Simultaneous Model Identification and Task Satisfaction in the Presence of Temporal Logic Constraints

Sandeep P. Chinchali Scott C. Livingston Marco Pavone Joel W. Burdick

***Abstract*— Recent proliferation of cyber-physical systems, ranging from autonomous cars to nuclear hazard inspection robots, has exposed several challenging research problems on automated fault detection and recovery. This paper considers how recently developed formal synthesis and model verification techniques may be used to automatically generate information-seeking trajectories for anomaly detection. In particular, we consider the problem of how a robot could select its actions so as to maximally disambiguate between different model hypotheses that govern the environment it operates in or its interaction with other agents whose prime motivation is a priori unknown. The identification problem is posed as selection of the most likely model from a set of candidates, where each candidate is an adversarial Markov decision process (MDP) together with a linear temporal logic (LTL) formula that constrains robot-environment interaction. An adversarial MDP is an MDP in which transitions depend on both a (controlled) robot action and an (uncontrolled) adversary action. States are labeled, thus allowing interpretation of satisfaction of LTL formulae, which have a special form admitting satisfaction decisions in bounded time. An example where a robotic car must discern whether neighboring vehicles are following its trajectory for a surveillance operation is used to demonstrate our approach.**

I. Introduction

Autonomous robots programmed to carry out physical tasks must often enter an information gathering mode in order to acquire the physical data needed to complete a task, such as manipulation of a novel object whose properties are a priori unknown. Autonomous robots working in the presence of humans or other autonomous agents must often determine the intent of these agents–are they friendly, adversarial, or neutral? Because of the importance of information seeking behavior to many robotic tasks, there is a growing literature on information theoretic methods to incorporate information acquisition into the planning or action-selection process.

In the domain of mobile robot exploration and SLAM, a number of works have explored how single robots or a team of robots should select their exploring movements so as to optimize a number of exploration criteria [5], [20], [25]. Action selection for industrial assembly robots [6] has also been considered. More recently there has been work on "planning for sensing" in the context of dexterous manipulation [11], [12], [22]. In general, these problems can be seen as a type of system identification [19].

The novel contribution of this paper is the formulation of a class of information-seeking problems in the language of formal systems and model verification theory. In this approach, high level task definitions (or "specifications") are used to automatically construct automata which form a hybrid control system whose design guarantees that the specification is faithfully executed. Formal methods offer the promise of automated contruction of high-level controllers and strong guarantees on performance. Research in computer-aided verification of software and embedded systems has considered problems requiring synthesis of automata from temporal logic formulae for several decades [2], [10], [23], and the hybrid control community has adopted these methods over the last decade [7], [9]. Only recently has this approach gained attention by the robotics community [3], where formal methods have been used to construct sensor-based motion planning strategies [16], including finger-gaiting algorithms [4].

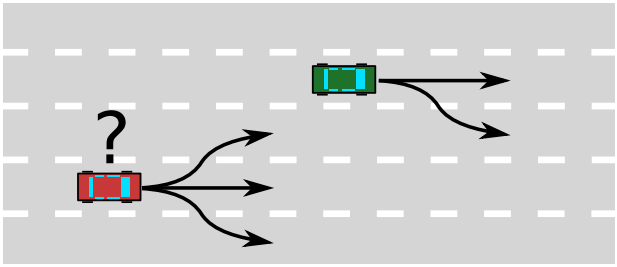

Fig. 1. Illustration of car-following detection scenario. We control the green leading car and want to decide which of a set of models is consistent with motion of the following red car. Temporal logic formulae constrain the follower's response to actions of the leading car.

A significant challenge of applying formal methods in robotics is the prevalence of uncertainty. This includes sensor noise, imperfect actuation, and changes in robot surroundings. There is a rapidly growing literature on treating formal methods amid uncertainty. A well known stochastic model is the Markov decision process, which can be naturally used if probabilistic satisfaction of a task specification is allowed [2]. A class of methods known to work well in practice for constrained and uncertain problems is model predictive control [21]; a similarly motivated method has recently been explored in a temporal logic context by Ding *et al.* [8]. Sarid *et al.* propose a method for synthesizing a reactive robot controller based on task specifications written in terms of classes of objects that may be discovered online while building a map [24]. Wongpiromsarn and Frazzoli model an uncertain environment using switched Markov chains, where, while a model of each chain is known, which one is active is not known [26].

Recent work by Jones, Schwager, and Belta has addressed

S.P. Chinchali and M. Pavone are with Stanford University, Stanford, CA, USA, {csandeep, pavone}@stanford.edu. S.C. Livingston and J.W. Burdick are with the California Institute of Technology, Pasadena, CA, USA, {slivingston@cds, jwb@robotics}.caltech.edu

http://resolver.caltech.edu/CaltechCDSTR:2016.002

synthesis of informative motion plans under temporal logic constraints [14], [15] and anomaly detection from unsupervised learning of formal specifications [13]. Rather than focus on informative motion plans for monitoring tasks, we instead consider control sequences designed to *elicit informative responses* from dynamic environments that may attempt to obfuscate their identities. Such adversarial environments may not immediately expose faults, and the problem of anomaly detection depends on informative "probing" of anomalous behavior to detect faults early and safely.

We study the problem of identifying which of several candidate strategies an environment is following, where each candidate is associated with a temporal logic specification. The overall system specification also governs behaviors that are permissible by the robot and its unknown and possibly adversarial environment, and possibly a task to be completed during the process. We provide an algorithm that formalizes how robots should visit states that distinguish between candidate models, gather observations that disqualify wrong candidates, and lead to information gain while adhering to high-level temporal logic formulae.

This paper is organized as follows. Section II introduces bounded time linear temporal logic (BLTL) as a high-level specification language and adversarial Markov decision processes to capture model uncertainty. Section III provides the problem formulation of model identification coupled with temporal logic constraints and introduces the motivating example of detecting anomalous drivers in robotic navigation. Section IV introduces our key contribution, an algorithm for robotic inference with temporal logic constraints. Section V provides a simulation of the algorithm for autonomous driving. Finally, Section VI summarizes our results and presents directions for future research.

II. Background

In this paper we consider discrete-time systems on finite state spaces. Note that for more general control systems it may be possible to construct a discrete abstraction [1], [18], whence our analysis applies. The primary specification language, which describes properties over finite time horizons, is referred to as bounded linear temporal logic (BLTL) and is defined below. The following running example, which is illustrated in Figure 1, will be used throughout the paper. Consider an autonomous vehicle being followed by other cars on a several-lane highway. These cars can closely pursue the robot, such as in a high-speed car chase, keep their distance to "tail" the robot during surveillance, or simply be benign (incidentally following the target). The robot must safely swerve lanes to avoid obstacles and determine if a pursuant car is tailing it, requiring strict adherence to temporal logic formulae while determining the nature of following cars.

As the above example suggests, robots must often interact with a dynamic environment so as to complete some task. Our development models this interaction in two respects: as a two-player game in which an adversarial strategy is to be identified, and as a Markov decision process that captures stochasticity of transitions among states.

A. Bounded linear-time temporal logic (BLTL)

In this paper we are concerned with modeling interaction between a robot and an adversary over finite durations. While transitions among states are modeled using so-called adversarial MDPs (AMDPs, defined below) (defined below), constraints on sequences of states are expressed using a temporal logic that we now introduce.

Our goal is to describe properties about finite sequences of states. To achieve this, we consider formulae that are built from an operator, $\mathcal{U}_I$, that describes constrained reachability over bounded time durations. The subscript $I = [a, b]$ is a bounded interval on the nonnegative integers $\mathbb{N}$. Note that because every interval I is bounded, every such formula can be decided using a finite sequence of time steps. We refer to formulae described here as bounded LTL, or BLTL.

More precisely, let Π be a finite set of *atomic propositions*. Elements of Π can be regarded as Boolean-valued variables that, at each discrete time, take either the value `True` or `False`. In the example of autonomous highway driving, atomic propositions associated with the robot include $\mathcal{Y} = \{C_m\}_{m=1}^M$ to indicate the lane occupancy of the robot (car) in one of M total lanes, where index m refers to a specific lane. Correspondingly, there are atomic propositions associated with the (adversarial) environment, $\mathcal{X} = \{F_m\}_{m=1}^M$, which indicate the lane occupancy of a following vehicle. The set $\{C_i, F_j\} \subset \Pi$ represents a state where C_i and F_j are both `True` while other atomic propositions are `False`, which have the intuitive interpretation that the robot car occupies lane i while the environment follower occupies lane j.

The syntax of BLTL formulae over interval I is given by the context-free grammar

$$\varphi ::= \texttt{True} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi \mathcal{U}_I \varphi, \qquad (1)$$

where p is an atomic proposition $p \in \Pi$. Notice that (1) is essentially the basic syntax of LTL but with an interval associated with the $\mathcal{U}$ operator.

Before defining the semantics of BLTL, we fix notation about finite strings. Let Σ be a set (or alphabet) over which string concatenation is defined, while Σ^+ denotes the set of all finite strings of length at least 1. For a string $\sigma = \sigma_0\sigma_1\cdots\sigma_n \in \Sigma^+$, a fragment from index i through j, with $i < j$, is written $\sigma_{i:j} = \sigma_i\sigma_{i+1}\cdots\sigma_j$. If $j = n$, i.e., to express the suffix of σ beginning at index i, we may abbreviate the subscript by writing $\sigma_{i:}$.

Take the alphabet to be subsets of atomic propositions, i.e., $\Sigma = 2^\Pi$. Let $t \in \mathbb{N}$. We denote the offset of $I = [a, b]$ from t by $t + I = [a + t, b + t]$. Satisfaction of a BLTL formula φ by a finite sequence $\sigma \in \Sigma^+$ beginning at time $t \in \mathbb{N}$, which is denoted by $\sigma_{t:} \models \varphi$, is defined inductively as follows.

1) $\sigma_{t:} \models \texttt{True}$.
2) for a single atomic proposition $p \in \Pi$, $\sigma_{t:} \models p$ if and only if $p \in \sigma_t$.
3) $\sigma_{t:} \models \neg\varphi$ if and only if $\sigma_{t:} \not\models \varphi$.

4) $\sigma_{t:} \models \varphi_1 \wedge \varphi_2$ if and only if $\sigma_{t:} \models \varphi_1$ and $\sigma_{t:} \models \varphi_2$.
5) $\sigma_{t:} \models \bigcirc \varphi$ if and only if $\sigma_{(t+1):} \models \varphi$.
6) $\sigma_{t:} \models \varphi_1 \mathcal{U}_I \varphi_2$ if and only if there exists some $j \in t + I$ so that $\sigma_{j:} \models \varphi_2$ and for all $i \in (t + I) \cap [0, j]$, $\sigma_{i:} \models \varphi_1$.

If $t = 0$, we also write $\sigma \models \varphi$. Other operators can be derived from those defined above. In particular, $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \implies \psi \equiv \neg\varphi \vee \psi$, $\diamondsuit_I \varphi \equiv \texttt{True}\,\mathcal{U}_I \varphi$ and $\square_I \varphi \equiv \neg\diamondsuit_I \neg\varphi$. Notice that σ must be long enough to observe if a BLTL formula is satisfied. For any BLTL formula φ, denote by $T(\varphi)$ the minimum time at which the satisfaction of φ by any $\sigma \in \Sigma^+$ can be decided. That is, $T(\varphi)$ is constant and associated with each φ. $T(\varphi)$ always exists and is finite. Indeed, it is at most the sum of the upper bounds of all intervals appearing in φ.

BLTL can be used to specify robot tasks over bounded time durations. For example, we can achieve *positional safety* (remain within a set of certain "safe" states) with the operator $\square_I$ and *finite time responses* with the operator $\diamondsuit_I$. In the autonomous driving example, safety formulae require cars to persist in the same lane or move only to an adjacent lane at the next time step. Such an example LTL formula is $\bigwedge_m \square_I \left(C_m \implies \bigcirc \left(F_{m-1} \vee F_m \vee F_{m+1}\right)\right)$.

B. Labeled Markov decision processes

A *labeled Markov chain* $\mathcal{M}$ is a tuple $(S, \mathrm{Init}, \mathbf{P}, \Pi, L)$ where S is a finite set of states, $\mathbf{P} : S \times S \to [0, 1]$ is a transition probability function, Init is a probability mass function over S, and $L : S \to 2^\Pi$ is a labeling that assigns subsets of atomic propositions in set Π to each state. The transition probabilities from a state s to subsequent states s' satisfy $\Sigma_{s' \in S} P(s, s') = 1$.

A *labeled Markov decision process* (MDP) is a tuple $(S, \mathrm{Init}, \mathrm{Act}, \mathbf{P}, \Pi, L)$ where S, Init, Π, and L are as for Markov chains, Act is a finite set of actions, and $\mathbf{P} : S \times \mathrm{Act} \times S \to [0, 1]$ is a transition probability function.

As defined by Wongpiromsarn and Frazzoli [26], an *adversarial MDP* (AMDP) $\mathcal{M}$ is a tuple $(S, \mathrm{Init}, \mathrm{Act}^c, \mathrm{Act}^u, \mathbf{P}, \Pi, L)$, where S is a finite set of states, Init is a probability mass function over S, Act^c is a mapping from states into sets of *controlled actions*, Act^u is a mapping into sets of *uncontrolled actions* (or *adversarial actions*), $L : S \to 2^\Pi$ is a labelling function from states to atomic propositions, and $\mathbf{P} : S \times \mathrm{Act}^c \times \mathrm{Act}^u \times S \to [0, 1]$ defines transition probabilities where for each state $s \in S$, $a \in \mathrm{Act}^c(s)$, and $b \in \mathrm{Act}^u(s)$, $\sum_{s' \in S} \mathbf{P}(s, a, b, s') = 1$. For each state $s \in S$, the controlled action $a \in \mathrm{Act}^c(S)$ (respectively, adversarial action $b \in \mathrm{Act}^u(S)$) is said to be *enabled at* s if $a \in \mathrm{Act}^c(s)$ (respectively, $b \in \mathrm{Act}^u(s)$).

C. BLTL satisfaction on AMDPs

We now define the reactive synthesis problem for BLTL formulae on adversarial MDPs. Recall from the previous section (Section II-B) that there is a joint sequence of states that arise from policies (or strategies) for action selection by the robot and the adversary. Each state in the sequence is labeled with a set of atomic propositions from Π. Since there is only one set of atomic propositions, our treatment may superficially appear different from the original setting of reactive synthesis for LTL proposed in [23], in which there are distinct controlled and uncontrolled variables. However, observe that we recover the usual setting of reactive synthesis by partitioning the set of atomic propositions and constructing an AMDP so that control actions Act^c can only affect values of one set of atomic propositions, and that adversarial actions Act^u can only affect values of the other set.

Our development follows that of [26] except that we consider time-bounded properties, as defined in Section II-A. Let φ be a BLTL formula over the set of atomic propositions Π, and let $\mathcal{M}$ be an adversarial MDP with states labeled by subsets of the same set of atomic propositions Π. A finite sequence of states $s_0 \cdots s_t$ of $\mathcal{M}$ is said to *satisfy* φ, denoted by $s_0 \cdots s_t \models \varphi$, if $L(s_0) \cdots L(s_t) \models \varphi$, where L is the labeling of $\mathcal{M}$ that associates sets of atomic propositions with states. Thus, through the labeling we are able to decide satisfaction as defined in Section II-A.

We denote the set of finite state sequences of length at least one to be S^+. Let $\mathcal{C} : S^+ \to \mathrm{Act}^c$ be a control policy, and let $\mathcal{E} : S^+ \to \mathrm{Act}^u$ be an adversary policy. From a state s of the adversarial MDP, the probability of satisfying φ is defined as

$$\Pr_{\mathcal{M}}^{\mathcal{C},\mathcal{E}}(s \models \varphi) = \sum_{s_0 \cdots s_{T(\varphi)} \in \mathrm{Sat}_s^{T(\varphi)}(\varphi)} \prod_{t=0}^{T(\varphi)-1} \mathbf{P}^{\mathcal{C},\mathcal{E}}(s_0, \ldots, s_{t+1}), \quad (2)$$

where we define the set of state sequences of length $T(\varphi)$ that begin at s and that satisfy φ as

$$\mathrm{Sat}_s^{T(\varphi)}(\varphi) = \Big\{ s_0 \cdots s_{T(\varphi)} \in S^{T(\varphi)} \mid s_0 = s \wedge s_0 \cdots s_{T(\varphi)} \models \varphi \Big\}, \quad (3)$$

and for any sequence of states $s_0 \cdots s_t$, we define

$$\mathbf{P}^{\mathcal{C},\mathcal{E}}(s_0, \ldots, s_{t+1}) = \mathbf{P}(s_t, \mathcal{C}(s_0 \cdots s_t), \mathcal{E}(s_0 \cdots s_t), s_{t+1}). \quad (4)$$

Intuitively, $\mathbf{P}^{\mathcal{C},\mathcal{E}}$ is just the transition probability function $\mathbf{P}$ of $\mathcal{M}$ during application of the control policy $\mathcal{C}$ and the adversary policy $\mathcal{E}$. Recall from Section II-A that $T(\varphi)$ is a sufficient length to decide satisfaction of φ by a sequence of sets of atomic propositions, hence it is also a sufficient length for sequences of states in the MDP.

III. Problem formulation

In this section we formulate the problem we wish to solve in this paper. We first define the set of candidate environment models the robot must disambiguate and how the robot interacts with them, and key assumptions on their interaction.

A. System - environment interaction model

The interaction between a controlled system and adversarial environment is modelled as an adversarial MDP, as defined in Section II-B, with an associated temporal logic

formula the interaction must adhere to. We consider a set of such adversarial MDPs corresponding to possible environmental models. Henceforth, a candidate adversarial MDP, or candidate MDP, refers to the interaction between the system and a possible environment model as an adversarial MDP.

Consider a set

$$\mathbf{M} = \{(\mathcal{M}_i, \varphi_i)\}_{i=1}^{N} \tag{5}$$

of N candidate environment models. Each model $(\mathcal{M}_i, \varphi_i)$, for $1 \leq i \leq N$, is a pair consisting of an adversarial MDP $\mathcal{M}_i$ and temporal logic specification φ_i. The formula is of a special form of LTL that describes interaction using BLTL subformulae. Precisely, φ_i is

$$\bigwedge_r \Box \left(\psi_i^{\mathrm{req},r} \implies \psi_i^{\mathrm{res},r}\right), \tag{6}$$

where $\psi_i^{\mathrm{req},r}$ and $\psi_i^{\mathrm{res},r}$ are both BLTL formulae, and $\Box$ is the "always" LTL operator. Often, $\psi_i^{\mathrm{req},r}$ is a system control sequence that guarantees a response from an environment in the form $\psi_i^{\mathrm{res},r}$. In the pursuant car example, we take $\psi^{\mathrm{req},1} = C_1$ and $\psi^{\mathrm{res},1} = \diamondsuit_{I_{[0,2]}} F_1$, which specifies that, if the robot car enters lane 1, it will observe the environment car follow it to lane 1 within 2 time steps. The precise syntax and semantics of LTL are not needed here; it is enough to know the BLTL preliminaries of Section II-A and to interpret (6) as requiring that the subformula

$$\psi_i^{\mathrm{req},r} \implies \psi_i^{\mathrm{res},r}$$

is satisfied at every time step, for all r. The motivation for this assumed form of interaction formulae is that it allows us to solve the problem through repeated finite-duration interactions. The robot can probe for a particular behavior r of model i by satisfying $\psi_i^{\mathrm{req},r}$, from where we expect a trajectory satisfying $\psi_i^{\mathrm{res},r}$ in response, provided that model i is the ground-truth.

The adversarial MDP for model $(\mathcal{M}_i, \varphi_i)$ is defined as $\mathcal{M}_i = (S, \mathrm{Init}, \mathrm{Act}^c, \mathrm{Act}^u, \mathbf{P}_i, \Pi, L)$. The adversarial MDPs for all models i share the same state set S and initial state distribution Init, set of atomic propositions Π, labeling function $L : S \to 2^{\Pi}$, and system (controller) and environment (adversary) action sets Act^c and Act^u. However, as indicated by the subscripts, the transition probabilities $\mathbf{P}_i$ may be different. The assumption of common states, initial distributions, adversarial actions, and atomic propositions, and labellings is without loss of generality because any differences only render easier the identification problem, which is stated below in Section III-E.

A candidate model $(\mathcal{M}_i, \varphi_i)$ is said to be *self-consistent* if, for each $s \in S$ and for any robot controller $\mathcal{C}$, there is some adversarial policy $\mathcal{E}$ such that

$$\Pr_{\mathcal{M}_i}^{\mathcal{C},\mathcal{E}}(s \models \varphi_i) = 1.$$

The key intuition behind *self-consistency* is, if model $\mathcal{M}_i$ is indeed the true model, the environment must always enact a strategy that is concordant with the true model's specification φ_i. Self-consistency is a natural requirement of systems that satisfy high-level constraints in that environment assumptions for the true model encoded in φ_i must be adhered to in order to allow guarantees about interaction with the true model.

B. Distinguishability

Let $\mathbf{M} = \{(\mathcal{M}_i, \varphi_i)\}_{i=1}^{N}$ be a set of candidate models. For an LTL formula φ, $\mathcal{L}(\varphi)$ denotes the set of all sequences that satisfy φ. The models $(\mathcal{M}_i, \varphi_i)$ and $(\mathcal{M}_j, \varphi_j)$ are said to be *interactively distinguishable* if $\mathcal{L}(\varphi_i) \neq \mathcal{L}(\varphi_j)$. Let $s \in S$. The models are said to be *distinguishable in probability from* s if

$$\min_{\mathcal{C}} \max_{\mathcal{E}} \Pr_{\mathcal{M}_i}^{\mathcal{C},\mathcal{E}}(s \models \varphi_i \wedge \varphi_j) < 1 \quad \vee \quad \min_{\mathcal{C}} \max_{\mathcal{E}} \Pr_{\mathcal{M}_j}^{\mathcal{C},\mathcal{E}}(s \models \varphi_i \wedge \varphi_j) < 1. \tag{7}$$

If the property holds for all s, then the models are said to be *distinguishable in probability*.

Intuitively, being distinguishable in probability from some state s ensures that a control policy exists such that the adversary cannot simultaneously satisfy both candidate models with probability 1. Accordingly, if we can repeatedly reach s and repeatedly play an appropriate policy, one of the models must eventually be demonstrated as false.

C. Assumptions

We assume a finite set of self-consistent models $\mathbf{M} = \{(\mathcal{M}_i, \varphi_i)\}_{i=1}^{N}$ and fully observable system and robot states. The key assumption that facilitates model identification is that for each adversarial MDP $\mathcal{M}_i$ and for models $i \neq j$, there is a state s such that $(\mathcal{M}_i, \varphi_i)$ and $(\mathcal{M}_j, \varphi_j)$ are distinguishable in probability from s.

Such an assumption is justified because if models are identical or are not perceivably different from any state, purposeful robot motion towards distinguishing states s cannot be made for model identification. Further, we assume each adversarial MDP $\mathcal{M}_i$ has no transient states, namely that it is possible with positive probability to reach every state from every other state. Such an assumption is necessary to preclude scenarios where the environment always prohibits the controller from reaching a distinguishing state from which it can differentiate models, in which case model identification also cannot occur. Relaxation of such assumptions, including addressing Hidden Markov Models (HMMs) and randomized policies, is discussed in Section VI.

D. Belief distribution over models

The robot must identify the true model from other competing candidate models by judicious choice of information-seeking control actions in its system strategy or policy $\mathcal{C} : S^+ \to \mathrm{Act}^c$. Such actions are control inputs that elicit responses from the environment model and exploit transition probabilities for the MDPs to reach states that disambiguate models.

The robot can differentiate environment models by estimating the likelihood that gathered observations came from the interaction between the robot and a possible environment model. Since observations may be consistent with many

candidate models, we indicate the robot's current certainty at time t about the true model identity in a belief distribution. The belief distribution is a probability distribution over candidate models that evolves with time t as new observations are generated. Given a sequence of states, controlled actions, and adversarial actions, $s_0 a_0 b_0 s_1 a_1 b_1 s_2 \cdots s_T$, the belief for model i is updated as

$$B_i \leftarrow \mathbf{P}_i\left(s_0, a_0, b_0, s_1\right) \cdots \mathbf{P}_i\left(s_{T-1}, a_{T-1}, b_{T-1}, s_T\right) B_i \eta,$$

where η is a normalization factor to ensure that $\sum_j B_j = 1$.

E. Problem statement

A *control policy* for the robot is a mapping from finite sequences of states to actions, $\mathcal{C} : S^+ \to \mathrm{Act}^c$. Without loss of generality, we refer to the ground truth model as M_1 and assume it is one of the candidate models.

Problem 1: **Given** a set of self-consistent candidate models $\mathbf{M} = \{(\mathcal{M}_i, \varphi_i)\}_{i=1}^N$, where each adversarial MDP $\mathcal{M}_i$ has no transient states and where for $i \neq j$, there is a state s such that $(\mathcal{M}_i, \varphi_i)$ and $(\mathcal{M}_j, \varphi_j)$ are distinguishable in probability from s, and a tolerance ε for wrong models, **find** a control policy such the ground truth model is found with tolerance ε, i.e., the belief vector element $B_i \geq 1 - \varepsilon$, where $(\mathcal{M}_i, \varphi_i)$ is the ground truth.

F. Example: detecting pursuers in robotic navigation

We return to the running example of a robot vehicle being followed by other cars. These cars can closely pursue the robot, such as in a high-speed car chase, keep their distance to "tail" the robot during surveillance, or simply be a benign car that happens to follow the target by chance. In this paper, we perform numerical experiments for this example using the set of candidate models $\mathbf{M} = \{(\mathcal{M}_{\text{pursue}}, \varphi_{\text{pursue}}), (\mathcal{M}_{\mathbf{k}-\text{bound}}, \varphi_{\mathbf{k}-\text{bound}}), (\mathcal{M}_{\text{benign}}, \varphi_{\text{benign}})\}$ with $N = 3$ models.

Using the notation from the previous section, system variables $\mathcal{Y} = \{C_1, \ldots, C_M\}$ indicate the lane occupancy of the controlled robot car in one of M total lanes. Corresponding environment variables $\mathcal{X} = \{F_1, \ldots, F_M\}$ indicate the lane occupancy of a following vehicle. A state of the system at discrete time t is the lane occupancy of the robot and follower vehicle $s_t = (C_{m,t} F_{m,t})$.

The robot must enact a strategy $\mathcal{C}$ (choose and follow a trajectory) to move between lanes in response to the follower's lane choices F_m. This strategy $\mathcal{C}$ must allow the robot to best determine the follower's identity as `pursuant`, k−`bound` , or `benign`.

The candidate models have interaction formulae as follows.

1) $\varphi_{\text{benign}} = \texttt{True}$, i.e., the benign has no temporal logic constraint.
2) $\varphi_{k-\text{bound}}$ is
$$\bigwedge_r \Box\left(C_r \implies \Diamond_{I_k} F_{r-k} \vee F_{r-k+1} \vee \cdots \vee F_{r+k}\right).$$
3) $\varphi_{\text{pursue}} = \bigwedge_r \Box\left(C_r \implies \Diamond_{I_k} F_r\right)$

The k−`bound` car must be within k lanes of the robot car at all times and can only move one lane in each discrete timestep. BLTL is well-suited to describe the k−`bound` behavior since choice of a k length interval, denoted by I_k, indicates how quickly the follower must enter the same lane as the car. The requirement that the car must be within k lanes, coupled with single lane moves per timestep, imposes that the follower has k timesteps to reach the robotic car's lane. Specifically, response-request formula $\varphi = (C_m \implies \Diamond_{I_k} F_m)$ encodes that a system request (the car being in lane m or C_m) will necessitate a response where the follower reaches lane m within k steps. We simulated with $k = 1$ lanes. Note that the "pursuer" is a k−`bound` car with $k = 0$.

IV. Algorithm for interaction with bounded LTL constraints

A. Description

We first establish notation used in the algorithms. Let f and g be two probability mass functions over the same sample space. Denote the Kullback-Leibler (KL) divergence [17] between them by $\mathcal{D}(f, g)$. Given an index i, the set of indices excluding i is denoted by

$$-i = \{1, 2, \ldots, n\} \setminus \{i\}.$$

For a state s, reference model $(\mathcal{M}_i, \varphi_i)$, and control policy $\mathcal{C}$, define `InfoGain` for a robot action a

$$\min_{b \in \mathrm{Act}^u} \sum_{j \in -i} \mathcal{D}(f_i^{s,a,b}, f_j^{s,a,b}) \tag{8}$$

where, for control and adversarial actions a and b, each $f_j^{s,a,b}(\cdot)$ is a probability mass function over the set of MDP states S based on the transition probability of model j, i.e.,

$$f_j^{s,a,b}(s') = \mathbf{P}_j(s, a, b, s')$$

for $s' \in S$.

We now present Algorithm 1 that chooses system control actions to maximally disambiguate models. Beginning with a set of candidate models and uniform initial belief distribution, the robot assumes the most likely model i (line 6) with the goal of gathering further evidence to substantiate model i. The robot considers possible models j with non-zero belief elements in the set `Candidates` (line 7). From such candidates, the robot tries to find another model j for which the assumed model i and model j are distinguishable in probability from state s in the hope of observing a state sequence that disqualifies candidate model j by violating its specification φ_j. The min-max formulation in line 9 indicates that the environment may choose an adversarial controller $\mathcal{E}$ to "obfuscate" or minimize the observed deviation between models but the system must choose the most informative controller $\mathcal{C}^*$ to embark on a finite horizon trajectory from state s that is least likely to be concordant with assumed model i and other competing model j. If such a candidate model j cannot be found in the hopes of violating its specification φ_j, the robot chooses the optimal controller $\mathcal{C}^*$ that maximizes the resulting difference in state distributions measured in terms of the Kullback-Leibler (KL) divergence

Algorithm 1 Bounded LTL interaction

Require: Set of candidate models $\mathbf{M} = \{(\mathcal{M}_i, \varphi_i)\}_{i=1}^N$, wrong model probability tolerance ε

1: $T := \max_j T(\varphi_j)$
2: Initialize at state s
3: $t := 0$ // Start time
4: $B := (\frac{1}{N}, \ldots, \frac{1}{N})$ // Uniform initial belief vector
5: **repeat**
6: $\quad i := \text{rand}\arg\max_j B_j$ // Randomly break ties
7: $\quad \text{Candidates} := \{j \mid B_j > 0\}$
8: $\quad$ **if** for some $j \in \text{Candidates}$, $(\mathcal{M}_j, \varphi_j)$ and $(\mathcal{M}_i, \varphi_i)$ are distinguishable in probability from s **then**
9: $\quad\quad \mathcal{C}^* := \arg\min_{\mathcal{C}} \max_{\mathcal{E}} \Pr_{\mathcal{M}_j}^{\mathcal{C},\mathcal{E}}(s \models \varphi_i \wedge \varphi_j)$
10: $\quad$ **else**
11: $\quad\quad \mathcal{C}^* := \texttt{InfoGain}(s, B, \{\mathcal{M}_1, \ldots, \mathcal{M}_N\})$
12: $\quad$ **end if**
13: $\quad$ Apply controller $\mathcal{C}^*$.
14: $\quad$ Receive resulting sequence of states $s_1 \cdots s_T$ and observe adversary actions $b_0, \ldots b_{T-1}$
15: $\quad$ For each j, if $s_0 \cdots s_T \not\models \varphi_j$, then $B_j := 0$
16: $\quad$ Update belief vector using transition probabilities for $(s_0, \mathcal{C}^*(s_0), b_0, s_1), (s_1, \mathcal{C}^*(s_0 s_1), b_1, s_2), \ldots$
17: $\quad$ Normalize belief vector so that $\sum_{j=1}^N B_j = 1$
18: $\quad t := t + 1$; $s := s_T$
19: **until** for some model $(\mathcal{M}_i, \varphi_i)$, $B_i \geq 1 - \varepsilon$

to enable maximum information gain (line 11). We note one example of the `InfoGain` function in terms of the KL divergence but our algorithm is general and other measures of information gain are possible. The novelty of our approach lies in the exploitation of distinguishing states based on LTL formulae.

Once the policy given by optimal controller $\mathcal{C}^*$ is enacted and observations $b_0 \ldots b_{T-1}$ are gathered, the robot can immediately discount models j where the observed transitions do not satisfy the model's BLTL specification φ_j (line 15). Such transitions that violate BLTL specifications are inherently encoded as zero probability transitions in the model's adversarial MDP. Then, Bayesian updates of the belief vector given the observations guide the choice of the most likely model at the next algorithm iteration (line 16).

Algorithm 2 is a greedy version of Algorithm 1 with single timestep iterations and where temporal logic formulae are disregarded in the selection of control actions – potentially allowing for much faster computation times. At each iteration, the robot enacts a greedy decision to enact a single, optimal control action a^*, that regardless of adversarial environmental action b, maximizes the KL divergence between distributions over next s' between model i and competing models j. We note that in the single timestep case, an example of the `InfoGain` routine is to maximize the cumulative KL divergence under assumed model i as opposed to other models j.

Algorithm 2 Greedy one-step interaction

Require: Wrong model probability tolerance ε

1: Initialize at state s
2: $t := 0$ // Start time
3: $B := (\frac{1}{N}, \ldots, \frac{1}{N})$ // Uniform initial belief vector
4: **repeat**
5: $\quad i := \text{rand}\arg\max_j B_j$ // Randomly break ties
6: $\quad a^* := \arg\max_{a \in \text{Act}^c} \min_{b \in \text{Act}^u} \sum_{j \in -i} \mathcal{D}(f_i^{s,a,b}, f_j^{s,a,b})$
7: $\quad$ Apply controller action a^*
8: $\quad$ Receive resulting new state s'
9: $\quad$ Update belief vector using transition probabilities for (s, a, b, s')
10: $\quad$ Normalize belief vector so that $\sum_{j=1}^N B_j = 1$
11: $\quad t := t + 1$; $s := s'$
12: **until** for some model $\mathcal{M}_i$, $B_i \geq 1 - \varepsilon$

B. Correctness analysis

In this section, we present a correctness result about Algorithm 1 that provides conditions for convergence to the ground-truth model when the candidate models are mutually distinguishable in probability. In this case, we take $\epsilon = 0$ because violating traces, i.e., finite sequences of states, will eventually occur, causing corresponding elements of the belief vector to be assigned exactly 0 (at line 15).

Theorem 1: Let $\mathbf{M} = \{(\mathcal{M}_i, \varphi_i)\}_{i=1}^N$ be a set of self-consistent candidate models such that for each pair (i, j), $(\mathcal{M}_i, \varphi_i)$ and $(\mathcal{M}_j, \varphi_j)$ are distinguishable in probability. Suppose that the ground-truth model is in $\mathbf{M}$ (but unknown). If $\varepsilon = 0$, then Algorithm 1 will terminate with probability 1 and return the ground-truth model.

Proof: In each iteration, line 17 ensures that the elements of B sum to 1 before the exit condition $B_i \geq 1 - \varepsilon$ (line 19) is reached. Thus, if $\varepsilon = 0$, the exit condition can be rewritten as $B_i = 1$ for some i. Consider an iteration of the loop in Algorithm 1 (lines 5–19). Let i be selected as in line 6 of Algorithm 1. From the exit condition on line 19, it follows that a loop-invariant is $|\text{Candidates}| \geq 2$. By hypothesis each pair of candidate models is distinguishable in probability, and therefore the condition on line 8 must be satisfied. Let C^* be selected according to line 9. From the definition of distinguishable in probability (see (7)), we have for any adversarial policy $\mathcal{E}$,

$$\Pr_{\mathcal{M}_i}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_i \wedge \varphi_j) < 1 \vee \Pr_{\mathcal{M}_j}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_i \wedge \varphi_j) < 1.$$

Suppose the left disjunct, having probability with respect to $\mathcal{M}_i$ holds, i.e., $\Pr_{\mathcal{M}_i}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_i \wedge \varphi_j) < 1$. The case where the right disjunct holds is entirely similar. Hence,

$$\Pr_{\mathcal{M}_i}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_i) < 1 \quad \vee \quad \Pr_{\mathcal{M}_i}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_j) < 1. \tag{9}$$

There are now two cases. First, if $(\mathcal{M}_i, \varphi_i)$ is the ground-truth, then the adversary must play such that $\Pr_{\mathcal{M}_i}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_i) = 1$, which is possible by the assumption of self-consistency. Thus the other disjunct in (9) must be true, i.e.,

$$\Pr_{\mathcal{M}_i}^{\mathcal{C}^*,\mathcal{E}}(s \models \varphi_j) < 1. \tag{10}$$

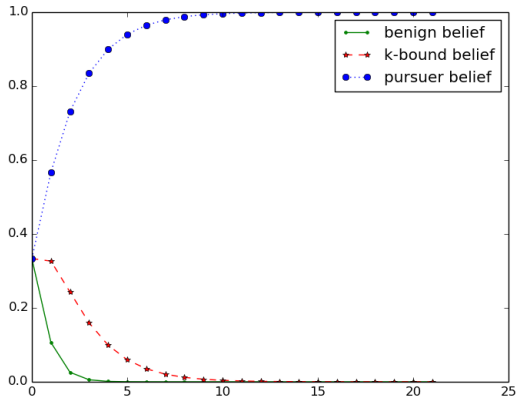

Fig. 2. Trial in which the `pursuant` car model is the ground truth. Simulation is with 5 lanes. Notice that the mass in the belief vector quickly accumulates on the correct model, and the benign model is dismissed due to violation within a few trials.

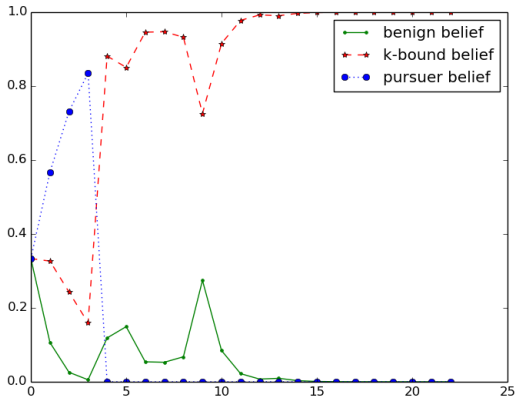

Fig. 3. Trial in which the k-`bound` car model is the ground truth, where $k = 1$. Simulation is with 5 lanes.

Second, if $(\mathcal{M}_i, \varphi_i)$ is not the ground-truth, then we may conservatively suppose that the adversary plays so as to ensure that every φ_j is satisfied, and thus that no update occurs on line 15. From line 6, it follows that the first case, i.e., where i is the ground-truth, will occur infinitely often because it will always have corresponding element of the belief vector as strictly positive. Because there are finitely many candidate models, at least one of them, say j, will be chosen infinitely often. Let p be the probability on the left-side of the inequality in (10). After n iterations of choosing model $(\mathcal{M}_j, \varphi_j)$, the probability that φ_j is satisfied in all iterations is p^n. Since $p < 1$, $\lim_{n\to\infty} p^n = 0$, i.e., the event of all iterations having state sequences that satisfy φ_j has probability 0. Therefore, with probability 1 there is an iteration in which the resulting state sequence does not satisfy φ_j. Upon that iteration, at line 15, $B_j = 0$, and therefore the candidate model $(\mathcal{M}_j, \varphi_j)$ cannot be in Candidates again. Because there are finitely many candidate models, the result follows. ■

We note that Algorithm 1 can also handle the case when models are not distinguishable in probability. In this case, the `InfoGain` subroutine on line 11 would select control actions to maximally differentiate models. The analysis and correctness proof for this case is the subject of future work.

V. Numerical Experiments

We return to the pursuant car example from Section III-F to illustrate that Algorithm 1 can deduce how the robot car must change lanes to distinguish pursuers. The simulations provide intuition to understand the proof in Section IV-B.

In our implementation of Algorithm 1, we take as input BLTL candidate models and an environment simulator for the ground truth model which selects among permissible moves (that respect the BLTL specifications)

The adversarial MDP in this example is formed as a product of two MDPs, each having state space that is just a chain graph where each state represents a lane of the road. There are two agents: the robot and another car. Intuitively, we want to construct a useful description about whether the car is following the robot, and if so, to find a concise model of the behavior that is amenable to other control synthesis methods. Here, deciding qualitatively how the other car (the environment) is moving can be thought of as "fault detection." For each agent, from every position there are three possible actions: stay-in-current-lane, move left, and move right. Associated with each agent is a parameter $p \in [0, 1]$ that provides the probability that the requested action fails. If attempting to stay in the current lane, with probability $\frac{p}{2}$ the outcome is to move into the lane to the left of the current one, and with probability $\frac{p}{2}$ the outcome is to move into the right lane. If attempting to move to the left, with probability p the outcome is to stay in the current lane. If attempting to go right, with probability p the outcome is to stay in the current lane.

If the agent is in the left-most (respectively, right-most) lane, then the action to move to the left (respectively, to the right) always (i.e., with probability 1) results in the agent staying in its current lane. In the left-most lane (respectively, right-most) lane, the action to stay in the current lane fails with probability $\frac{p}{2}$, having the outcome to move right (respectively, left). This assumption causes the action sets Act^c and Act^u to be the same at every state.

We ran the simulation on a road with 5 lanes. The value of the failure probability varied depending on the candidate MDP.

We use $p = 0.5$ for the benign car, $p = 0.2$ for the k-bound car, and $p = 0.0$ for the pursuant car to describe a scenario where the pursuer follows the robotic car without failure. It is important to observe that, while it is clear that $\varphi_{\text{benign}} = \texttt{True}$ permits emulation of any other candidate model, the failure probability ensures that the models can be distinguished.

A trial in which the ground-truth is the pursuant car is shown in Figure 2. The pursuant car has very restrictive behavior since it must always follow the robot, allowing it

to quickly be differentiated from a benign car, explaining the extremely quick, monotonic convergence in belief illustrated in Figure 2. A trial in which the ground-truth is the k-bound car with $k = 1$ is shown in Figure 3. Since the k-bound car need not immediately pursue the robot, it can be harder to distinguish from other models, explaining the slower convergence in Figure 3.

VI. Conclusions and Future Work

A general problem has been posed for identification of a model described by both dynamics and a temporal logic specification. A practical method for purposeful robot movement toward distinguishing states—all while satisfying feasible candidate task specifications—was presented and its convergence proven for a class of probabilistic environments.

In future work, we plan to address the optimality of our algorithm and consider variations featuring robust MDPs and chance-constrained MDPs. Further, we will extend the problem to the case of hidden environment states, necessitating the use of Hidden Markov Models (HMMs), and also consider randomized strategies.

To address the less restrictive problem settings described above, we will consider several heuristics and algorithm optimality/time-complexity. Even in the current algorithm, one can employ heuristics to hasten convergence. For example, in line 8 of Algorithm 1, if several candidate models j exist, we can select the controller which minimizes the probability in line 9 over all contending models. We can also exploit the bounded nature of BLTL to select the candidate j which minimizes the time $T(\varphi_i)$ or $T(\varphi_j)$ required to distinguish between models i and j and only apply the controller for this minimal duration in line 13.

Our work is a small but exciting step towards the ambitious goal of automated fault detection and recovery for cyber-physical systems. Reaching this goal may one day allow, for example, detection of hacked cars in autonomous transportation networks or compromised drones in networks of aerial vehicles. Our work makes the valuable insight that increasingly complex cyber-physical systems may not immediately exhibit key faults, requiring future robots to synthesize informative controllers that dynamically interact with adversarial environments to expose anomalies.

Acknowledgments

This work was partially supported by United Technologies Corporation and IBM, through the industrial cyberphysical systems (iCyPhy) consortium, and by NASA under the Space Technology Research Grants Program, Grant NNX12AQ43G. S.P Chinchali is supported by a Stanford Graduate Fellowship and National Science Foundation Fellowship. The authors thank Richard M. Murray for comments on an early draft.

References

[1] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proc. of the IEEE*, 88(7):971–984, July 2000.

[2] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[3] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas. Symbolic planning and control of robot motion: Finding the missing pieces of current methods and ideas. *IEEE Robotics & Automation Magazine*, pages 61–70, March 2007.

[4] S. Chinchali, S. C. Livingston, U. Topcu, J. W. Burdick, and R. M. Murray. Towards formal synthesis of reactive controllers for dexterous robotic manipulation. In *Proc. of International Conference on Robotics and Automation (ICRA)*, pages 5183–5189, Saint Paul, Minnesota, USA, May 2012.

[5] T. H. Chung and J. W. Burdick. A decision-making framework for control strategies in probabilistic search. In *Proc. of International Conference on Robotics and Automation (ICRA)*, pages 4386–4393, Roma, Italy, April 2007.

[6] R. Cole and C. K. Yap. Shape from probing. *Journal of Algorithms*, 8:19–38, March 1987.

[7] J. M. Davoren and A. Nerode. Logics for hybrid systems. *Proc. of the IEEE*, 88(7):985–1010, July 2000.

[8] X. C. Ding, C. Belta, and C. G. Cassandras. Receding horizon surveillance with temporal logic specifications. In *49th IEEE Conf. on Decision and Control (CDC)*, pages 256–261, December 2010.

[9] G. E. Fainekos, A. Girard, and G. J. Pappas. Hierarchical synthesis of hybrid controllers from temporal logic specifications. In *Proc. of Hybrid Systems: Computation and Control (HSCC)*, 2007.

[10] E. Grädel, W. Thomas, and T. W. (Eds.). *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[11] P. Hebert, J. W. Burdick, T. Howard, N. Hudson, and J. Ma. Action inference: the next best touch. In *Proc. of ICRA, submitted*, 2013.

[12] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez. Task-driven tactile exploration. In *Proc. of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[13] A. Jones, Z. Kong, and C. Belta. Anomaly detection in cyber-physical systems: A formal methods approach. In *Proc. of the 53rd IEEE Conference on Decision and Control (CDC)*, pages 848–853, 2014.

[14] A. Jones, M. Schwager, and C. Belta. A receding horizon algorithm for informative path planning with temporal logic constraints. In *Proc. of 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5019–5024. IEEE, 2013.

[15] A. Jones, M. Schwager, and C. Belta. Information-guided persistent monitoring under temporal logic constraints. In *American Control Conference (ACC), 2015*, pages 1911–1916. IEEE, 2015.

[16] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal logic-based reactive mission and motion planning. *IEEE Trans. on Robotics*, 25(6):1370–1381, 2009.

[17] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, pages 79–86, 1951.

[18] J. Liu and N. Ozay. Abstraction, discretization, and robustness in temporal logic control of dynamical systems. In *Proceedings of Hybrid Systems: Computation and Control (HSCC)*, pages 293–302, April 2014.

[19] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.

[20] J. Ma. *Real-Time Applications of 3D Object Detection and Tracking*. PhD thesis, Caltech, Pasadena, CA, USA, 2010.

[21] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36:789–814, 2000.

[22] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake. Simultaneous localization and grasping as a belief space control problem. In *15th Int. Symposium on Robotics Research (ISRR)*, Flagstaff, Arizona, USA, August 28–September 1 2011.

[23] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '89, pages 179–190, New York, NY, USA, 1989. ACM.

[24] S. Sarid, B. Xu, and H. Kress-Gazit. Guaranteeing high-level behaviors while exploring partially known maps. In *Proc. of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[25] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2006.

[26] T. Wongpiromsarn and E. Frazzoli. Control of probabilistic systems under dynamic, partially known environments with temporal logic specifications. In *Proc. of CDC*, pages 7644–7651, December 2012.