Temporal logic planning in uncertain environments with probabilistic roadmaps and belief spaces

S. Haesaert[1], R. Thakker[2], P. Nilsson[3], A. Agha-mohammadi[2], R.M. Murray[3]

***Abstract*— Navigation problems expressed via temporal logics show promise for autonomous robot applications due to their versatility. In this paper, we introduce a method for planning with these specifications in uncertain environments that yields guaranteed satisfaction probabilities. We show that point-based value iteration can be combined with probabilistic roadmaps to solve this planning problem over the belief space of the uncertain environment.**

I. INTRODUCTION

Probabilistic planning problems such as the rover science exploration problem (Fig.1) and the planning of search-and-rescue missions executed by autonomous robots are planning problems that pose two main challenges, firstly they require long term autonomy combined with intelligent, complex behavior, and secondly, they require the planning to work well in very uncertain environments. The first challenge has been addressed by using temporal logics, which adds new specification facets that are more complex and carry the promise for more long term autonomy. Tackling both challenge, this paper introduces a temporal logic planning approach for uncertain environments around the exemplary problem of rover science exploration depicted in Fig.1. The objective of the rover is to collect samples of science targets while avoiding hazards such as large rocks, deep sand or steep slopes. The presence of science targets and hazards regions is uncertain and needs to be explored by the rover while executing the science mission. For basic science missions, this partially observable Markov decision problem (POMDP) has been solved as a simple discounted reward problem in [21].

The majority of high-level specifications that expresses missions in terms of formulae in a temporal logic of choice have been developed and applied to planning problems in a deterministic setting. This includes the temporal logic planning [12], and reactive planning in [13], [26] for deterministic models. Equivalent results for general stochastic models with guarantees are computationally intractable for continuous space systems and require using approximate abstractions [10], [23], [27] to obtain solutions.

Leveraging sampling-based planning or probabilistic roadmaps, there are many promising solutions to complex planning problems specified with temporal logics. This includes inter alia [5], [6], [19]. Convergence to optimality is proven for sampling-based algorithms that solve motion planing problems using process algebra specifications in [24]. These methods can be extended to handle state uncertainty for partially observable rovers [2], [25], but they cannot yet be used to deal with partially observable environment states.

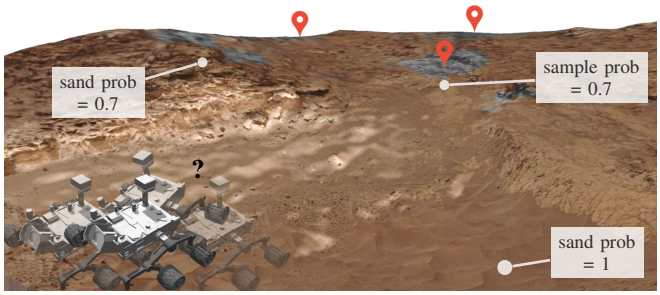

Fig. 1. The figure illustrates the knowledge available to the rover while planning science missions. The goal is to obtain samples and avoid hazard regions such as sandy spots. The rover does not recognize a priori whether a region does or does not contain a sample or an obstacle.

In contrast, the quest to solve planning problems in partially observable and stochastic domains [11] has led to the development of the tailored approximate dynamic programming methods referred to as *point-based value iteration* [18]. These solvers have been used to solve reachability and safety problems for finite, partially observable Markov decision problem [15], [16], and to solve probabilistic specifications [17]. Originally developed for finite-state problems, these methods need computationally expensive abstractions to work for continuous-state models. As such, we prefer to not use pure point-based value iteration to solve complex navigation problems. In this work, we give an algorithm that combines the merits of roadmaps for planning with the rover's state, and point-based value iterations for dealing with (finite) partially observable environments.

The article is organized as follows. In the next section, the temporal logic navigation problem in uncertain environments is introduced. Section III details the conceptual idea of the integration of temporal logic specifications, planning and belief points into a unified method. Subsequently, Section IV contains the technical details and proofs of the theoretical results needed to give the complete method in Section V. Section VI presents a Mars rover case study, before conclusions are given in the last section.

II. FRAMEWORK AND PROBLEM STATEMENT

As a concrete example of our navigation problem in uncertain environments, we focus on a typical environment

[1]Sofie Haesaert is with Faculty of Electrical Engineering, Eindhoven University of Technology, The Netherlands s.haesaert@tue.nl
[2]Rohan Thakker and Ali Agha-mohammadi are with the Jet Propulsion Laboratory, California, USA [3]Petter Nilsson and R.M. Murray are with California Institute of Technology.

that the Mars rover navigates through consisting of different terrain types including sand, rocks, etc.

The rover model: The rovers movements are modeled as an ordinary differential equation with $\dot{x} = f(x,u_r) + w$, where $x \in \mathbb{X}$ is the state of the rover, u_r the input, and w is a stochastic disturbance. Beyond the current location of the rover, its state can include its velocity and orientation.

The rover environment: The rover navigates through different types of regions. Without loss of generality, we assume that prior knowledge of the map will tell us that the k-the uncertain area could be sand or not, but it will not tell us that it could be either sand or rock, or nothing. That is, we assume that the different region types are always distinguishable. Therefore, the *environment state* x_e for an environment with n_e uncertain regions takes valuations in the finite set $x_e \in \{0,1\}^{n_e} =: \mathbb{X}_e$. As such $x_e(k) = 1$ if the region type holds in that region and $x_e(k) = 0$ if not. Knowledge of the environment expressed by the value of x_e is not available and can only be inferred from noisy measurements or observations made by the rover. As

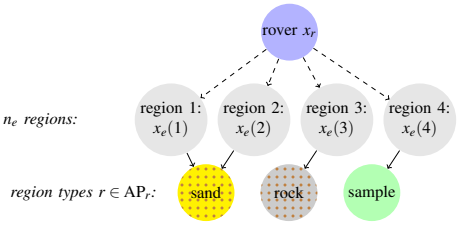

Fig. 2. The rover state x_r is modeled together with unknown environment state x_e and the types of the individual regions.

in Figure 1, we can express the prior belief that the k-th region actually contains the samples or obstacles as a belief distribution b that assigns to each possible environment state x_e a probability $b(x_e)$. We refer to this distribution as the belief state b of the environment, and this state is such that $b : \mathbb{X}_e \to [0,1]$ and $\sum_{x_e \in \mathbb{X}_e} b(x_e) = 1$. The space of possible belief is denoted as $\mathbb{B}$. An equivalent vector-based notation $\mathbf{b} := \begin{bmatrix} b(x_e^1) & b(x_e^2) & b(x_e^3) & \dots \end{bmatrix}^T$ based on an enumeration of the environment states x_e^i for which $\mathbb{X}_e =: \{x_e^1, x_e^2, x_e^3, \dots, x_e^n\}$ with $n = 2^{n_e} = |\mathbb{X}_e|$ can also be used.

The rover interacts with the environment via observations. The interaction is modeled via an observation model $(\mathbb{O}, \mathsf{ob})$, where $\mathbb{O} := \{o_1, \dots, o_m\}$ denotes the finite set of possible observations; and $\mathsf{ob}(\cdot)$ denotes the stochastic kernel that assigns probabilities to observations $o \in \mathbb{O}$ given $x_e \in \mathbb{X}_e$:

$$\mathsf{ob} : \mathbb{X}_e \times (\mathbb{X} \times \mathbb{U}_e) \times \mathbb{O} \to [0,1]. \tag{1}$$

The observation action $u_e \in \mathbb{U}_e$ is the choice of region(s) that the rover is observing and the current state of the rover is given as $x \in \mathbb{X}$. The observation probability depends on x via the distance to the observed region.

Based on observation o, the rover state x, and action $(x,u) \in \mathbb{X} \times \mathbb{U}_e$, the belief $b = \mathbb{P}(x_e | \{\text{ past info}\})$ is updated as

$$b^{(x,u),o}(x_e) = \frac{\mathbb{P}(o|x_e,(x,u))}{\mathbb{P}(o|b,(x,u))} b(x_e). \tag{2}$$

Denote $\mathbf{O} \in \mathbb{R}^{m \times n}$ with the ij-th element defined as $[\mathbf{O}(x,u)]_{ij} := \mathsf{ob}(o_i | x_e^j, (x,u))$ and denote with $\mathbf{O}^o(x,u)$ the row vector of $\mathbf{O}(x,u)$ for observation o. Remark that

$$\mathbb{P}(o|b,(x,u)) = \textstyle\sum_{x_e \in \mathbb{X}_e} \mathsf{ob}(o|x_e,(x,u)) b(x_e) = \mathbf{O}^o(x,u)\mathbf{b}. \tag{3}$$

Adopting the notation of [3], the updated belief becomes

$$\mathbf{b}^{(x,u),o} = \frac{\operatorname{diag}[\mathbf{O}^o(x,u)]\,\mathbf{b}}{\|\operatorname{diag}[\mathbf{O}^o(x,u)]\,\mathbf{b}\|_1} \tag{4}$$

with the 1-norm $\|\mathbf{a}\|_1 := \sum |a_i|$.

Specifications in linear temporal logic: Consider a set $AP = \{p_1, \dots, p_L\}$ of atomic propositions that defines an *alphabet* $\Sigma := 2^{AP}$ where each *letter* $\pi \in \Sigma$ is composed of a set of atomic propositions. An infinite string of letters forms a *word* $\boldsymbol{\pi} = \pi_0 \pi_1 \pi_2 \ldots \in \Sigma^{\mathbb{N}}$. Specifications imposed on the behavior the rover are defined as formulas composed of atomic propositions and operators. We consider the syntactically co-safe subset of linear-time temporal logic (scLTL) properties [14]. This subset of interest consists of temporal logic formulae constructed according to the following syntax

$$\psi ::= p \mid \neg p \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \psi_1 \mathscr{U} \psi_2 \mid \bigcirc \psi. \tag{5}$$

where $p \in \mathrm{AP}$ is an atomic proposition. The *semantics* of scLTL are defined recursively over $\boldsymbol{\pi}_i$ as $\boldsymbol{\pi}_i \models p$ iff $p \in \pi_i$; $\boldsymbol{\pi}_i \models \psi_1 \wedge \psi_2$ iff $(\boldsymbol{\pi}_i \models \psi_1) \wedge (\boldsymbol{\pi}_i \models \psi_2)$; $\boldsymbol{\pi}_i \models \psi_1 \vee \psi_2$ iff $(\boldsymbol{\pi}_i \models \psi_1) \vee (\boldsymbol{\pi}_i \models \psi_2)$; $\boldsymbol{\pi}_i \models \psi_1 \mathscr{U} \psi_2$ iff $\exists j \geq i$ s.t. $(\boldsymbol{\pi}_j \models \psi_2)$ and $\boldsymbol{\pi}_k \models \psi_1, \forall k \in \{i, \dots j-1\}$; and $\boldsymbol{\pi}_i \models \bigcirc \psi$ iff $\boldsymbol{\pi}_{i+1} \models \psi$. The eventually operator $\diamondsuit\ \psi$ is used in the sequel as a shorthand for $\top \mathscr{U} \psi$. We say that $\boldsymbol{\pi} \models \psi$ iff $\boldsymbol{\pi}_0 \models \psi$.

Specifying rover behavior: Together with the rover location, the observations define the set of atomic propositions that hold true at a given time instant. A labeling function $\mathsf{L}_r : \mathbb{X} \to \mathrm{AP}_r$ maps the rover location to the region type $r \in \mathrm{AP}_r$. Similarly, the observations $o \in \mathbb{O}$ map to AP_o with the labeling $\mathsf{L}_o : \mathbb{O} \to 2^{\mathrm{AP}_o}$. Therefore, the rover generates words $\boldsymbol{\pi}$ with letters $\pi_i \in \Sigma := 2^{\mathrm{AP}}$ with $\mathrm{AP} := \mathrm{AP}_r \cup \mathrm{AP}_o$. The mission executed by the rover satisfies ψ if $\boldsymbol{\pi} \models \psi$.

Problem statement: The objective of this work is to design a policy μ such that a specification ψ is satisfied with a guaranteed lower bound on the satisfaction probability, i.e.,

$$\mathbb{P}^{\mu}(\boldsymbol{\pi} \models \psi). \tag{6}$$

The policy μ maps the history of the rover state and its observations to the rover control and observation actions.

III. Beliefs, Roadmaps and Temporal Logics

A. Planning with temporal logic specifications

Consider a *deterministic finite-state automaton* (DFA), defined by the tuple $\mathscr{A} = (Q, q_0, \Sigma, \tau_{\mathscr{A}}, Q_f)$, where Q is a finite set of states, $q_0 \in Q$ is an initial state, Σ is an input alphabet, $\tau_{\mathscr{A}} : Q \times \Sigma \to Q$ is a transition function, and $Q_f \subseteq Q$ is a set of accepting states. A word $\boldsymbol{\pi} = \pi_0 \pi_1 \pi_2 \ldots$ *is accepted* by a DFA if there exists a sequence $q_0 q_1 q_2 \ldots q_f$ with $q_f \in Q_f$, that starts with the initial state q_0 and for which $q_{k+1} = \tau_{\mathscr{A}}(q_k, \pi_k)$. In other words, a sequence of letters is accepted if the resulting execution in the DFA *reaches* the set

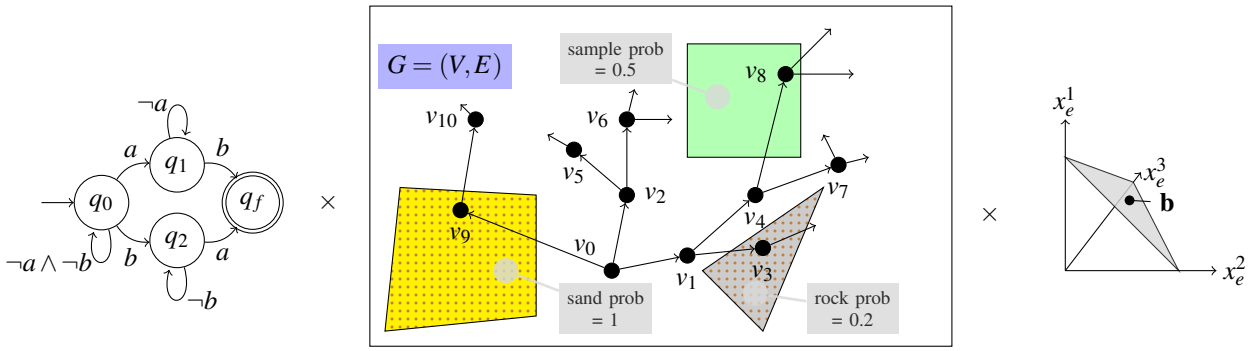

Fig. 3. The figure depicts the idea for belief space temporal logic planning. Properties are expressed as reach(-avoid) problems over deterministic finite state automata (DFA). This reach problem is solved over the probabilistic roadmap for the rover navigation with the point-based value iteration over the belief spaces.

of accepting states. We denote the set of words accepted by a DFA $\mathscr{A}$ as $\mathscr{L}(\mathscr{A})$. For every scLTL property ψ there exists a DFA $\mathscr{A}_\psi$ modeling it [4], [14]. In particular, $\boldsymbol{\pi} \models \psi$ iff $\boldsymbol{\pi} \in \mathscr{L}(\mathscr{A}_\psi)$. *Problem* (6) *can be converted into the problem of constructing a reachability-enforcing policy over the DFA* $\mathscr{A}_\psi$.

B. Probabilistic roadmap for rover navigation

We abstract the dynamics of the rover by using a probabilistic roadmap (PRM) [8] as depicted in the middle of Fig. 3. A collection of useful states/configurations v_i is selected in a random fashion. Then, a graph $G = (V, E)$ is built with vertices $V := \{v_1, \ldots, v_n\}$ and edge set $E := \{e_{ij}, \ldots\}$ with e_{ij} the edge connecting nodes v_i and v_j. Each vertex v_i is associated to a rover state or configuration. Between nodes, the rover is controlled by a low-level edge controller and follows a precomputed path. We limit the roadmap G as follows. Firstly, at each vertex, the rover choose which neighboring node to drive to and what sensor to use to observe the environment. Secondly, we require that for any two neighboring states (v_1, v_2) there exists a local control $u_r(t)$ and risk $1 - p_{v_1,v_2}$ such that the realized trajectory between v_1, v_2 does not change its label $\mathsf{L}_r(x(\cdot))$ more than once with the probability p_{v_1,v_2}. This requirement is commonly used for deterministic temporal logic planning [7], [24] as it simplifies the computations while it does not affect the existence of solutions. Additionally, we require that there is a vertex $v_0 \in V$ that represents the initial state of the rover at the beginning of the planning problem.

Practically, we can split the computations for the PRM up in a *high-level planner* that plans over the PRM based on the uncertainty in the regions and prunes and resamples useful states, and a *low-level planner* that computes the risks associated to connecting vertices and that maps the local route of the rover. The low-level planner can be solved as for example a stochastic reach-avoid problem, see [1], [23].

C. Planning over the scLTL-PRM with belief states

While navigating with the rover, the state q_i of the DFA, the state v_i, of the PRM and the current belief state $\mathbf{b}_i$ of the environment are available. We refer to this combined system, given in Fig. 3 and 4, as the scLTL-PRM with belief states.

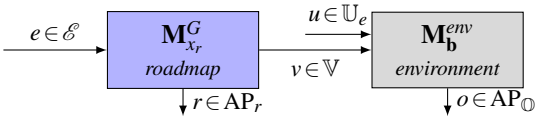

Fig. 4. Composition of the roadmap model $\mathbf{M}_{x_r}^G$ and the belief model of the environment $\mathbf{M}_{\mathbf{b}}^{env}$.

In conclusion, to solve the planning problem for scLTL specifications in uncertain hazardous environments, we will compute a navigation policy based on the current state $(q, v, \mathbf{b})$ that decide which $e \in \mathscr{E}$ to take towards the next vertex v' in the roadmap and the next observation action $u \in \mathbb{U}_e$. The objective is to find the policy or strategy that yields a guaranteed probability on the satisfaction of the specification ψ. In the next section, we will define the value functions for this navigation problem and show that they have a local piece-wise affine structure. This enables the use of point-based value iteration schemes [20] that use a finite number of beliefs.

IV. Exact value iteration

Consider a given Markov policy $\mu = (\mu_0, \ldots, \mu_{K-1})$, with $\mu_i : Q \times \mathbb{V} \times \mathbb{B} \to \mathscr{E} \times \mathbb{U}_e$. The probability of reaching Q_f starting from s_0 within K time steps can be expressed as

$$\mathbf{V}_\mu^K(s) = \mathbb{E}_\mu \left[\sum_{i=0}^{K} \mathbf{1}_{Q_f}(q_i) \prod_{j=0}^{i-1} \mathbf{1}_{Q \setminus Q_f}(q_j) \middle| s_0 = (q_0, v_0, \mathbf{b}_0) \right], \quad (7)$$

with $s = (q, v, \mathbf{b})$. The value function $\mathbf{V}_\mu^K(x)$ can be computed recursively as $\mathbf{V}_\mu^K = \mathsf{B}_\mu(\mathbf{V}_\mu^{K-1})$ with $\mathbf{V}_\mu^0 = \mathbf{1}_{Q_f}(q)$, and with the Bellman recursion

$$\mathsf{B}_\mu(\mathbf{V})(q, v, \mathbf{b}) = \max \left\{ \mathbf{1}_{Q_f}(q), \mathbb{E}_\mu^{\mathbb{S}} \left[\mathbf{V}\left(q', v', \mathbf{b}'\right) \mid (q, v, \mathbf{b}) \right] \right\}.$$

For the prefix $\boldsymbol{\pi}^{\leq K}$ of length K it holds that $\mathbb{P}_\mu(\boldsymbol{\pi}^{\leq K} \models \psi | s_0) = \mathbf{V}^K(s_0)$. Denote the optimal Bellman recursion as

$$\mathsf{B}_*(\mathbf{V})(q, v, \mathbf{b}) := \max_\mu \mathsf{B}_\mu(\mathbf{V})(q, v, \mathbf{b}) \quad (8)$$

with the resulting fixed point $\mathbf{V}_*^\infty := \lim_{K \to \infty} \mathsf{B}_*^K(\mathbf{V}^0)$ for which $\mathsf{B}_*^K(\mathbf{V}^0)$ defines the K-th iterated application of the

Bellman mapping. The maximal satisfaction probability is given as

$$\mathbb{P}^{\mu^*}(\pi \models \psi) = \mathbf{V}_*^{\infty}(q_0, v_0, \mathbf{b}_0). \tag{9}$$

If B_* is contractive, then $\mathbf{V}^{\infty}$ is the unique fixed point and optimal policy μ^* is the stationary Markov policy given as

$$\mu^* := \arg\max_{\mu} \mathsf{B}_{\mu}(\mathbf{V}_*^{\infty})(q, v, \mathbf{b}). \tag{10}$$

Theorem 1 (Properties of the optimal Bellman recursion): The Bellman operator B_* defined in equation (8) satisfies the following properties: if for all q, v: $\mathbf{V}(q, v, \mathbf{b})$ is *convex* with respect to $\mathbf{b} \in \mathbb{B}$, then $\mathsf{B}_*(\mathbf{V})(q, v, \mathbf{b})$ is convex with respect to $\mathbf{b} \in \mathbb{B}$. And if $\mathbf{V}$ is a convex piecewise affine function given as

$$\mathbf{V}(q, v, \mathbf{b}) = \max_{\alpha \in \Gamma(q,v)} \alpha \cdot \mathbf{b} \tag{11}$$

for which $\Gamma(q, v)$ are sets of vectors, then for all (q, v) there exists sets $\Gamma_{\mathsf{B}}(q, v)$ such that $\mathsf{B}_*(\mathbf{V})(q, v, \mathbf{b}) = \max_{\alpha \in \Gamma_{\mathsf{B}}(q,v)} \alpha \cdot \mathbf{b}$.

To prove that Theorem 1 holds, we analyze **1)** choosing and transitioning over a PRM edge and **2)** receiving an observation after choosing an observation action.

1) PRM edge: For a given input $e \in \mathscr{E}$, the partial Bellman operator can be denoted as

$$\mathsf{B}_e(\mathbf{V})(q, v, \mathbf{b}) = \max\Big\{\mathbf{1}_{Q_f}(q), \mathbb{E}_e[\mathbf{V}(q, v', \mathbf{b}') | q, v, \mathbf{b}, e]\Big\}. \tag{12}$$

This operator is simple since for the given edge choice e the next node v' is known and the state in the DFA remains constant. At the same time, the belief about the environment is preserved. Thus, the expectation in Eq. (12) reduces to

$$\mathbb{E}_e[\mathbf{V}(q, v', \mathbf{b}') \mid \mathbf{b}, v, q] = p_{v,v'} \mathbf{V}(q, v', \mathbf{b}) \tag{13}$$

for which probability $p_{v,v'}$ is associated to $v \xrightarrow{e} v'$. The convexity with respect to $\mathbf{b}$ is preserved and

$$\Gamma_{\mathsf{B}_e}(q, v) := \{p_{v,v'} \alpha | \alpha \in \Gamma(q, v')\} =: p_{v,v'} \Gamma(q, v').$$

Since $\mathbf{1}_{Q_f}(q)$ is constant with respect to $\mathbf{b}$, the maximization step preserves convexity and the piecewise affine structure.

2) Observation action: For a given observation action $u \in \mathbb{U}_e$, we can again express the partial Bellman recursion $\mathsf{B}_u(\mathbf{V})$ for the observation as

$$\mathbb{E}_u[\mathbf{V}(q', v, \mathbf{b}') \mid \mathbf{b}, v, q] = \textstyle\sum_{o \in \mathbb{O}} \mathbf{V}(q', v, \mathbf{b}') \mathbb{P}(o | \mathbf{b}, (v, u)) \tag{14}$$

for which q' is such that $q \xrightarrow{\mathsf{L}_r(v) \cup \mathsf{L}_o(o)} q'$. Consider Lemma 2 in [3], which states that if g is convex in $\mathbf{b}$, so is $g\big(\mathbf{b}^{(x,u),o}\big)\mathbb{P}(o|\mathbf{b}, (v, u))$. This implies that Eq. (14) is convex in $\mathbf{b}$, because the sum operator preserves convexity. If $\mathbf{V}$ is piecewise affine for each pair (v, q), then the above expectation can be written as follows

$$\sum_{o \in \mathbb{O}} \max_{\alpha \in \Gamma(v,q')} \alpha \cdot \mathbf{b}^{(v,u),o} \mathbb{P}(o \mid \mathbf{b}, (v, u)). \tag{15}$$

Here the set of alpha vectors $\Gamma(v, q')$ is given based on the pair v, q', which can be computed given observation o, node v, DFA state q, and the applied input $u \in \mathbb{U}_e$. For ease of notation, we index the updated belief $\mathbf{b}^{(v,u),o}$, with v instead of the corresponding location x. Denote $\alpha^{(v,u),o} = \mathrm{diag}[\mathbf{O}^o(u, v)]\alpha$, where for the given constant value of v the alpha vector is indexed based on the applied action and the returned observation, then it holds that

$$\Big(\alpha \cdot \mathbf{b}^{(v,u),o}\Big) \mathbb{P}(o \mid \mathbf{b}, (v, u)) \equiv \alpha^{(v,u),o} \cdot \mathbf{b}.$$

This reduces equation (15), to $\sum_{o \in \mathbb{O}} \max_{\alpha \in \Gamma(v,q')} \alpha^{(v,u),o} \cdot \mathbf{b}$. For a given u define

$$\Gamma_{\mathsf{B}_u}(v, q) := \Big\{ \sum_o \alpha^{(v,u),o} | \alpha \in \Gamma(v', q') \Big\} \text{ for } q \notin Q_f, \tag{16}$$

for which the expectation (14) equates to $\max_{\alpha \in \Gamma_{\mathsf{B}_u}} \alpha \cdot \mathbf{b}$.

We are now ready to compose the proof of Theorem 1.

Proof: [Proof of Theorem 1] The Bellman operators $\mathsf{B}_u(\mathbf{V})(\mathbf{b}, v, q)$ for $e \in \mathscr{E}$ or $u \in \mathbb{U}_e$ preserve convexity and preserve the piecewise affine nature of a value function. For a given composed action $(e, u) \in \mathscr{E} \times \mathbb{U}_e$, their sequential application yields a composed Bellman operator $\mathsf{B}_{(e,u)} := \mathsf{B}_u \circ \mathsf{B}_e$ that still preserves convexity and preserves the piecewise affine nature of a value function. The optimal Bellman recursion given as $\mathsf{B}_*(\mathbf{V})(\mathbf{b}, v, q) = \max_{(e,u)} \mathsf{B}_{(e,u)}(\mathbf{V})(\mathbf{b}, v, q)$. Since also the maximization over all $\mathscr{E} \times \mathbb{U}_e$ preserves these properties, we have proven Theorem 1. ∎

Since the value iterations are initialized with $\mathbf{1}_{Q_f}(q)$, this implies that for K finite iterations $\mathbf{V}_*^K(q, v, \mathbf{b})$ can be expressed as a convex piecewise affine function

$$\mathbf{V}_*^K(q, v, \mathbf{b}) = \max_{\alpha \in \Gamma(q,v)} \alpha \cdot \mathbf{b}.$$

With $\mathbf{V}_*^K \leq \mathbf{V}_*^{\infty}$ and, for $K \to \infty$, the value function $\mathbf{V}_*^K(q, v, \mathbf{b})$ converges to $\mathbf{V}_*^{\infty}$. For these value functions, the use of a roadmap has already introduced an approximation that leads to a type of sparseness, as only limited configurations or states of the rover need to be used to compute the value function. Next, we show that by working with a point-based value function also a reduced representation over the belief space is used, leading to an overall efficient method.

V. Point-based value iteration over scLTL-PRMs

To solve the value iteration approximately, we implement a point-based value iteration [18], [22]. Starting from a limited set of belief points $\mathscr{B}(v, q) := \{\mathbf{b}_i\}$ at each vertex, we now use approximate Γ sets that only include the α vectors associated to belief points in $\mathscr{B}(v, q)$. As a consequence of Theorem 1, the resulting approximate value function will still be a lower bound of the true value function.

For each given point $\mathbf{b}$, we compute the associated value function and the optimizing α-vector using the backup operation $\mathrm{backup}(\mathbf{b}, v, q, \Gamma)$ for $q \in Q \setminus Q_f$ as given in Algorithm 1. Given a set of belief points $\mathscr{B}(q, v)$ and a set of alpha vectors Γ, this yields the operator denoted as $\mathsf{B}_{\mathscr{B}}$ given as

$$\mathsf{B}_{\mathscr{B}}(\Gamma)(q, v) := \{\mathrm{backup}(\mathbf{b}, v, q, \Gamma) | \mathbf{b} \in \mathscr{B}(q, v)\}. \tag{17}$$

The associated policy can be computed together with the alpha vectors. Thus, we have introduced back-ups typically

Algorithm 1: $\alpha = \text{backup}(\mathbf{b}, v, q, \Gamma)$

$\alpha^{(v',u),o} \leftarrow \arg\max_{\alpha \in \Gamma(q'',v')} \alpha^{(v',u),o} \cdot \mathbf{b}$;
$\alpha^{(v',u)} \leftarrow \sum_o \alpha^{(v',u),o}$;
$\alpha^{v'} \leftarrow \arg\max \alpha^{(v',u)} \cdot \mathbf{b}$;
$\alpha \leftarrow p_{v,v'}\left(\arg \max_{\alpha^{v'}, v \xrightarrow{e} v'} p_{v,v'}\left(\alpha^{v'} \cdot \mathbf{b}\right)\right)$

used for POMDP problems [18], [21] and we have adapted them to solve a probabilistic reachability over the PRM and the specification DFA. Iterative improvement to the probabilistic roadmaps can be used in combination with iterative expansion of the set of belief points to increase the computational precision.

Pruning vertices $v \in V$ that do not influence the value iterations reduces the computational burden of the algorithm. Elements of $\mathbb{V}$ and $\mathscr{E}$ can only be pruned if they do not affect the soundness of the value function. That is, we require that the updated roadmap $G' = (\mathbb{V}', \mathscr{E}')$ is such that for any $v \in \mathbb{V}$

$$\mathbf{V}(q, v, \mathbf{b}) = \max_{\alpha \in \Gamma(q,v)} \alpha \cdot \mathbf{b}$$

stills represent a lower-bound on the true probabilistic reachability over the pruned $G' = (\mathbb{V}', \mathscr{E}')$ and that the initial location of the rover v_0 is still in $\mathbb{V}$. This implies that we preserve all vertices that can be reached from v_0 via transitions that belong to the maximizing policy. In this paper, we use a basic grid-based uniform sampling algorithm [9] that excludes samples in known obstacles. More advanced sampling methods can either be used to create a better uniform distribution or to increase samples in more difficult regions. The belief point set $\mathscr{B}(v,q) := \{\mathbf{b}_i\}$ is built based on the reachable belief points. Reachable points are iteratively added based on their distance to $\mathscr{B}(v,q)$ as explained in [18].

VI. Case Study

In this section, we introduce a case study and apply the point-based value iteration over an scLTL-PRM to solve it. After introducing models for the rover, the uncertain environment, and measurements, we showcase the use of point-based value iteration for scLTL planning.

Abstract robot models. Many types of low-level controllers are available to control the robot over a roadmap. For clarity, we use a simple linear model that captures the essential dynamics of the rover that moves around in a workspace. That is, the rover is modeled as $x^+ = x + u + w$, with $x \in \mathbb{R}^2$. The state takes values in the $[-5,5] \times [-5,5]$ configuration space of the robot. Here we restrict attention to *risk regions* that may contain obstacles that the rover can not traverse, and *target regions* that are likely places where scientific samples can be extracted. These uncertain regions are denoted as A_k the potential target regions with $k = \{1,2\}$ and R_k the uncertain risk regions with $k := \{3,4,5\}$ as in Fig. 5. Additionally, the configuration space includes regions that clearly contain obstacles. These regions are denoted as R or depicted with the full brick pattern in Fig. 5. The belief that an obstacle or, respectively, a sample is present in an uncertain region is modeled with the environment model and inferred through measurements as detailed next.

Environment and measurement model. After choosing with $u_k \in \mathbb{U}_e$ to observe the k-th region, a true or false measurement is obtained, denoted respectively as t_k and f_k. The probability of these samples is given as

$$\mathbb{P}(t_k) = \begin{cases} 1 - P_{\text{falserate}}(x,k) & \text{if } x_e(k) = 1 \\ P_{\text{falserate}}(x,k) & \text{if } x_e(k) = 0 \end{cases} \tag{18}$$

and $\mathbb{P}(f_k) := 1 - \mathbb{P}(t_k)$ and $\mathbb{P}(f_j), \mathbb{P}(t_j) = 0$ for all $j \neq k$. This yields the set of observations $\mathbb{O} := \bigcup_{k=1}^{n_e} \{t_k, f_k\}$. The false rate of the measurement $P_{\text{falserate}}(x)$ is a function of the distance of the rover to the region, that is

$$P_{\text{falserate}}(x,k) := \begin{cases} 0 & x \in \mathscr{P}_k, \text{ else} \\ 0.2 & d(x,\mathscr{P}_k) \leq 1, \\ 0.1 + 0.1 d(x,\mathscr{P}_k) & 1 \leq d(x,\mathscr{P}_k) \leq 4, \\ 0.5 & d(x,\mathscr{P}_k) \geq 4. \end{cases}$$

where $d(x,\mathscr{P}_k)$ is the distance betweem the rover and the k-th region.

Specification. The rover needs to *collect a sample* and avoid *obstacles*. This former part is specified as being in uncertain sample/target regions (A_i) and finding a sample in it (t_i), i.e., $\phi_{target} := \Diamond(A_1 \wedge t_1) \vee \Diamond(A_2 \wedge t_2)$. Additionally, the rover should not enter a region with obstacles. For the uncertain regions this is equivalent to finding an obstacle while being in that region. This type of failure is expressed as $\phi_{unsafe} := R \vee \bigvee_{k=3}^{5}(R_k \wedge t_k)$ with R the proposition for the known obstacle regions and R_k the uncertain obstacle regions. The overall mission specification can hence be expressed as

$$\psi := \neg\phi_{unsafe} \;\mathscr{U}\; \phi_{target}. \tag{19}$$

Results. For this case study the objective is to compute a strategy that maximizes the guaranteed probability $\mathbb{P}(\pi \vDash \psi)$. In Fig. 5, the initial vertices of the roadmap are given

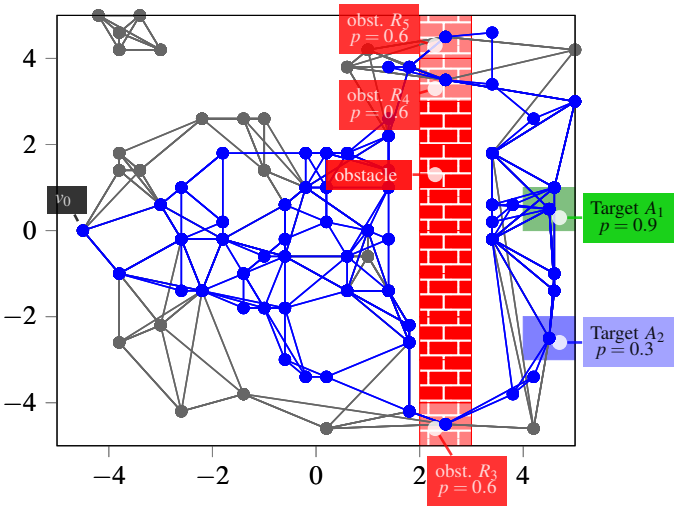

Fig. 5. The initial roadmap from which the computations start is given in grey, whereas the the roadmap after pruning and resampling 4 times with 100 belief points in each vertex is given in blue.

together with the initial beliefs in the uncertain regions. In

each of the vertices, the belief point set $\mathscr{B}(v,q) := \{\mathbf{b}_i\}$ contains 100 belief points. In between the value iterations, the roadmap is pruned and resampled 4 times. After 4 times increasing the number of pruning and resampling times did not increase the accuracy much more. This yields the more optimized roadmap given in blue Fig. 5. In this roadmap, nodes that lead toward the regions of interest have been automatically preserved.

After these computations, the synthesized strategy yields a satisfaction probability of at least 0.61. In Fig. 6, a single execution of this policy is given. The rover moves over the nodes of the roadmap and observes the different regions. As shown in the figure, the true and false measurements of a region (up/down sign) impact the satisfaction probability. The bottom plot depicts the evolution of the beliefs for each of the individual regions.

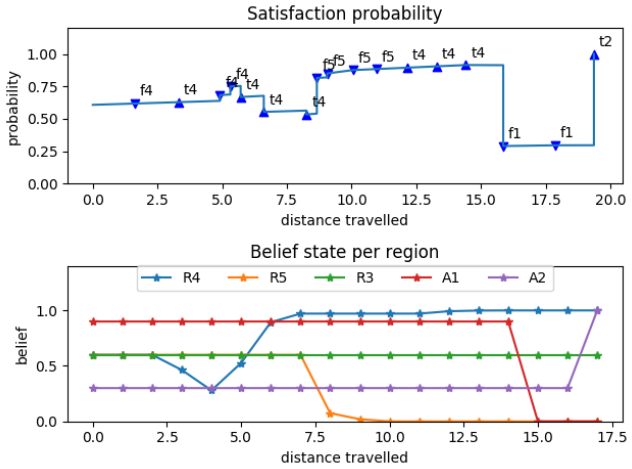

Fig. 6. The distance travelled versus the current mission probability are given in the top figure and the beliefs in the uncertain regions are given in the bottom plot.

VII. Conclusions

We have shown that it is possible to combine the benefits of probabilistic roadmaps and point-based value iterations to solve planning problems in uncertain environments. The given method is sound as it gives a lower bound on the satisfaction probability.

Acknowledgment

A part of this research was carried out at the Jet Propulsion Lab (JPL) and the California Institute of Technology (Caltech) under a contract with the NASA and funded through the President's and Director's Fund 105275 -17AW0044.

References

[1] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems. *Automatica*, 44(11):2724–2734, 2008.

[2] A. Agha-mohammadi, S. Chakravorty, and N. Amato. FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *International Journal of Robotics Research (IJRR)*, 33(2):268–304, 2014.

[3] K. Astrom. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

[4] C. Belta, B. Yordanov, and E. A. Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Springer, 2017.

[5] A. Bhatia, L. Kavraki, and M. Vardi. Sampling-based motion planning with temporal goals. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 2689–2696. IEEE, 2010.

[6] A. Bhatia, M. R. Maly, L. E. Kavraki, and M. Y. Vardi. Motion planning with complex goals. *IEEE Robotics & Automation Magazine*, 18(3):55–64, 2011.

[7] L. I. R. Castro, P. Chaudhari, J. Tumova, S. Karaman, E. Frazzoli, and D. Rus. Incremental sampling-based algorithm for minimum-violation motion planning. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3217–3224. IEEE, 2013.

[8] R. Geraerts and M. H. Overmars. A comparative study of probabilistic roadmap planners. In *Algorithmic Foundations of Robotics V*, pages 43–57. Springer, 2004.

[9] R. J. Geraerts and M. H. Overmars. Sampling techniques for probabilistic roadmap planners. *Intelligent Autonomous Systems 8*, page 600, 2004.

[10] S. Haesaert, S. E. Zadeh Soudjani, and A. Abate. Verification of general Markov decision processes by approximate similarity relations and policy refinement. *SIAM Journal on Control and Optimization*, 55(4):2333–2367, 2017.

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

[12] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008.

[13] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Where's Waldo? Sensor-based temporal logic motion planning. In *IEEE International Conference on Robotics and Automation*, pages 3116–3121, 2007.

[14] O. Kupferman and M. Y. Vardi. Model checking of safety properties. *Formal Methods in System Design*, 19(3):291–314, 2001.

[15] K. Lesser and M. Oishi. Reachability for partially observable discrete time stochastic hybrid systems. *Automatica*, 50(8):1989–1998, 2014.

[16] K. Lesser and M. Oishi. Finite State Approximation for Verification of Partially Observable Stochastic Hybrid Systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 159–168, New York, NY, USA, 2015. ACM.

[17] G. Norman, D. Parker, and X. Zou. Verification and control of partially observable probabilistic systems. *Real-Time Systems*, 53(3):354–402, 2017.

[18] J. Pineau, G. Gordon, S. Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003.

[19] E. Plaku. Path planning with probabilistic roadmaps and co-safe linear temporal logic. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2269–2275. IEEE, 2012.

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

[21] T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 520–527. AUAI Press, 2004.

[22] T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. *arXiv preprint arXiv:1207.1412*, 2012.

[23] S. E. Z. Soudjani, C. Gevaerts, and A. Abate. Faust 2: Formal abstractions of uncountable-state stochastic processes. In *TACAS*, volume 15, pages 272–286, 2015.

[24] V. Varricchio, P. Chaudhari, and E. Frazzoli. Sampling-based algorithms for optimal motion planning using process algebra specifications. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5326–5332. IEEE, 2014.

[25] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta. Control in belief space with Temporal Logic specifications. In *Proc. IEEE CDC*, pages 7419–7424. IEEE, 2016.

[26] T. Wongpiromsarn, U. Topcu, and R. M. Murray. Receding Horizon Temporal Logic Planning for Dynamical Systems. In *Conference on Decision and Control (CDC) 2009*, pages 5997 –6004, 2009.

[27] M. Zamani, I. Tkachev, and A. Abate. Towards scalable synthesis of stochastic control systems. *Discrete Event Dynamic Systems*, 27(2):341–369, June 2017.