2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Barrier Functions for Multiagent-POMDPs with DTL Specifications

Mohamadreza Ahmadi, Andrew Singletary, Joel W. Burdick, and Aaron D. Ames

Abstract— **Multi-agent partially observable Markov decision processes (MPOMDPs) provide a framework to represent heterogeneous autonomous agents subject to uncertainty and partial observation. In this paper, given a nominal policy provided by a human operator or a conventional planning method, we propose a technique based on barrier functions to design a minimally interfering *safety-shield* ensuring satisfaction of high-level specifications in terms of linear distribution temporal logic (LDTL). To this end, we use sufficient and necessary conditions for the invariance of a given set based on discrete-time barrier functions (DTBFs) and formulate sufficient conditions for finite time DTBF to study finite time convergence to a set. We then show that different LDTL mission/safety specifications can be cast as a set of invariance or finite time reachability problems. We demonstrate that the proposed method for safety-shield synthesis can be implemented online by a sequence of one-step greedy algorithms. We demonstrate the efficacy of the proposed method using experiments involving a team of robots.**

I. Introduction

Decision making under uncertainty and partial observation is an important branch of artificial intelligence (AI) and probabilistic robotics that has received attention in the recent years. A popular formalism that can capture the decision making, uncertainty, and partial observation associated with such systems is the partially observable Markov decision process (POMDP). In a POMDP framework, an autonomous agent is not aware of the exact state of the environment and, through a sequence of actions and observations, it updates its *belief* in the current state of the environment. Decision making is then carried out based on the history of the observations or the current belief. Despite the fact that POMDPs provide a unique modeling paradigm, they are notoriously hard to solve. In particular, it was shown that the infinite-horizon total undiscounted/discounted/average reward problem for a single agent is undecidable [24] and even the finite-horizon problem for multiple agents with full communication is PSPACE-complete [26]. However, methods based on discretization of the belief space (known as point-based methods) [28], heuristics [7], finite-state controllers [4], [9], [20], or abstractions [16] are shown to be successful to handle relatively large problems.

However, for safety-critical systems, such as Mars rovers and autonomous vehicles, safe operation is as (if not more) important than optimality and it is often cumbersome to design a policy to guarantee both safety and optimality. In particular for high-level safety specifications in terms of temporal logic, synthesizing a policy satisfying the specification is undecidable [11] and requires heuristics and ad-hoc methods [10]. Run-time enforcement of linear temporal logic (LTL) specifications in the absence of partial observability, *i.e.*, in Markov decision processes (MDPs), was considered in [6], [17], [18], where the authors use automaton representations of the LTL specifications. Though effective for MDPs, the latter approach is not applicable to POMDPs for two main reasons. First, given an LTL formula, the automaton representation can introduce finite but arbitrary large number of additional states. Therefore, model checking of even abstraction-based methods [33] for POMDPs may not be suitable for run-time enforcement. Second, LTL may not be a suitable logic for describing safety specifications for systems subject to unavoidable uncertainty and partial observation [19]. Therefore, we use LDTL [19], [32], which can be used for specifying tasks for stochastic systems with partial state information.

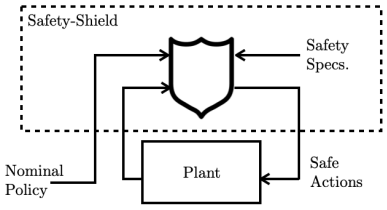

Fig. 1. Schematic diagram of the proposed *Safety-Shield* framework.

In this paper, instead of automaton representations, we employ discrete time barrier functions (DTBFs) to enforce safety/mission specifications in terms of LDTL specifications in Multi-agent POMDPs (plant) in *run time* with *minimum interference* (see Fig. 1). To this end, we represent the joint belief evolution of an MPOMDP as a discrete-time system [3]. The main contributions of this paper are then as follows: (i) We enrich the DTBFs for enforcing invariance formulated in our preliminary work [5] with finite-time DTBFs for assuring finite time reachability and (ii) we propose Boolean compositions of these finite-time DTBFs. (iii) We propose a LDTL safety-shield method based on one-step greedy algorithms [12, Chapter 16] to synthesize a safety-shield for an MPOMDP given a nominal planning policy. We illustrate the efficacy of the proposed approach by applying it to an exploration scenario of a team of heterogeneous robots in ROS simulation environment.

The rest of the paper is organized as follows. The next section reviews some preliminary notions and definitions used in the sequel. In Section III, we propose DTBFs for invariance and finite-time reachability as well as their Boolean compositions. In Section IV, we design a safety-shield based on DTBFs for LDTL specifications. In Section V, we elucidate our results with a multi-robot case study. Finally, in Section VI, we conclude the paper.

This work was supported by DARPA Subterranean Challenge and Raytheon Technologies. The authors are with the California Institute of Technology, 1200 E. California Blvd., MC 104-44, Pasadena, CA 91125, e-mail: ({mrahmadi,asingletary, jwb,ames}@caltech.edu).)

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

Notation: $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. $\mathbb{R}_{\geqslant 0}$ denotes the set $[0,\infty)$. $\mathbb{N}_{\geqslant l}$ denotes the set of integers greater that equal to l, i.e., $\mathbb{N}_{\geqslant l} = \{l, l+1, \ldots\}$. For a finite-set A, $|A|$ and 2^A denote the number of elements in A and the power set of A, respectively. A continuous function $\alpha : [0,a) \to \mathbb{R}_{\geqslant 0}$ is a class $\mathcal{K}$ function if $\alpha(0) = 0$ and it is strictly increasing. Similarly, a continuous function $\beta : [0,a) \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ is a class $\mathcal{KL}$ function if $\beta(r,\cdot) \in \mathcal{K}$ and if $\beta(\cdot,s)$ is decreasing with respect to s and $\lim_{s\to\infty} \beta(\cdot,s) \to 0$. For two functions $f : \mathcal{G} \to \mathcal{F}$ and $g : \mathcal{X} \to \mathcal{G}$, $f \circ g : \mathcal{X} \to \mathcal{F}$ denotes the composition of f and g and $\mathrm{Id} : \mathcal{F} \to \mathcal{F}$ denotes the identity function satisfying $\mathrm{Id} \circ f = f$ for all functions $f : \mathcal{X} \to \mathcal{F}$. The Boolean operators are denote by $\neg$ (negation), $\vee$ (conjunction), and $\wedge$ (disjunction). The temporal operators are denoted by $\bigcirc$ (next), $\mathfrak{U}$ (until), $\Box$ (always), and $\Diamond$ (eventuality).

II. Preliminaries

In this section, we briefly review some notions and definitions used throughout the paper.

A. Multi-Agent POMDPs

An MPOMDP [7], [25] provides a sequential decision-making formalism for high-level planning of multiple autonomous agents under partial observation and uncertainty. At every time step, the agents take actions and receive observations. These observations are shared via (noise and delay free) communication and the agents decide in a centralized framework.

***Definition** 1: An MPOMDP is a tuple $\big(I, Q, p^0, \{A_i\}_{i\in I}, T, R, \{Z_i\}_{i\in I}, O\big)$, wherein*

- *I denotes a index set of agents;*
- *Q is a finite set of states with indices $\{1, 2, \ldots, n\}$;*
- *$p^0 : Q \to [0,1]$ defines the initial state distribution;*
- *A_i is a finite set of actions for agent i and $A = \times_{i\in I} A_i$ is the set of joint actions;*
- *$T : Q \times A \times Q \to [0,1]$ is the transition probability, where $T(q,a,q') := P(q^t = q' | q^{t-1} = q, a^{t-1} = a)$, $\forall t \in \mathbb{N}_{\geqslant 1}, q, q' \in Q, a \in A$, i.e., the probability of moving to state q' from q when the joint actions a are taken;*
- *$R : Q \times A \to \mathbb{R}$ is the immediate reward function for taking the joint action a at state q;*
- *Z_i is the set of all possible observations for agent i and $Z = \times_{i\in I} Z_i$, representing outputs of discrete sensors, e.g. $z \in Z$ are incomplete projections of the world states q, contaminated by sensor noise;*
- *$O : Q \times A \times Z \to [0,1]$ is the observation probability (sensor model), where $O(q',a,z) := P(z^t = z | q^t = q', a^{t-1} = a), \forall t \in \mathbb{Z}_{\geqslant 1}, q \in Q, a \in A, z \in Z$, i.e., the probability of seeing joint observations z given joint actions a were taken and resulting in state q'.*

Since the states are not directly accessible in an MPOMDP, decision making requires the history of joint actions and joint observations. Therefore, we must define the notion of a joint *belief* or the posterior as sufficient statistics for the history. Given an MPOMDP, the joint belief at $t = 0$ is defined as $b^0(q) = p^0(q)$ and $b^t(q)$ denotes the probability of the system being in state q at time t. At time $t+1$, when joint action $a \in A$ is taken and joint observation $z \in Z$ is observed, the belief is updated via a Bayesian filter [21] as

$$b^t(q') := BU(z^t, a^{t-1}, b^{t-1}) = \frac{O(q', a^{t-1}, z^t) \sum_{q\in Q} T(q, a^{t-1}, q') b^{t-1}(q)}{\sum_{q'\in Q} O(q', a^{t-1}, z^t) \sum_{q\in Q} T(q, a^{t-1}, q') b^{t-1}(q)} \quad (1)$$

where the beliefs belong to the belief unit simplex

$$\mathcal{B} = \left\{ b \in [0,1]^{|Q|} \mid \sum_{q\in Q} b^t(q) = 1, \ \forall t \right\}.$$

A policy in an MPOMDP setting is then a mapping $\pi : \mathcal{B} \to A$, i.e., a mapping from the continuous joint beliefs space into the discrete and finite joint action space. When I is just a singleton (only one agent), we have a POMDP [30].

B. Linear Distribution Temporal Logic

We formally describe high-level mission specifications that are defined in temporal logic. Temporal logic has been used as a formal way to allow the user to intuitively specify high-level specifications, in for example, robotics [22]. The temporal logic we use in this paper can be used for specifying tasks for stochastic systems with partial state information. This logic is suitable for problems involving significant state uncertainty, in which the state is estimated on-line. The syntactically co-safe linear distribution temporal logic (scLDTL) describes co-safe linear temporal logic properties of probabilistic systems [19]. We consider a modified version to scLDTL, the linear distribution temporal logic (LDTL), which includes the additional temporal operator $\Box$ "always". The latter operator is important since it can be used to describe notions such as *safety*, *liveness*, and *invariance*.

LDTL has predicates of the type $\zeta < 0$ with $\zeta \in \mathcal{F}_Q = \{f \mid f : \mathcal{B} \to \mathbb{R}\}$, i.e., $\mathcal{F}_Q$ is the class of (nonlinear) functions mapping from the belief simplex into reals, and state predicates $q \in A$ with $A \in 2^Q$.

***Definition** 2 (LDTL Syntax): An LDTL formula over predicates $\mathcal{F}_Q$ and Q is inductively defined as*

$$\varphi := A | \neg A | \zeta | \neg\zeta | \varphi \vee \varphi | \varphi \wedge \varphi | \varphi \ \mathfrak{U} \ \varphi | \bigcirc \varphi | \Diamond\varphi | \Box\varphi, \quad (2)$$

where $A \in 2^Q$ is a set of states, $\zeta \in \mathcal{F}_Q$ is a belief predicate, and φ is an LDTL formula.

Satisfaction over pairs of hidden state paths and sequences of belief states can then be defined as follows.

***Definition** 3 (LDTL Semantics): The semantic of LDTL formulae is defined over words $\omega \in (Q \times \mathcal{B})^\infty$. Let (q^i, b^i) be the ith letter in ω. The satisfaction of a LDTL formula φ at position i in ω, denoted by $\omega^i \models \varphi$ is recursively defined as follows*

- *$\omega^i \models A$ if $q^i \in A$,*
- *$\omega^i \models \neg A$ if $q^i \notin A$,*
- *$\omega^i \models f$ if $f(b^i) < 0$,*
- *$\omega^i \models \neg f$ if $f(b^i) \geqslant 0$,*
- *$\omega^i \models \varphi_1 \wedge \varphi_2$ if $\omega^i \models \varphi_1$ and $\omega^i \models \varphi_2$,*
- *$\omega^i \models \varphi_1 \vee \varphi_2$ if $\omega^i \models \varphi_1$ or $\omega^i \models \varphi_2$,*
- *$\omega^i \models \bigcirc\varphi$ if $\omega^{i+1} \models \varphi$,*

- $\omega^i \models \varphi_1 \mathfrak{U} \varphi_2$ *if there exists a* $j \geqslant i$ *such that* $\omega^j \models \varphi_2$ *and for all* $i \leqslant k < j$ *it holds that* $\omega^k \models \varphi_1$,
- $\omega^i \models \Diamond \varphi$ *if there exists* $j \geqslant i$ *such that* $\omega^j \models \varphi$, *and*
- $\omega^i \models \Box \varphi$ *if, for all* $j \geqslant i$, $\omega^j \models \varphi$.

The word ω *satisfies a formula* φ, *i.e.,* $\omega \models \varphi$, *iff* $\omega^0 \models \varphi$.

Designing policies that guarantee LDTL formulas as defined in Definition 1 can only be carried if the system is linear and subject to Guassian noise [32]. We show later in Section IV that DTBFs can be used to enforce LDTL formulas for any finite POMDP.

III. Discrete-Time Barrier Functions

In order to guarantee the satisfaction of the LDTL formulae in MPOMDPs, we use barrier functions [8] rather than automatons and model checking. Hence, our method does not rely on automaton representations, discretizations of the belief space, or finite memory controllers. These barrier functions ensure that the solutions to the joint belief update equation remain inside or reach subsets of the belief simplex that is induced by the LDTL formula. Noting that the joint belief evolution of an MPOMDP (1) can be described by a discrete-time system [3], in this section, we propose conditions based on DTBFs for verifying invariance and finite-time reachability properties.

Given an MPOMDP as defined in Definition 1, at every time-step $t \in \mathbb{N}_{\geqslant 0}$, the joint belief update equation (1) can be described by the following discrete-time system

$$b^{t+1} = F(b^t), \tag{3}$$

with $F : \mathcal{B} \to \mathcal{B} \subset \mathbb{R}^n$ given an observation and an action. We consider subsets of the belief simplex defined as

$$\mathcal{S} := \{b \in \mathcal{B} \mid h(b) \geqslant 0\}, \tag{4a}$$
$$\mathrm{Int}(\mathcal{B}) := \{b \in \mathcal{B} \mid h(b) > 0\}, \tag{4b}$$
$$\partial\mathcal{S} := \{b \in \mathcal{B} \mid h(b) = 0\}. \tag{4c}$$

We then have the following definition of a DTBF.

***Definition** 4 (Discrete-Time Barrier Function): For the discrete-time system* (3)*, the continuous function* $h : \mathbb{R}^n \to \mathbb{R}$ *is a discrete-time barrier function for the set* $\mathcal{S}$ *as defined in* (4)*, if there exists* $\alpha \in \mathcal{K}$ *satisfying* $\alpha(r) < r$ *for all* $r > 0$ *such that*

$$h(b^{t+1}) - h(b^t) \geqslant -\alpha(h(b^t)), \quad \forall b^t \in \mathcal{B}. \tag{5}$$

In fact, the DTBF defined above is a discrete-time zeroing barrier function per the literature [8] (see also the reciprocal DTBF proposed in [2]), but we drop the "zeroing" as it is the only form of barrier function that will be considered in this paper.

We can show that the existence of a DTBF is both necessary and sufficient for invariance. We later show in Section V that such DTBF can be used to verify a class of LDTL specifications.

***Theorem** 1 ([5]): Consider the discrete-time system* (3)*. Let* $\mathcal{S} \subseteq \mathcal{B} \subset \mathbb{R}^n$ *with* $\mathcal{S}$ *as described in* (4)*. Then,* $\mathcal{S}$ *is invariant if and only if there exists a DTBF as defined in Definition 4.*

A. Finite Time DTBFs

Another class of problems we are interested in involve checking whether the solution of a discrete time system can reach a set in finite time. We will show in Section IV that such problems arise when dealing with "eventuality" type LDTL specifications. To this end, we define a finite time DTBF (see [31] for the continuous time variant).

***Definition** 5 (Finite Time DTBF): For the discrete-time system* (3)*, the continuous function* $\tilde{h} : \mathcal{B} \to \mathbb{R}$ *is a finite time DTBF for the set* $\mathcal{S}$ *as defined in* (4)*, if there exist constants* $0 < \rho < 1$ *and* $\varepsilon > 0$ *such that*

$$\tilde{h}(b^{t+1}) - \rho\tilde{h}(b^t) \geqslant \varepsilon(1-\rho), \quad \forall b^t \in \mathcal{B}. \tag{6}$$

We then have the following result to check finite time reachability of a set for a discrete-time system.

***Theorem** 2: Consider the discrete-time system* (3)*. Let* $\mathcal{S} \subset \mathcal{B} \subset \mathbb{R}^n$ *be as described in* (4)*. If there exists a finite time DTBF* $\tilde{h}$ *as in Definition 5, then for all* $b^0 \in \mathcal{B}\backslash\mathcal{S}$, *there exists a* $t^* \in \mathbb{N}_{\geqslant 0}$ *such that* $b^{t^*} \in \mathcal{S}$. *Furthermore,*

$$t^* \leqslant \log\left(\frac{\varepsilon - \tilde{h}(b^0)}{\varepsilon}\right)/\log\left(\frac{1}{\rho}\right), \tag{7}$$

where the constants ρ *and* ε *are as defined in Definition 5.*

Proof: We prove by induction. With some manipulation inequality (6) can be modified to $\tilde{h}(b^{t+1}) - \varepsilon \geqslant \rho\tilde{h}(b^t) - \rho\varepsilon = \rho\left(\tilde{h}(b^t) - \varepsilon\right)$. Thus, for $t = 0$, we have $\tilde{h}(b^1) - \varepsilon \geqslant \rho\left(\tilde{h}(b^0) - \varepsilon\right)$. For $t = 1$, we have $\tilde{h}(b^2) - \varepsilon \geqslant \rho\left(\tilde{h}(b^1) - \varepsilon\right) \geqslant \rho^2\left(\tilde{h}(b^0) - \varepsilon\right)$, where we used the inequality for $t = 0$ to obtain the last inequality above. Then, by induction, we have $\tilde{h}(b^t) - \varepsilon \geqslant \rho^t\left(\tilde{h}(b^0) - \varepsilon\right)$. Hence, $\tilde{h}(b^t) \geqslant \rho^t(\tilde{h}(b^0) - \varepsilon) + \varepsilon$. Since $0 < \rho < 1$ and $b^0 \in \mathcal{B}\backslash\mathcal{S}$, i.e., $\tilde{h}(b^0) < 0$, as t increases b^t approaches $\mathcal{S}$ because by definition $h(b^t) \geqslant 0$ implies $b^t \in \mathcal{S}$. Re-arranging the terms gives

$$\varepsilon - \tilde{h}(b^t) \leqslant \rho^t\left(\varepsilon - \tilde{h}(b^0)\right). \tag{8}$$

Since $b^0 \in \mathcal{B}\backslash\mathcal{S}$, i.e., $\tilde{h}(b^0) < 0$, $\varepsilon - \tilde{h}(b^0)$ is a positive number. Dividing both sides of (8) with the positive quantity $\varepsilon - \tilde{h}(b^0)$ yields $\frac{\varepsilon - \tilde{h}(b^t)}{\varepsilon - \tilde{h}(b^0)} \leqslant \rho^t$. Taking the logarithm of both sides of the above inequality gives $\log\left(\frac{\tilde{h}(b^t) - \varepsilon}{\tilde{h}(b^0) - \varepsilon}\right) \leqslant t\log(\rho)$, or equivalently

$$-\log\left(\frac{\tilde{h}(b^0) - \varepsilon}{\tilde{h}(b^t) - \varepsilon}\right) \leqslant -t\log(\frac{1}{\rho}).$$

Since $0 < \rho < 1$, $\log(\frac{1}{\rho})$ is a positive number. Dividing both sides of the inequality above with the negative number $-\log(\frac{1}{\rho})$ obtains $t \leqslant \log\left(\frac{\varepsilon - \tilde{h}(b^0)}{\varepsilon - \tilde{h}(b^t)}\right)/\log\left(\frac{1}{\rho}\right)$. Also, by definition, b^t reaches $\mathcal{S}$ at least at the boundary at t^* when $\tilde{h}(b^t) = 0$. Substituting $\tilde{h}(b^t) = 0$ in the last inequality for t gives $t^* \leqslant \log\left(\frac{\varepsilon - \tilde{h}(b^0)}{\varepsilon}\right)/\log\left(\frac{1}{\rho}\right)$, which gives an upper bound for the first time $b^t \in \mathcal{S}$. ∎

B. Boolean Composition of Finite Time DTBFs

In order to assure specifications involving conjunction or disjunction of LDTL formulae in Definition 3, we need to consider properties of sets defined by Boolean composition of DTBFs. In this regard, in [14], the authors proposed non-smooth barrier functions as a means to analyze composition of barrier functions by Boolean logic, i.e., $\vee$, $\wedge$, and $\neg$. Similarly, in this study, we propose non-smooth DTBFs. The negation operator is trivial and can be shown by checking if $-h$ satisfies the corresponding property.

In the following, we propose conditions for checking Boolean compositions of finite time DTBFs. Fortunately, since we are concerned with discrete time systems, this does not require non-smooth analysis (for a similar result pertaining compositions of DTBFs see Proposition 1 in [5]).

***Proposition** 1: Let $\mathcal{S}_i = \{b \in \mathcal{B} \mid \tilde{h}_i(b) \geqslant 0\}$, $i = 1, \ldots, k$ denote a family of sets defined analogous to $\mathcal{S}$ in* (4)*. Consider the discrete-time system* (3)*. If there exist constants $0 < \rho < 1$ and $\varepsilon > 0$ such that*

$$\min_{i=1,\ldots,k} \tilde{h}_i(b^{t+1}) - \rho \min_{i=1,\ldots,k} \tilde{h}_i(b^t) \geqslant \varepsilon(1-\rho), \ \forall b \in \mathcal{B}, \quad (9)$$

then there exists

$$t^* \leqslant \log\Big(\frac{\varepsilon - \min_{i=1,\ldots,k} \tilde{h}_i(b^0)}{\varepsilon}\Big)/\log\Big(\frac{1}{\rho}\Big). \quad (10)$$

such that if $b^0 \in \mathcal{B}\backslash\bigcup_{i=1}^k \mathcal{S}_i$ then $b^{t^*} \in \Big\{b \in \mathcal{B} \mid \wedge_{i=1,\ldots,k}\Big(\tilde{h}_i(b) \geqslant 0\Big)\Big\}$. Similarly, the disjunction case follows by replacing min *with* max *in* (9) *and* (10)*.*

Proof: We prove the conjunction case and the disjunction case follows the same lines. If (9) holds, from the proof of Theorem 2, we can infer that $\min_{i=1,\ldots,k} \tilde{h}_i(b^t) - \varepsilon \geqslant \rho^t\Big(\min_{i=1,\ldots,k} \tilde{h}_i(b^0) - \varepsilon\Big)$, which implies that $t \leqslant \log\Big(\frac{\varepsilon - \min_{i=1,\ldots,k} \tilde{h}_i(b^0)}{\varepsilon - \min_{i=1,\ldots,k} \tilde{h}_i(b^t)}\Big)/\log\Big(\frac{1}{\rho}\Big)$. If $b^0 \in \mathcal{B}\backslash\bigcup_{i=1}^k \mathcal{S}_i$, then by definition $\tilde{h}_i(b^0) < 0$, $i = 1, \ldots, k$. Hence, $\min_{i=1,\ldots,k} \tilde{h}_i(b^0) < 0$. Moreover, because t is a positive integer, $\varepsilon - \min_{i=1,\ldots,k} \tilde{h}_i(b^0) \geqslant \varepsilon - \min_{i=1,\ldots,k} \tilde{h}_i(b^t)$. That is, $\min_{i=1,\ldots,k} \tilde{h}_i(b^0) \leqslant \min_{i=1,\ldots,k} \tilde{h}_i(b^t)$ along the solutions b^t of the discrete-time system (3). Furthermore, $b^t \in \Big\{b \in \mathcal{B} \mid \wedge_{i=1,\ldots,k}\Big(\tilde{h}_i(b) \geqslant 0\Big)\Big\}$ whenever $\min_{i=1,\ldots,k} \tilde{h}_i(b^t) \geqslant 0$. The upper-bound for this t happens when $\min_{i=1,\ldots,k} \tilde{h}_i(b^t) = 0$, i.e., when all $\tilde{h}_i(b^t)$ are either positive or zero. This by definition implies that $b^t \in \Big\{b \in \mathcal{B} \mid \wedge_{i=1,\ldots,k}\Big(\tilde{h}_i(b) \geqslant 0\Big)\Big\}$. Then, setting $\min_{i=1,\ldots,k} \tilde{h}_i(b^t) = 0$ gives $t^* \leqslant \log\Big(\frac{\varepsilon - \min_{i=1,\ldots,k} \tilde{h}_i(b^0)}{\varepsilon}\Big)/\log\Big(\frac{1}{\rho}\Big)$. ∎

IV. Safety-Shield Synthesis

Since the states are not directly observable in MPOMDPs, we are interested in guaranteeing safety specifications in terms of LDTL in a probabilistic setting in the joint belief space. We denote by $\pi_n : \mathcal{B} \to \mathcal{A}$ a deterministic nominal joint policy mapping each joint belief into a joint action (for example, policies for infinite horizon discounted cost problems). At every time step t, the nominal policy assigns a nominal action, i.e., $\pi_n(b^t) = a_n^t$. The immediate reward at that time step can then be computed as $r_n^t(b^t, a_n^t) = \sum_{q \in Q} b^t(q) R(q, a_n^t)$.

We are interested in solving the following problem.

***Problem** 1 (Safety-Shield Synthesis): Given an MPOMDP as defined in Definition 1, a corresponding belief update equation* (1)*, a safety LDTL formula φ, and a nominal planning policy π_n, determine a sequence of actions a^t, $t \in \mathbb{N}_{\geqslant 0}$ such that $\omega^0 = (q^0, b^0) \models \varphi$ and the quantity $\|r^t - r_n^t\|^2$ is minimized for all $t \in \mathbb{N}_{\geqslant 0}$, where r_n^t denotes the nominal immediate reward at time step t.*

Note that choice of the 2-norm squared of the error is arbitrary and other metrics, such as ℓ_1-norm of error between r^t and r_n^t can be studied as well.

LDTL Specification	DTBF Implementation
$\omega^i \models A$	$h(b^i) = \sum_{q \in A} b^i(q) - 1$
$\omega^i \models \neg A$	$h(b^i) = \sum_{q \in Q\backslash A} b^i(q) - 1$
$\omega^i \models f$	$h(b^i) = -f(b^i) + \delta$
$\omega^i \models \neg f$	$h(b^i) = f(b^i)$
$\omega^i \models \varphi_1 \wedge \varphi_2$	$h(b^i) = \min\{h_1(b^i), h_2(b^i)\}$
$\omega^i \models \varphi_1 \vee \varphi_2$	$h(b^i) = \max\{h_1(b^i), h_2(b^i)\}$
$\omega^i \models \bigcirc\varphi$	$h(b^{i+1}) = h_\varphi(b)$
$\omega^i \models \varphi_1 \mathfrak{U} \varphi_2$	$h_2(b^j) < 0 \implies h = h_1(b^j), \forall j \geqslant i$
$\omega^i \models \Diamond\varphi$	$h(b^j) = \tilde{h}(b^j),\ \forall i \leqslant j \leqslant t^*$
$\omega^i \models \Box\varphi$	$h(b^j) = h_\varphi(b^j), \forall j \geqslant i$

TABLE I. LDTL specifications and the DTBF implementation.

A. Enforcing LDTL via DTBFs

In this section, we describe how the semantics of LDTL as given in Definition 3 can be represented as set invariance and reachability conditions over the belief simplex. The structure of the DTBFs for each specification are summarized in Table I.

We describe each row of the table as follows. (1) $\omega^i \models A \subset Q$: can be encoded as verifying whether $q^i \in A$. In the belief simplex, this is equivalent to checking whether $b^i \in \mathcal{B}_s = \{b^i \in \mathcal{B} \mid \sum_{q \in A} b^i(q) \geqslant 1\}$, which can be checked by considering the DTBF $h(b^i) = \sum_{q \in A} b^i(q) - 1$. (2) $\omega^i \models \neg A \subset Q$: can be cast as checking whether $b^i \in \mathcal{B}_s = \{b^i \in \mathcal{B} \mid \sum_{q \in Q\backslash A} b^i(q) \geqslant 1\}$ by considering the DTBF $h(b^i) = \sum_{q \in Q\backslash A} b^i(q) - 1$. (3),(4) $\omega^i \models f$ and $\omega^i \models \neg f$: these formulas are defined in the belief space, since $\omega^i \models f$ implies $f(b^i) < 0$ and $\omega^i \models \neg f$ implies $f(b^i) \geqslant 0$. They can be checked by considering DTBFs $h(b^i) = -f(b^i) + \delta$ with $0 < \delta << 1$ for $\omega^i \models f$ and $h(b^i) = f(b^i)$ for $\omega^i \models \neg f$. (5),(6) $\omega^i \models \varphi_1 \wedge \varphi_2$ and $\omega^i \models \varphi_1 \vee \varphi_2$: can be implemented by Boolean composition of the barrier functions as discussed in Section III-B. (7) $\omega^i \models \bigcirc\varphi$: can be implemented by checking whether φ is satisfied in the next step. (8) $\omega^i \models \varphi_1 \mathfrak{U} \varphi_2$: can be enforced by checking whether formula φ_1 is satisfied until φ_2. To this end, we can check whether formula φ_2 is not satisfied at every time step i by checking inequality $h_2(b) < 0$ where h_2 is the DTBF for formula φ_2. If φ_2 is not satisfied, then φ_1 is checked via a corresponding DTBF h_1. (9) $\omega^i \models \Diamond\varphi$: can be checked using the finite time DTBF given by Theorem 2. Note that the property is checked until t^*, since after t^* the formula φ is ensured to hold. (10) $\omega^i \models \Box\varphi$: can be simply

Algorithm 1 The one-step greedy algorithm for safety-shield synthesis given the nominal policy at every time-step t.

Input: System information I, Q, A, T, R, Z, O, nominal policy π_n, safety specifications defined by LDTL formula φ, current observation z^t, the past belief b^{t-1}

1: $b^t = BU(z^t, a_n^{t-1}, b^{t-1})$
2: **if** $h(b^t) - h(b^{t-1}) \geqslant -\alpha(h(b^{t-1}))$ and/or $\tilde{h}(b^t) - \rho\tilde{h}(b^{t-1}) \geqslant \varepsilon(1-\rho)$ **then**
3: **return** $a^* = a_n^t$
4: **else**
5: $i = 1$
6: **for** $i = 1, 2, \ldots, |A|$ **do**
7: $b^t = BU(z^t, a(i), b^{t-1})$
8: **if** $h(b^t) - h(b^{t-1}) \geqslant -\alpha(h(b^{t-1}))$ and/or $\tilde{h}(b^t) - \rho\tilde{h}(b^{t-1}) \geqslant \varepsilon(1-\rho)$ **then**
9: $r(i) = \left(\sum_{q' \in Q} b(q') R(q', a(i))\right)$
10: $i_* = \arg\min_{i=1,2,\ldots,|A|} \|r(i) - r_n^t\|^2$
11: **return** $a^* = a(i_*)$.

enforced by checking whether φ is satisfied for all time using the corresponding DTBF h_φ.

B. Safety-Shield Synthesis

Algorithm 1 illustrates how DTBFs can shield the agent actions to ensure LDTL safety. Note that, depending on the specification that needs to be enforced, either or both of (5) and (6) should be checked at every time step as described in Table I. The algorithm is described as follows. 1: At every time step t, it first computes the next joint belief b^t given the nominal action a_n designed based on the nominal policy π_n. 2: It then checks whether that action leads to a safe joint belief update. 3: If yes, the algorithm returns a_n for implementation. 5: If no, the algorithm picks a joint action $a(i)$ from $|A|$ combinations of actions (recall that $\times_{i \in I} A_i = A$). 6-7: For each joint action $a(i)$, it computes the next joint belief and 8: checks whether the next joint belief satisfies the LDTL specification. If the safety specification is satisfied, 9: it computes the corresponding reward function $r(i)$ for the joint action $a(i)$. 10: It then picks a safe joint action that minimally changes the immediate reward from the nominal immediate reward r_n^t in a least squares sense. This ensures that the decision making remains as much faithful as possible to the nominal policy (see [15] for analogous formulations for systems described by nonlinear differential equations).

V. Case Study: Multi-Robot Exploration

To demonstrate our method, we consider high-fidelity simulations of three heterogeneous ground-air robots [27], namely, a drone and two ground vehicles (a Rover Robotics Flipper and a modified Segway) exploring an unknown environment in ROS (see Figure 2(a)). The drone can rapidly explore the environment from above and it is used to locate a desired sample (goal). The Flipper is a small, tracked vehicle capable of traversing in rough terrain, whose job is to locate obstacles in the area. The Segway is larger, wheeled robot without external sensing capabilities, whose purpose is to retrieve the sample without colliding any obstacles. For the MPOMDP representation and more details on the setup, we refer the interested reader to Section V in [5].

The nominal policy used for the drone and the Flipper is a simple implementation of A*, that tries to maximize information gain by moving in new regions of the state space [29], hence exploring the environment. The Segway, on the other hand, is fed a constant action repeatedly, and relies on the safety-shield to reach the sample.

The first mission objective given to the Segway (located at q_S) is to not collide with the Flipper (located at q_F) with probability 0.9. This can be represented as $\square\neg f_1$, for $f_1 = 0.1 - b(q_S)b(q_F)$. The next requirement is that the Segway must not collide with the three obstacles (located at q_{o_i}, $i = 1, 2, 3$), again with probability 0.9. This can be enforced with the formula $\square\neg f_2$, for $f_2 = 0.1 - \wedge_{i=1}^{3} b(q_S)b(q_{o_i})$. To enforce these objectives as a single specification, the formula is $\square\neg(f_1 \vee f_2)$. Note that, if the agents meet this specification at time $t = 0$, then there always exists an action that meets this specification, as the agents can stop or remain in place.

In order to enforce LDTL formula $\square\neg(f_1 \vee f_2)$, we use the DTBF $h(b) = \min(f_1, f_2)$, where we used De Morgan's laws to obtain $\neg(f_1 \vee f_2) = \neg f_1 \wedge \neg f_2$, the fourth row of Table I, and Proposition 1. Figure 2(b) illustrates the safety shield enforcing this specification over the beliefs of the agents and the obstacles. Despite the obstacle being one cell away from the desired Segway position, the uncertainty stemming from the Flipper measurements of the obstacles as well as the state estimator of the Segway prevent the robot from moving into the desired position.

While the safety shield is able to keep the robots safe under this specification, there is no requirement of progression towards the objective, to retrieve the sample (located at q_G). Retrieving the sample with probability 0.5 can be written as $\diamondsuit f_3$, with $f_3 = 0.5 - b(q_S)b(q_G)$.

Combining all three objectives into one yields the final mission specification, given by the formula:

$$\varphi = \square\neg(f_1 \vee f_2) \wedge \diamondsuit f_3, \tag{11}$$

which ensures that the Segway *always* avoids the Flipper and the three obstacles with more than 0.90 probability and *eventually* reaches the goal with more than 0.5 probability.

The finite time DTBF $\tilde{h}(b) = b(q_S)b(q_G) - 0.5$ where we used the third and ninth rows of Table I to enforce $\diamondsuit f_3$. For the finite time DTBF condition (6), the parameters ρ and ϵ must be set to tune how quickly the set must be reached. To allow for more freedom of operation, we choose $\rho = 0.99$ and $\epsilon = 0.1$.

Figure 2(c-d) shows the results in our high-fidelity simulation environment. In particular, Figure 2(d) depicts the evolution of the DTBFs over the whole experiment. As it can be seen, for the nominal policy, the Segway fails to satisfy the mission specifications (since h becomes negative in many instances). However, with the safety-shield the satisfaction of mission specifications is guaranteed (h is always positive). Furthermore, the finite time DTBF becomes positive at the end of the experiment, which shows that the *eventually*

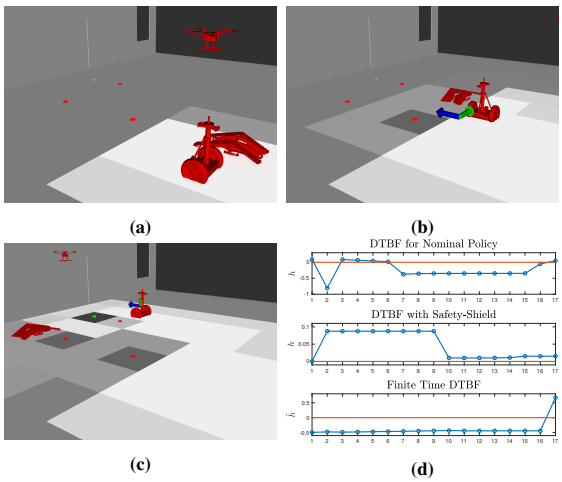

Fig. 2. Simulation results of the multi-agent system. **(a)** The initial positions of the three agents, obstacles (red), and sample (green). **(b)** Example of the nominal action (blue) being overwritten by the safety shield (green). **(c)** Updated costmaps reflected in grayscale after a longer period of exploration. **(d)** The plots of the DTBFs for the experiment, as explained above.

specification in (11) is satisfied. More information on this simulation can be found in the video here [1].

VI. Conclusions

We proposed a technique based on DTBFs to enforce safety/mission specifications in terms of LDTL formulas. Future research will explore policy synthesis for POMDPs ensuring both safety and optimality. To this end, we use receding horizon control with imperfects information [13] and control invariant set estimation of MPOMDPs via Lyapunov functions [3]. Multi-agent learning under partial observation [23] with safety guarantees is also a future direction.

References

[1] Video of the simulation. https://www.youtube.com/watch?v=7VyF7P-oM9I.

[2] A. Agrawal and K. Sreenath. Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Robotics: Science and Systems*, 2017.

[3] M. Ahmadi, N. Jansen, B. Wu, and U. Topcu. Control theory meets POMDPs: A hybrid systems approach. *arXiv preprint arXiv:1905.08095*, 2019.

[4] M. Ahmadi, R. Sharan, and J. W. Burdick. Stochastic finite state control of POMDP with LTL specifications. *arXiv preprint arXiv:2001.07679*, 2020.

[5] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames. Safe Policy Synthesis in Multi-Agent POMDPs via Discrete-Time Barrier Functions. *58th IEEE Conference on Decision and Control*, Dec 2019.

[6] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] C. Amato and F. A. Oliehoek. Scalable planning and learning for multiagent pomdps. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[8] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

[9] S. Carr, N. Jansen, R. Wimmer, A. C. Serban, B. Becker, and U. Topcu. Counterexample-guided strategy improvement for POMDPs using recurrent neural networks. *IJCAI*, 2019.

[10] K. Chatterjee, M. Chmelík, R. Gupta, and A. Kanodia. Qualitative analysis of POMDPs with temporal logic specifications for robotics applications. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 325–330. IEEE, 2015.

[11] K. Chatterjee, M. Chmelík, and M. Tracol. What is decidable about partially observable markov decision processes with ω-regular objectives. *Journal of Computer and System Sciences*, 82(5):878–911, 2016.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[13] N. E. Du Toit and J. W. Burdick. Robotic motion planning in dynamic, cluttered, uncertain environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 966–973, May 2010.

[14] P. Glotfelter, J. Cortés, and M. Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315, 2017.

[15] T. Gurriet, M. Mote, A. D. Ames, and E. Feron. An online approach to active set invariance. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3592–3599. IEEE, 2018.

[16] S. Haesaert, P. Nilsson, C. Vasile, R. Thakker, A. Agha-mohammadi, A.D. Ames, and R. M. Murray. Temporal logic control of pomdps via label-based stochastic simulation relations. *IFAC-PapersOnLine*, 51(16):271–276, 2018.

[17] M. Hasanbeig, Y. Kantaros, A. Abate, D. Kroening, G. J. Pappas, and I. Lee. Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees. In *IEEE Conference on Decision and Control*, 2019.

[18] N. Jansen, B. Könighofer, S. Junges, and R. Bloem. Shielded decision-making in MDPs. *arXiv preprint arXiv:1807.06096*, 2018.

[19] A. Jones, M. Schwager, and C. Belta. Distribution temporal logic: Combining correctness with quality of estimation. In *52nd IEEE Conference on Decision and Control*, pages 4719–4724. IEEE, 2013.

[20] S. Junges, N. Jansen, R. Wimmer, T. Quatmann, L. Winterer, J-P. Katoen, and B. Becker. Finite-state controllers of POMDPs via parameter synthesis. *Corvallis: AUAI Press*, 2018.

[21] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[22] M. Lahijanian, J. Wasniewski, S. B. Andersson, and C. Belta. Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees. In *2010 IEEE International Conference on Robotics and Automation*, pages 3227–3232. IEEE, 2010.

[23] M. Liu, K. Sivakumar, S. Omidshafiei, C. Amato, and J. P. How. Learning for multi-robot cooperation in partially observable stochastic environments with macro-actions. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1853–1860, 2017.

[24] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

[25] J. V. Messias, M. Spaan, and P. U. Lima. Efficient offline communication policies for factored multiagent pomdps. In *Advances in Neural Information Processing Systems*, pages 1917–1925, 2011.

[26] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[27] L. Rosa, M. Cognetti, A. Nicastro, P. Alvarez, and G. Oriolo. Multi-task cooperative control in a heterogeneous ground-air robot team. *IFAC-PapersOnLine*, 48(5):53–58, 2015.

[28] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

[29] A. Singletary, T. Gurriet, P. Nilsson, and A. Ames. Safety-critical rapid aerial exploration of unknown environments. *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[30] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[31] M. Srinivasan, S. Coogan, and M. Egerstedt. Control of multi-agent systems with finite time control barrier certificates and temporal logic. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1991–1996, Dec 2018.

[32] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta. Control in belief space with temporal logic specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7419–7424. IEEE, 2016.

[33] L. Winterer, S. Junges, R. Wimmer, N. Jansen, U. Topcu, J. Katoen, and B. Becker. Motion planning under partial observability using game-based abstraction. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2201–2208, Dec 2017.