Reinforcement Learning with Temporal Logic Constraints for Partially-Observable Markov Decision Processes

Yu Wang, Alper Kamil Bozkurt, and Miroslav Pajic

Abstract—This paper proposes a reinforcement learning method for controller synthesis of autonomous systems in unknown and partially-observable environments with subjective time-dependent safety constraints. Mathematically, we model the system dynamics by a partially-observable Markov decision process (POMDP) with unknown transition/observation probabilities. The time-dependent safety constraint is captured by iLTL, a variation of linear temporal logic for state distributions. Our Reinforcement learning method first constructs the belief MDP of the POMDP, capturing the time evolution of estimated state distributions. Then, by building the product belief MDP of the belief MDP and the limiting deterministic Büchi automaton (LDBA) of the temporal logic constraint, we transform the time-dependent safety constraint on the POMDP into a state-dependent constraint on the product belief MDP. Finally, we learn the optimal policy by value iteration under the state-dependent constraint.

I. INTRODUCTION

Reinforcement learning methods are widely used to synthesize control policies for autonomous systems that work in unknown environments (e.g., robotics and unmanned vehicles) [1]. Among them, model-free reinforcement learning methods are of particular interest since they can derive control policies without identifying the complete system model [1]. Instead, they can find the best control policy for a given discounted reward function by iteratively rolling out a tentative control policy and improving it based on observed system behaviors. For systems with fully observable states, algorithms have been developed for various tasks [2]–[4] including linear temporal logic tasks [5]–[9].

More often than not, the states of real-world autonomous systems such as self-driving cars [10] and robots [11] are not fully observable due to system/environment uncertainty (e.g., sensor noise). For these applications, a widely-used mathematical model for control synthesis is the partially observable Markov decision process (POMDP) [12]. A POMDP generalizes a Markov decision process (MDP), whose states are partially-observable through a probabilistic relation to a set of observations. Due to the partial observability, control synthesis is considerably harder for POMDP than fully observable models like Markov decision processes. Most of the exact decision problems on POMDP are either undecidable or PSPACE-complete [13], [14].

This work studies the control synthesis for POMDPs in the Bayesian framework. Instead of exhaustively consider all states agreeing with the observation (e.g., in [13], [14]), we based the control actions on the posterior state estimation, i.e., beliefs. For a given observation, the evolution of beliefs under the control actions is captured by a Markov decision process with infinitely many states. Thus, the Bayesian framework “lifts” the POMDP control problem into a (fully-observable) MDP control problem at the cost of expanding the state space. Accordingly, we value iteration methods to deal with the infinite product belief space [15], [16].

A crucial concern in learning-based control synthesis is dynamical safety. For safety-critical systems, such as self-driving cars [10] and robots [11], designing a policy that guarantees both safety and optimality is necessary. Typically, the safety constraints are time-dependent and expressible by linear temporal logic (LTL), a set of symbols and rules for formally representing and reasoning about time-dependent properties [17]. For different scenarios, variations of LTL in syntax and semantics are used [18].

This work considers safety constraints expressed by iLTL, a variation of LTL for state distributions. It has found applications in wireless sensor network [19] and cyber-physical systems [20], [21]. Compare to barrier certificates [22], iLTL is more expressive for time-dependent safety constraints. We use iLTL to capture safety constraints for posterior state estimations (i.e., beliefs) of the POMDP. Namely, when synthesizing the optimal control for a given discounted reward, if we “believe” an action will violate the iLTL safety constraint, we should not take it.

We propose a reinforcement learning method to derive the optimal control policy for a given discounted reward under an iLTL safety constraint. By constructing the belief MDP of the POMDP, we first lift the control synthesis problem to the belief space. Then, we build the product belief MDP of the belief MDP and the limiting deterministic Büchi automaton (LDBA) of the temporal logic constraint. This transforms the time-dependent iLTL constraint to a state-dependent Büchi constraint on the product belief MDP [8]. Finally, we propose a value iteration method to learn the optimal policy for the discounted reward under the Büchi constraint. An overview of our approach is shown by Figure 1.

The rest of the paper runs as follows. We give the definition of POMDP in Section II and formula the control synthesis problem Section III. We introduce the LDBA and build the product belief MDP in Section IV. Then, we introduce the value iteration under constraints and the learning algorithm in Section V. Finally, we conclude this work in Section VI.

The authors are with the Pratt School of Engineering of Duke University, Durham, NC 27708, USA. Emails: {yu.wang094, alper.bozkurt, miroslav.pajic}@duke.edu

This work is sponsored in part by the ONR under agreements N00014-17-1-2504, N00014-20-1-2745 and N00014-18-1-2374, AFOSR award number FA9550-19-1-0169, and the NSF CNS-1652544 grant.
We model an autonomous system’s dynamics in the unknown and partially-observable environment by a partially-observable Markov decision process (POMDP), where the underline dynamics is a Markov decision process, but the control can only depend on observations probabilistically related to the states.

For a finite $S$, we denote the set of probability distributions on $S$ by Dist$(S)$. A POMDP is a tuple $\mathcal{M} = (S, A, T, p_0, r, \gamma, O, \Omega)$, where

- $S = \{s_1, \ldots, s_n\}$ is a set of $n$ states.
- $A = \{a_1, \ldots, a_m\}$ is a set of $m$ actions.
- $T : S \times A \times S \rightarrow [0, 1]$ is a set of transition probabilities between states,\(^1\) satisfying for any $s \in S$ and any $a \in A,$

$$\sum_{s' \in S} T(s, a, s') = 1.$$ 

By taking the action $a \in A$ on the state $s \in S$, the probability of transitioning to the state $s' \in S$ is $T(s, a, s')$.

- $p_0 \in \text{Dist}(S)$ is an initial distribution.
- $r : S \rightarrow \mathbb{R}$ is the immediate reward function.
- $\gamma : S \rightarrow (0, 1)$ is a discount factor.
- $O = \{o_1, \ldots, o_l\}$ is a set of $l$ observations.
- $\Omega : S \times O \rightarrow \mathbb{R}$ is a set of observation probabilities,\(^2\) satisfying for any $s \in S$

$$\sum_{o \in O} \Omega(s, o) = 1.$$ 

On the state $s$, the probability of yielding an observation $o$ is $\Omega(s, o)$.

When $S = O$ and $\Omega(s, o) = 1$ if and only if $s = o$, the POMDP $\mathcal{M}$ reduces to a Markov decision process (MDP). We call a sequence of states $\sigma : \mathbb{N} \rightarrow S$ a path of the POMDP if for any $t \in \mathbb{N}$, there exists $a \in A$ such that $T(s(t), a, s(t+1)) > 0$. In addition, we call a sequences of state distributions $\sigma : \mathbb{N} \rightarrow \text{Dist}(S)$ as an execution of the POMDP.

### III. Problem Formulation

We consider the control synthesis problem on the POMDP from Section II. Mathematically, the control policy

$$\pi : O^* \rightarrow \text{Dist}(A) \tag{1}$$

decides (probabilistically) the action to take from the history of observations. We denote the POMDP under the control policy by $\mathcal{M}_\pi$ and a random path drawn from the controlled POMDP by $\sigma \sim \mathcal{M}_\pi$. The control goal is to maximize the expected cumulative reward of the random path $\sigma$; equivalently, we look for the optimal control policy $\pi^*$ such that

$$\pi^* = \arg\max_\pi \mathbb{E}_{\sigma \sim \mathcal{M}_\pi} R(\sigma), \tag{2}$$

where

$$R(\sigma) = \sum_{t=1}^{\infty} \gamma^t r(\sigma(t)). \tag{3}$$

### A. Belief Markov Decision Processes

The states are not directly observable in a POMDP. But, from a history of observation, one can derive a probabilistic estimation of the states from the history of observations is called a belief $b \in \text{Dist}(S)$. At $t = 0$, the belief $b_0$ is the initial distribution, i.e., $b_0 = p_0$. For $t > 0$, the belief $b_t$ updates upon observing $o$ by the Bayes rule

$$b_{t+1}(s') = \frac{\Omega(s', o) \sum_{s \in S} T(s, a, s') b_t(s)}{\sum_{s' \in S} \Omega(s', o) \sum_{s \in S} T(s, a, s') b_t(s)} \tag{4}$$

Following (4), the POMDP $\mathcal{M}$ induces a Markov decision process (MDP) on the belief space Dist$(S)$ where each belief (i.e., state distribution) is a state.\(^3\) It shows how the belief changes for taking different actions. The belief MDP is fully observable.

**Definition 1.** The belief MDP $\mathcal{D}$ of the POMDP $\mathcal{M} = (S, A, T, p_0, r, \gamma, O, \Omega)$ is defined by $\mathcal{D} = (\text{Dist}(S), A, T_D, p_0, r_D, \gamma)$ where

$$T_D(b, a, b') = \sum_{o \in O} \sum_{s \in S} \sum_{s' \in S} \eta(b, o, b') \Omega(s', o) T(s, a, s') b(s)$$

with

$$\eta(b, o, b') = \begin{cases} 1, & \text{if the belief update for } b, o \text{ by } (4) \text{ returns } b', \\ 0, & \text{otherwise}, \end{cases}$$

and

$$r_D(b, a) = \sum_{s \in S} b(s) r(s, a)$$

for $b, b' \in \text{Dist}(S)$ and $a \in A$.

### B. Time-Dependent Safety Constraints

We formally capture time-dependent safety constraints by temporal logic. For the probabilistic safety related to state distributions, we introduce a variant of the linear temporal logic (LTL) to capture time-dependent specifications for wireless sensor network [19] and cyber-physical systems [20, [21].

An inequality LTL (iLTL) formula is derived recursively from the rules

$$\varphi ::= f \mid \lnot \varphi \mid \varphi \land \varphi \mid \bigotimes \varphi \mid \varphi U_T \varphi \tag{5}$$

\(^3\)The resulting belief MDP is a continuous state space, even if the “originating” POMDP has a finite number of states.
where
- \( f : \mathbb{R}^n \to \mathbb{R} \) is a (given) function and is called an atomic proposition.
- \( T \in \mathbb{N} \) and \( \bigcirc \) and \( U \) are temporal operators, meaning “next” and “until”, respectively.

Other common logic operators can be derived as follows:
\[
\begin{align*}
\text{True} & \equiv \varphi \wedge \neg \varphi, \\
\varphi \wedge \varphi' & \equiv \neg(\neg \varphi \wedge \neg \varphi'), \\
\varphi \rightarrow \varphi' & \equiv \neg \varphi \wedge \varphi', \\
\bigcirc \neg \varphi & \equiv \text{True} U \neg \varphi, \quad \text{and} \quad \bigcirc \bigcirc \varphi \equiv \neg \bigcirc \bigcirc \neg \varphi.
\end{align*}
\]

Also, we denote \( U \sigma, \bigcirc \sigma \), and \( \bigcirc \bigcirc \sigma \) by \( U \), \( \bigcirc \) and \( \bigcirc \bigcirc \), respectively. Our definition of iLTL is more general than [19]–[21], since we allow atomic propositions to be present in the \( \bigcirc \) operator.

Given the MDP from Section II and the iLTL constraint to maximize the discounted reward (3), we formally introduce the problem formulation below.

\[
\text{Problem 1. Given the MDP from Section II and the iLTL constraint } \varphi \text{ from Section III-B, find a policy } \pi \text{ in the form of (1) maximizing the expected value of the discounted reward (3), while ensuring the belief sequence } b_0b_1 \ldots \text{ derived from (4)} \text{ satisfies } \varphi.
\]

In Problem 1, the constraint \( b_0b_1 \ldots \models \varphi \) means when synthesizing the optimal control for a given discounted reward, if we “believe” an action will violate the iLTL safety constraint, we should not take it. Besides, in our Bayesian framework discussed in Section III-A, the control policy \( \pi : O^* \to \text{Dist}(A) \) depends on past observations through belief updates (which are a sufficient statistic). Thus, to maximize the discounted reward (3) (without the safety constraint \( \varphi \)), we only need the belief sequence \( b_0b_1 \ldots \).

In addition, since the satisfaction of the safety constraint \( \varphi \) also involves the belief sequence \( b_0b_1 \ldots \), it suffices to consider a policy mapping belief sequences to actions. This is summarized by the following lemma.

**Lemma 1.** To solve Problem 1, it suffices to find a policy
\[
\pi : \text{Dist}(S)^* \to A.
\]

**IV. Product Belief MDP**

Following Lemma 1, the control policy that solves Problem 1 depends on the past belief sequence, so it is memory-dependent; this is beyond the capability of reinforcement learning. To remove the memory dependency, we generalize the product technique from [8] to belief MDPs that have infinite states.

A. Limiting-Deterministic Büchi Automata

An LDBA is a tuple \( A = (Q, \Sigma, \delta, q_0, B) \) where
- \( Q \) is a finite set of states;
- \( \Sigma \) is a finite set of alphabets;
- \( \delta : Q \times (\Sigma \cup \{\varepsilon\}) \to Q \) is a (partial) transition function (i.e., all alphabets are allowed on each state) with \( \varepsilon \) standing for the empty alphabet;
- \( q_0 \in Q \) is an initial state;
- \( B \) is a set of accepting states.

The LDBA satisfies that
- the transition \( \delta \) is total except for the empty alphabet, i.e., \( |\delta(q, \cdot)| = 1 \) for any \( q \in Q \);
- there exists a bipartition of into an initial and accepting component, i.e., \( Q = Q_A \cup Q_I \) such that
  - transitions from the accepting component stay within it, i.e., \( \delta(q, \cdot) \subseteq Q_A \) for any \( q \in Q_A \);
  - the accepting states are in the accepting component, i.e., \( B \subseteq Q_A \);
- the \( \varepsilon \)-moves are not allowed in the accepting component, i.e., \( \delta(q, \varepsilon) = \emptyset \) for any \( q \in Q_A \).

We call \( q : \mathbb{N} \to Q \) a path of the LDBA if \( q(0) = q_0 \) and for any \( t \in \mathbb{N} \), there exists \( \sigma \in \Sigma \) such that \( q(t+1) = \delta(q(t), \sigma) \).

B. Product Belief MDP

**Definition 2.** The product belief MDP \( \prod^\times = \mathcal{D} \times A \) of the belief MDP \( \mathcal{D} = (\text{Dist}(S), A, T_D, p_0, r_D, \gamma) \) and an LDBA

\[\]
Theorem 1. The product belief MDP has an infinite states space $\text{Dist}(S) \times Q$. Thus the tabular reinforcement learning method does not apply. Here, we generalize the value iteration method [15], [25] to solve for the optimal control policy under the constraint.

A. Bellman Equation on Belief Space

We define the value function for the reward $r^\pi$ on the product belief space by

$$V_r((b, q)) = \max_{\pi^\pi} \mathbb{E}_{\pi^\pi} \sum_{t \in \mathbb{N}} \gamma^t r^\pi(\sigma(t))$$

where $\pi^\pi \sim \mathcal{D}^\pi_{\pi^\pi}(b, q)$ is a random path drawn from $\mathcal{D}^\pi_{\pi^\pi}(b, q)$ under the policy $\pi$ from the product state $(b, q)$. The value function $V_r$ captures the maximal expected value of the reward (3) if started from the product state $(b, q)$. By Theorem 1, $V_r(b, q)$ is the maximal expected reward of (3) on the POMDP $\mathcal{M}$.

In addition, we define the value function for the Büchi constraint on the product belief space by

$$V_p((b, q)) = \max_{\pi^\pi} \mathbb{P}_{\pi^\pi} \sum_{t \in \mathbb{N}} \gamma^t \mathbb{P}(\sigma(t))$$

and similarly $\pi^\pi$. From Corollary 1, it suffices to consider pure and memoryless policies for maximizing the two value functions (14) and (15). Accordingly, they satisfy the following Bellman equations

$$V_r((b, q)) = \max_{\sigma \in A} Q_r((b, q), a)$$

$$Q_r((b, q), a) = r^\pi((b, q)) + \gamma \sum_{q' \in Q} \int_{b' \in \text{Dist}(S)} T^\pi(((b, q), a), ((b', q'))) V_r((b', q')) db'$$

and

$$V_p((b, q)) = \max_{\sigma \in A} Q_p((b, q), a)$$

$$Q_p((b, q), a) = \sum_{q' \in Q} \int_{b' \in \text{Dist}(S)} T^\pi(((b, q), a), ((b', q'))) V_p((b', q')) db'$$

Furthermore, the maxima of the right-hand side of (12) and (13) can be memoryless. Despite the existence of $\varepsilon$ actions, this memoryless policy maps to a memory-dependent policy on the belief MDP, which maximizes the left-hand side of (12) and (13), as formally stated below.

Corollary 1. A memoryless policy maximizes the right-hand side of (12) and (13). It induces a memory-dependent policy that maximizes the left-hand side of (12) and (13).

Proof. Follow from the proof Theorem 3 in [24].

V. LEARNING UNDER CONSTRAINTS

The product belief MDP has an infinite states space $\text{Dist}(S) \times Q$. Thus the tabular reinforcement learning method does not apply. Here, we generalize the value iteration method [15], [25] to solve for the optimal control policy under the constraint.

A. Bellman Equation on Belief Space

We define the value function for the reward $r^\pi$ on the product belief space by

$$V_r((b, q)) = \max_{\pi^\pi} \mathbb{E}_{\pi^\pi} \sum_{t \in \mathbb{N}} \gamma^t r^\pi(\sigma(t))$$

where $\pi^\pi \sim \mathcal{D}^\pi_{\pi^\pi}(b, q)$ is a random path drawn from $\mathcal{D}^\pi_{\pi^\pi}(b, q)$ under the policy $\pi$ from the product state $(b, q)$. The value function $V_r$ captures the maximal expected value of the reward (3) if started from the product state $(b, q)$. By Theorem 1, $V_r(b, q)$ is the maximal expected reward of (3) on the POMDP $\mathcal{M}$.

In addition, we define the value function for the Büchi constraint on the product belief space by

$$V_p((b, q)) = \max_{\pi^\pi} \mathbb{P}_{\pi^\pi} \sum_{t \in \mathbb{N}} \gamma^t \mathbb{P}(\sigma(t))$$

and similarly $\pi^\pi$. From Corollary 1, it suffices to consider pure and memoryless policies for maximizing the two value functions (14) and (15). Accordingly, they satisfy the following Bellman equations

$$V_r((b, q)) = \max_{\sigma \in A} Q_r((b, q), a)$$

$$Q_r((b, q), a) = r^\pi((b, q)) + \gamma \sum_{q' \in Q} \int_{b' \in \text{Dist}(S)} T^\pi(((b, q), a), ((b', q'))) V_r((b', q')) db'$$

and

$$V_p((b, q)) = \max_{\sigma \in A} Q_p((b, q), a)$$

$$Q_p((b, q), a) = \sum_{q' \in Q} \int_{b' \in \text{Dist}(S)} T^\pi(((b, q), a), ((b', q'))) V_p((b', q')) db'$$

Furthermore, the maxima of the right-hand side of (12) and (13) can be memoryless. Despite the existence of $\varepsilon$ actions, this memoryless policy maps to a memory-dependent policy on the belief MDP, which maximizes the left-hand side of (12) and (13), as formally stated below.

Corollary 1. A memoryless policy maximizes the right-hand side of (12) and (13). It induces a memory-dependent policy that maximizes the left-hand side of (12) and (13).

Proof. Follow from the proof Theorem 3 in [24].
For solving (18), we also have
\[ V_p((b, q)) = 1 \text{ for } (b, q) \in B^\times \] (19)
where \( B^\times \) is given by (11), according to iLTL syntax.

**B. Value Iteration**

Our learning method aims to find solutions to the Bellman equations (17) and (18) by sampling, without using knowledge on the transition probabilities \( T^\times \). Since the product belief space \( \text{Dist}(S) \times Q \) is infinite, tabular learning methods are not applicable. Instead, we generalize the value iteration method [25] to the product belief space. Our method is based on the fact that the value functions and Q-functions are piecewise linear, as stated below.

**Lemma 3.** The value functions \( V_r((b, q)) \) and \( V_p((b, q)) \) and Q-functions \( Q_r((b, q), a) \) and \( Q_p((b, q), a) \) are convex and piecewise linear in \( b \).

**Proof.** It suffices to prove the statement for each \( q \in Q \), which follows directly from [25].

Based on Lemma 3, we can represent the Q-function \( Q_r \) (and the same for \( Q_p \)) on the product space by
\[ Q_r((b, q), a) = \max_{\theta \in \Theta_{q,a}} \sum_{s \in S} \theta(s)b(s) \] (20)
\[ V_r((b, q)) = \max_{\theta \in \Theta_q} \sum_{s \in S} \theta(s)b(s) \] (21)
where each \( \Theta_q \) and \( \Theta_{q,a} \) (for \( q \in Q \) and \( a \in A \)) is a finite set of \( |S| \)-dimensional vectors, which define hyperplanes on the belief space. Following (16), we can take
\[ \Theta_q = \bigcup_{a \in A} \Theta_{q,a}. \] (22)

Our goal is to identify the set \( \Theta_{q,a} \) (thus \( \Theta_q \)). By plugging (20)-(22) into the Bellman equation (17) and using Definition 2, we have
\[ Q_r((b, q), a) = \sum_{s \in S} r(s)b(s) + \gamma \int_{b' \in \text{Dist}(S)} T_D(b, a, b') \max_{\theta \in \Theta_{q,a}} \sum_{s \in S} \theta(s)b'(s) \lesssim \theta \] (23)
where \( q' = \delta(q, L(b)) \). The updated \( Q_r \) by (23) is again piecewise linear and convex in \( b \).

The update rule (23) is not directly usable since the belief transition probabilities \( T_D \) are unknown. For learning, we can replace \( T_D \) with its empirical estimation. Suppose starting from the belief state \( b \), we take the action \( a \) repeatedly for \( n \) times, and derive \( n \) observations \( o_1, \ldots, o_n \) and correspondingly \( n \) updated beliefs \( b'_1, \ldots, b'_n \) by (4). Then, the empirical estimation of \( T_D \) is given by
\[ \hat{T}_D = \frac{1}{n} I(b, a, b'_i) \] (24)
where \( I \) is the indicator function.

By applying (24) to (23), we derive
\[ Q_r((b, q), a) \leftarrow \sum_{s \in S} r(s)b(s) + \gamma \sum_{i=1}^n \max_{\theta \in \Theta_{q,a}} \sum_{s \in S} \theta(s)b'_i(s) \] (25)
where \( q' = \delta(q, L(b)) \). Accordingly, we update \( \Theta_{q,a} \) by
\[ \Theta_{q,a} \leftarrow \Theta_{q,a} \cup \left\{ \gamma \sum_{i=1}^n \arg\max_{\theta \in \Theta_{q,a}} \sum_{s \in S} \theta(s)b'_i(s) \right\} \] (26)
The updated \( \Theta_{q,a} \) may contain duplicated vectors dominated by others in taking the maximum in (20). We can prune them by linear programming.

**Theorem 2.** By iteratively updating \( \Theta_{q,a} \) by (26) for all \( q \in Q \) and pruning, the Q-function \( Q_r \) and value function \( V_r \) converges. The same holds for \( Q_p \) and \( V_p \).

**Proof.** First, as the number of samples \( n \to \infty \), we have \( \hat{T}_D \to T_D \). Then by repeatedly taking all \( q \in Q \), the value functions \( V_r((\cdot, q)) \) (or \( V_p \)) converges for each \( q \) by [25].

**Remark 1.** We can reduce the pruning complexity by choosing a finite set of witness beliefs \( W \subseteq \text{Dist}(S) \). We keep \( \theta \in \Theta_{q,a} \) if and only if it defines the value functions at some \( w \in W \), i.e., \( \theta = \arg\max_{\theta \in \Theta_{q,a}} \sum_{s \in S} \theta(s)w(s) \). The point-based pruning method is computationally simpler at the cost of introducing a bounded error related to the density of \( W \) (see [15] for details).

**C. Learning Algorithm**

We now present a learning method to solve Problem 1 when the transition probabilities of the POMDP \( \mathcal{M} \) is unknown. Our approach simultaneously runs two learning algorithms on the product belief MDP \( D^\times \) to solve for the Q-functions \( Q_r \) and \( Q_p \) (and the value functions \( V_r \) and \( V_p \)). For \( Q_r \), we use Q-learning with \( \epsilon \)-greedy policy exploration. Meanwhile, we use the Q-learning method from [8] to update \( Q_p \). The Q-function \( Q_p \) determines the maximal satisfaction probability of the iLTL constraint \( \varphi \). Therefore, to ensure the absolute satisfaction of \( \varphi \), only the actions from
\[ A_{safe}((b, q)) = \{ a \in A \mid Q_p((b, q), a) = 1 \} \] (27)
are allowable on the product belief state \((b, q)\) for the learning of \( Q_p \) (excluding those \( \epsilon \)-greedy policy explorations).

We implement our learning method in an off-policy fashion for generality. We keep track of all previously-sampled episodes
\[ \Xi \leftarrow \Xi \cup \{ ((b, q), a, (b', q')) \} \] (28)
and update the empirical transition probabilities of beliefs \( T_D \) accordingly. The new samples are drawn by randomly choosing a product belief \((b, q)\) that has appeared in \( \Xi \). The corresponding action is selected as described above.

The overall learning method is presented by Algorithm 1. Finally, we can derive the policy that solves Problem 1 by the discussion in Section IV-B using the learned Q-functions \( Q_r \) and \( Q_p \) from Algorithm 1.
Algorithm 1 Reinforcement learning on product belief MDP

1: **Input** iLTL constraint \( \varphi \), POMDP \( \mathcal{M} \)
2: Build Belief MDP \( \mathcal{D} \) for POMDP \( \mathcal{M} \)
3: Build LDBA \( \mathcal{A} \) for iLTL formula \( \varphi \)
4: Build product belief MDP \( \mathcal{D} \times A \)
5: Set parameters \( T \gg 1 \) and \( \epsilon \ll 1 \)
6: Initialize \( \Theta_{q,a} \) for \( Q_r \) and \( \Theta_{q,a} \) for \( Q_p \)
7: \((b,q) \leftarrow (p_0,q_0)\)
8: while not converge do
9: \( \text{Get } A_{\text{safe}} \text{from (27)} \)
10: \( \text{Choose } a \in A_{\text{safe}}((b,q)) \text{ with probability } 1 - \epsilon \)
11: or any other in \( A \) with probability \( \epsilon \)
12: \( \text{Take } a \text{ and observe } o \)
13: \( \text{Compute } b' \text{ by (4) and } q' \leftarrow \delta(q,L(b)) \text{ with (9)} \)
14: \( \text{Update } \Xi \text{ by (28) and } T_D \text{ by (24)} \)
15: \( \text{Update and prune } \Theta_{q,a} \text{ (and } \Theta'_{q,a} \text{) by (26)} \)
16: \( \text{Randomly pick new } (b,q) \text{ that has appeared in } \Xi \)
17: end while
18: Return \( \Theta_{q,a} \) and \( \Theta'_{q,a} \)

Theorem 3. For a given POMDP \( \mathcal{M} \) and iLTL safety constraints \( \varphi \), Algorithm 1 converges.

Proof. By Theorem 2, both \( Q_r \) and \( Q_p \) converges in Algorithm 1, thus the claim holds. \( \square \)

Remark 2. In the pruning step of Algorithm 1, we may use the beliefs in \( \Xi \) as the witness beliefs (as discussed in Remark 1), since those beliefs are the most important for learning. We will study this problem in future work.

VI. CONCLUSION

This paper proposed a reinforcement learning method for controller synthesis of autonomous systems in unknown and partially-observable environments with subjective time-dependent safety constraints. We modeled the system dynamics by a partially-observable Markov decision process (POMDP) with unknown transition/observation probabilities and the time-dependent safety constraint by linear temporal logic formulas. Our Reinforcement learning method first constructed the belief MDP of the POMDP. Then, by building the product belief MDP of the belief MDP and the LDBA of the temporal logic constraint, we transformed the time-dependent safety constraint on the POMDP into a state-dependent constraint on the product belief MDP. Finally, we proposed a learning method for the optimal policy under the state-dependent constraint.

REFERENCES

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. MIT press, 2018.
[2] J. Koher, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” The International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.
[3] J. Koher and J. Peters, “Learning motor primitives for robotics,” in 2009 IEEE International Conference on Robotics and Automation, May 2009, pp. 2112–2118.
[4] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-End Training of Deep Visuomotor Policies,” Journal of Machine Learning Research, vol. 17, no. 1, pp. 1334–1373, 2016.
[5] X. Li, C.-I. Vasile, and C. Belca, “Reinforcement Learning With Temporal Logical Rewards,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017, pp. 3834–3839.
[6] E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak, “Omega-Regular Objectives in Model-Free Reinforcement Learning,” in Tools and Algorithms for the Construction and Analysis of Systems, T. Vojnar and L. Zhang, Eds. Cham: Springer International Publishing, 2019, vol. 11427, pp. 395–412.
[7] 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,” IEEE 58th Conference on Decision and Control (CDC), 2019, pp. 5338–5343.
[8] A. K. Bozkurt, Y. Wang, M. Zavlanos, and M. Pajic, “Control synthesis from linear temporal logic specifications using model-free reinforcement learning,” in IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 10349–10355.
[9] ——, “Model-Free Reinforcement Learning for Stochastic Games with Linear Temporal Logic Objectives,” in International Conference on Robotics and Automation, 2021.
[10] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, “Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction,” IEEE Transactions on Intelligent Vehicles, vol. 3, no. 1, pp. 5–17, Mar. 2018.
[11] A. K. Bozkurt, Y. Wang, and M. Pajic, “Secure Planning Against Stealthy Attacks via Model-Free Reinforcement Learning,” in IEEE International Conference on Robotics and Automation (ICRA), 2021.
[12] L. P. Kaebibling, M. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” Artificial Intelligence, vol. 101, no. 1, pp. 99–134, May 1998.
[13] O. Madani, S. Hanks, and A. Condon, “On the Uncertainty of Probabilistic Planning and Infinite-Horizon Partially Observable Markov Decision Problems,” in AAAI Conference on Artificial Intelligence, 1999, pp. 541–548.
[14] C. H. Papadimitriou and J. N. Tsitsiklis, “The Complexity of Markov Decision Processes,” Mathematics of Operations Research, vol. 12, no. 3, pp. 441–450, Aug. 1987.
[15] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for POMDPs,” in International Joint Conference on Artificial Intelligence, vol. 3, 2003, pp. 1025–1032.
[16] S. Ji, R. Parr, H. Li, X. Liao, and L. Carin, “Point-Based Policy Iteration,” in AAAI Conference on Artificial Intelligence, 2007, pp. 1243–1249.
[17] A. Pnueli, “The temporal logic of programs,” in 18th Annual Symposium on Foundations of Computer Science, 1977, pp. 46–57.
[18] A. Alur, A. Singletary, J. W. Burdick, and A. D. Ames, “Barrier Functions for Multiagent-POMDPs with DTL Specifications,” in IEEE Conference on Decision and Control, 2020, pp. 1380–1385.
[19] Y. Kwon and G. Agha, “Linear Inequality LTL (iLTL): A Model Checker for Discrete Time Markov Chains,” in Formal Methods and Software Engineering, ser. Lecture Notes in Computer Science, J. Davies, W. Schulte, and M. Barnett, Eds. Springer Berlin Heidelberg, 2004, vol. 3308, pp. 194–208.
[20] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, “Statistical verification of dynamical systems using set oriented methods,” in ACM International Conference on Hybrid Systems: Computation and Control (HSCC), New York, NY, USA, 2015, pp. 169–178.
[21] ——, “Verifying continuous-time stochastic hybrid systems via Morizwanzig model reduction,” in IEEE Conference on Decision and Control (CDC), 2016, pp. 3012–3017.
[22] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control Barrier Functions: Theory and Applications,” in European Control Conference, 2019.
[23] Y. Wang and M. Pajic, “Hyperproperties for robotics: Motion planning via HyperLTL,” in IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 8462–8468.
[24] S. Sickert, J. Esparza, S. Jaax, and J. Křetínský, “Limit-Deterministic Büchi Automata for Linear Temporal Logic,” in Computer Aided Verification, S. Chaudhuri and A. Farzan, Eds. Cham: Springer International Publishing, 2016, vol. 9780, pp. 312–322.
[25] R. D. Smallwood and E. I. Sondik, “The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon,” Operations Research, vol. 21, no. 5, pp. 1071–1088, 1973.