Proceedings of Machine Learning Research vol 120:1–10, 2020

Tractable Reinforcement Learning of Signal Temporal Logic Objectives

Harish Venkataraman KUMAA001@UMN.EDU
Derya Aksaray DAKSARAY@UMN.EDU
Peter Seiler SEILE017@UMN.EDU
Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA

Abstract

Signal temporal logic (STL) is an expressive language to specify bounded time real-world robotic tasks and safety specifications. Recently, there has been an interest in learning optimal policies to satisfy STL specifications via reinforcement learning. Learning to satisfy STL specifications often needs a sufficient length of state history to compute reward and the next action. The need for history results in exponential state-space growth for the learning problem. Thus the learning problem becomes computationally intractable for most real-world applications. In this paper, we propose a compact means to capture state history in a new augmented state-space representation. An approximation to the objective (maximizing probability of satisfaction) is proposed and solved for in the new augmented state-space. We show the performance bound of the approximate solution and compare it with the solution of an existing technique via simulations.
Keywords: Reinforcement Learning (RL), Formal Methods, Signal Temporal Logic (STL)

1. Introduction

Reinforcement learning (RL) for controlling unknown or partially known stochastic dynamical systems to satisfy complex bounded time objectives has gained good momentum in the robotics community, e.g., use of an aerial vehicle for infrastructure inspection or environmental monitoring while maintaining a sufficient state of charge throughout the mission. Such complex objectives can be rigorously expressed by temporal logics (TL).

In the literature, there are numerous works on model-based control synthesis for the satisfaction of TL specifications (e.g., Ding et al. (2014); Sadigh et al. (2014); Fu and Topcu (2014); Lahijanian et al. (2015); Aksaray et al. (2015, 2016b)). Recently, model-free learning paradigm to satisfy TL specifications has also gained a significant interest (e.g., using RL to find policies that maximize the probability of satisfying a given linear temporal logic (LTL), Brazdil et al. (2014); Sadigh et al. (2014); Fu and Topcu (2014); Li et al. (2017); Li and Belta (2019); Xu and Topcu (2019)). *Q*-learning, a variant of RL has also been shown to be successful in learning policies satisfying signal temporal logic specifications [Aksaray et al. (2016a)].

Signal temporal logic (STL) is a rich predicate logic that can specify bounds on physical parameters and time intervals [Maler and Nickovic (2004)]. For example, an autonomous UAV needs to recharge itself periodically (every 15 minutes in a 2-hour mission) to avoid crashing onto the ground. Such a bounded time objective can be expressed by STL as $G_{[0,105\ min]}F_{[0,15\ min]}f(x,y) \in \mathscr{C}$ where $f(x,y) \in \mathscr{C}$ refers to the vehicle position (x,y) being inside a recharging station.

STL does not have a graph representation such as an automaton to book-keep history. Thus, Aksaray et al. (2016a) constructed a higher dimensional Markov Decision Process (MDP) model, known as τ-MDP, for learning. This τ-MDP model stores the state history (including the current state) over a finite window of length τ and enables to compute a reward and action at each time step. For instance, if the specification requires visiting region B after region A within 10-time steps, then STL satisfaction can be verified with the knowledge of at least 10-time steps. The number of states in the τ-MDP model grows exponentially with the size of τ. For example, if the original MDP has m states, τ-MDP has m^τ states. This state-space explosion renders learning on τ-MDP model impractical for real-world robotics problems with large state-space and long STL horizons.

The primary focus of this paper is to provide a new augmented system on which learning to satisfy STL specifications is more computationally tractable and thus could scale to problems with longer STL horizons. The basic idea is that both rewards and actions can be computed without exact state history. The reward and the next action can be computed based on the current state and newly defined notion of flags. The flags capture the historic knowledge of the partial satisfaction for each STL sub-formula constituting the STL specification. The new augmented system is defined as a new MDP known as F-MDP, which holds the actual system states and the flag states.

We formulate a learning problem over F-MDP in place of τ-MDP [Aksaray et al. (2016a)] and propose a technique to learn the optimal policy maximizing the probability of satisfaction. The proposed technique is shown to have polynomial space complexity as a function of τ. Empirical results also support faster learning due to the compactness of F-MDP. The rest of this paper is organized as follows: Sections 2 introduces the key concepts, Section 3 defines the problem formally, Section 4 describes the proposed technique in detail, analyzes the performance and computational complexity of the proposed technique, Section 5 provides simulation results and finally Section 6 concludes with future prospects.

2. Preliminaries

2.1. Signal Temporal Logic (STL)

In this paper, the desired system behavior is described by an STL fragment with the following *syntax*

$$\begin{aligned} \Phi \quad &:= F_{[a,b]}\phi | G_{[a,b]}\phi \\ \phi \quad &:= \phi \wedge \phi | \phi \vee \phi | F_{[c,d]}\varphi | G_{[c,d]}\varphi \\ \varphi \quad &:= \psi | \neg\varphi | \varphi \wedge \varphi | \varphi \vee \varphi, \end{aligned} \tag{1}$$

where $a,b,c,d \in \mathbb{R}_{\geq 0}$ are finite non-negative time bounds; Φ, ϕ, and φ are STL formulae; ψ is predicate in the form of $f(\mathbf{s}) < d$ where $\mathbf{s} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is a signal, $f : \mathbb{R}^n \to \mathbb{R}$ is a function, and $d \in \mathbb{R}$ is a constant. The Boolean operators $\neg$, $\wedge$, and $\vee$ are negation, conjunction (i.e., *and*), and disjunction (i.e., *or*), respectively. The temporal operators F and G refer to *Finally* (i.e., eventually) and *Globally* (i.e., always), respectively. The reader is referred to Maler and Nickovic (2004) for details on STL.

For any signal $\mathbf{s}$, let s_t denote the value of $\mathbf{s}$ at time t and let $(\mathbf{s},t)$ be the part of the signal that is a sequence of $s_{t'}$ for $t' \in [t,\infty)$. Accordingly, the *Boolean semantics* of STL is recursively defined as follows:

$$\begin{array}{llllll}
(\mathbf{s},t) \models (f(\mathbf{s}) < d) & \Leftrightarrow & f(s_t) < d & \Leftrightarrow & ((\mathbf{s},t) \models f(s_t) < d) = 1, \\
(\mathbf{s},t) \models \neg(f(\mathbf{s}) < d) & \Leftrightarrow & \neg\big((\mathbf{s},t) \models (f(\mathbf{s}) < d)\big) & \Leftrightarrow & ((\mathbf{s},t) \models f(s_t) < d) = 0, \\
(\mathbf{s},t) \models \phi_1 \wedge \phi_2 & \Leftrightarrow & (\mathbf{s},t) \models \phi_1 \text{ and } (\mathbf{s},t) \models \phi_2 & \Leftrightarrow & (\min\{(\mathbf{s},t) \models \phi_1, (\mathbf{s},t) \models \phi_2\}) = 1, \\
(\mathbf{s},t) \models \phi_1 \vee \phi_2 & \Leftrightarrow & (\mathbf{s},t) \models \phi_1 \text{ or } (\mathbf{s},t) \models \phi_2 & \Leftrightarrow & (\max\{(\mathbf{s},t) \models \phi_1, (\mathbf{s},t) \models \phi_2\}) = 1, \\
(\mathbf{s},t) \models G_{[a,b]}\phi & \Leftrightarrow & (\mathbf{s},t') \models \phi \quad \forall t' \in [t+a, t+b] & \Leftrightarrow & (\min_{t' \in [t+a,t+b]} (\mathbf{s},t') \models \phi) = 1, \\
(\mathbf{s},t) \models F_{[a,b]}\phi & \Leftrightarrow & \exists t' \in [t+a, t+b] \text{ s.t. } (\mathbf{s},t') \models \phi & \Leftrightarrow & (\max_{t' \in [t+a,t+b]} (\mathbf{s},t') \models \phi) = 1.
\end{array}$$

where $((\mathbf{s},t) \models f(s_t) < d) = 1$ implies that $f(s_t) < d$ is true and $((\mathbf{s},t) \models f(s_t) < d) = 0$ implies that $f(s_t) < d$ is false, with $0,1 \in \mathbb{R}$. For a signal $(\mathbf{s},0)$, i.e., the whole signal starting from time 0, satisfying $F_{[a,b]}\phi$ means that "there exists a time within $[a,b]$ such that ϕ will eventually be true", and satisfying $G_{[a,b]}\phi$ means that "ϕ is true for all times between $[a,b]$".

As in Dokhanchi et al. (2014), let $hrz(\phi)$ denote the *horizon* of an STL formula ϕ, which is the required number of samples to resolve any (future or past) requirements of ϕ. The horizon can be computed recursively as

$$\begin{array}{rcll}
hrz(\psi) & = & 0, & \\
hrz(\phi) & = & b & \text{if } \phi = G_{[a,b]}\psi \text{ or } F_{[a,b]}\psi, \\
hrz(F_{[a,b]}\phi) = hrz(G_{[a,b]}\phi) & = & b + hrz(\phi), & \\
hrz(\neg\phi) & = & hrz(\phi), & \\
hrz(\phi_1 \wedge \phi_2) = hrz(\phi_1 \vee \phi_2) & = & \max\{hrz(\phi_1), hrz(\phi_2)\}, &
\end{array}$$

where $a,b \in \mathbb{R}_{\geq 0}$, ψ is a predicate, and ϕ, ϕ_1, ϕ_2 are STL formulae.

STL formula Φ as defined in (1) allows at most three layers of nesting. The first layer Φ constitutes of just $F_{[a,b]}\phi$ or $G_{[a,b]}\phi$. The second layer ϕ constitutes of more than one STL fragment ϕ_i in conjunction using logical operators and their order of precedence of operation. The third layer φ_i in ϕ_i allows more than one predicate in conjunction using logical operators and their order of precedence of operation. The index variable i is used to specify each STL fragment in the second layers of Φ. This formulation allows for specifying most complex bounded time objectives or safety specifications, involving asymptotic and/or periodic behavior. Throughout the paper, we call Φ an STL formula and its constituent ϕ_i is called the i^{th} STL sub-formula.

2.2. Markov Decision Process

A Markov Decision Process (MDP) is used to model discrete-time, finite state and action space stochastic dynamical systems as $M = \langle \Sigma, A, P, R \rangle$, where Σ denotes the state-space, A denotes the action-space, $P : \Sigma \times A \times \Sigma \to [0,1]$ is the probabilistic transition relation, and $R : \Sigma \to \mathbb{R}$ is the reward function. If $\sigma \in \Sigma$ is the current state, then the next state $\sigma' \in \Sigma$ on taking an action $a \in A$ is determined using the probabilistic transition function P. Given an initial state σ_0 and action sequence $a_{0:T-1}$, we denote the state sequence generated by M for T time steps as $\sigma_{0:T}$. Every state $\sigma \in \Sigma$ has a scalar measure of value given by the reward function R.

2.3. Reinforcement Learning: Q-learning

For systems with unknown stochastic dynamics, reinforcement learning can be used to design optimal control policies, i.e., the system learns how to take actions by trial and error interactions with the environment [Sutton and Barto (1998)]. Q-learning is an off-policy model-free reinforcement learning method [Watkins and Dayan (1992)], which can be used to find the optimal policy for a finite MDP. In particular, the objective of an agent at state σ_t is to maximize $V(\sigma_t)$, its expected (discounted) cumulative reward in finite or infinite horizon, i.e., $E\left[\sum_{k=0}^{T} R(\sigma_{k+t+1})\right]$ or $E\left[\sum_{k=0}^{\infty} \gamma^k R(\sigma_{k+t+1})\right]$, where $R(\sigma)$ is the reward obtained at state σ, and γ is the discount factor. Also, $V^*(\sigma) = \max_a Q^*(\sigma, a)$, where $Q^*(\sigma, a)$ is the optimal Q-function for every state-action pair (σ, a). Starting from state σ, the system chooses an action a, which takes it to state σ' and results in a reward $R(\sigma)$. Then, the Q-learning update rule is defined as $Q(\sigma, a) := (1-\alpha)Q(\sigma, a) + \alpha[R(\sigma) + \gamma \max_{a^* \in A} Q(\sigma', a^*)]$, where $\gamma \in (0,1)$ is the discount factor and $\alpha \in (0,1]$ is the learning rate. Accordingly, if each action $a \in A$ is repetitively implemented in each state $\sigma \in \Sigma$ for infinite number of times and α decays appropriately, then Q converges to Q^* with probability 1 [Tsitsiklis (1994)]. Thus, we can find the optimal policy $\pi^* : \Sigma \to A$ as $\pi^*(\sigma) = \arg\max_a Q^*(\sigma, a)$.

3. Problem Statement

In this paper, we assume that the dynamical system is abstracted as an MDP $M = \langle \Sigma, A, P, R \rangle$, where Σ denotes the set of partitions over a continuous space, A is the set of motion primitives, and each motion primitive $a \in A$ drives the system from the centroid of a partition to the centroid of an adjacent partition. In real-world applications, many systems (e.g., robotic platforms) have uncertainty in their dynamics that are difficult to model (e.g., uncertainty in actuation, gusts in the environment, noises in sensors). In this aspect, we assume that the transition probability function P is unknown in MDP M. In other words, given a state-action pair σ_t, a, the probability distribution of state at next time step σ_{t+1} is unknown. Accordingly, a learning problem can be defined as follows:

Problem 1 (Maximizing Probability of Satisfaction) *Given an STL specification $\Phi = F_{[0,T]}\phi$ or $G_{[0,T]}\phi$ with a horizon $hrz(\Phi) = T$, a stochastic system model $M = \langle \Sigma, A, P, R \rangle$ with unknown P and an initial partial state trajectory $\sigma_{0:\tau}$ for $\tau = \left\lceil \frac{hrz(\phi)}{\Delta t} \right\rceil + 1$, find a control policy*

$$\pi_1^* = \arg\max_\pi Pr^\pi[\sigma_{0:T} \models \Phi] = \begin{cases} \arg\max_\pi E^\pi\Big[\max_{t \in [\tau-1, T]} \sigma_{t-\tau+1:t} \models \phi\Big], & \text{if } \Phi = F_{[0,T]}\phi \\ \arg\max_\pi E^\pi\Big[\min_{t \in [\tau-1, T]} \sigma_{t-\tau+1:t} \models \phi\Big], & \text{if } \Phi = G_{[0,T]}\phi \end{cases} \tag{2}$$

where $Pr^\pi[\sigma_{0:T} \models \Phi]$ is the probability of $\sigma_{0:T}$ satisfying Φ under policy π, and $\sigma_{t-\tau+1:t} \models \phi$ is 0 or 1 depending on the satisfaction of $\sigma_{t-\tau+1:t}$ with respect to ϕ.

It is shown in Aksaray et al. (2016a) that the objective function in (2) is not in the standard form of Q-learning. Hence, they have proposed an approximation of Problem 1.

Problem 2 (Maximizing Approximate Probability of Satisfaction, Aksaray et al. (2016a)) *Given an STL specification $\Phi = F_{[0,T]}\phi$ or $G_{[0,T]}\phi$ with a horizon $hrz(\Phi) = T$, a stochastic system model $M = \langle \Sigma, A, P, R \rangle$ with unknown P, known reward function R, a log-sum-exp approximation constant $\beta > 0$ and an initial partial state trajectory $\sigma_{0:\tau}$ for $\tau = \left\lceil \frac{hrz(\phi)}{\Delta t} \right\rceil + 1$, find a control policy*

$$\pi_2^* = \begin{cases} \arg\max_\pi E^\pi\Big[\sum_{t=\tau-1}^{T} e^{\beta(\sigma_{t-\tau+1:t} \models \phi)}\Big], & \text{if } \Phi = F_{[0,T]}\phi \\ \arg\max_\pi E^\pi\Big[-\sum_{t=\tau-1}^{T} e^{-\beta(\sigma_{t-\tau+1:t} \models \phi)}\Big], & \text{if } \Phi = G_{[0,T]}\phi \end{cases} \tag{3}$$

A new system model is defined using a τ-MDP $M^\tau = \langle \Sigma^\tau, A, P^\tau, R^\tau \rangle$, with states $\sigma_t^\tau = \sigma_{t-\tau+1:t}$ and solved using Q-learning in Aksaray et al. (2016a). This τ-MDP representation suffers from exponential state-space growth with horizon τ (Curse of history). Hence, we intend to solve the approximate objective (3), in a new system representation without the curse of history.

4. Proposed Technique

The proposed technique is based on the definition of a new compact system representation, which is rich enough to capture the necessary history within a finite window horizon. The new representation is sufficient to compute reward and action at each time step.

Let Φ be $G_{[0,T]}\phi$ or $F_{[0,T]}\phi$, where Φ is the STL formula with the syntax given in (1). Suppose Φ has n STL sub-formulae ϕ_i as $G_{[0,t]}\varphi_i$ or $F_{[0,t]}\varphi_i$ with horizon $hrz(\phi_i) = \tau_i, \forall i \in [1,n]$. Then, we assign one discrete valued flag variable ($f_i \in \mathfrak{F}_\mathrm{i} := \{k/(\tau_i - 1), k \in [0, \tau_1 - 1]\}$) for each ϕ_i and proceed with defining a flag state augmented MDP know as F-MDP, which captures current state and flags to test for satisfaction of each STL sub-formula ϕ_i. $\mathfrak{F}_\mathrm{i}$ is the flag state set of the flag $f_i, i \in [1,n]$.

Definition 1 (F-MDP) *Given MDP $M = (\Sigma, A, P, R)$ and flag state sets $\mathfrak{F}_\mathrm{i}, \forall i \in [1,n]$, an F-MDP is a tuple $M^F = (\Sigma^F, A, P^F, R^F)$, where*

- $\Sigma^F \subseteq (\Sigma \times \prod_{i=1}^{n} \mathfrak{F}_\mathrm{i})$ *is the set of finite states, obtained by the cartesian product between state set and all n flag state sets. Each state $\sigma^F \in \Sigma^F$ holds the current $\sigma \in \Sigma$ and $f_i \in \mathfrak{F}_\mathrm{i}, \forall i \in [1,n]$.*
- $P^F : \Sigma^F \times A \times \Sigma^F \to [0,1]$ *is a probabilistic transition relation. Let $\sigma^F = \sigma, f_1, f_2, .., f_n$ and $\sigma^{F\prime} = \sigma', f_1', f_2', .., f_n'$. $P^F(\sigma^F, a, \sigma^{F\prime}) > 0$ if and only if $P(\sigma, a, \sigma') > 0$ and $f_i' = update(f_i, \sigma), \forall i \in [1,n]$. where $update(.)$ is the flag update rule:*

$$f_i' = \begin{cases} 1, & \text{if } \phi_i = F_{[0,t]}\varphi_i \text{ and } \sigma' \vDash \varphi_i \\ min(f_i - 1/(\tau_i - 1), 0), & \text{if } \phi_i = F_{[0,t]}\varphi_i \text{ and } \sigma' \nvDash \varphi_i \\ max(f_i + 1/(\tau_i - 1), 1), & \text{if } \phi_i = G_{[0,t]}\varphi_i \text{ and } \sigma' \vDash \varphi_i \\ 0, & \text{if } \phi_i = G_{[0,t]}\varphi_i \text{ and } \sigma' \nvDash \varphi_i \end{cases} \quad \forall i \in [1,n]. \tag{4}$$

- $R^F : \Sigma^F \to \mathbb{R}$ *is a reward function.*

From the update rule (4), we can see how each flag f_i can only take on discrete values between 0 and 1 with a step size of $1/(\tau_i - 1)$. Number of states the flag f_i can take is τ_i and thus $\mathfrak{F}_\mathrm{i}$, has a size equal to horizon τ_i of the STL sub-formula $\phi_i, \forall i \in [1,n]$.

Given a $\sigma^F = \sigma, f_1, f_2..f_n$, the satisfaction function $sat(\sigma^F, \phi)$ used to test for satisfaction of STL formula ϕ and its constituent STL sub-formulae ϕ_i, is recursively defined as follows:

$$\begin{aligned} sat(\sigma^F, \phi_i) &= \begin{cases} 1, & \text{if } f_i > 0 \text{ or } \sigma \vDash \varphi_i \text{ for } \phi_i = F_{[0,t]}\varphi_i \\ 0, & \text{if } f_i = 0 \text{ and } \sigma \nvDash \varphi_i \text{ for } \phi_i = F_{[0,t]}\varphi_i \\ 1, & \text{if } f_i = 1 \text{ and } \sigma \vDash \varphi_i \text{ for } \phi_i = G_{[0,t]}\varphi_i \\ 0, & \text{if } f_i < 1 \text{ or } \sigma \nvDash \varphi_i \text{ for } \phi_i = G_{[0,t]}\varphi_i \end{cases} \quad \forall i \in [1,n], \\ sat(\sigma^F, \phi_j \wedge \phi_k) &= min(sat(\sigma^F, \phi_j), sat(\sigma^F, \phi_k)), \\ sat(\sigma^F, \phi_j \vee \phi_k) &= max(sat(\sigma^F, \phi_j), sat(\sigma^F, \phi_k)), \end{aligned} \tag{5}$$

where ϕ_j, ϕ_k can be STL sub-formulae or their conjunction using logical operators $\wedge, \vee$. For example, if $\phi = ((\phi_1 \wedge \phi_2) \vee \phi_3)$, first current state σ^F is evaluated with respect to sub-formulae $\phi_i, \forall i \in [1,3]$. Then σ^F is evaluated with respect to $\phi_j \wedge \phi_k, j = 1, k = 2$. Finally σ^F is evaluated with respect to new $\phi_j \vee \phi_k$, where $\phi_j = \phi_1 \wedge \phi_2$ and $k = 3$.

The reward function R^F of the problem in the new MDP M^F is as follows:

$$r = \begin{cases} e^{\beta sat(\sigma^F, \phi)}, & \text{if } \Phi = F_{[0,T]}\phi \\ -e^{-\beta sat(\sigma^F, \phi)}, & \text{if } \Phi = G_{[0,T]}\phi \end{cases} \tag{6}$$

where $\beta > 0$ is the log-sum-exp approximation constant.

The overview of the complete technique to solve (3) using F-MDP is as follows:
1) For any STL formula Φ in accordance with (1) (i.e., $G_{[0,T]}\phi$ or $F_{[0,T]}\phi$), create one flag per STL sub-formula ϕ_i in Φ and redefine the learning problem in a new flag state augmented state-space Σ^F which has new state dimensions corresponding to the flags.
2) Define the objective function such that the agent observes an immediate reward as a function of current state in Σ^F. After executing these steps, one can use standard Q-learning algorithm to find the optimal policy $\pi^* : \Sigma^F \to A$ in the new F-MDP state-space. Overall, we aim to solve the following problem.

Problem 3 (Maximizing Approximate Probability of Satisfaction with F-MDP) *Let Φ be STL formula with the syntax in (1), made up of STL sub-formulae $\phi_i, \forall i \in [1,n]$. Let $T = hrz(\Phi)$, $\tau_i = \left\lceil \frac{hrz(\phi_i)}{\Delta t} \right\rceil + 1, \forall i \in [1,n]$ and $\tau = \max_{i \in [1,n]}(\tau_i)$. Given an unknown MDP M, and flag state sets $\mathfrak{F}_i, \forall i \in [1,n]$, F-MDP $M^F = \langle \Sigma^F, A, P^F, R^F \rangle$ can be constructed. Assume that initial τ-states $\sigma_{0:\tau-1}$ are given from which $\sigma^F_{\tau-1}$ can be obtained. Let $\beta > 0$ be a known approximation parameter. Find a control policy $\pi_3^* : \Sigma^F \to A$ such that*

$$\pi_3^* = \begin{cases} \arg\max_{\pi} E^{\pi}\left[\sum_{t=\tau-1}^{T} e^{\beta sat(\sigma_t^F, \phi)}\right], & \text{if } \Phi = F_{[0,T]}\phi \\ \arg\max_{\pi} E^{\pi}\left[-\sum_{t=\tau-1}^{T} e^{-\beta sat(\sigma_t^F, \phi)}\right], & \text{if } \Phi = G_{[0,T]}\phi \end{cases} \tag{7}$$

where $sat(\sigma^F, \phi)$ is the satisfaction function as defined in (5).

4.1. Theoretical Results

The optimal policy π_1^* of Problem 1, can be related to π_3^* of Problem 3 by the following theorem.

Theorem 2 *Let Φ and ϕ be STL formula with the syntax in (1) such that $\Phi = F_{[0,.]}\phi$ or $\Phi = G_{[0,.]}\phi$. Let $hrz(\Phi) = T$. Assume that a partial state trajectory $s_{0:\tau-1}$ is initially given where $\tau = \left\lceil \frac{hrz(\phi)}{\Delta t} \right\rceil + 1$. For some $\beta > 0$ and $\Delta t = 1$[1], let π_1^* and π_3^* be the optimal policies obtained by solving Problems 1 and 3 respectively. Then, $Pr^{\pi_1^*}[s_{0:T} \models \Phi] - \frac{1}{\beta}\log(T - \tau + 2) \leq Pr^{\pi_3^*}[s_{0:T} \models \Phi] \leq Pr^{\pi_1^*}[s_{0:T} \models \Phi]$.*

1. $\Delta t = 1$ is selected due to clarity in presentation, but it can be any time step.

Proof Proof 1 in Appendix of Venkataraman et al. (2019) ∎

For a given MDP M and STL formula Φ as in (1), the F-MDP Σ^F over which the Q-learning is solved has $|\Sigma^F| = |\Sigma| \times \Pi_1^n |\mathfrak{F}_i|$ number of states. Q-table used for Q-learning on F-MDP has $|\Sigma^F| \times |A|$ number of entries. The τ-MDP [Aksaray et al. (2016a)] on the other hand has $|\Sigma|^\tau$ elements in Σ^τ and $|\Sigma^\tau| \times |A|$ entries in its Q-table. For most real-world problems, it is safe to assume that both the number of states ($|\Sigma|$) and horizon ($hrz(\phi) = \tau$) is more than STL sub-formulae horizons ($hrz(\phi_i) = \tau_i, \forall i \in [1,n]$) and the number of STL sub-formulae (n) in ϕ respectively. The above reasoning shows how the proposed technique has only polynomial growth of state-space with horizon τ. Moreover, smaller Q-table also results in faster exploration. Thus for problems with large horizon τ, the proposed technique with F-MDP, convergence to the optimal policy faster than that with τ-MDP.

5. Simulation Results

Suppose that an agent moves over a discretized environment illustrated in Fig. 2 part (a). The set of motion primitives at each state is $A = \{N, NW, W, SW, S, SE, E, NE, stay\}$. We model the motion uncertainty as in Fig. 1 where, for any selected feasible action in A, the agent follows the corresponding blue arrow with probability 0.93 or a red arrow with probability 0.023. Moreover, the resulting state after taking an infeasible action (i.e., the agent tries to move towards a wall) is the current state. All simulations were implemented in MATLAB on a laptop with a quad core 2.4 GHz processor and 8.0 GB RAM.

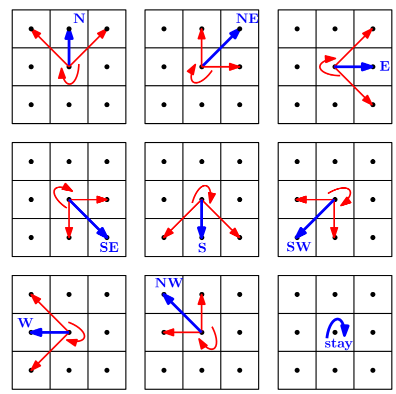

Figure 1: Motion uncertainty (red arrows) for an action (blue arrow).

We consider an STL defined over the environment as $\Phi = G_{[0,12]}\big(F_{[0,h]}(region\ A) \wedge F_{[0,h]}(region\ B)\big)$, where *region A* represents $x > 1 \wedge x < 2 \wedge y > 3 \wedge y < 4$ and *region B* represents $x > 2 \wedge x < 3 \wedge y > 2 \wedge y < 3$. Note that Φ expresses the following: "for all $t \in [0,12]$, eventually visit *region A* every $[t, t+h]$ and eventually visit *region B* every $[t, t+h]$". Note that $\Phi = G_{[0,12]}\phi$ where $\phi = \phi_1 \wedge \phi_2$ with $\phi_1 = F_{[0,h]}(region\ A)$, $hrz(\phi_1) = h$ and $\phi_2 = F_{[0,h]}(region\ B)$, $hrz(\phi_2) = h$. $\tau = h+1$ if Δt is 1.

In this case study, we chose three values for $h := 2,4,5$ with rest of the parameters remaining the same. The sizes of the state-spaces are $|\Sigma| = 36, 36, 36$ and $|\Sigma^F| = 324, 900, 1296$[2] for each $\tau = 3,5,6$ respectively. To implement the Q-learning algorithm, the number of episodes is chosen as 10000 (i.e., $1 \leq k \leq 10000$), with $\beta = 50$, $\gamma = 0.9999$, and $\alpha_k = 0.95^k$. After 10000 trainings, the resulting policies π_3^* is used to generate 500 trajectories, which leads to $Pr^{\pi_3^*}[s_{0:14} \models \Phi] = 0.6794$ for $h = 2$, $Pr^{\pi_3^*}[s_{0:16} \models \Phi] = 0.8237$ for $h = 4$, and $Pr^{\pi_3^*}[s_{0:17} \models \Phi] = 0.8432$ for $h = 5$.

The time and space requirements for 10000 episode trainings on F-MDP and τ-MDP for each of $\tau = 3,5,6$ are provided in Table 1. As predicated, F-MDP state-space is very compact even for large τ. Reduced space requirement also translates into faster execution time. Fig. 2 part (b) is histogram plot of probability of satisfaction by 75 roll-outs with 10000 episodes each for $\tau = 3$ using F-MDP

2. This indicates that there are $36 \times (h+1) \times (h+1)$ F-MDP states, 36 system states, $(h+1)$ f_1 flag states and $(h+1)$ f_2 flag states.

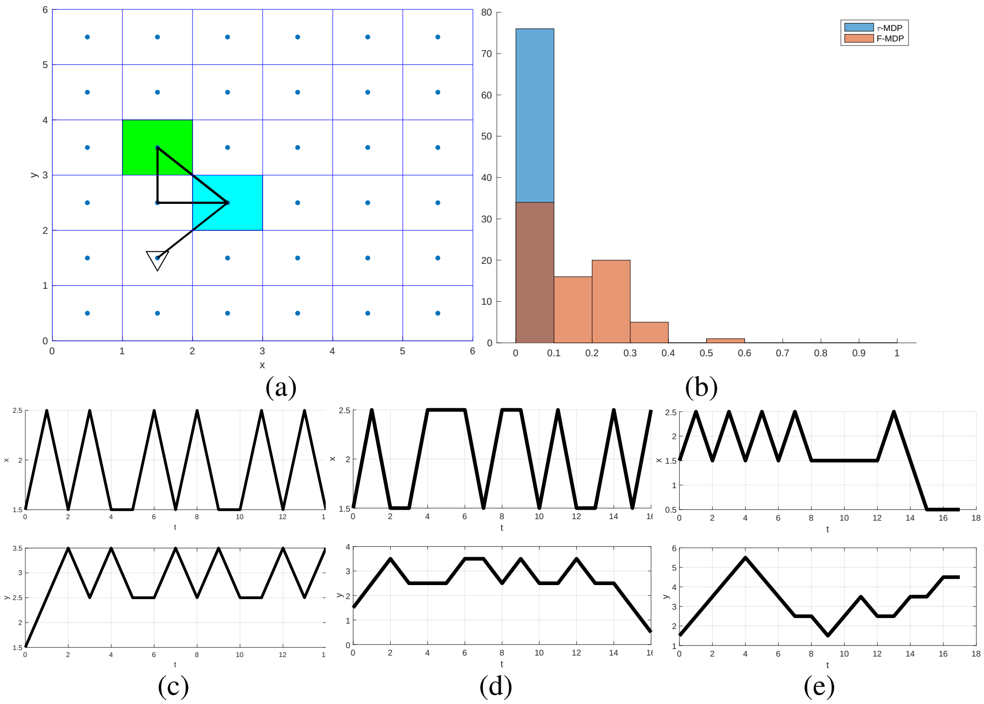

Figure 2: (a) Initial state and the desired regions (b) Distribution of probability of satisfaction of 75 different roll-outs with 10000 episodes each for $\tau = 3$ using F-MDP in orange and τ-MDP in blue (c) Sample trajectory generated by π_A^* for $\tau = 3$ (d) $\tau = 5$ (e) $\tau = 6$

in orange and τ-MDP in blue. In the problem with $\tau = 3$, learning on τ-MDP required $10\times$ longer episode length to achieve similar return distribution as learning on F-MDP. Fig. 2 part (c), (d) and (e) show sample trajectories generated by the optimal policy for $\tau = 3, 5$ and 6.

Table 1: Execution time (in minutes) and space requirements

Technique	$\tau = 3$	$\tau = 5$	$\tau = 6$
F-MDP	6	18	24
τ-MDP	21	290	*†

Technique	$\tau = 3$	$\tau = 5$	$\tau = 6$
F-MDP	$2.9\times e^3$	$8.1\times e^3$	$1.1\times e^4$
τ-MDP	$1.7\times e^{4\ddagger}$	$1.0\times e^{6\ddagger}$	$8.1\times e^{6\ddagger}$

. †. Matlab run time error: preferred array size exceeded

. ‡. Pruned based on feasibility of transition

6. Conclusion

We have proposed a model-free learning technique to synthesize control policies for satisfying STL specifications. The proposed technique remodels the system as an F-MDP, to capture the current system state and history. The learning objective (maximizing probability of satisfaction) is approximated and solved using Q-learning. We also proved that the computed optimal policy is arbitrarily close to that of the desired policy. Finally, we demonstrated the tractability of the proposed technique in simulation. Future work will explore solving this problem using robustness degree metric.

References

Derya Aksaray, Kevin Leahy, and Calin Belta. Distributed multi-agent persistent surveillance under temporal logic constraints. *IFAC-PapersOnLine*, 48(22):174–179, 2015.

Derya Aksaray, Austin Jones, Zhaodan Kong, Mac Schwager, and Calin Belta. Q-learning for robust satisfaction of signal temporal logic specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6565–6570. IEEE, 2016a.

Derya Aksaray, Cristian-Ioan Vasile, and Calin Belta. Dynamic routing of energy-aware vehicles with temporal logic constraints. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3141–3146. IEEE, 2016b.

Toma Brazdil, Krishnendu Chatterjee, Martin Chmelik, M.k, Vojtech Forejt, Jan Kretinsky, Marta Kwiatkowska, David Parker, and Mateusz Ujma. Verification of markov decision processes using learning algorithms. In Franck Cassez and Jean-Franois Raskin, editors, *Automated Technology for Verification and Analysis*, volume 8837 of *Lecture Notes in Computer Science*, pages 98–114. Springer International Publishing, 2014. ISBN 978-3-319-11935-9. doi: 10.1007/978-3-319-11936-6_8. URL http://dx.doi.org/10.1007/978-3-319-11936-6_8.

Xu Chu Ding, Stephen L Smith, Calin Belta, and Daniela Rus. Optimal control of markov decision processes with linear temporal logic constraints. *IEEE Trans. on Automatic Control*, 59(5):1244–1257, 2014.

Adel Dokhanchi, Bardh Hoxha, and Georgios Fainekos. On-line monitoring for temporal logic robustness. In *Runtime Verification*, pages 231–246. Springer, 2014.

Jie Fu and Ufuk Topcu. Probably approximately correct MDP learning and control with temporal logic constraints. *CoRR*, abs/1404.7073, 2014. URL http://arxiv.org/abs/1404.7073.

Morteza Lahijanian, Sean B. Andersson, and Calin Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Trans. on Automatic Control*, 6(8):2031–2045, 2015. doi: 10.1109/TAC.2015.2398883.

Xiao Li and Calin Belta. Temporal logic guided safe reinforcement learning using control barrier functions. *arXiv preprint*, 2019.

Xiao Li, Cristian-Ioan Vasile, and Calin Belta. Reinforcement learning with temporal logic rewards. *arXiv preprint*, 2017.

Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In Yassine Lakhnech and Sergio Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004. doi: 10.1007/978-3-540-30206-3_12. URL http://dx.doi.org/10.1007/978-3-540-30206-3_12.

Dorsa Sadigh, Eric S Kim, Samuel Coogan, S Shankar Sastry, and Sanjit A Seshia. A learning based approach to control synthesis of markov decision processes for linear temporal logic specifications. In *IEEE Conf. on Decision and Control*, pages 1091–1096. IEEE, 2014.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16 (3):185–202, 1994.

Harish Venkataraman, Derya Aksaray, and Peter Seiler. Tractable reinforcement learning of signal temporal logic objectives. *arXiv preprint*, 2019.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Zhe Xu and Ufuk Topcu. Transfer of temporal logic formulas in reinforcement learning. *arXiv preprint*, 2019.