag{5}$$

where

- $\mathbf{m}^* = \mu_{0:t}$,

- $\gamma|_{\mathbf{m}^*}(x_{0:t}, u_{0:t-1}) = \sum_{\tau=0}^{t-1} K_\tau^{(\mathbf{m}^*)} \hat{w}_\tau + k^{(\mathbf{m}^*)}$, and
- $\hat{w}_\tau$ is defined according to the following rule:

$$\hat{w}_\tau \triangleq \hat{w}(x_{0:t}, u_{0:t-1}) = x_{\tau+1} - A^{(\theta')}x_\tau - B^{(\theta')}u_\tau \quad (6)$$

for a fixed $\theta' \in \mu_t$ and $\tau < t$. Note that this controller is composed of a set of linear, disturbance feedback functions with memory, parametrized by $(K_\tau^{(\mathbf{m}^*)}, k_\tau^{(\mathbf{m}^*)})$, that it switches between based on the output $\mathbf{m}^*$ of μ.

At each time τ, μ_τ is one of a finite number of values (i.e., $\mu_\tau \in 2^\Theta \setminus \emptyset$). Thus, the sequence $\mu_{0:T}$ also belongs to a finite set $((2^\Theta)^{T+1})$. With this, we can define the closed-loop versions of consistency sets and reachable behavior sets that are related as follows:

$$\mathcal{R}^* = \bigcup_{\mathbf{m}\in(2^\Theta)^{T+1}} \mathcal{C}(\mathbf{m}, \gamma|_\mathbf{m}) = \bigcup_{\mathbf{m}\in(2^\Theta)^{T+1}} \mathcal{R}(\mathbf{m}, \gamma|_\mathbf{m}) \quad (7)$$

where

$$\mathcal{R}^* \triangleq \left\{ \begin{bmatrix} x_{0:T-1} \\ u_{0:T-2} \end{bmatrix} \middle| \begin{array}{l} x_0 \in \mathcal{X}_0, u_0 = \gamma(x_0) \\ \exists \theta \in \Theta : \forall \tau \in [1, T-2] : \\ \quad x_\tau - A^{(\theta)}x_{\tau-1} + B^{(\theta)}u_{\tau-1} \in \mathcal{W}^{(\theta)} \\ \quad u_\tau = \gamma(x_{0:\tau}, u_{0:\tau-1}) \\ x_{T-1} - A^{(\theta)}x_{T-2} + B^{(\theta)}u_{T-2} \in \mathcal{W}^{(\theta)} \end{array} \right\},$$

$$\mathcal{C}(\mathbf{m}', \gamma|_\mathbf{m}) \triangleq \left\{ \begin{bmatrix} x_{0:t} \\ u_{0:t-1} \end{bmatrix} \in \mathcal{C}(\mathbf{m}') \middle| \begin{array}{l} \forall \tau \in [0, t] : \\ \quad u_\tau = \gamma|_\mathbf{m}(x_{0:\tau}, u_{0:\tau-1}) \end{array} \right\},$$

and $\mathcal{R}(\mathbf{m}', \gamma|_\mathbf{m})$ is defined similarly. The ability to decompose the set of all possible trajectories of a system using either a partition made up of consistency sets or a set cover made up of reachable behavior sets will be used to guarantee properties of all trajectories of a system. The set cover of reachable behavior sets, specifically, is easier to analyze than the set of all possible trajectories because each reachable behavior set is a polytope.

III. Problem Statement

This section first introduces a form of Linear Temporal Logic with the Knowledge operator (KLTL) designed for application to discrete-time dynamical systems like (2). Then, we introduce our controller synthesis problem in terms of a discrete-time linear system's satisfaction of a given KLTL formula.

A. KLTL

Temporal logics like LTL specify the desired behaviors of software and cyber-physical systems when the state of the world is completely measured. When the state of the world is *partially* observed, these temporal logics can be used in conservative ways where enough state uncertainty can lead to infeasibility of the formula. On the other end of the spectrum, temporal logics like KLTL embrace the existence of partial state observability. KLTL, specifically, defines a knowledge operator which encodes whether or not a formula's satisfaction can be inferred or learned from the current data.

KLTL was initially introduced for discrete systems like Transition Systems [16], but is understood to be a form of Dynamic Epistemic Logic [17] which has existed for much longer. This paper is the first that we are aware of which seeks to apply KLTL to linear dynamical systems. For these systems, the standard grammar and semantics of KLTL must be slightly modified and we discuss their definitions here.

Definition 3 (KLTL Grammar)**.** *The grammar of KLTL is as follows:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi \mathcal{U} \varphi \mid \mathcal{K}\varphi \quad (8)$$

in which $p \in AP$ is an atomic proposition, while $\bigcirc$ and $\mathcal{U}$ are the "next" and "until" operators from linear temporal logic. Formulas of the type $\mathcal{K}\varphi$ are read as "the system knows that the formula φ holds". We define the temporal modalities $\Diamond$ (eventually) and $\Box$ (always) as usual.

To develop semantics for this KLTL system, we introduce the concept of the labelling function and the set of trajectory-mode pairs.

Definition 4 (Labelling Function)**.** *A labelling function for a set of atomic propositions AP for the system* (2) *is a function that describes which atomic propositions are satisfied at a given state and with a given mode, (i.e. $L : \mathcal{X} \times \Theta \to 2^{AP}$).*

Assumption 1. *We assume that each atomic proposition p is associated with a polytopic set $\mathcal{S}_p \subseteq \mathcal{X}$ or a discrete set $\Theta_p \subseteq \Theta$ such that either:*

- $x \in \mathcal{S}_p \iff p \in L(x, \theta)$*, or*
- $\theta \in \Theta_p \iff p \in L(x, \theta)$.

A proposition that has an associated polytopic set representation in the state space will be called a *state-based proposition* and a proposition that has an associated discrete set Θ_p will be called a *model-based proposition*.

Let the set of all feasible trajectory-mode pairs (x, θ) that (2) can produced over time horizon T (i.e. $x = x_{0:T}$) with a given controller γ be:

$$\mathcal{T}(\Theta, \gamma) \triangleq \left\{ (x, \theta) \middle| \begin{array}{l} \theta \in \Theta,\ x_0 \in \mathcal{X}_0,\ u_0 = \gamma(x_0) \\ \forall \tau \in [1, T-1] : \\ \quad x_\tau - A^{(\theta)}x_{\tau-1} - B^{(\theta)}u_{\tau-1} \in \mathcal{W}^{(\theta)} \\ \quad u_\tau = \gamma(x_{0:\tau}, u_{0:\tau-1}) \\ x_T - A^{(\theta)}x_{T-1} - B^{(\theta)}u_{T-1} \in \mathcal{W}^{(\theta)} \end{array} \right\}$$

With this labelling function and the set $\mathcal{T}(\Theta, \gamma)$ in mind, we now define the KLTL semantics for systems with unknown mode.

Definition 5 (KLTL Semantics for Unknown-Mode Linear Systems)**.** *For any pair of a state trajectory x and unknown parameter θ, interpret the KLTL operators as follows:*

- $(x, \theta, i) \models p$ *if* $p \in L(x_i, \theta)$
- $(x, \theta, i) \models \neg p$ *if* $(x, \theta, i) \not\models p$
- $(x, \theta, i) \models \varphi_1 \wedge \varphi_2$ *if* $(x, \theta, i) \models \varphi_1$ *and* $(x, \theta, i) \models \varphi_2$
- $(x, \theta, i) \models \bigcirc\varphi$ *if* $(x, \theta, i+1) \models \varphi$
- $(x, \theta, i) \models \varphi_1 U \varphi_2$ *if* $\exists \tau \geq 0$ *such that* $(x, \theta, j) \models \varphi_2$ *and* $\forall 0 \leq k < \tau$, $(x, \theta, k) \models \varphi_1$,

- $(x, \theta, i) \models \mathcal{K}\varphi$ *if for all* $(x', \theta') \in \mathcal{T}(\Theta, \gamma)$ *s.t.* $(x, \theta) \sim_i (x', \theta')$*, we have* $(x', \theta', i) \models \varphi$.

In our case, we define the similarity relationship $\sim_i$ *as* $(x, \theta) \sim_i (x', \theta')$ *if and only if* $x_{0:i} = x'_{0:i}$.

The semantics of $(x, \theta, i) \models \mathcal{K}\varphi$ state that "the system knows φ" is satisfied if for any *similar* trajectory-mode pairs of the closed-loop system, (x', θ'), the formula φ is satisfied. To simplify the exposition, we introduce the following shorthand $(x, \theta) \models \varphi \iff (x, \theta, 0) \models \varphi$.

B. Formal Problem Statement

The problem class that is addressed in this paper can now be written.

Problem 1. *Consider an unknown-mode linear system* Σ *and a KLTL formula* φ*. Find a controller* $\gamma : \mathcal{X} \times (\mathcal{X} \times \mathcal{U})^* \to \mathcal{U}$ *that guarantees all closed-loop trajectory-mode pairs of the system satisfy the formula* φ.

Importantly, we do not develop a single algorithm which can satisfy an arbitrary KLTL formula for linear systems. Instead, we present a number of formula templates in the following sections and provide recipes for how to guarantee that these useful templates are satisfied. Those interested in more complicated formulas will need to use the intuition developed here to design adaptive controllers for their own KLTL formulas and we demonstrate how this can be done with more complicated formulas later on.

IV. Approach

In this section, we will discuss how to design adaptive controllers that satisfy several template KLTL formulas. The template formulas involve most of the operators introduced in Definition 5, but frequently contain atomic propositions and not complex compositions of formulas. The types of atomic propositions that we consider in this paper are categorized as either state-based or model-based, but combinations of these atomic propositions can be used to express a broad variety of goals:

- $(x, \theta) \models p$
- $(x, \theta) \models \bigcirc^\tau \varphi$
- $(x, \theta) \models \varphi_1 U \varphi_2$
- $(x, \theta) \models \mathcal{K}\varphi$

For the sake of illustrating our sketch of a solution to Problem 1, we discuss what constraints need to be satisfied in order for the system (2) with controller γ to satisfy such formulas. Similar formulas (e.g., $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $\Diamond\varphi$) can all be interpreted from our definitions of these initial operators.

A. Satisfying Atomic Propositions

First, we will consider the constraints that must be satisfied for a controller that solves Problem 1 where $\varphi = p$. In order to do this, we must verify that *every* trajectory-mode pair of the system satisfies the conditions outlined in Definition 5. The set $\mathcal{T}(\Theta, \gamma)$ of all trajectory-mode pairs for system (2) with controller γ can be represented by the union of a finite number of convex sets as follows.

Lemma 2. *The set of all feasible, closed-loop trajectory-mode pairs* $(x_{0:T}, \theta)$ *that a system* (2) *can produce for a given controller* γ *over time horizon* T *is equivalent to the following union of polytopes:*

$$\mathcal{T}(\Theta, \gamma) = \bigcup_{\mathbf{m} \in (2^\Theta)^{T+1}} \bigcup_{\theta \in \mathbf{m}_T} R_{[0:T]} \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) \times \{\theta\} \quad (9)$$

where $R_{[0:T]}$ *is a matrix that selects the state from time* 0 *to the end of the trajectory* $(x_{0:T-1})$ *of the vector* $\begin{bmatrix} x_{0:T-1}^\top & u_{0:T-2}^\top \end{bmatrix}^\top$.

Proof: This lemma holds given (7) and the fact that x_T can be exactly defined as belonging to a polytopic projection of $\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}})$. □

Now, we can present constraints that guarantee a controller satisfies the formula $\varphi = p$ when obeyed.

Proposition 1. *Consider an uncertain linear system* (2) *and a state-based atomic proposition* p*. The system under controller* γ *satisfies the formula* $\varphi = p$ *if for all* $\mathbf{m} \in (2^\Theta)^{T+1}$

$$R_0 \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) \subseteq \mathcal{S}_p \quad (10)$$

where R_0 *is a matrix that selects the state at time 0* (x_0) *of the vector* $\begin{bmatrix} x_{0:T}^\top & u_{0:T-1}^\top \end{bmatrix}^\top$.

Proof: Consider the relationship (7). If each of the reachable sets satisfies (10), then the full set $\mathcal{R}^*$ satisfies the set containment as well. Thus, all trajectories of the system satisfy the single state-based formula. □

The constraint (10) is a polytope containment constraint and can be written as shown in Lemma 1. The constraints in Lemma 1 are linear if the polytope matrices (e.g., $H_\mathcal{X}$ and $h_\mathcal{Y}$) are constant, but in our work the polytope matrices are linear functions of the decision variables (e.g. $(K_\tau^{(\mathbf{m}^*)}, k_\tau^{(\mathbf{m}^*)})$) as discussed in [12]. Thus, we know these polytope containment constraints are bilinear. So, for state-based atomic proposition p, finding a controller that satisfies the bilinear constraint (10) guarantees that the proposition p is satisfied.

Proposition 2. *Consider an uncertain linear system* (2) *and a model-based atomic proposition* p*. The system under controller* γ *satisfies the formula* $\varphi = p$ *if and only if* $\Theta_p = \Theta$.

Proof: The formula is only satisfied if $\theta \in \Theta_p$ for all $(x, \theta) \in \mathcal{R}^*$. By definition, every mode $\theta \in \Theta$ is represented in $\mathcal{R}^*$, so the formula is only satisfied if Θ_p also contains all modes (i.e. $\Theta_p = \Theta$). □

Thus, only one model-based atomic proposition can be satisfied in this way. This makes sense as the mode is not influenced by the controller and can take any value in Θ.

B. Satisfying Formulas with the Repeated Next Operator

If Problem 1 is solved for a formula $(\bigcirc\varphi)$, then all trajectories of system (2) satisfy formula φ at time $t = 1$. Such trajectories satisfy the formula at the "next" time after

$t = 0$. So, when the next operator is repeated more than once:

$$\bigcirc^{\tau} = \underbrace{\bigcirc \bigcirc \cdots \bigcirc}_{\tau \text{ times}}$$

then the formula φ should be satisfied at time step $t = \tau$ in order to satisfy $\bigcirc^{\tau}\varphi$.

This is equivalent to simply applying a linear transform to the set $\mathcal{R}^*$ and thus can be easily incorporated into optimization-based approaches.

Proposition 3. *Consider an uncertain linear system* (2) *and a state-based proposition p. The system under controller γ satisfies the formula $\varphi = \bigcirc^{\tau} p$ if for all $\mathbf{m} \in (2^{\Theta})^{T+1}$*

$$R_0 \cdot R_{[\tau:]}\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) \subseteq \mathcal{S}_p \tag{11}$$

where $R_{[\tau:]}$ is a matrix that selects the state from time τ to the end of the trajectory ($x_{\tau:T-1}$) of the vector $\begin{bmatrix} x_{0:T-1}^{\top} & u_{0:T-2}^{\top} \end{bmatrix}^{\top}$.

Again, (11) is a polytope containment constraint. Thus, we know it can be transformed into a set of bilinear constraints using Lemma 1.

C. Satisfying Formulas with the Until Operator

If Problem 1 is solved for a formula $(\varphi_1 U \varphi_2)$, then all trajectories of system (2) satisfy formula φ_1 for all times up until some time $\tau \geq 0$ where φ_2 must be satisfied.

When the formulas φ_1 and φ_2 are both state-based atomic propositions, this is equivalent to a sequence of many polytope containment constraints which can be easily incorporated into optimization-based approaches.

Proposition 4. *Consider an uncertain linear system* (2) *and two state-based atomic propositions p_1 and p_2. The system under controller γ satisfies the formula $p_1 U p_2$ if there exists a $\tau \geq 0$ such that for all $\mathbf{m} \in (2^{\Theta})^{T+1}$*

$$R_i \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) \subseteq \mathcal{S}_{p_1} \quad \forall i < \tau \tag{12}$$

and

$$R_{\tau} \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) \subseteq \mathcal{S}_{p_2}. \tag{13}$$

Proof: Consider the semantics of the until operator defined in Definition 5. With the until semantics in mind as well as Proposition 1, it follows that the existence of a τ such that (12) and (13) proves that $p_1 U p_2$. □

For the sake of space, we refrain from discussing other conditions under which formulas containing the until operator are satisfied (i.e., for other combinations of state- and model-based propositions).

D. Satisfying Formulas with the Knowledge Operator

The following results describe what constraints the closed-loop sets $\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}})$ should satisfy to guarantee that learning-based formulas are satisfied. We say that a formula φ which includes the $\mathcal{K}$ operator is learning-based and will show how useful these formulas can be.

The first formula template that we analyze are for the formula $\varphi = \mathcal{K}p$ which behaves predictably given our results for the formula $\varphi' = p$.

Proposition 5. *Consider an uncertain linear system* (2) *and a model-based proposition p. The system under controller γ satisfies the formula $\varphi = \mathcal{K}p$ if and only if $\Theta_p = \Theta$.*

This result can then be extended for a formula template that defines how a system should "react" to new information.

1) Adaptation to New Knowledge: Consider the following template formula:

$$\varphi = \bigcirc^{\tau_1}(\mathcal{K}p_1 \implies \bigcirc^{\tau_2} p_2).$$

The formula is satisfied if, on all trajectories where the controller knows that the true mode of the system is in Θ_p at time τ_1, the trajectory eventually reaches a region $\mathcal{X}_{p_2}$ at time $\tau_1 + \tau_2$.

This more complicated formula uses the $\mathcal{K}$ operator as a precondition in an implies statement. Thus, this statement is only applied to trajectories that satisfy $\mathcal{K}p_1$ at time τ_1 and no other trajectories. We can extract this set of trajectories using the following lemma:

Lemma 3. *Consider an uncertain linear system* (2) *and model-based atomic proposition p with $\Theta_p \subseteq \Theta$. For such a system and for formula $\varphi = \bigcirc^{\tau}\mathcal{K}p$, the following sets are equivalent:*

$$\{(x,\theta) \mid (x,\theta) \models \varphi\} = \bigcup_{\substack{\mathbf{m} \in (2^{\Theta})^{T+1} \\ \textit{s.t. } \mathbf{m}_{\tau} \subseteq \Theta_p}} \mathcal{C}(\mathbf{m}, \gamma|_{\mathbf{m}}) \times \mathbf{m}_T. \tag{14}$$

Proof: First, consider a trajectory mode pair (x, θ) such that $(x,\theta) \models \varphi$. By the definition, this trajectory-mode pair satisfies $\varphi = \bigcirc^{\tau}\mathcal{K}p$ if and only if $\mu(x_{0:\tau}, u_{0:\tau-1}) \subseteq \Theta_p$, where u_t is the output of the controller (5). This output of the estimator is feasible if and only if there exists $\mathbf{m} = \mu_{0:T} \in (2^{\Theta})^{T+1}$ such that $(x,\theta) \in \mathcal{C}(\mathbf{m}, \gamma|_{\mathbf{m}}) \times \mathbf{m}_T$. Thus, the left hand side is equivalent to the right hand side. □

Proposition 6. *Consider an uncertain linear system* (2)*, a model-based atomic proposition p_1 and a state-based atomic proposition p_2. The system under controller γ satisfies the formula $\varphi = \bigcirc^{\tau_1}(\mathcal{K}p_1 \implies \bigcirc^{\tau_2} p_2)$ if for all $\mathbf{m} \in (2^{\Theta})^{T+1}$ such that $\mathbf{m}_{\tau_1} \subseteq \Theta_{p_1}$:*

$$R_{\tau_1 + \tau_j} \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) \subseteq \mathcal{S}_{p_2}. \tag{15}$$

Proof: The formula is satisfied if all trajectory-mode pairs (x, θ) of the system that satisfy $\bigcirc^{\tau_1}\mathcal{K}p_1$ also satisfy $\bigcirc^{\tau_1+\tau_2} p_2$. The set of trajectory-mode pairs (x, θ) that satisfy $\bigcirc^{\tau_1}\mathcal{K}p_1$ is given in Lemma 3. Furthermore, using the properties of consistency sets:

$$\bigcup_{\substack{\mathbf{m} \in (2^{\Theta})^{T+1} \\ \text{s.t. } \mathbf{m}_{\tau_1} \subseteq \Theta_{p_1}}} \mathcal{C}(\mathbf{m}, \gamma|_{\mathbf{m}}) \subseteq \bigcup_{\substack{\mathbf{m} \in (2^{\Theta})^{T+1} \\ \text{s.t. } \mathbf{m}_{\tau_1} \subseteq \Theta_{p_1}}} \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}).$$

Therefore, if (15) is satisfied for $\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}})$ then it also holds for $\mathcal{C}(\mathbf{m}, \gamma|_{\mathbf{m}})$. Then, applying the results from Proposition 3, we can verify that all such trajectory-mode pairs satisfy the final formula $\bigcirc^{\tau_1+\tau_2} p_2$. □

V. Satisfying More Complex Formulas

The templates given in the last section offer a blueprint for how to guarantee that linear systems satisfy arbitrarily complex formulas. In this section, the templates are used to design controllers that satisfy two complex formulas of practical interest. The first formula is the "learn, then adapt" formula which adapts the goal of the controller depending on what is learned at run-time. The second formula is a formula which guarantees model information is hidden by the controller and from any external adversary. The design problem for each of these formulas is shown to be equivalent to a bilinear optimization problem which can be solved using off-the-shelf optimization toolboxes such as YALMIP [18].

A. Learn, Then Adapt

The "learn, then adapt" class of formulas has the following form:

$$\varphi_L = \bigwedge_{i=1}^{|\Theta|} \left(\Diamond \mathcal{K}\{\theta_i\} \implies \Diamond \mathcal{X}_T^{(i)} \right) \tag{16}$$

where we slightly abuse the grammar of KLTL by using the sets Θ_p and $\mathcal{S}_p$ in the place of propositions (e.g., p). For example, in the above formula $\{\theta_i\}$ is meant to represent the atomic proposition that is satisfied only when the trajectory was generated by mode i.

This formula can be relevant in tasks where the controller might perform a health check and if a fault is detected due to internal or external issues, an alternative task is attempted instead of the original task. Similarly, this formula is also relevant in situations where the task is conditioned on the a priori unknown discrete state of the system (e.g., the type of cargo being carried). The next result shows how to synthesize controllers that guarantee satisfaction of "learn-then adapt" formulas.

Proposition 7. *Assume that the system's* (2) *mode is guaranteed to be known in* T *or fewer steps (i.e.,* $\mathcal{C}(\mathbf{m}) = \emptyset$ *when* $|\mathbf{m}_{T-1}| > 1$*). If the following optimization problem is feasible*

$$\begin{aligned}
&\textit{Find } \left\{ \mathbf{K}^{(\mathbf{m})}, \mathbf{k}^{(\mathbf{m})} \right\}_{\mathbf{m} \in (2^\Theta)^T} \\
&\quad \textit{s.t. } \forall \mathbf{m} \in (2^\Theta)^{T+1} : \\
&\qquad \exists \tau \in [1, T-1] \textit{ s.t. } \mathbf{m}_\tau = \{i\} \implies && (17a)\\
&\qquad\quad G_T(i, \mathbf{m}) \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) + B^{(i)}(k^{(\mathbf{m})}) \oplus \mathcal{W}^{(i)} \subseteq \mathcal{X}_T^{(i)} && (17b)\\
&\qquad \forall t \in [0, T-1] : && (17c)\\
&\qquad\quad G_u(i, \mathbf{m}, t) \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) + (k^{(\mathbf{m}_{0:t})}) \subseteq \mathcal{U} && (17d)\\
&\quad \forall \mathbf{m}, \mathbf{m}' \in (2^\Theta)^{T+1} : \\
&\qquad \mathbf{m}_{0:t} = \mathbf{m}'_{0:t} \implies \\
&\qquad\quad K^{(\mathbf{m}_{0:t})} = K^{(\mathbf{m}'_{0:t})},\ k^{(\mathbf{m}_{0:t})} = k^{(\mathbf{m}'_{0:t})}. && (17e)
\end{aligned}$$

where

- $G_T(\theta, \mathbf{m}) = \begin{bmatrix} A^{(\theta)} + B^{(\theta)} K^{(\mathbf{m})} \hat{R}_x^{(\theta)} & B^{(\theta)} K^{(\mathbf{m})} \hat{R}_u^{(\theta)} \end{bmatrix}$
- $\hat{R}_x^{(\theta)}$ *and* $\hat{R}_u^{(\theta)}$ *are defined to reconstruct the disturbances according to the disturbance estimator from [19] (i.e.,* $\hat{w}_{0:T-2} = \hat{R}_x^{(\theta)} x + \hat{R}_u^{(\theta)} u$ *where* $\hat{w}$ *is defined as in* (6)*)*
- $G_u(\theta, \mathbf{m}, t) = R_t \begin{bmatrix} \mathbf{K}^{(\mathbf{m})} \hat{R}_x^{(\theta)} & \mathbf{K}^{(\mathbf{m})} \hat{R}_u^{(\theta)} \end{bmatrix}$

then there exists an adaptive controller which solves Problem 1 for $\varphi = \varphi_L$*. The matrix* $\mathbf{K}^{(\mathbf{m})}$ *is a block matrix where the* (i,j)*-th* $m \times n$ *block is defined as follows:*

$$\mathbf{K}^{(\mathbf{m})}_{[i],[j]} = \begin{cases} 0_{m \times n} & i \leq j \\ K_j^{(\mathbf{m}_{0:i})} & \textit{otherwise.} \end{cases} \tag{18}$$

Proof: Assume that optimization (17) is feasible.

Now, consider an arbitrary trajectory $(x_{0:T}, u_{0:T-1})$ created by the system (2) in mode $\theta \in \Theta$ and the controller γ defined by feasible variables of (17). The trajectory $(x_{0:T}, u_{0:T-1})$ is an element of the set $\mathcal{C}(\overline{\mathbf{m}}, \gamma|_{\overline{\mathbf{m}}})$ for some $\overline{\mathbf{m}} \in (2^\Theta)^{T+1}$.

By the assumptions of Proposition 7, it is not possible for $\overline{\mathbf{m}}_\tau = \Theta$ for all $\tau \in [1, T-1]$. In other words, $|\overline{\mathbf{m}}_{T-1}| = 1$. Let's suppose that $\overline{\mathbf{m}}_{T-1} = \{\theta_1\}$ without loss of generality.

By definition of $\mathcal{R}$, $R_T \mathcal{R}(\overline{\mathbf{m}}, \gamma|_{\overline{\mathbf{m}}}) = G_T(\theta, \mathbf{m}) \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) + B^{(\theta)}(k^{(\mathbf{m})}) \oplus \mathcal{W}^{(\theta)}$ where $\mathbf{m} = \overline{\mathbf{m}}_{0:T-1}$. This equality allows us to simplify the constraints (17a) and (17b) to $R_T \mathcal{R}(\overline{\mathbf{m}}, \gamma|_{\overline{\mathbf{m}}}) \subseteq \mathcal{X}_T^{(i)}$ for this particular $\overline{\mathbf{m}}$. This condition is now in the form from Proposition 6 and thus we can guarantee that this trajectory-mode pair satisfies $\bigcirc^{\tau_1}(\mathcal{K} p_1 \implies \bigcirc^{\tau_2} p_2)$ for some $\tau \in [1, T-1]$. Trajectory-mode pair $(x_{0:T}, u_{0:T-1})$ satisfying that formula is a sufficient condition for the pair satisfying φ_L.

It follows that for any trajectory of the closed-loop system, the mode is learned before time T and then the system reaches its mode-dependent goal. Thus, is a sufficient condition for satisfaction of φ_L is provided. □

Note that for any sequence $\overline{\mathbf{m}} \in (2^\Theta)^{T+1}$ that does not have a τ where $\overline{\mathbf{m}}_\tau = \{\theta_i\}$ for some i, the formula φ_L does not require anything. It is also possible to satisfy the formula φ_L if one can hide information as discussed in the next section but the sufficient conditions in Proposition 7 do not cover this case.

As discussed in some of the previous sections, this optimization contains bilinear constraints. Specifically, the bilinearity comes from all constraints where a set containment constraint is created with an inbody that is a parameterized set, (e.g. $\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}})$) and a circumbody that is another polytopic set. The parameterized set $\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}})$ is defined by matrices that are a linear combination of the control gains, which leads to a multiplication of the control gains with dual variables (i.e., Λ).

B. Accomplish Goal While Hiding Information

The "controller-hidden information" formula template is also a specification that can be written in terms of KLTL and has practical application. It is formally written as:

$$\varphi_H = \neg \left(\bigvee_{\theta_i \in \Theta'} \mathcal{K}\{\theta_i\} \right) U \mathcal{X}_T \tag{19}$$

where the same abuse of notation from φ_L is used here.

This formula is useful in situations where the controller would like to keep some model information hidden (i.e., the models within Θ') to external observers until a task is achieved. For example, an autonomous vehicle may wish to hide the state of its internal damage to outside observers while completing a trip. The next result shows how to synthesize controllers that guarantee satisfaction of "controller-hidden information" formulas.

Proposition 8. *If the following optimization problem is feasible*

$$
\begin{aligned}
&\textit{Find } \{\mathbf{K}^{(\mathbf{m})}, \mathbf{k}^{(\mathbf{m})}\}_{\mathbf{m}\in(2^{\Theta})^T} \\
&\quad \textit{s.t. } \forall \mathbf{m} \in (2^{\Theta})^T \\
&\qquad \forall \tau \in [1, T-1] \;\; \mathbf{m}_\tau \neq \{\theta_i\} \implies \\
&\qquad\quad G_T(i,\mathbf{m})\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) + B^{(i)}(k^{(\mathbf{m})}) \oplus \mathcal{W}^{(i)} \subseteq \mathcal{X}_T \qquad (20a) \\
&\qquad \exists \tau \in [1, T-1] \textit{ s.t. } \mathbf{m}_\tau = \{\theta_i\} \implies \\
&\qquad\quad \mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) = \emptyset \qquad (20b) \\
&\qquad \forall t \in [0, T-1]: \\
&\qquad\quad G_u(i,\mathbf{m},t)\mathcal{R}(\mathbf{m}, \gamma|_{\mathbf{m}}) + (k^{(\mathbf{m})}) \subseteq \mathcal{U} \qquad (20c) \\
&\qquad \forall \mathbf{m}, \mathbf{m}' \in (2^{\Theta})^T: \\
&\qquad \mathbf{m}_{0:t} = \mathbf{m}'_{0:t} \implies \\
&\qquad\quad K^{(\mathbf{m}_{0:t})} = K^{(\mathbf{m}'_{0:t})},\; k^{(\mathbf{m}_{0:t})} = k^{(\mathbf{m}'_{0:t})}. \qquad (20d)
\end{aligned}
$$

where $G_T(\theta,\mathbf{m})$, $\hat{R}_x^{(\theta)}$, $\hat{R}_x^{(\theta)}$, $G_u(\theta,\mathbf{m},t)$ and $\mathbf{K}^{(\mathbf{m})}$ are defined as in Proposition 7, then there exists an adaptive controller that solves Problem 1 for $\varphi = \varphi_H$ with $\Theta' = \{\theta_i\}$.

Proof: Assume that optimization (20) is feasible. Now, consider an arbitrary trajectory $(x_{0:T}, u_{0:T-1})$ created by the system (2) in mode $\theta \in \Theta$ and the controller γ defined by feasible variables of (20). The trajectory $(x_{0:T}, u_{0:T-1})$ is an element of the set $\mathcal{C}(\overline{\mathbf{m}}, \gamma|_{\overline{\mathbf{m}}})$ for some $\bar{\mathbf{m}} \in (2^{\Theta})^{T+1}$. The specific sequence $\overline{\mathbf{m}}$ is constrained by (20b). The constraint would be infeasible if a trajectory existed that produced $\mathbf{m}$ where the true model is in the hidden models Θ'. Given that the constraint was feasible, the sequence $\overline{\mathbf{m}}$ guarantees that the hidden model *remains* hidden until at least time T. At time T, the trajectory is guaranteed to reach the target region $\mathcal{X}_T$ as guaranteed by (20a). Finally, we can conclude that any trajectory under the controller defined by (20) is guaranteed to hide the modes from Θ' until time T at which point the state-based atomic proposition $\mathcal{X}_T$ will be satisfied. φ_H is guaranteed to be satisfied. □

VI. Results

In this section, we demonstrate controllers synthesized with formulas from Section V on a drone system. The optimization problems are solved using a 2017 Dell XPS laptop with 2.8 GHz Processor and 16 GB RAM.The performance of the synthesized controllers are shown both in simulation and on a Crazyflie 2.1 quadcopter.

A. The Drone System

We model the drone as a two-dimensional single integrator (i.e. we assume that we control the drone's velocity).

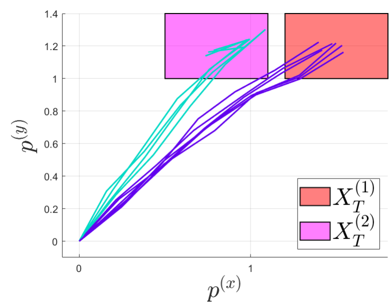

Fig. 2. Ten trajectories of the simulated drone system under the adaptive controller that is the solution to (17) (Section V-A). The solution satisfies task ϕ_L, guaranteeing that any trajectories under normal (purple) or corrupted (cyan) dynamics are steered into the correct regions ($\mathcal{X}_T^{(1)}$ in red, or $\mathcal{X}_T^{(2)}$ in magenta, respectively).

Consider the system as a two mode instance (i.e. $|\Theta| = 2$) of (2), where the state is a two-dimensional vector $x_t = [p_t^{(x)}\; p_t^{(y)}]^\top$ representing the drone's position in the x- and y-axes of the plane. The initial state set of the system is $\mathcal{X}_0 = \{[0\;0]^\top\}$. The matrices defining the system, the set of disturbances $\mathcal{W}$, and the and admissible control inputs will be slightly different for each formula.

The velocity controller for the drone is emulated in hardware using a position controller which receives the point $(p^{(x)} + u_x\Delta t, p^{(y)} + u_y\Delta t)$ at each time step.

B. Learn, Then Adapt

Let the drone system be represented by two tuples $(A, B_1, \mathcal{W}_1) = (I, R_0, \mathcal{W})$ and $(A, B_2, \mathcal{W}_2) = (I, R_{\pi/16}, \mathcal{W})$. The set of allowable inputs is $\mathcal{U} = \{u \in \mathbb{R}^2 \mid \|u\|_\infty \leq 0.5\}$. Note that the two systems are nearly identical except for the input matrix B which is either a counterclockwise rotation by $\frac{\pi}{16}$ radians ($R_{\pi/16}$) or no rotation (R_0). This indicates that either velocity commands are corrupted (i.e. rotated by $\frac{\pi}{16}$ radians) or not. Corruption of the velocity command can be the result of damage to the drone, software errors, and much more.

The first task for this system is φ_L (16), where

- $\mathcal{X}_T^{(1)} = [1.2, 1.8] \times [1.0, 1.4]$ and
- $\mathcal{X}_T^{(2)} = [0.5, 1.1] \times [1.0, 1.4]$.

In other words, if the drone learns that its control commands are corrupted then it should reach region $\mathcal{X}_T^{(2)}$. If the drone learns that its control commands are not corrupted, then it should reach region $\mathcal{X}_T^{(1)}$.

The optimization in Proposition 7 is used to guarantee that the drone's trajectories satisfy φ_L with $T = 5$. The optimization is solved in 0.38 seconds (with a constraint setup time of 56.34 seconds). The controller successfully satisfies the task both in simulations (see Figure 2), but also in real world experiments (see Figure 1).

C. Accomplish Goal While Hiding Information

For illustrating this task, let us again consider the drone system with the following two modes: $(A_1, B_1, \mathcal{W}_1)$ and $(A_2, B_2, \mathcal{W}_2)$, where $\mathcal{W}_1 = \mathcal{W}_2 = [0.8, 1.2] \times [-0.1, 0.1]$

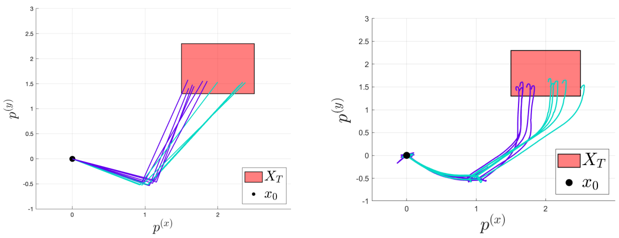

Fig. 3. Ten simulated trajectories of the drone (left) and ten experimental trajectories of the Crazyflie drone (right) when its controlled by the adaptive controller that is the solution to (20) (Section V-B). The solution satisfies task ϕ_H, guaranteeing that any trajectories flowing according to mode 1 (purple) or mode 2 (cyan) will reach the target region (red) while not revealing the mode of the system to outside observers.

$$A_1 = A_2 = B_1 = I \text{ and } B_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

As before, the states are x and y position of the drone and the control inputs are velocities in these directions. The velocity inputs are limited to the set $\mathcal{U} = \{u \in \mathbb{R}^2 \mid \|u\| \leq 2.0\}$ The task is to hide the true mode of the system, i.e., the sign of the actuator in x-direction, from any observer with access to the external behavior $(x_{0:t}, u_{0:t-1})$ while reaching a target region of the state space $\mathcal{X}_T = [1.5, 2.5]^2$. This formally can be stated with the "controller-hidden information" formula φ_H where $\Theta' = \{\theta_1, \theta_2\}$.

Note that the only difference in the two modes is one entry of the gain matrix, B, of each system. This difference and the definition of the task indicate that the adaptive controller must determine how to satisfy the reachability task using one input while also acting conservatively to remain undetectable due to other input. Our algorithm successfully does this after solving the optimization in Proposition 8 in 0.673 seconds with YALMIP (with a constraint setup time of 6.24 seconds). As shown in Figure 3, the controller successfully satisfies the task both in simulation and in hardware experiments.

VII. Conclusion and Future Work

In this paper, we developed optimization-based sufficient conditions for designing adaptive controllers for unknown linear systems that guarantee the system's closed-loop behavior satisfies certain classes of KLTL formula. KLTL's grammar and semantics for unknown linear systems are presented. With these semantics in mind, constraints are defined which enforce that an adaptive, disturbance-feedback controller that uses the consistency estimator satisfies simple and more complex formula templates. Specifically, the constraints on adaptive controllers (and the optimization problems that find such controllers) are presented for the complex "learn, then adapt" and "controller-hidden information" formulas. Instances of these two formulas are then created for a real-world drone system and our method is applied to design controllers that lead to behaviors that satisfy each formula.

In future work, we will consider an expanded form of the KLTL semantics for multi-agent systems. In such multi-agent systems, the knowledge operator is agent-dependent and thus the ability to hide or *share* information is a much more complicated and important task. The information asymmetry in such multi-agent settings makes hiding information feasible more often than in the setting presented in this paper.

References

[1] William Langewiesche. *Fly by wire: the geese, the glide, the miracle on the hudson*. Farrar, Straus and Giroux, 2009.

[2] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. *Formal methods for discrete-time dynamical systems*, volume 15. Springer, 2017.

[3] Yuanqi Mao, Behcet Acikmese, Pierre-Loic Garoche, and Alexandre Chapoutot. Successive convexification for optimal control with signal temporal logic specifications. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '22, 2022.

[4] Chuchu Fan, Zengyi Qin, Umang Mathur, Qiang Ning, Sayan Mitra, and Mahesh Viswanathan. Controller synthesis for linear system with reach-avoid specifications. *IEEE Transactions on Automatic Control*, 67(4):1713–1727, 2021.

[5] Suda Bharadwaj, Rayna Dimitrova, and Ufuk Topcu. Synthesis of surveillance strategies via belief abstraction. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4159–4166. IEEE, 2018.

[6] Aaron D Ames, Paulo Tabuada, Austin Jones, Wen-Loong Ma, Matthias Rungger, Bastian Schürmann, Shishir Kolathaya, and Jessy W Grizzle. First steps toward formal controller synthesis for bipedal robots with experimental implementation. *Nonlinear Analysis: Hybrid Systems*, 25:155–173, 2017.

[7] Juergen E. Ackermann. Parameter space design of robust control systems. *IEEE Transactions on Automatic Control*, 25(6):1058–1072, 1980.

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

[9] Karl J. Åström and Björn Wittenmark. Adaptive control. 1995.

[10] Brett T Lopez, Jean-Jacques E Slotine, and Jonathan P How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3):1031–1036, 2020.

[11] Sadra Sadraddini and Calin Belta. Formal methods for adaptive control of dynamical systems. In *2017 IEEE Conference on Decision and Control (CDC)*, pages 1782–1787, 2017.

[12] Kwesi Rutledge and Necmiye Ozay. Correct-by-construction exploration and exploitation for unknown linear systems using bilinear optimization. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '22, 2022.

[13] Liren Yang and Necmiye Ozay. Fault-tolerant output-feedback path planning with temporal logic constraints. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4032–4039. IEEE, 2018.

[14] Pian Yu and Dimos V Dimarogonas. Hierarchical control for uncertain discrete-time nonlinear systems under signal temporal logic specifications. In *2021 IEEE Conference on Decision and Control (CDC)*, pages 1450–1455. IEEE, 2021.

[15] Olvi Mangasarian. Set containment characterization. *Journal of Global Optimization*, 24(4):473–480, 2002.

[16] Rodica Bozianu, Cătălin Dima, and Emmanuel Filiot. Safraless synthesis for epistemic temporal specifications. In *International Conference on Computer Aided Verification*, pages 441–456. Springer, 2014.

[17] John-Jules Ch Meyer and Wiebe Van Der Hoek. *Epistemic logic for AI and computer science*. Number 41. Cambridge University Press, 2004.

[18] Johan Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *2004 IEEE International Symposium on Computer Aided Control System Design*, Taipei, Taiwan, 2004.

[19] Andrew Wintenberg and Necmiye Ozay. Implicit invariant sets for high-dimensional switched affine systems. In *2020 IEEE Conference on Decision and Control (CDC)*, pages 3291–3297. IEEE, 2020.