2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Time-Optimal Navigation in Uncertain Environments with High-Level Specifications

Ugo Rosolia, Mohamadreza Ahmadi, Richard M. Murray, and Aaron D. Ames

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683486

***Abstract*— Mixed observable Markov decision processes (MOMDPs) are a modeling framework for autonomous systems described by both fully and partially observable states. In this work, we study the problem of synthesizing a control policy for MOMDPs that minimizes the expected time to complete the control task while satisfying syntactically co-safe Linear Temporal Logic (scLTL) specifications. First, we present an exact dynamic programming update to compute the value function. Leveraging this result, we propose a point-based approximation, which allows us to compute a lower bound of the closed-loop probability of satisfying the specifications. The effectiveness of the proposed approach and comparisons with standard strategies are shown on high-fidelity navigation tasks with partially observable static obstacles.**

I. Introduction

Autonomous systems take actions based on observations of the environment surrounding them. When the environment includes both fully observable and partially observable regions, mixed observable Markov decision processes (MOMDPs) can be used as a framework for decision making under uncertainty [1]. In MOMDPs, the state space is partitioned into fully observable and partially observable states. Decisions are taken based on the fully observable states and the *belief* representing a probability distribution over the partially observable states. Compared to partially observable Markov decision processes (POMDPs), which maintain a belief for all possible states [2], MOMDPs allow us to reduce the computational complexity of the policy synthesis process when both partial and full state observations are available [1].

In POMDPs and MOMDPs, the control objective is usually expressed as a reward maximization problem [2]. However, reward maximization alone cannot fully encode the desired high-level objectives. Thus, researchers have focused on constrained POMDPs (CPOMDPs), where the synthesis goal is to compute a policy that maximizes the expected reward, while satisfying expected constraints. This problem was first studied in [3], where the authors presented an exact dynamic programming update to compute the optimal deterministic policy. The computational complexity of solving this problem is double exponential in the time horizon. But, the optimal solution can be approximated in polynomial time using point-based [4] and finite-state [5] approximations.

Whenever temporal properties of the system are of interest, control objectives can also be expressed using Linear Temporal Logic (LTL) formulas [6]. The *qualitative problem* of synthesizing a policy, which guarantees satisfaction of LTL formulas for POMDPs, is undecidable when searching over the set of feedback policies and EXPTIME-complete when designing finite-state controllers [7]–[11]. When the system is uncertain, it may be impossible to design a policy that guarantees satisfaction of the specifications for all possible uncertainty realizations. In this case, it is desirable to solve the *quantitative problem*, where the objective is to synthesize a policy that maximizes the probability of satisfying LTL specifications. The solution to this quantitative problem can be approximated by discretizing the belief space [12], leveraging finite state controllers [11] or using point-based and simulation-based strategies [13]–[18]. The optimal solution to quantitative problems is usually not unique; instead there exists a set of optimal control policies [19]. For this reason, it is often preferable to compute an optimal policy, which maximizes an expected reward while satisfying LTL specifications [19]–[21].

In this work, we consider *time-optimal quantitative* problems, where the goal is to minimize the expected time to complete the task while satisfying syntactically co-safe LTL (scLTL) specifications. These problems have been studied for deterministic systems in [20], [21] and in [19], [22]–[26] for Markov decision processes. To the knowledge of the authors, this is the first work that studies time-optimal quantitative problems for mixed observable Markov decision processes. Our contribution is threefold. First, we present a dynamic programming update to compute the value function associated with the time-optimal quantitative problem. Second, we propose a point-based strategy to approximate the optimal value function and we show that our approach maximizes a lower bound of the closed-loop probability of satisfying the specifications. Finally, we compare our method with standard time-optimal and quantitative policies. We show that the proposed strategy allows us to minimize the expected time to complete the task without compromising the probability of satisfying the specifications.

Notation: For a vector $\alpha \in \mathbb{R}^n$ and an integer $s \in \{1, \dots, n\}$ we use $\alpha(s)$ to denote the sth component of the vector α and $\alpha^\top$ to indicate its transpose. For a function $V : \mathbb{R}^n \to \mathbb{R}$, $V(\alpha)$ denotes the value of the function V at α. Throughout the paper, we will use capital letters to indicate functions and lower letters to indicate vectors. Given two sets $\mathcal{A}$ and $\mathcal{B}$, the set minus operation is denoted as $\mathcal{A} \setminus \mathcal{B}$ and the Cartesian product as $\mathcal{A} \times \mathcal{B}$. Furthermore, we define the indicator function $\mathbb{1}_{\mathcal{A}}(x) = 1$ if $x \in \mathcal{A}$ and $\mathbb{1}_{\mathcal{A}}(x) = 0$ otherwise. The vectors of ones is written as $1_n \in \mathbb{R}^n$ and zeros as $0_n \in \mathbb{R}^n$. Finally, given two sets of vectors $\Gamma = \{\gamma_i | \forall i \in \{1, \dots, n_\gamma\}\}$ and $\Lambda = \{\lambda_j | \forall j \in \{1, \dots, n_\lambda\}\}$ we denote $\Gamma \oplus \Lambda = \{\gamma_i + \lambda_j | \forall i \in \{1, \dots, n_\gamma\}, \forall j \in \{1, \dots, n_\lambda\}\}$.

Work supported by the NSF award #1932091. The authors are with the California Institute of Technology, e-mails: {urosolia, mrahmadi, murray, ames}@caltech.edu.

II. Background

In this section, we introduce some definitions and assumptions used throughout the paper.

Mixed Observable Markov Decision Process

A MOMDP provides a sequential decision-making formalism for high-level planning under mixed full and partial observations [1]. More formally, a MOMDP $\mathcal{M}$ is a tuple $(\mathcal{S}, \mathcal{E}, \mathcal{A}, \mathcal{Z}, T_s, T_e, O)$, where

- $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$ is a set of fully observable states;
- $\mathcal{E} = \{1, \ldots, |\mathcal{E}|\}$ is a set of partially observable states;
- $\mathcal{A} = \{1, \ldots, |\mathcal{A}|\}$ is a set of actions;
- $\mathcal{Z} = \{1, \ldots, |\mathcal{Z}|\}$ is the set of observations for the partially observable state $e \in \mathcal{E}$;
- The function $T_s : \mathcal{S} \times \mathcal{E} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ describes the probability of transitioning to a state s' given the action a and the system's state (s, e), i.e., $T_s(s, e, a, s') := \mathbb{P}(s_{k+1} = s' | s_k = s, e_k = e, a_k = a)$;
- The function $T_e : \mathcal{S} \times \mathcal{E} \times \mathcal{A} \times \mathcal{S} \times \mathcal{E} \to [0, 1]$ describes the probability of transitioning to a state e' given the action a, the successor observable state s' and the system's current state (s, e), i.e., $T_e(s, e, a, s', e') := \mathbb{P}(e_{k+1} = e' | s_k = s, e_k = e, a_k = a, s_{k+1} = s')$;
- The function $O : \mathcal{S} \times \mathcal{E} \times \mathcal{A} \times \mathcal{Z} \to [0, 1]$ describes the probability of observing the measurement $z \in \mathcal{Z}$, given the current state of the system $(s', e') \in \mathcal{S} \times \mathcal{E}$ and the action a applied at the previous time step, i.e., $O(s', e', a, z) := P(z_k = z | s_k = s', e_k = e', a_{k-1} = a)$;

MOMDPs were introduced in [1] to model systems where a subspace of the state space is perfectly observable. The advantage of distinguishing between fully and partially observable states is that a belief state is needed only for the partially observable states. Thus, we introduce the belief vector $b_{\mathcal{E}} \in \mathcal{B}_{\mathcal{E}} = \{b_{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|} : \sum_{e=1}^{|\mathcal{E}|} b_{\mathcal{E}}(e) = 1\}$, where each entry $b_{\mathcal{E}}(e)$ represents the posterior probability that the partially observable state e_k equals $e \in \mathcal{E}$.

Syntactically Co-Safe LTL Specifications

We consider objectives which are expressed using scLTL specifications. For a set of atomic proposition $\mathcal{AP}$, an scLTL specification is defined as follows:

$$\psi := p \mid \neg p \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \psi_1 U \psi_2 \mid \bigcirc \psi,$$

where the atomic proposition $p \in \mathcal{AP}$ and ψ, ψ_1, ψ_2 are scLTL formulas, which can be defined using the logic operators negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$). Furthermore, scLTL formulas can be specified using the temporal operators until (U) and next ($\bigcirc$). Each atomic proposition p_i is associated with a subset of the MOMDP state space $\mathcal{P}_i \subset \mathcal{S} \times \mathcal{E}$, and a state $\omega_k = (s_k, e_k)$ of the MOMDP $\mathcal{M}$ satisfies the atomic proposition p_i if $\omega_k \in \mathcal{P}_i$. Finally, satisfaction of a specification ψ for the trajectory $\boldsymbol{\omega}_k = [\omega_k, \omega_{k+1}, \ldots]$, denoted by

$$\boldsymbol{\omega}_k \models \psi$$

is recursively defined as follows: *i)* $\boldsymbol{\omega}_k \models p \iff \omega_k \in \mathcal{P}$, *ii)* $\boldsymbol{\omega}_k \models \psi_1 \wedge \psi_2 \iff (\boldsymbol{\omega}_k \models \psi_1) \wedge (\boldsymbol{\omega}_k \models \psi_1)$, *iii)* $\boldsymbol{\omega}_k \models \psi_1 \vee \psi_2 \iff (\boldsymbol{\omega}_k \models \psi_1) \vee (\boldsymbol{\omega}_k \models \psi_1)$, *iv)* $\boldsymbol{\omega}_k \models \psi_1 U \psi_2 \iff \boldsymbol{\omega}_l \models \psi_2$ and $\boldsymbol{\omega}_j \models \psi_2,\ \forall j \in \{k, \ldots, l-1\}$, *v)* $\boldsymbol{\omega}_k \models \bigcirc \psi \iff \boldsymbol{\omega}_{k+1} \models \psi$.

Assumption 1. We consider reachability specifications, which are satisfied when the observable state $s \in \mathcal{S}$ of a MOMDP $\mathcal{M}$ reaches a target set $\mathcal{T} \subset \mathcal{S}$.

The above assumption is not restrictive for finite time problem, as the problem of checking if a finite time trajectory of a MOMDP satisfies any scLTL specification can be recasted as a reachability problem over an extended MOMDP. Please refer to [27, Chapter 3], [12], [13] for further details on how to construct such extended MOMDP.

III. Time-Optimal Quantitative MOMDP

Problem Formulation

In this section, we introduce the problem under study. Given a MOMDP $\mathcal{M}$ with observable states $\mathcal{S}$, partially observable states $\mathcal{E}$, and target set $\mathcal{T}$ associated with the specification ψ, we consider the finite-horizon problem of maximizing the probability of satisfying the specification ψ, while minimizing the expected time to complete the task. In particular, we define the following time-optimal quantitative constrained MOMDP (CMOMDP)

$$\begin{aligned} \pi^{\text{TOQ}} = \underset{\pi}{\text{argmin}} \quad & \mathbb{E}^{\pi}\left[\sum_{t=0}^{N} \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s_t)\right] \\ \text{subject to} \quad & \pi \in \underset{\kappa}{\text{argmax}}\, \mathbb{P}^{\kappa}[\boldsymbol{\omega} \models \psi], \end{aligned} \tag{1}$$

where $\mathbb{E}^{\pi}[\cdot]$ denotes the expectation under the policy π, N represents the duration of the task and the indicator function $\mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s) = 1$ when $s \in \mathcal{S} \setminus \mathcal{T}$ and $\mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s) = 0$ when $s \notin \mathcal{S} \setminus \mathcal{T}$. In the above problem, $\mathbb{P}^{\kappa}[\boldsymbol{\omega} \models \psi]$ represents the probability that the closed-loop trajectory under the policy $\kappa : \mathcal{S} \times \mathcal{B}_{\mathcal{E}} \to \mathcal{A}$ will satisfy the specifications. Therefore, the optimal policy $\pi^{\text{TOQ}} : \mathcal{S} \times \mathcal{B}_{\mathcal{E}} \to \mathcal{A}$ from (1) maximizes the probability of satisfying the specifications while minimizing the expected time to complete the control task, i.e., reaching the set $\mathcal{T} \times \mathcal{E} \subset \mathcal{S} \times \mathcal{E}$.

Motivating Example

Problem (1) is motivated by the example shown in Figure 1, where a Segway has to collect science samples which may be located in the goal region (green) while avoiding known obstacle regions (dark brown) and exploring uncertain regions (light brown). The control problem can be formulated as a MOMDP, where the Segway's position is perfectly observed and only partial observations about the traversability of the uncertain regions are available. Figure 1 shows an example with one goal region $\mathcal{G}$ and four uncertain regions $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$, which may be traversable with probability 0.9, 0.4, 0.3, and 0.5, respectively. The Segway receives a perfect measurement when it is next to an uncertain region, otherwise the measurement is corrupted as described in the example section. In this example, the task has a duration of $N = 30$ time steps. The control objective is given by the scLTL formula $\psi = \neg \texttt{Collision} U \texttt{Goal}$, where the atomic proposition `Collision` is satisfied when the system is in a cell occupied by an obstacle and the atomic

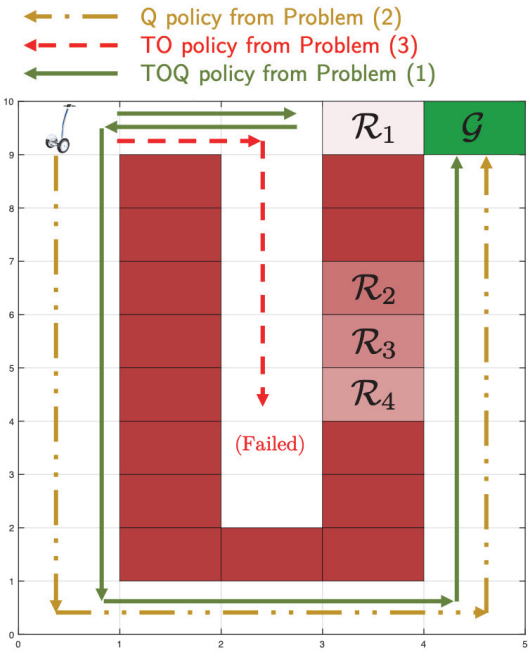

Fig. 1. This figure shows a navigation example with several obstacles (dark brown), one goal region $\mathcal{G}$ (green) and four uncertain regions (light brown) $\mathcal{R}_1$, $\mathcal{R}_2$ $\mathcal{R}_3$, and $\mathcal{R}_4$. In this example, all uncertain regions are not traversable.

proposition `Goal` is satisfied when the system reached the goal cell.

As discussed in [12], [14], [15], a control policy can be computed maximizing the probability that ψ is satisfied, i.e.,

$$\pi^{\mathrm{Q}} = \underset{\kappa}{\operatorname{argmax}} \quad \mathbb{P}^{\kappa}[\boldsymbol{\omega} \models \psi]. \tag{2}$$

Alternatively, a control policy can be synthesized minimizing the expected time to complete the task. The time-optimal problem is given by the standard reward minimization:

$$\pi^{\mathrm{TO}} = \underset{\pi}{\operatorname{argmin}} \quad \mathbb{E}^{\pi}\left[\sum_{t=0}^{N} \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s_t)\right], \tag{3}$$

where the indicator function $\mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}$ is defined as in (1). Notice that the solution to the above minimization problem can be approximated with point-based methods [28] or finite state controllers [29].

Figure 1 shows the closed-loop behaviors associated with the control policies from Problems (1)–(3). The Time-Optimal (TO) policy from Problem (3) steers the system beside the uncertain regions to collect perfect measurements about the traversability of the terrain. In this example, all uncertain regions are not traversable and therefore the control policy from Problem (3) fails to reach the goal state $\mathcal{G}$ in $N = 30$ time steps. On the other hand, the Time-Optimal Quantitative (TOQ) policy from Problem (1) and the Quantitative (Q) policy from Problem (2), which are designed to maximize the probability of satisfying the specification, reach the goal set $\mathcal{G}$. Finally, we notice that the TOQ policy first explores region $\mathcal{R}_1$ and then takes the path around the obstacle to reach the goal. This behavior minimizes the expected time to complete the task, as region $\mathcal{R}_1$ may be traversable with probability 0.9. Thus, this example shows the advantage of synthesizing TOQ policies, which minimize the expected time to complete the task, while guaranteeing that the probability of satisfying the specifications is maximized.

IV. Exact Dynamic Programming Update

In this section, we first show that the optimal value function $V_k^*(s,\cdot) : \mathcal{B}_{\mathcal{E}} \to \mathbb{R}$ of the time-optimal quantitative problem (1) is piecewise affine for all $s \in \mathcal{S}$ and $k \in \{0,\dots,N\}$. Afterwards, following the approach presented in [3], we define a pair of support vectors which characterize the optimal value function $V_k^*(s,\cdot)$ for all $s \in \mathcal{S}$ and $k \in \{0,\dots,N\}$.

As shown in [1], the synthesis problem can be reformulated as a stochastic optimal control problem for a fully observable uncertain system, where the states are the belief $b_{\mathcal{E}}$ and the fully observable state s of the MOMDP. Indeed, the belief evolves accordingly to the following update equation:

$$\begin{aligned} b'_{\mathcal{E}}(e') =& \eta O(s',e',a,z) \\ & \times \sum_{e\in\mathcal{E}} T_s(s,e,a,s') T_e(s,e,a,s',e') b_{\mathcal{E}}(e), \end{aligned} \tag{4}$$

where the scalar $\eta = 1/P(s',z|s,b_{\mathcal{E}},a)$ is a normalization constant [1], [30].

Next, we introduce two lemmas that allow us to reformulate Problem (2) and Problem (3) as standard reward maximization problems. Afterwards, we will leverage these results to derive an exact dynamic programming update for the time-optimal quantitative Problem (1).

Lemma 1. *Consider a MOMDP $\mathcal{M}$ with terminal set $\mathcal{T}\times\mathcal{E}$ and a finite horizon N. The probability that the quantitative policy π^{Q} from Problem (2) satisfies the specifications is*

$$\max_{\kappa} \mathbb{P}^{\kappa}[\boldsymbol{\omega} \models \psi] = \bar{J}_0(s, b_{\mathcal{E}}),$$

where the optimal value function $\bar{J}_0$ is given by the following dynamic programming recursion:

$$\bar{J}_k(s,b_{\mathcal{E}}) = \mathbb{1}_{\mathcal{T}}(s) + \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s) \max_{a\in\mathcal{A}} \mathbb{E}[\bar{J}_{k+1}(s',b'_{\mathcal{E}})|s,b_{\mathcal{E}},a] \tag{5}$$

with $\bar{J}_N(s,\cdot) = \mathbb{1}_{\mathcal{T}}(s)$ for all $s \in \mathcal{S}$. Furthermore, the optimal value function $\bar{J}_k(s,\cdot) : \mathcal{B}_{\mathcal{E}} \to [0,1]$ is piecewise-affine for all $k \in \{0,\dots,N\}$ and for all $s \in \mathcal{S}$.

Proof: Notice that, given a policy $\kappa : \mathcal{S}\times\mathcal{B}_{\mathcal{E}} \to \mathcal{A}$, the probability of satisfying the specification is given by the probability of reaching the terminal set $\mathcal{T}$, i.e.,

$$\begin{aligned} \mathbb{P}^{\kappa}[\boldsymbol{\omega}\models\psi] &= \mathbb{P}^{\kappa}\big[\exists k \in \{0,\dots,N\} : s_k \in \mathcal{T}, \\ &\qquad s_j \in \mathcal{S}\setminus\mathcal{T}, \forall j \in \{0,\dots,k-1\}\big] \\ &= \mathbb{E}^{\kappa}\left[\sum_{j=0}^{N}\left(\prod_{i=0}^{j-1} \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s_i)\right)\mathbb{1}_{\mathcal{T}}(s_j)\right], \end{aligned}$$

where $\mathbb{E}^{\kappa}[\cdot]$ denotes the expectation under the policy κ. For more details on the above stochastic reachability problem please refer to [31].

Furthermore from [31, Theorem 4], we have that the optimal value function $\bar{J}_k : \mathcal{S} \times \mathcal{B}_\mathcal{E} \to \mathbb{R}$, which is associated with the optimal policy that maximizes the probability of reaching the set $\mathcal{T}$, is given by the following recursion

$$\bar{J}_k(s, b_\mathcal{E}) = \mathbb{1}_\mathcal{T}(s) + \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s) \max_{a\in\mathcal{A}} \mathbb{E}[\bar{J}_{k+1}(s', b'_\mathcal{E})|s, b_\mathcal{E}, a] \tag{6}$$

where $\bar{J}_N(s,\cdot) = \mathbb{1}_\mathcal{T}(s)$. Next, we show by induction that $\bar{J}_k(s,\cdot) : \mathcal{B}_\mathcal{E} \to \mathbb{R}$ is piecewise affine for all $k \in \{0,\dots,N\}$ and $s \in \mathcal{S}$. Assume that $\bar{J}_{k+1}(s,\cdot)$ is piecewise affine for a set of support vectors Λ_s, i.e., $\bar{J}_{k+1}(s, b_\mathcal{E}) = \max_{\beta\in\Lambda_s} \beta^\top b_\mathcal{E} = \max_{\beta\in\Lambda_s} \sum_e \beta(e) b_\mathcal{E}(e)$. Then, using the belief update (4), the definition $\eta = 1/P(s', z|s, b_\mathcal{E}, a)$ and the definition of the value function $\bar{J}_{k+1}(s,\cdot)$, we have that

$$\begin{aligned}
\mathbb{E}[\bar{J}_{k+1}(s', b'_\mathcal{E})|s, b_\mathcal{E}, a] &= \sum_{s',z} P(s', z|s, b_\mathcal{E}, a) J_{k+1}(s', b'_\mathcal{E}) \\
&= \sum_{s',z} P(s', z|s, b_\mathcal{E}, a) \max_{\beta\in\Lambda_s} \sum_{e'} \beta(e') b'_\mathcal{E}(e') \\
&= \sum_{s',z} \frac{1}{\eta} \max_{\beta\in\Lambda_s} \sum_{e'} \beta(e') \eta O(s', e', a, z) \sum_e T_s(s, e, a, s') \\
&\qquad \times T_e(s, e, a, s', e') b_\mathcal{E}(e).
\end{aligned} \tag{7}$$

Now define $\beta'(e) = \sum_{e'} F(s, e, a, s', e', z)\beta(e')$ for

$$\begin{aligned}
F(s, e, a, s', e', z) = T_s(s, e, a, s') T_e(s, e, a, s', e') \\
\times O(s', e', a, z),
\end{aligned} \tag{8}$$

then equation (7) can be rewritten as

$$\begin{aligned}
&\mathbb{E}[\bar{J}_{k+1}(s', b'_\mathcal{E})|s, b_\mathcal{E}, a] \\
&\quad = \sum_{s',z} \max_{\beta\in\Lambda_s} \sum_e b_\mathcal{E}(e) \sum_{e'} F(s, e, a, s', e', z)\beta(e') \\
&\quad = \sum_{s',z} \max_{\beta\in\Lambda_s} b_\mathcal{E}^\top \beta'.
\end{aligned} \tag{9}$$

Equation (9) implies that the conditional expectation in (6) is a piecewise affine function of $b_\mathcal{E}$. Therefore, $\bar{J}_k(s,\cdot)$ is piecewise affine as it is given by the summation and the point-wise maximization of picecewise affine functions for all $s \in \mathcal{S}$. The proof is concluded by induction on k as the value function $\bar{J}_N(s,\cdot)$ is piecewise affine for all $s \in \mathcal{S}$. ■

Lemma 2. *Consider a MOMDP $\mathcal{M}$ with terminal set $\mathcal{T} \times \mathcal{E}$ and a finite horizon N. The optimal control policy π^{TO} from Problem* (3) *is the optimizer of the following problem:*

$$\max_\pi \ \mathbb{E}^\pi\left[\sum_{t=0}^{N} \mathbb{1}_\mathcal{T}(s_t)\right].$$

Proof: Notice that by definition $\mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s_t) = 1 - \mathbb{1}_\mathcal{T}(s_t), \forall s \in \mathcal{S}$. Thus, we have that

$$\begin{aligned}
\operatorname*{argmin}_\pi \mathbb{E}^\pi\left[\sum_{t=0}^{N} \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s_t)\right] &= \operatorname*{argmin}_\pi \mathbb{E}^\pi\left[\sum_{t=0}^{N} (1 - \mathbb{1}_\mathcal{T}(s_t))\right] \\
= \operatorname*{argmin}_\pi \mathbb{E}^\pi\left[\sum_{t=0}^{N} -\mathbb{1}_\mathcal{T}(s_t)\right] &= \operatorname*{argmax}_\pi \mathbb{E}^\pi\left[\sum_{t=0}^{N} \mathbb{1}_\mathcal{T}(s_t)\right],
\end{aligned}$$

which concludes the proof. ■

Optimal Value Function

In what follows, we leverage the dynamic programming update from Lemma 1 and the maximization problem from Lemma 2 to design an exact dynamic programming update for the time-optimal quantitative problem (1). We modify the strategy presented in [3] to solve the CMOMDP from (1). The key idea is to construct a set of vector pairs $\langle \alpha^i_{s,k}, \beta^i_{s,k} \rangle$, which define the optimal value function

$$\begin{aligned}
V^*_k(s, b_\mathcal{E}) = \max_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k}} \quad & \alpha^\top b_\mathcal{E} \\
\text{subject to} \quad & \langle\alpha,\beta\rangle \in \operatorname*{argmax}_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k}} \beta^\top b_\mathcal{E},
\end{aligned} \tag{10}$$

where at time k the set $\Gamma^*_{s,k}$ collects the support vector pairs associated with the observable state $s \in \mathcal{S}$.

Next, we show that the support vectors can be updated using the following recursion:

$$\begin{aligned}
\alpha^{a,z,s',i}_{s,k}(e) &= \frac{\mathbb{1}_\mathcal{G}(s)}{|\mathcal{Z}||\mathcal{S}|} + \sum_{e'} F(s, e, a, s', e', z) \alpha^i_{s',k+1}(e'), \\
\beta^{a,z,s',i}_{s,k}(e) &= \frac{\mathbb{1}_\mathcal{G}(s)}{|\mathcal{Z}||\mathcal{S}|} + \mathbb{1}_{\mathcal{Q}\setminus\mathcal{G}}(s) \sum_{e'} F(s, e, a, s', e', z) \\
&\qquad \times \beta^i_{s',k+1}(e') \\
\Gamma^{*,a}_{s,k} &= \oplus_{z\in\mathcal{Z}, s'\in\mathcal{S}} \{\langle \alpha^{a,z,s',i}_{s,k}, \beta^{a,z,s',i}_{s,k} \rangle | \\
&\qquad \forall i \in \{1, \dots, |\Gamma^*_{s,k+1}|\}\} \\
\Gamma^*_{s,k} &= \cup_{a\in\mathcal{A}} \Gamma^{*,a}_{s,k}.
\end{aligned} \tag{11}$$

where the function F is defined as in (8) and

$$\Gamma^*_{s,N} = \begin{cases} \langle 1_{|\mathcal{Z}|}, 1_{|\mathcal{Z}|} \rangle & \text{if } s \in \mathcal{T}, \\ \langle 0_{|\mathcal{Z}|}, 0_{|\mathcal{Z}|} \rangle & \text{otherwise.} \end{cases} \tag{12}$$

The backup update of the α-vector is used to compute the support vectors associated with the cost and it was presented in [32]. On the other hand, the backup update of the β-vector is designed based on the dynamic programming update (5) from Lemma 1 and it is a key contribution of this work. The following lemma illustrates that the backup update of the β-vector, which defines the set of support vectors $\Gamma^*_{s,k}$ from (11), allows us to compute the probability that the time-optimal quantitative policy satisfies the specifications.

Lemma 3. *Let $\Gamma^*_{s,k}$ be the set of support vectors constructed using the dynamic programming recursion from* (11)*. Then, the constraint value function*

$$J^*_k(s, b_\mathcal{E}) = \max_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k}} \beta^\top b_\mathcal{E} \tag{13}$$

represents the probability that the time-optimal quantitative policy π^{TOQ} satisfies the specifications, i.e., $J^*_k(s, b_\mathcal{E}) = \mathbb{P}^{\pi^{TOQ}}\big[[\omega_k, \dots, \omega_N] \models \psi\big], \forall k \in \{0, \dots, N\}$.

Proof: First, we show by induction that $\bar{J}_k(s,\cdot) = J^*_k(s,\cdot)$ for all $s \in \mathcal{S}$. Assume that $\bar{J}_{k+1}(s,\cdot) = J^*_{k+1}(s,\cdot)$ for all $s \in \mathcal{S}$, which implies that

$$\max_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k+1}} b_\mathcal{E}^\top \beta = J^*_{k+1}(s, b_\mathcal{E}) = \bar{J}_{k+1}(s, b_\mathcal{E}) = \max_{\beta\in\Lambda_s} b_\mathcal{E}^\top \beta. \tag{14}$$

Then, from equations (6), (9) and (14), we have that

$$\begin{aligned}\bar{J}_k(s,b_{\mathcal{E}}) &= \mathbb{1}_{\mathcal{T}}(s) + \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s)\max_{a\in\mathcal{A}}\sum_{s',z}\max_{\beta\in\Lambda_s} b_{\mathcal{E}}^{\top}\beta' \\ &= \max_{a\in\mathcal{A}}\sum_{s',z}\left[\frac{\mathbb{1}_{\mathcal{T}}(s)}{|\mathcal{Z}||\mathcal{S}|} + \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s)\max_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k+1}} b_{\mathcal{E}}^{\top}\beta'\right] \\ &= \max_{a\in\mathcal{A}}\sum_{s',z}\max_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k+1}}\left[\frac{\mathbb{1}_{\mathcal{T}}(s)}{|\mathcal{Z}||\mathcal{S}|}1_{|\mathcal{E}|}^{\top} + \mathbb{1}_{\mathcal{S}\setminus\mathcal{T}}(s)(\beta')^{\top}\right]b_{\mathcal{E}} \\ &= \max_{a\in\mathcal{A}}\max_{\langle\alpha,\beta\rangle\in\Gamma^{*,a}_{s,k}}\beta^{\top}b_{\mathcal{E}} = \max_{\langle\alpha,\beta\rangle\in\Gamma^{*}_{s,k}}\beta^{\top}b_{\mathcal{E}} = J^*_k(s,b_{\mathcal{E}}),\end{aligned}$$

where $\beta'(e) = \sum_{e'} F(s,e,a,s',e',z)\beta(e')$, $1_{|\mathcal{E}|} \in \mathbb{R}^{|\mathcal{E}|}$ is a vector of ones, $1_{|\mathcal{E}|}^{\top} b_{\mathcal{E}} = 1$ and the sets of support vectors $\Gamma^{*,a}_{s,k}$ and $\Gamma^{*}_{s,k}$ are defined by the backup update (11) for the set of support vectors $\Gamma^*_{s,k+1}$ from equation (14). Finally, as $J^*_N(s,\cdot) = \bar{J}_N(s,\cdot), \forall s \in \mathcal{S}$ by induction we have that $J^*_k(s,\cdot) = \bar{J}_k(s,\cdot), \forall s \in \mathcal{S}\ \forall k \in \{0,\dots,N\}$, which together with Lemma 2 and the definitions of Problems (1) and (2) imply that $J^*_k(s,\cdot) = \bar{J}_k(s,\cdot) = \mathbb{P}^{\pi^{\mathrm{Q}}}\big[[\omega_k,\dots,\omega_N] \models \psi\big] = \mathbb{P}^{\pi^{\mathrm{TOQ}}}\big[[\omega_k,\dots,\omega_N] \models \psi\big]$, $\forall k \in \{0,\dots,N\}$ and $\forall s \in \mathcal{S}$. ■

V. Point-Based Approximation

At each time step the dynamic programming update from (11) generates in the worst case $|\mathcal{A}||\Gamma^*_{s,k+1}|^{(|\mathcal{Z}|+|\mathcal{S}|)}$ new support vector pairs [28]. In this section, we present a point-based update, where the optimal value function is approximated by a constant number of vectors computed for a set $\mathcal{D}_b = \{b_{\mathcal{E}}^{(1)},\dots,b_{\mathcal{E}}^{(n)}\}$ of n discrete beliefs.

The proposed point-based strategy is based on the update from equation (11). In particular, Algorithm 1 computes one pair of vectors $\langle\alpha^{a^*},\beta^{a^*}\rangle$ that approximates the optimal value function (10) at a point $(s,b_{\mathcal{E}})$. In line 1 of Algorithm 1, we compute the active pair of support vectors using the following expression[1]

$$\begin{aligned}&\langle\alpha^{s',a,z},\beta^{s',a,z}\rangle \\ &\quad= \operatorname*{argmax}_{\langle\alpha,\beta\rangle\in\Gamma_{s,k+1}} \alpha^{\top}F_v(s,b_{\mathcal{E}},a,s',z) \\ &\quad\text{subject to}\quad \langle\alpha,\beta\rangle \in \operatorname*{argmax}_{\langle\alpha,\beta\rangle\in\Gamma_{s,k+1}} \beta^{\top}F_v(s,b_{\mathcal{E}},a,s',z),\end{aligned} \tag{15}$$

where $F_v : \mathcal{S}\times\mathcal{B}_{\mathcal{E}}\times\mathcal{A}\times\mathcal{S}\times\mathcal{Z}\to\mathcal{B}_{\mathcal{E}}$ is the belief vector update, i.e., $b'_{\mathcal{E}} = F_v(s,b_{\mathcal{E}},a,s',z)$. Afterwards, we update the support vectors pair associated with an action a (line 2). These vectors are then used to compute the set of admissible actions (line 3) and the optimal action a^* (line 4). Finally, we add the optimal pair $\langle\alpha^{a^*},\beta^{a^*}\rangle$ to the set of support vector pairs $\Gamma_{s,k}$, which approximate the optimal value function $V^*_k(s,\cdot)$ from (10) for all $s \in \mathcal{S}$.

The `Backup` function from Algorithm 1 is used to update the sets $\Gamma_{s,k}$, which define the value function approximation:

$$\begin{aligned}V_k(s,b_{\mathcal{E}}) = \max_{\langle\alpha,\beta\rangle\in\Gamma_{s,k}}\ &\alpha^{\top}b_{\mathcal{E}} \\ \text{subject to}\quad &\langle\alpha,\beta\rangle \in \operatorname*{argmax}_{\langle\alpha,\beta\rangle\in\Gamma_{s,k}} \beta^{\top}b_{\mathcal{E}}.\end{aligned} \tag{16}$$

[1]For more details on the belief propagation please refer to [1].

Algorithm 1: `Backup`, (α,β)-vectors computation

Input: $s, b_{\mathcal{E}}, \Gamma_{s,k+1}, \Gamma_{s,k}$

1 For all $s' \in \mathcal{S}$, $a \in \mathcal{A}$, $z \in \mathcal{Z}$
  $\langle\alpha^{s',a,z},\beta^{s',a,z}\rangle \leftarrow$ from Equation (15) ;

2 For all $a \in \mathcal{A}$, $e \in \mathcal{E}$
  $\alpha^a(e) \leftarrow \mathbb{1}_{\mathcal{G}}(s) + \sum_{s',z,e'} F(s,e,a,s',e',z)\alpha^{s',a,z}(e)$
  $\beta^a(e) \leftarrow \mathbb{1}_{\mathcal{G}}(s) + \mathbb{1}_{\mathcal{S}\setminus\mathcal{G}}(s)\sum_{s',z,e'} F(s,e,a,s',e',z) \times\alpha^{s',a,z}(e)$;

3 Compute $\mathcal{C} = \operatorname{argmax}_{a\in\mathcal{A}}(b_{\mathcal{E}}^{\top}\beta^a)$;

4 Compute $a^* = \operatorname{argmax}_{a\in\mathcal{C}}(b_{\mathcal{E}}^{\top}\alpha^a)$;

5 Add $\langle\alpha^{a^*},\beta^{a^*}\rangle$ to $\Gamma_{s,k}$;

Output: $\Gamma_{s,k}$

Algorithm 2: Value function update

Input: $\Gamma_{s,k+1}$

for $s \in \mathcal{S}$ **do**
  Initialize $\Gamma_{s,k} = \emptyset$;
  for $b_{\mathcal{E}} \in \mathcal{D}_b$ **do**
    $\Gamma_{s,k} \leftarrow$ `Backup`$(s, b_{\mathcal{E}}, \Gamma_{s,k+1}, \Gamma_{s,k})$;
  end
end

Output: $\Gamma_{s,k}$

For time $k \in \{0,\dots,N-1\}$, the sets $\Gamma_{s,k}$ are recursively computed using Algorithm 2, which for all $s \in \mathcal{S}$ computes the support vector pairs at all belief points $b_{\mathcal{E}} \in \mathcal{D}_b$. The recursion is initialized setting $\Gamma_{s,N}$ equal to $\Gamma^*_{s,N}$.

Finally, we show that the sets of support vector pairs $\Gamma_{s,k}$ computed using Algorithms 1 and 2 allow us to define an approximated constraint value function, which is a lower-bound of the probability of satisfying the specifications.

Theorem 1. *Let $\Gamma_{s,k}$ be the set of support vectors constructed using the point-based strategy from Algorithms 1 and 2. Then, the approximated constraint value function*

$$J_k(s,b_{\mathcal{E}}) = \max_{\langle\alpha,\beta\rangle\in\Gamma_{s,k}} \beta^{\top}b_{\mathcal{E}} \tag{17}$$

is a lower-bound of the probability that the control policy π^{TOQ} satisfies the specifications, i.e., $J_k(s,b_{\mathcal{E}}) \le \mathbb{P}^{\pi^{TOQ}}\big[[\omega_k,\dots,\omega_N] \models \psi\big], \forall k \in \{0,\dots,N\}$.

Proof: The β-vectors computed by the backup Algorithms 1–2 are a subset of the β-vectors from (11), which define the optimal value function from (13). Therefore, as $\Gamma_{s,k} \subseteq \Gamma^*_{s,k}$ we have that

$$J_k(s,b_{\mathcal{E}}) = \max_{\langle\alpha,\beta\rangle\in\Gamma_{s,k}} \beta^{\top}b_{\mathcal{E}} \le \max_{\langle\alpha,\beta\rangle\in\Gamma^*_{s,k}} \beta^{\top}b_{\mathcal{E}} = J^*_k(s,b_{\mathcal{E}}),$$

$\forall k \in \{0,\dots,N\}, \forall s \in \mathcal{S}$ and $\forall b_{\mathcal{E}} \in \mathcal{B}_{\mathcal{E}}$. The above equation and Lemma 3 imply that $J_k(s,b_{\mathcal{E}}) \le \mathbb{P}^{\pi^{\mathrm{TOQ}}}\big[[\omega_k,\dots,\omega_N] \models \psi\big], \forall k \in \{0,\dots,N\}, \forall s \in \mathcal{S}$ and $\forall b_{\mathcal{E}} \in \mathcal{B}_{\mathcal{E}}$. ■

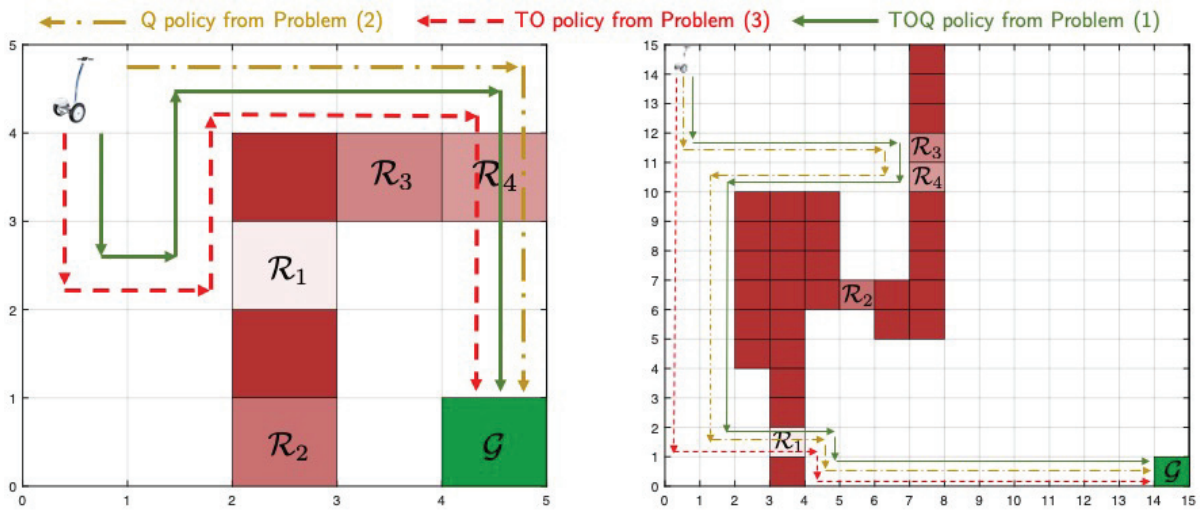

Fig. 2. Grid worlds with several obstacles (dark brown), one goal region $\mathcal{G}$ (green) and four uncertain regions (light brown) $\mathcal{R}_1$, $\mathcal{R}_2$ $\mathcal{R}_3$, and $\mathcal{R}_4$, which may be free with probability 0.9, 0.4, 0.3, and 0.5. The closed-loop trajectories are associated with different environment realizations. In particular, $\mathcal{R}_4$ and $\mathcal{R}_1$ are traversable in the 5x5 and 15x15 grid worlds, respectively.

VI. Examples

A. Grid Worlds

The proposed strategy is tested on three grid worlds shown in Figures 1 and 2. We compared the proposed Time-Optimal Quantitative (TOQ) policy approximated using a one-step look ahead and the value function from Section V with the Quantitative (Q) and Time-Optimal (TO) policies from Problems (2)–(3), which are approximated using standard point-based approaches for reward maximization[2]. In all simulations, the Segway receives a perfect measurement when adjacent to an uncertain region, a measurement which is correct with probability 0.8 when one grid cell away in the diagonal direction and an uninformative measurement otherwise. The uncertain regions $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$, may be traversable with probability 0.9, 0.3, 0.4, and 0.5. Finally, in order to analyze the effect of the number of uncertain regions on the computational complexity, we also tested a scenario where region $\mathcal{R}_4$ is a known obstacle.

Figures 1 and 2 show the closed-loop trajectories for different realizations of the uncertain regions. We notice that the TOQ policy behaves similar to the TO one, when the constraint from Problem (1) does not restrict the search space. Consider the 5x5 grid world in Figure 2, where the agent can explore all uncertain regions in different orders, as the task horizon is $T = 30$. In this example, the TOQ policy first explores region $\mathcal{R}_1$, and then it steers the agent through region $\mathcal{R}_4$. This behavior minimizes the expected time to complete the task, as region $\mathcal{R}_1$ has the highest probability of being free. Thus, the closed-loop trajectories associated with the TO and TOQ policies overlap. On the other hand, in the 15x15 grid world from Figure 2, the task horizon is $T = 40$ and the agent cannot explore all regions. Therefore, the TOQ policy maximizes the number of visited regions and behaves as the Q policy. Indeed, in this 15x15 grid world, first visiting region $\mathcal{R}_1$, which has the highest probability of being free, would lead to a lower probability of mission success. In general, the TOQ policy minimizes the expected time to complete the task, without compromising the probability of satisfying the specifications, as we have seen in Figure 1.

Table I shows the expected time to complete the task, the probability of failure, the upper-bound of the probability of failure[3], the total time to approximate the value function and the backup time required to approximate the value function at a belief point $b_{\mathcal{E}} \in \mathcal{D}_b$. The TO and Q policies are approximated using a standard point-based backup update and the TOQ policy is computed using the `backup` function from Algorithm 1. The TO policy from Problem (3) minimizes the expected time to complete the task but, as a result, it incurs in the highest probability of failure. On the other hand, the proposed TOQ policy has a probability of failure equal to the Q policy, which is computed maximizing the probability of satisfying the specifications. Therefore, the proposed strategy is able to minimize the expected time to complete the task, without compromising the probability of mission success. Notice that as a trade-off the computational burden of synthesizing the proposed TOQ policy is higher compared to the one needed to synthesize the Q and TO policies. This result is expected as we are approximating the solution to Problem (1) using a pair of vectors; whereas, the point-based strategy used to approximate Problems (2)–(3) maintains a single support vector per belief point. Finally, we underline that the backup time shown in Table I is associated with the computation of the support vectors at a discrete belief point $b_{\mathcal{E}} \in \mathcal{D}_b$. Thus, it mostly depends on the dimension of the belief space, which grows exponentially

[2]Code available online: `https://github.com/urosolia/MOMDP`. All simulations are run on a 2015 Macbook Pro with a 2.5GHz Quad-Core Intel Core i7 and 16GB of memory.

[3]For the TOQ policy the probability of failure $(1 - \mathbb{P}^{\pi}[\boldsymbol{\omega} \models \phi]) \leq 1 - J_0(s, b_{\mathcal{E}})$ where $J_0(s, b_{\mathcal{E}})$ is defined in Lemma 1.

Grid World	Exp. Time	Prob. Failure	Failure Bound	Total Time [s]	Backup Time [ms]
$[5x5]_3^{TO}$	8.12	4.2%	N/A	4.09	0.45
$[5x5]_3^{Q}$	27.78	4.2%	$\leq 4.2\%$	3.49	0.39
$[5x5]_3^{TOQ}$	8.12	4.2%	$\leq 4.2\%$	8.29	0.92
$[5x5]_4^{TO}$	8.2	2.1%	N/A	7.09	0.62
$[5x5]_4^{Q}$	28.39	2.1%	$\leq 2.1\%$	6.57	0.58
$[5x5]_4^{TOQ}$	8.2	2.1%	$\leq 2.1\%$	16.86	1.48
$[10x5]_3^{TO}$	4.23	4.2%	N/A	4.69	0.33
$[10x5]_3^{Q}$	29.0	0%	$\leq 0\%$	4.63	0.32
$[10x5]_3^{TOQ}$	6.2	0%	$\leq 0\%$	11.71	0.82
$[10x5]_4^{TO}$	4.53	2.1%	N/A	8.58	0.48
$[10x5]_4^{Q}$	29.0	0%	$\leq 0\%$	9.34	0.52
$[10x5]_4^{TOQ}$	6.2	0%	$\leq 0\%$	24.42	1.37
$[15x15]_3^{TO}$	25.2	10%	N/A	36.15	0.33
$[15x15]_3^{Q}$	36.66	6%	$\leq 6\%$	36.21	0.34
$[15x15]_3^{TOQ}$	31.72	6%	$\leq 6\%$	86.32	0.81
$[15x15]_4^{TO}$	25.2	10%	N/A	75.66	0.58
$[15x15]_4^{Q}$	37.83	3%	$\leq 8.9\%$	73.97	0.56
$[15x15]_4^{TOQ}$	29.86	3%	$\leq 3.5\%$	186.52	1.43

TABLE I

COMPARISON BETWEEN THE TOQ, Q AND TO POLICIES COMPUTED APPROXIMATING PROBLEMS (1)–(3), RESPECTIVELY. IN THE TABLE, THE GRID WORLD $[\text{XxY}]_i^j$ IS DEFINED BY XxY GRID CELLS, i UNCERTAIN REGIONS AND THE CONTROL POLICY $j \in \{\text{TO}, \text{Q}, \text{TOQ}\}$.

with the number of uncertain regions. Clearly, the total time needed to synthesize the control policy depends also on the grid size and number of belief points, as the `backup` update from Algorithm 2 is used repeatedly to approximate the value function. Indeed, when parallel computing is not available, the total computational cost scales linearly with the number of observable states $|\mathcal{S}|$ and discrete belief points $|\mathcal{D}_b|$.

B. Navigation Task

In this section, we use the proposed time-optimal quantitative policy (1) as high-level decision maker for the navigation problem shown in Figure 3, where a Segway has to explore a partially known environment to locate science samples that may be located in the goal regions $\mathcal{G}_i$. The specification $\psi = \neg\texttt{collision} U((\texttt{Goal}_1 \wedge \texttt{sample}_1) \vee (\texttt{Goal}_2 \wedge \texttt{sample}_2))$, where the atomic proposition $\texttt{sample}_i$ is satisfied if the region $\mathcal{G}_i$ contains a science sample and the atomic proposition $\texttt{Goal}_i$ is satisfied if the Segway is in a goal cell $\mathcal{G}_i$. We implemented a hierarchical controller, where the proposed time-optimal quantitative policy (1) computes high-level commands and a model predictive controller [33] is used to compute low-level inputs. The high-level commands are move North, South, East and West and they are used to compute the cell where the Segway should move next. Then, the low-level control problem is solved as a standard regulation problem [33], where the goal is to steer the Segway

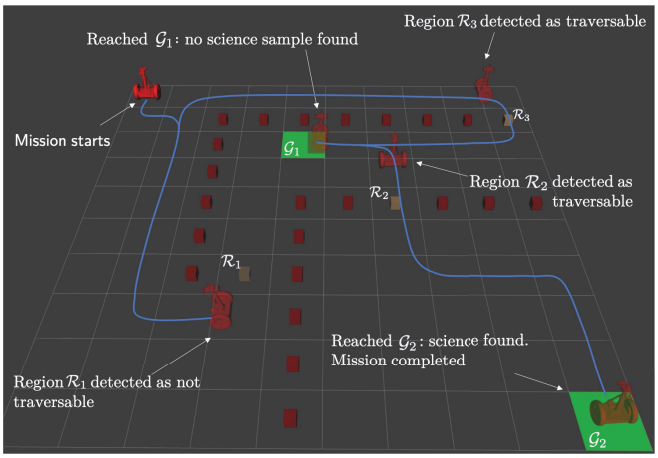

Fig. 3. Evolution of the Segway (blue) in the high-fidelity simulator. The TOQ policy decides to first explore region $\mathcal{R}_1$, which in this example is not traversable. Afterwards, the controller explores regions $\mathcal{R}_2$, $\mathcal{G}_1$ and $\mathcal{R}_3$. Finally, the Segway reaches region $\mathcal{G}_2$, which in this example contains the science sample.

to the center of the goal cell. When a transition from cell i to cell j occurs, we update the belief about the environment and the observable state of the MOMDP, which represents the cell where the Segway is located. The accuracy of the environment observations decays exponentially as a function of the distance between the Segway and the measured region. In particular, for the binary variable $r^{(i)} \in \{0, 1\}$, which represents the traversability of the region $\mathcal{R}_i$, we receive a measurement $z_r^{(i)}$ which is accurate with the following probability:

$$P(z_r^{(i)} = 1 | r^{(i)} = 1, s) = \begin{cases} 1 & \text{if } d(s, \mathcal{R}_i) \leq 1, \\ 0.5 + 0.3e^{-(d(s,\mathcal{R}_i)-2)/2.5} & \text{otherwise,} \end{cases}$$

where $d(s, \mathcal{R}_i)$ represents the Manhattan distance between the Segway and region $\mathcal{R}_i$. Similarly, we define the binary variable $g^{(i)} \in \{0, 1\}$, which equals to one when region $\mathcal{G}_i$ contains a science sample and zero otherwise, and we receive an observation $z_g^{(i)}$ which has the following accuracy:

$$P(z_g^{(i)} = 1 | g^{(i)} = 1, s) = \begin{cases} 1 & \text{if } d(s, \mathcal{R}_i) = 0, \\ 0.5 + 0.25e^{-d(s,\mathcal{R}_i)/1.5} & \text{otherwise.} \end{cases}$$

Figure 3 shows the closed-loop trajectory of the Segway. At the beginning of the simulation, the probability that regions $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$, may be traversable is 0.7, 0.5, and 0.4. Furthermore, the probability that regions $\mathcal{G}_1$ and $\mathcal{G}_2$ contain the science sample is 0.8 and 0.6, respectively. The controller first explores region $\mathcal{R}_1$, which has the highest probability of being traversable. However, in this example region $\mathcal{R}_1$ is not traversable and therefore the Segway steers to region $\mathcal{R}_3$. As shown in Figure 4, the environment observations are used to update environment beliefs and the probability of mission success, which represents the

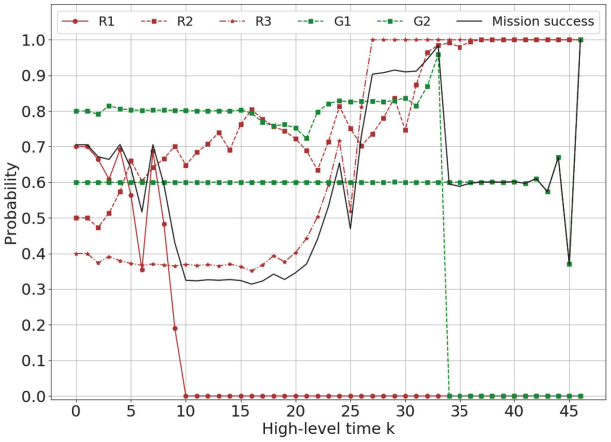

Fig. 4. Probability of satisfying the specification. The figure shows also the evolution of the belief for the uncertain and goal regions.

probability of satisfying the mission specifications. Notice that, when the Segway detects that region $\mathcal{G}_1$ does not contain a science sample at the high-level time $k = 34$, the probability of mission success drops, as shown in Figure 4. Afterwards, the controller explores region $\mathcal{R}_3$ and steers the Segway to region $\mathcal{G}_2$, which contains a science sample. Finally, we notice that for all high-level time steps $k \geq 37$ the controller is uncertain only about the state of region $\mathcal{G}_2$, therefore the probability of mission success overlaps with the probability that region $\mathcal{G}_2$ contains a science sample.

VII. Conclusions

In this work, we studied time-optimal quantitative problems for MOMDPs. First, we presented a dynamic programming update to compute the value function of time-optimal quantitative problems. Afterwards, we leveraged the piecewise-affine nature of the optimal value function to define a point-based approximation strategy, which allows us to compute a lower bound of the probability of satisfying the specifications. Finally, we compared the proposed strategy with time-optimal and quantitative policies.

References

[1] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[2] E. J. Sondik, "The optimal control of partially observable markov processes over the infinite horizon: Discounted costs," *Operations research*, vol. 26, no. 2, pp. 282–304, 1978.

[3] J. D. Isom, S. P. Meyn, and R. D. Braatz, "Piecewise linear dynamic programming for constrained POMDPs." in *AAAI*, vol. 1, 2008, pp. 291–296.

[4] D. Kim, J. Lee, K.-E. Kim, and P. Poupart, "Point-based value iteration for constrained POMDPs," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[5] P. Poupart, A. Malhotra, P. Pei, K.-E. Kim, B. Goh, and M. Bowling, "Approximate linear programming for constrained partially observable markov decision processes," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[6] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. IEEE, 1977, pp. 46–57.

[7] K. Chatterjee, M. Chmelik, 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)*. IEEE, 2015, pp. 325–330.

[8] K. Chatterjee, M. Chmelik, and M. Tracol, "What is decidable about partially observable markov decision processes with ω-regular objectives," *Journal of Computer and System Sciences*, vol. 82, no. 5, pp. 878–911, 2016.

[9] S. Junges, N. Jansen, R. Wimmer, T. Quatmann, L. Winterer, J.-P. Katoen, and B. Becker, "Finite-state controllers of pomdps via parameter synthesis," 2018.

[10] S. Junges, N. Jansen, and S. A. Seshia, "Enforcing almost-sure reachability in POMDPs," *arXiv preprint arXiv:2007.00085*, 2020.

[11] M. Ahmadi, R. Sharan, and J. W. Burdick, "Stochastic finite state control of POMDPs with ltl specifications," *arXiv preprint arXiv:2001.07679*, 2020.

[12] P. Nilsson, S. Haesaert, R. Thakker, K. Otsu, C.-I. Vasile, A.-A. Agha-Mohammadi, R. M. Murray, and A. D. Ames, "Toward specification-guided active Mars exploration for cooperative robot teams," *Robotics: Science and Systems (RSS)*, 2018.

[13] M. Bouton, J. Tumova, and M. J. Kochenderfer, "Point-based methods for model checking in partially observable markov decision processes." in *AAAI*, 2020, pp. 10 061–10 068.

[14] S. Haesaert, R. Thakker, R. Nilsson, A. Agha-mohammadi, and R. M. Murray, "Temporal logic planning in uncertain environments with probabilistic roadmaps and belief spaces," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 6282–6287.

[15] 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)*. IEEE, 2016, pp. 7419–7424.

[16] S. Haesaert, P. Nilsson, C. I. Vasile, R. Thakker, A.-a. Agha-mohammadi, A. D. Ames, and R. M. Murray, "Temporal logic control of pomdps via label-based stochastic simulation relations," *IFAC-PapersOnLine*, vol. 51, no. 16, pp. 271–276, 2018.

[17] Y. Wang, S. Chaudhuri, and L. E. Kavraki, "Bounded policy synthesis for POMDPs with safe-reachability objectives," *arXiv preprint arXiv:1801.09780*, 2018.

[18] K. Lesser and M. Oishi, "Approximate safety verification and control of partially observable stochastic hybrid systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 81–96, 2016.

[19] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of markov decision processes with linear temporal logic constraints," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[20] S. Karaman, R. G. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 2117–2122.

[21] E. M. Wolff and R. M. Murray, "Optimal control of nonlinear systems with temporal logic specifications," in *Robotics Research*. Springer, 2016, pp. 21–37.

[22] F. Teichteil-Königsbuch, "Path-constrained markov decision processes: bridging the gap between probabilistic model-checking and decision-theoretic planning." 2012.

[23] ——, "Stochastic safest and shortest path problems." in *AAAI*, 2012.

[24] F. W. Trevizan, F. Teichteil-Königsbuch, and S. Thiébaux, "Efficient solutions for stochastic shortest path problems with dead ends." in *UAI*, 2017.

[25] B. Lacerda, F. Faruq, D. Parker, and N. Hawes, "Probabilistic planning with formal performance guarantees for mobile service robots," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1098–1123, 2019.

[26] A. Kolobov, D. Weld *et al.*, "A theory of goal-oriented MDPs with dead ends," *arXiv preprint arXiv:1210.4875*, 2012.

[27] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 89.

[28] J. Pineau, G. Gordon, S. Thrun *et al.*, "Point-based value iteration: An anytime algorithm for POMDPs," in *IJCAI*, vol. 3, 2003, pp. 1025–1032.

[29] P. Poupart and C. Boutilier, "Bounded finite state controllers," in *NIPS*, 2003.

[30] M. Péron, K. Becker, P. Bartlett, and I. Chadès, "Fast-tracking stationary MOMDPs for adaptive management problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[31] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.

[32] M. Araya-López, V. Thomas, O. Buffet, and F. Charpillet, "A closer look at MOMDPs," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2. IEEE, 2010, pp. 197–204.

[33] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.