Distribution Temporal Logic: Combining Correctness with Quality of Estimation

Austin Jones, Mac Schwager, Calin Belta

***Abstract*— We present a new temporal logic called Distribution Temporal Logic (DTL) defined over predicates of belief states and hidden states of partially observable systems. DTL can express properties involving uncertainty and likelihood that cannot be described by existing logics. A co-safe formulation of DTL is defined and algorithmic procedures are given for monitoring executions of a partially observable Markov decision process with respect to such formulae. A simulation case study of a rescue robotics application outlines our approach.**

I. Introduction

Temporal logics (TLs) provide a rigorous framework for describing complex, temporally ordered tasks for dynamical systems. Temporal logic formulae can be used to describe relevant properties such as safety (“Always avoid collisions”), reliability (“Recharge infinitely often”), or achievement (“eventually reach destination”) [?]. In this work, we define Distribution Temporal Logic (DTL), a new kind of TL for specifying tasks for stochastic systems with partial state information. The logic is well-suited to problems in which the uncertainty of an on-line state estimate is significant and unavoidable. Many such systems arise in robotics applications, where a robot may be uncertain of, for example, its own position in its environment, the location of objects in its environment, or the classification of objects.

We represent the system as a Partially Observable Markov Decision Process (POMDP) [?], whose state is described by the current probability distribution over the hidden state of the system (the *belief state*). We define DTL over properties of belief states as well as hidden states. With DTL, we can describe such tasks as “Measure the system state until estimate variance is less than v while minimizing the probability of entering a failure mode” or “If the most likely card to be drawn next is an Ace, increase your bet”. DTL is a promising framework for high-level tasks over POMDPs as it can be used to describe the value of taking observations as well as describe complex tasks defined over the hidden states of the system.

Current research on temporal logic specifications for dynamical systems can be broadly divided into three common problems of increasing difficulty: (i) *monitoring* whether a single execution of a system satisfies a TL formula, (ii) *model checking* whether some or all executions of a system satisfy a formula, and (iii) *synthesis* of control policies to ensure formula satisfaction. Solutions for all three of these problems have been heavily studied for both deterministic and stochastic systems See the technical report [?] and [?], [?], [?], [?], [?], [?], [?].

Our focus in this paper is on stochastic systems with a hidden state. This paper introduces DTL as a means to formally pose these standard problems over such systems, and provides monitoring results by giving a procedure to verify *ex post facto* with what probability a particular execution of a POMDP satisfies a particular DTL specification. The more difficult problems of DTL model checking and synthesis will be investigated in future research.

Recent development of point-based approximation methods [?], [?], [?], [?] and bisimulation-based reduction methods [?], [?] have made it possible to maximize the expected reward defined over hidden states in high-dimensional POMDPs with low computational overhead. It is well known, however, that maximizing the actual reward gathered in an execution of a POMDP is undecidable [?]. Synthesizing policies over POMDPs to maximize the probability of satisfying a TL formula over hidden states is thus a hard problem, though some results exist for synthesis over short time horizons [?] and in systems where TL satisfaction can be guaranteed [?].

The best action to take in a POMDP to increase the probability of satisfaction depends intimately on the quality of knowledge of the system. Information-theoretic measures defined over belief states can quantify the certainty (i.e. Shannon entropy) of the current estimate or the expected informativeness (i.e. mutual information) of future actions [?], [?]. Considering these two measures in mobile robots have increased environmental estimation quality [?], [?], [?], [?]; incorporating them into TL-based planning for POMDPs will possibly yield similar results.

Our intention in this work is to introduce a new logic to leverage the richness of information conveyed in the belief state. Specifically, our contributions in this work are:

- We define syntactically co-safe linear DTL (scLDTL), a DTL that can be used to prescribe finite-time temporal logic behaviors of POMDPs.
- We demonstrate that DTL can describe behaviors in partially observable systems that are not describable by current TLs
- We provide an algorithmic procedure for evaluating the probability of satisfaction of an scLDTL formula with respect to an execution of a POMDP.

We intend to extend these results to synthesis in future work.

Austin Jones is with the Division of Systems Engineering, Mac Schwager and Calin Belta are with the Division of Systems Engineering and the Department of Mechanical Engineering at Boston University, Boston, MA 02115. Email: {austinmj,schwager,cbelta}@bu.edu

This work was partially supported by ONR under grant ONR MURI N00014-09-1051 and by NSF under grant CNS-1035588

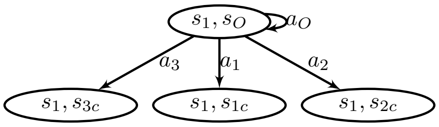


Fig. 1. A representation of the hidden state dynamics of the multiple hypothesis testing POMDP given that the true source is s_1.

II. Preliminaries

In this work, we assume that the reader is familiar with notions of set theory, probability, and the probability simplex representation of probability mass functions.

A *partially observable Markov decision process* (POMDP) [**?**], [**?**], [**?**] is a tuple $POMDP = (S, \hat{p}^0, P, Act, Obs, h)$ where S is a set of (hidden) states of the system, Act is a collection of actions, and $P : S \times Act \times S \to \mathbb{R}$ is a probabilistic transition relation such that taking the action a in state s will drive the system to state s' with probability $P(s, a, s')$. After an action a drives $POMDP$ to a state $s \in S$, , the system generates an observation $o \in Obs$ with probability $h(s, a, o)$. The system maintains a belief state $\hat{p}^t$ of the current state of $POMDP$, where $\hat{p}^t(s) = Pr[POMDP$ in state $s | a^{0:t-1}$ taken, $o^{1:t}$ seen], via recursive Bayesian estimation initialized with the prior distribution $\hat{p}^0$.

In this paper, we use syntactically co-safe linear TL (scLTL) as a basis for the definition of a new temporal logic. An scLTL formula is inductively defined as follows [**?**]:

$$\phi := \pi | \neg\pi | \phi \vee \phi | \phi \wedge \phi | \phi \mathcal{U} \phi | \bigcirc \phi | \diamondsuit \phi, \tag{1}$$

where π is an atomic proposition, $\neg$ (negation), $\vee$ (disjunction), and $\wedge$ (conjunction) are Boolean operators, and $\bigcirc$ ("next"), $\mathcal{U}$ ("until"), and $\diamondsuit$ ("eventually") are temporal operators.

We will assume that the reader is familiar with systems modeled as deterministic transition systems [**?**], a graph-like abstraction with labeled edges and states, in addition to scLTL formulae and the use of finite state automata to perform model checking of deterministic [**?**] and stochastic [**?**] systems.

III. Motivating Example: Hypothesis Testing

In this section, we use a simple multiple hypothesis testing example to motivate the introduction of the logic scLDTL described in Section **??**. Consider an experiment in which one of three coins, each with different expected frequency of heads, is flipped repeatedly. The unknown states of the system are $S_h = \{s_1, s_2, s_3\}$, where s_i is a coin with heads frequency p_i. The set of observations is $Obs = \{o_1, o_2\}$ where o_1 is heads and o_2 is tails. At each time step, a deciding agent can either flip a coin or choose a hypothesis in S_h. Let $S = S_h \times S_d$, where $S_d = \{s_{1c}, s_{2c}, s_{3c}, s_O\}$ is the state space of the deciding agent. s_O means that coin flips are still being observed and s_{ic} means that the hidden state s_i is chosen as the most likely hypothesis. The process is illustrated in Figure **??**. This is described formally by the POMDP $MHT = (S, \hat{p}^0, P, \{a_O, a_1, a_2, a_3\}, \{o_1, o_2\}, h)$ where P and h are given by

$$\begin{array}{ll} P([s_i, s_0], a_0, [s_i, s_0]) = 1, & \\ P([s_i, s_0], a_j, [s_i, s_{jc}]) = 1 & \forall i, j \in \{1, 2, 3\} \\ P(s, a, s') = 0, & \text{otherwise} \end{array} \tag{2a}$$

$$\begin{array}{ll} h([s_i, s_O], a_O, o_1) = p_i, & h([s_i, s_O], a_O, o_2) = 1 - p_i \\ h(s, a, o) = 0, & \text{otherwise} \end{array} \tag{2b}$$

Consider the problem in which we are given an infinite number of observations from MHT, but must estimate the state of the system in finite time. One solution method is to prescribe a threshold on the entropy of the belief state and terminate observation and select the most likely hypothesis when it is reached. In plain English, this is "When the entropy of the belief state is below h, select the most likely hypothesis."

This can easily be described by the new Distribution Temporal Logic (DTL) we define in Section **??**. As it will become clear later, this predicate logic is defined over two types of predicates: belief predicates and state predicates. "When the entropy of the belief state is below h" is equivalent to the belief predicate $H(\hat{p}) < h$ where $H(\cdot)$ denotes entropy "The most likely hypothesis" is equivalent to s_i such that $\hat{p}([s_i, s_O]) > \hat{p}([s, s_O])$ $\forall s \in S_h \setminus \{s_i\}$. Each comparison between components of $\hat{p}$ is a belief predicate. The selection of hypothesis s_i means the state is in the set $\{[s_j, s_{ic}]\}_{j \in \{1,2,3\}}$, and such sets will be referred to as state predicates. As it will become clear in Section **??**, the overall specification translates to the following DTL formula

$$\begin{array}{l} (H(\hat{p}) < h) \Rightarrow \\ (\bigwedge_{s_i \in S_h} (\bigwedge_{s_j \in S_h \setminus \{s_i\}} (\hat{p}([s_j, s_O]) > \hat{p}([s_i, s_O])) \Rightarrow \\ \bigcirc \{[s_j, s_{ic}]\}_{j \in \{1,2,3\}}), \end{array} \tag{3}$$

where the temporal and logical operators have roughly the same semantics as scLTL (see Section **??**, Definition **??**).

Neither the threshold on entropy nor the selection of the most likely hypothesis can be formulated using POCTL*, the most general existing temporal logic for partially observable systems [**?**], [**?**]. POCTL* can describe some properties with respect to a belief state, namely whether the probability under the initial belief state of a collection of sample paths of hidden states and observations occurring is greater than or less than some threshold, but this calculation is a linear function of the belief state. As entropy is a non-linear function of the belief state, entropy levels cannot be described in POCTL*. Further, POCTL* cannot compare the estimated probabilities of hypotheses to each other, meaning it cannot be used to formulate the selection of the most likely hypothesis.

Since the problem we consider here is readily addressed with tools from optimal estimation and information theory (see e.g. [**?**], [**?**]), constructing a new TL to describe the solution strategy may seem unnecessary. However, even considering only measures of uncertainty allows the description of novel behaviors, such as specifying low uncertainty levels as temporal goals or using uncertainty thresholds to trigger behaviors.

IV. Syntactically co-safe linear distribution temporal logic

In this section, we construct a new logic, syntactically co-safe linear distribution temporal logic (scLDTL), that describes co-safe temporal logic properties of probabilistic systems. scLTL is defined over two types of predicates: belief predicates of the type $f < 0$, with $f \in F_S : \{f : Dist(S) \to \mathbb{R}\}$ (denoted simply by f) where $Dist(S)$ is the set of all pmfs that can be defined over state space S and state predicates $s \in A$, with $A \in 2^S$ (denoted simply by A). In the example formulae used in this work, we abuse notation of belief predicates in order to enhance readability. Formally, we have:

Definition 1 (scLDTL syntax)**.** An scLDTL formula over predicates over F_S and state sets is inductively defined as follows:

$$\phi := A|\neg A|f|\neg f|\phi \vee \phi|\phi \wedge \phi|\phi \mathfrak{U} \phi| \bigcirc \phi| \diamondsuit \phi, \tag{4}$$

where $A \in 2^S$ is a set of states, $f \in F_S$ is a belief predicate, ϕ is an scLDTL formula, and $\neg$, $\vee$, $\wedge$, $\bigcirc$, $\mathfrak{U}$, and $\diamondsuit$ are as described in Section **??**.

We construct a basic notion of satisfaction over pairs of hidden state sample paths and sequences of belief states, given by Definition **??**.

Definition 2 (scLDTL semantics)**.** The semantics of scLDTL formulae is defined over words $w \in (S \times Dist(S))^\infty$. Denote the ith letter in w as $(s^i, \hat{p}^i)$ The satisfaction of a scLDTL formula at position i in w, denoted $(s^i, \hat{p}^i) \models \phi$, is recursively defined as follows:

- $(s^i, \hat{p}^i) \models A$ if $s^i \in A$,
- $(s^i, \hat{p}^i) \models f$ if $f(\hat{p}^i) < 0$,
- $(s^i, \hat{p}^i) \models \neg A$ if $s^i \notin A$,
- $(s^i, \hat{p}^i) \models \neg f$ if $f(\hat{p}^i) \geq 0$,
- $(s^i, \hat{p}^i) \models \phi_1 \wedge \phi_2$ if $(s^i, \hat{p}^i) \models \phi_1$ and $(s^i, \hat{p}^i) \models \phi_2$,
- $(s^i, \hat{p}^i) \models \phi_1 \vee \phi_2$ if $(s^i, \hat{p}^i) \models \phi_1$ or $(s^i, \hat{p}^i) \models \phi_2$,
- $(s^i, \hat{p}^i) \models \bigcirc\phi$ if $(s^{i+1}, \hat{p}^{i+1}) \models \phi$,
- $(s^i, \hat{p}^i) \models \phi_1 \mathfrak{U} \phi_2$ if there exists $j \geq i$ such that $(s^j, \hat{p}^j) \models \phi_2$ and for all $i \leq k < j$ $(s^k, \hat{p}^k) \models \phi_1$,
- $(s^i, \hat{p}^i) \models \diamondsuit \phi$ if there exists $j \geq i$ such that $(s^j, \hat{p}^j) \models \phi$.

The word $w \models \phi$, iff $(s^0, \hat{p}^0) \models \phi$.

We also define a notion of probabilistic satisfaction with respect to an execution of a POMDP in Definition **??**.

Definition 3 (scLDTL satisfaction with respect to a POMDP execution)**.** An execution of a POMDP (a sequence of belief states $\hat{p}^{0:t}$, the sequence of actions taken $a^{0:t-1}$, and the sequence of observations seen $o^{1:t}$) *probabilistically satisfies* the scLDTL formula ϕ with probability $Pr[\{s^{0:t}$ such that $(s^0, \hat{p}^0) \ldots (s^t, \hat{p}^t) \models \phi\} | \hat{p}^{0:t}, a^{0:t-1}, o^{1:t}]$, denoted in shorthand as $Pr[\phi | \hat{p}^{0:t}, a^{0:t-1}, o^{1:t}]$.

V. Monitoring POMDPs

Here we show how to solve the following problem.

Problem 1 (scLDTL monitoring of POMDPs)**.** Evaluate with what probability a given finite-length execution of a POMDP $POMDP = (S, \hat{p}^0, P, Act, Obs, h)$ satisfies a given scLDTL formula ϕ.

The solution to this problem could be used to evaluate the performance of a single execution of a POMDP or, as we show in Section **??**, can be used to compare the performance of control policies. More importantly, the tools developed for this problem are potentially useful for developing synthesis procedures.

The evaluation proceeds in two stages. In the first stage, called feasibility checking, we check for the possible existence of a sample path $s^{0:t}$ such that $(s^0, \hat{p}^0) \ldots (s^t, \hat{p}^t) \models \phi$ and $\prod_{i=0}^t \hat{p}^i(s^i) > 0$, which is a necessary but not sufficient condition for $Pr[\phi | a^{0:t-1}, o^{1:t}, \hat{p}^0] > 0$. The second stage is probabilistic satisfaction checking, in which $Pr[\phi | \hat{p}^{0:t}, a^{0:t-1}, o^{1:t}]$ is calculated.

A. Feasibility checking

First, we construct a deterministic transition system FTS whose labels correspond to the belief predicates involved in the scLDTL formula ϕ. We relax all state predicates by mapping them to belief predicates, e.g., state predicate A is relaxed to the belief predicate $Pr[s \in A] > 0$. We also create a mapping Ψ_F from each belief predicate to an atomic proposition. Then, for each f appearing in the relaxed scLDTL formula, we calculate the level set $f(\hat{p}) = 0$ in $Dist(S)$ and map it to a set of probability vectors in the probability simplex. We take the quotient of the partition given by the level sets to form a transition system and label each state with $\Psi_F(f)$ for each f that was satisfied in the corresponding region. We denote the region of the simplex corresponding to the state q_j in the transition system as $Reg(q_j)$.

More details of constructing this transition system may be found in technical report [**?**].

After FTS is constructed, we proceed to feasibility checking. From ϕ, we create an scLTL formula ϕ' by replacing every predicate in ϕ with its image in the mapping Ψ_F. We then transform the sequence $\hat{p}^{0:t}$ to a run over FTS (a finite sequence of states of FTS) and perform automata-based scLTL verification to check whether the word over FTS satisfies ϕ'. If verification succeeds, a deterministic transition system DTS, which is later used in probabilistic acceptance checking to describe the time evolution of the satisfaction of belief predicates, is constructed. DTS is a simple,“linear” transition system whose action set is a singleton and whose only possible run is $q_0 \ldots q_t$ where the label of state q_k is $\bigcup_{\{f | f(\hat{p}^k) > 0\}} \Psi_F(f)$.

If verification fails, then we do not proceed to probabilistic acceptance checking, as $Pr[\phi | \hat{p}^{0:t}, a^{0:t-1}, o^{1:t}] = 0$.

We illustrate this procedure in the following example

Example 1. Consider the multiple hypothesis testing POMDP MHT given in Section **??** with scLDTL specification (**??**) where the entropy level is 0.8 bits. In this example, we will suppress the second component of the hidden state (S_d) as it is fully known and evolves deterministically. Figure **??**(a) shows the partitioning of the probability simplex from the belief predicates in (**??**). The predicates involving maximum likelihood (red) and specified entropy level (blue) partition the simplex into six regions corresponding to discrete states $q_i, i \in \{1, \ldots, 6\}$. From this partition, we can form the transition system FTS shown in Figure **??**(b). A state q_i is labeled with proposition π_j if s_j is the most likely hypothesis according to the probability vectors in $Reg(q_i)$ and with π_4 if the entropy of any probability vector in $Reg(q_i)$ is less than 0.8 bits.

The green curve in Figure **??**(a) represents a single random execution of MHT with parameters $p_1 = 0.25$, $p_2 = 0.5$, $p_3 = 0.75$ where s_1 is the true state. Each point in the curve is the probability vector representation of the belief state $\hat{p}^i$ resulting from incorporating i observations. The transition system DTS is shown in Figure **??**(c). For the first three observations seen, the trajectory stays in $Reg(q_1)$. Thus the first three states in DTS are labeled with π_1. After the fourth measurement, the trajectory has gathered enough information to enter $Reg(q_4)$. Thus the fourth (and final) state in DTS is labeled with both π_1 and π_4.

B. Probabilistic acceptance checking

We begin the probability calculation by creating a mapping $\Psi_{sp} : 2^S \to \Pi_r$ that maps state predicates to atomic propositions. The scLDTL formula ϕ is mapped to a scLTL formula ϕ'' by applying the mapping Ψ_F to the belief predicates and the mapping Ψ_{sp} to the state predicates appearing in ϕ. An FSA is created from ϕ''. Next, we enumerate all of the sample paths consistent with the given execution of $POMDP$. We do this by creating a labeled Markov chain LMC for each possible initial state s_0 such that $\hat{p}^0(s_0) > 0$. A labeled Markov Chain (LMC) is given as a tuple $LMC = (S, s_0, P, AP, L)$ where S is a set of discrete states, s_0 is an initial state, P is a probabilistic transition relation such that LMC transitions from state s to state s' with probability $P(s, s')$, AP is a set of atomic propositions and $L : S \to 2^{AP}$ is a labeling function. The details of the LMC construction may be found in the technical report [**?**]. Each state s in the LMC is labeled with $\Psi_{sp}(A)$ for all A such that $s \in A$, i.e. according the state predicates satisfied by s. States at the ith level of the LMC are also labeled with the label of state q_i from DTS, i.e. according to the belief predicates satisfied by $\hat{p}^i$. For each LMC, we perform model checking using Bayesian smoothing [**?**] to calculate $Pr[\phi''$ is satisfied$|s^0 = s_0]$. The total acceptance probability is than calculated as $Pr[\phi|\hat{p}^{0:t}, a^{0:t-1}, o^{1:t}] = \sum_{s_0|\hat{p}^0(s_0)>0} Pr[\phi''$ is satisfied$|s^0 = s_0]$.

VI. Case Study: rescue robots

A proposed use of mobile robots is to perform rescue operations in areas that are too hazardous for human rescuers. A robot is deployed to a location such as an office building or school after a natural disaster and is tasked with finding all human survivors in the environment and with moving any immobilized survivors to safe areas. The robot must learn survivor locations and the safety profile of the building on-line by processing noisy measurements from its sensors. The combination of on-line estimation and time-sensitive decision-making indicates that scLDTL is a good framework for describing the mission specification at a high level.

A. Model

For simplicity, we consider a rescue robot acting in a two room environment. We model the robot as a POMDP $Rescue = (S, \hat{p}^0, P, Act, Obs, h)$. The state of the system is given by a vector $[s_q, s_O, s_{1,e}, s_{2,e}, s_{1,s}, s_{2,s}]$ in the state space $S = \{1, 2\} \times \{0, 1\}^5$. The element s_q corresponds to the room in which the robot currently resides and $s_O \in \{0, 1\}$ corresponds to whether ($s_O = 1$) or not ($s_O = 0$) the robot is carrying an object. The elements $s_{i,e} \in \{0, 1\}$ correspond to safety, i.e. if $s_{i,e} = 1$, then room i is safe to be occupied by a human. The elements $s_{i,s} \in \{0, 1\}$ correspond to survivor presence, i.e. if $s_{i,s} = 1$, a survivor is in room i.

The robot can stay in its current room and measure its surroundings, switch to the other room, pick up an object, or put down an object. Here we assume the motion model of the robot is deterministic, the safety of the environment is static, and the survivor locations change only if the robot moves a survivor. If the robot attempts to move a survivor, it fails with some probability p_{fail}.

If the robot takes action $Stay$, its sensors return observations in the set $Obs = \{0, 1\}^2$. The elements of Obs are binary reports of the safety and survivor occupancy of the current room. The sensor is parameterized by two independent false alarm and correct detection rates.

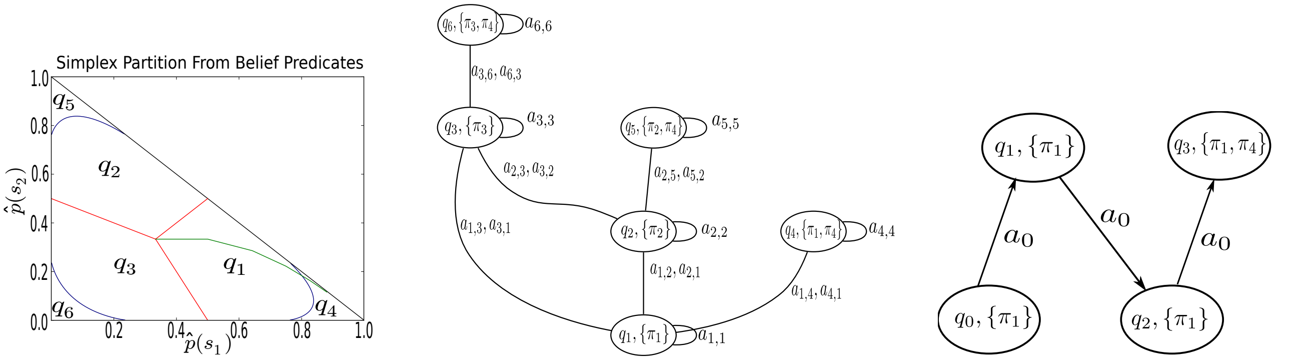
B. Problem statement

For convenience we establish the shorthand $\hat{p}_j(\sigma) = \sum_{\{s \in S | s_j = \sigma\}} \hat{p}(s)$ where s_j is a component of an element of S. We wish to find and move all of the survivors in the given area to safe regions.

The statement that describes the rescue robotics application is "Explore the environment and if the robot is in a state where it is sure with probability p_1 there is a survivor and with probability p_2 the state is unsafe, pick up the survivor, move to the other room and deposit the survivor. Perform these actions until the entropy of $\hat{p}_{i,e}$ is less than h_1 and the entropy of $\hat{p}_{i,s}$ is less than h_2 for $i = 1, 2$ and any identified survivors are in safe regions". This is encoded in the scLDTL formula $\phi_1 \mathcal{U} \phi_2$ where

$$\begin{aligned} \phi_1 = & \quad (\{s|s_q = j\} \wedge (\hat{p}_j(s) > p_1) \wedge (\hat{p}_{j,e}(0) > p_2)) \\ & \Rightarrow (\bigcirc(\{s|s_O = 1\} \mathcal{U} \{s|s_q \neq j\}) \wedge \bigcirc\{s|s_O = 0\} \\[1em] \phi_2 = & \quad \bigwedge_{i \in \{1,2\}} (H(\hat{p}_{i,e}) < h_1) \wedge (H(\hat{p}_{i,s}) < h_2) \\ & \wedge(\{s|s_{i,e} = 1\}) \wedge \{s|s_{i,s} = 1\}) \vee \{s|s_{i,s} = 0\}) \end{aligned} \tag{5}$$

Due to the time sensitive nature of survival, we consider the following time-constrained optimization problem.



(a) (b) (c)

Fig. 2. (a) The probability simplex for three hypotheses partitioned according to the belief predicates used in (**??**). The red lines divide the simplex into three regions corresponding to the most likely hypothesis. The blue curves are the level sets $H(\hat{p}) = 0.8$ bits. The green curve shows the probability trajectory corresponding to a sequence of belief states from a randomly generated execution of MHT. (b) The transition system FTS constructed by taking the quotient of the partition shown in (a). The edges are denoted by virtual actions $a_{i,j}$ which represent an action-observation pair that drives the belief state from a point in $Reg(q_i)$ to a point in $Reg(q_j)$ (c) The transition system DTS constructed from the given belief state sequence and FTS.

$$\max_{a^{0:t-1}} E_{\{o^{1:t}\}}[Pr[\phi_1 \mathcal{U} \phi_2 | \hat{p}^{0:t}, a^{0:t-1}, o^{1:t}]] \quad (6)$$

C. Acceptance checking

We consider two separate strategies to solve (**??**): time share, in which the robot switches rooms every $\lceil \frac{t}{a} \rceil$ observations, and entropy cutoff in which the robot switches rooms when the entropy of the estimate of the safety and survivor presence of the current room dips below h_3 and h_4, respectively. In entropy cutoff, the agent must wait ρ time units before switching if both estimates are at or below the specified uncertainties. Both strategies include the reactive behavior of attempting to pick up survivors when they are found.

The results from 250 Monte Carlo trials of length $t = 16$ are shown in Figure **??**. Simulation parameters are given in the caption of Figure **??**. Here we use $Pr[\phi]$ as shorthand for the statistic formed from samples of $Pr[\phi|\hat{p}^{0:t}, a^{0:t-1}, o^{1:t}]$ collected from the trials. For both methods, there are clusters of points around the lines $Pr[\phi] = 1$ and $Pr[\phi] = 0$. This is because by making the entropy of the belief state a temporal goal in the scLDTL formula, the probability calculation sets the satisfaction probability to 0 for executions after which the characterization of the environment is ambiguous.

The statistics resulting from our simulations are shown in Table **??**. The statistic $r(Pr[\phi], H(\hat{p}^t))$ is the correlation coefficient between the two variables. The success rate is given as the number of trials in which all survivors were moved to safety divided by the total number of trials. Note that the entropy cutoff method performs better in terms of acceptance probability, expected terminal entropy, and success rate. This matches intuition, as this method will drive the robot to stay in a room longer if it has not made any strong conclusions or it will move to the other room if it has already obtained a good estimate. In contrast, the time share method ignores estimate quality in its decision policy.

Further, note that for both methods, the correlation coefficient is weakly negative. This weakness is due to the clustering of points around $Pr[\phi] = 0$ and $Pr[\phi] = 1$. This negative correlation and the relative closeness of the average acceptance probability of the two methods to their respective success rates suggests that for some appropriately-defined scLDTL formulae, the probability $Pr[\phi|\hat{p}^{0:t}, a^{0:t-1}, o^{1:t}]$ is an appropriate metric for the dual consideration of estimate quality and system performance.

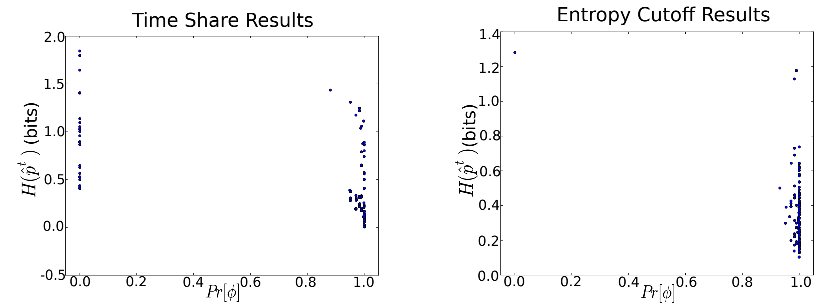
VII. Conclusions

We argued that a new type of temporal logic, generically denoted as Distribution Temporal Logic (DTL), is needed to express notions of uncertainty and ambiguity in partially observed systems. We have formalized a co-safe version of this logic and shown how to evaluate with what probability an execution of a POMDP satisfies a DTL formula. Our case study demonstrates that this probability is a relevant metric for the performance of control policies. In the future, we will extend these results to a procedure for synthesizing control policies that maximize this probability. The application of DTL to other probabilistic systems and further exploration of its expressivity are also planned areas of research.

Method	$E[Pr[\phi]]$	$var(Pr[\phi])$	$E[H(\hat{p}^t)]$	$var(H(\hat{p}^t)$	success rate	$r(Pr[\phi], H(\hat{p}^t))$
Timeshare	0.855	0.115	0.366 bits	0.150 bits^2	0.86	-0.547
Entropy Threshold	0.992	0.004	0.338 bits	0.034 bits^2	0.916	-0.341

TABLE I

STATISTICS FROM 250 MONTE CARLO TRIALS OF THE TWO-ROOM RESCUE ROBOTICS SIMULATION.



(a) (b)

Fig. 3. Scatter plots showing the results of 250 Monte Carlo trials of the two room rescue robot POMDP under policy (a) time share and (b) entropy cutoff. The control strategy parameters were a=3,$h_3 = h_4 = 0.3$, and $\rho = 2$. The parameters used in (**??**) are $h_1 = h_2 = 0.375$ and $p_1 = 0.9$, $p_2 = 0.25$. The probability that an agent fails to pick up a survivor was $p_{fail} = 0.4$. The false alarm rates for safety and survivor were both 0.1. The correct detection rates for safety and survivor were 0.8 and 0.9, respectively.