Optimal Temporal Logic Planning in Probabilistic Semantic Maps

Jie Fu, Nikolay Atanasov, Ufuk Topcu, and George J. Pappas

Abstract—**This paper considers robot motion planning under temporal logic constraints in probabilistic maps obtained by semantic simultaneous localization and mapping (SLAM). The uncertainty in a map distribution presents a great challenge for obtaining correctness guarantees with respect to the linear temporal logic (LTL) specification. We show that the problem can be formulated as an optimal control problem in which both the semantic map and the logic formula evaluation are stochastic. Our first contribution is to reduce the stochastic control problem for a subclass of LTL to a deterministic shortest path problem by introducing a confidence parameter δ. A robot trajectory obtained from the deterministic problem is guaranteed to have minimum cost and to satisfy the logic specification in the true environment with probability δ. Our second contribution is to design an admissible heuristic function that guides the planning in the deterministic problem towards satisfying the temporal logic specification. This allows us to obtain an optimal and very efficient solution using the A* algorithm. The performance and correctness of our approach are demonstrated in a simulated semantic environment using a differential-drive robot.**

I. Introduction

This paper addresses robot motion planning in uncertain environments with tasks specified by linear temporal logic (LTL) co-safe formulas. A map distribution, obtained from a semantic simultaneous localization and mapping (SLAM) algorithm [25, 1, 30, 3], facilitates natural robot task specifications in terms of objects and landmarks in the environment. For example, we can require a robot to "go to a room where there is a desk and two chairs" instead of giving it exact target coordinates. One could even describe tasks when the entire map is not available but is to be obtained as the robot explores its environment. Meanwhile, temporal logic allows one to specify rich, high-level robotic tasks. Hence, a meaningful question we aim to answer is the following. *Given a semantic map distribution, how does one design a control policy that enables the robot to efficiently accomplish temporal logic tasks with high probability, despite the uncertainty in the true environment?*

This question is motivated by two distinct lines of work, namely, control under temporal logic constraints and multi-task SLAM. Control synthesis with temporal logic specifications has been studied for both deterministic [15, 13, 4] and stochastic systems [19, 7]. Recent related work focuses on the design problem in the presence of unknown and uncertain environments. In general, three types of uncertainty are considered: sensor uncertainty [12], incomplete environment models [16, 11, 22, 23], or uncertainty in the robot dynamics [31, 9]. Johnson and Kress-Gazit [12] employ a model checking algorithm to evaluate the fragility of the control design with respect to temporal logic tasks when sensing is uncertain. To handle unexpected changes in the environment and incompleteness in the environment model, Kress-Gazit et al. [16] develop a sensor-based reactive motion planning method that guarantees the correctness of the robot behaviors under temporal logic constraints. Livingston et al. [22, 23] propose a way to efficiently modify a nominal controller through local patches for assume-guarantee LTL formulas. Guo et al. [11] develop a revision method for online planning in a gradually discovered environment. Probabilistic uncertainty is studied in [31, 9]. Wolff et al. [31] develop a robust control method with respect to temporal logic constraints in a stochastic environment modeled as an interval Markov decision process (MDP). Fu and Topcu [9] develop a method that learns a near-optimal policy for temporal logic constraints in an initially unknown stochastic environment. Generally, existing work abstracts the system and its environment into discrete models, such as, MDPs and two-player games, and plans in the discrete state space. In contrast, this work considers the new problem of control design within a probabilistic map obtained via a semantic SLAM algorithm. In this setting, the uncertainty in the landmark poses is modeled via a continuous distribution, while the uncertainty in their classes - via a discrete distribution.

While temporal logic can be used to specify a wide range of robot behaviors, recent advances in SLAM motivate the integration of task planning with simultaneous discovery of an unknown environment. Multi-task SLAM is proposed in Guez and Pineau [10]. The authors consider a planning problem in which a mobile robot needs to map an unknown environment, while localizing itself and maximizing long-term rewards. While the problem is formulated as a partially observable Markov decision process, planning is carried out using the mean of the robot pose and the mean of the map distribution. Bachrach et al. [2] develop a system for visual odometry and mapping using an RGB-D camera. The authors employ the Belief Roadmap algorithm [24] to generate the shortest path from the mean robot pose to a goal state, while propagating uncertainties along the path. It is difficult, however, to extend these approaches to temporal logic planning

J. Fu is with the Department of Electrical & Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01604, USA, jfu2@wpi.edu.

N. Atanasov, and G. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA, {atanasov, pappasg}@seas.upenn.edu.

U. Topcu is with the Department of Aerospace Engineering and Engineering Mechanics, University of Texas, Austin, TX 78712, USA, utopcu@utexas.edu.

This work is supported by AFRL # FA8650-15-C-2546, ONR # N000141310778, NSF # 1550212, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

with probabilistic semantic maps. Unlike reachability and reward maximization, the performance criteria induced by LTL formulas require a rigorous way to reason about the uncertainty in the map distribution. To satisfy quantitative temporal logic specifications within a probabilistic semantic map, our method brings together the notions of robustness and probabilistic correctness. This work makes the following **contributions**:

- A stochastic optimal control problem for planning robot motion in a probabilistic semantic map under temporal logic constraints is formulated.
- For a subclass of LTL, the stochastic problem is reduced to a deterministic shortest path problem that can be solved efficiently. We prove that for a given confidence parameter δ, the robot trajectory obtained from the deterministic problem, if it exists, satisfies the logic specification with probability δ in the true environment.
- An admissible heuristic is designed in order to compute the optimal solution of the deterministic problem efficiently via an A* planning algorithm.

II. Problem Formulation

In this section, we introduce models for the robot and its uncertain environment, represented by a semantic map distribution. Using temporal logic as the task specification language, we formulate a stochastic optimal control problem.

A. Robot and environment models

Consider a mobile robot whose dynamics are governed by the following discrete-time motion model:

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

where $x_t = (x_t^p, x_t^a) \in X$ is the robot state at time t, containing its pose x_t^p and other variables x_t^a such as velocity and acceleration and $u_t \in U$ is the control input, selected from a *finite* space of admissible controls. A trajectory of the robot, for $t \in \mathbb{N} \cup \{\infty\}$, is a sequence of states $x_{0:t} := x_0 x_1 \ldots x_t$.

The robot operates in an environment modeled by a semantic map $\mathcal{M} := \{l_1, \ldots, l_M\}$ consisting of M landmarks. Each landmark $l_i := (l_i^p, l_i^c) \in \mathcal{M}$ is defined by its pose l_i^p and class $l_i^c \in \mathcal{C}$, where $\mathcal{C}$ is a finite set of classes (e.g., table, chair, door, etc.). The robot does not know the true landmark poses but has access to a probability distribution $\mathcal{P}$ over the space of all possible maps. Such a distribution can be produced by a semantic SLAM algorithm [30, 3] and typically consists of a Gaussian distribution over the landmark poses and a discrete distribution over the landmark classes. More precisely, we assume $\mathcal{P}$ is determined by parameters $(\bar{l}^p, \Sigma^p, \{\rho_i^c\}_{i=1}^M)$ such that $l^p \sim \mathcal{N}(\bar{l}^p, \Sigma^p)$ and l_i^c is generated by the probability mass function ρ_i^c. We suppose that the class of each landmark is known and leave the case of uncertain landmark classes for future work.

B. Temporal logic specifications

We use linear temporal logic (LTL) to specify the robot's task in the environment. LTL formulas [29] can describe temporal ordering of events along the robot trajectories and are defined by the following grammar: $\phi := p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2$, where $p \in \mathcal{AP}$ is an atomic proposition and $\bigcirc$ and $\mathcal{U}$ are temporal modal operators for "next" and "until". Additional temporal logic operators are derived from basic ones: $\Diamond\varphi := \texttt{true}\,\mathcal{U}\varphi$ (eventually) and $\Box\varphi := \neg\Diamond\neg\varphi$ (always). We assume that the robot's task is given by an LTL co-safe formula [17], which allows checking its satisfaction using a finite-length robot trajectory.

The LTL formula is specified over a finite set of atomic propositions that are defined over the robot state space X and the environment map $\mathcal{M}$. Examples of atomic propositions include:

$$\begin{aligned} &\alpha_i^p(r) : d(x^p, l_i^p) \le r && \text{for } r \in \mathbb{R}, i \in \{1, \ldots, M\}, \\ &\alpha_i^c(Y) : l_i^c \in Y && \text{for } Y \subseteq \mathcal{C}, i \in \{1, \ldots, M\}. \end{aligned} \tag{2}$$

Proposition $\alpha_i^p(r)$ evaluates true when the robot is within r units distance from landmark i, while proposition $\alpha_i^c(Y)$ evaluates true when the class of the i-th landmark is in the subset Y of classes. In order to interpret an LTL formula over the trajectories of the robot system, we use a labeling function that determines which atomic propositions hold true for the current robot pose.

Definition 1 (Labeling function[1]). *Let $\mathcal{AP}$ be a set of atomic propositions and M be the set of all possible maps. A labeling function $L : X \times \mathsf{M} \to 2^{\mathcal{AP}}$ maps a given robot state $x \in X$ and map $\mathcal{M} \in \mathsf{M}$ to a set $L(x, \mathcal{M})$ of atomic propositions that evaluate true.*

For robot trajectory $x_{0:t}$ and map $\mathcal{M} \in \mathsf{M}$, the *label sequence of* $x_{0:t}$ in $\mathcal{M}$, denoted $L(x_{0:t}, \mathcal{M})$, is such that $L(x_{0:t}, \mathcal{M}) = L(x_0, \mathcal{M}) L(x_1, \mathcal{M}) L(x_2, \mathcal{M}) \ldots L(x_t, \mathcal{M})$. Given an LTL co-safe formula φ, one can construct a deterministic finite-state automaton (DFA) $\mathcal{A}_\varphi = (Q, 2^{\mathcal{AP}}, T, q_0, F)$ where $Q, 2^{\mathcal{AP}}, q_0, F$ are a finite set of states, the alphabet, the initial state, and a set of final states, respectively. $T : Q \times 2^{\mathcal{AP}} \to Q$ is a transition function such that $T(q, a)$ is the state that is reached with input a at state q. We extend the transition function in the usual way[2]: $T(q, uv) = T(T(q, u), v)$ for $u, v \in (2^{\mathcal{AP}})^*$. A word w is *accepted* in $\mathcal{A}_\varphi$ if and only if $T(q_0, w) \in F$. The set of words accepted by $\mathcal{A}_\varphi$ is the *language* of $\mathcal{A}_\varphi$, denoted $\mathcal{L}(\mathcal{A}_\varphi)$.

We say that a robot trajectory $x_{0:\infty}$ satisfies the LTL formula φ in the map $\mathcal{M}$ if and only if there is $k \ge 0$ such that $L(x_{0:k}, \mathcal{M}) \in \mathcal{L}(\mathcal{A}_\varphi)$. Then, $x_{0:k}$ is called a *good prefix* for the formula φ. Furthermore, $\mathcal{A}_\varphi$ accepts exactly the set of good prefixes for φ and for any state $q \in F$, it holds that $T(q, a) \in F$ for any $a \in 2^{\mathcal{AP}}$.

We are finally ready for a formal problem statement.

Problem 1. *Given an initial robot state $x_0 \in X$, a semantic map distribution $\mathcal{P}$, and an LTL co-safe formula φ*

[1] For a deterministic map, our labeling function definition reduces to the commonly used one in robotic motion planning under temporal logic constraints [7].

[2] Notation: Let A be a finite set. Let A^*, A^ω be the set of finite and infinite words over A. Let $\lambda = A^0$ be the empty string. Abusing notation, we use $\emptyset$ and λ interchangeably. For $w \in A^\omega$, if there exist $u \in A^*$ and $v \in A^\omega$ such that $w = uv$ then u is a *prefix* of w and v is a *suffix* of w.

represented by a DFA $\mathcal{A}_\varphi$, choose a stopping time τ and a sequence of control policies $\mu_t \in U$ for $t = 0, 1, \ldots, \tau$ that maximize the probability of the robot satisfying φ in the true environment $\mathcal{M}$ while minimizing its motion cost:

$$\begin{aligned}
\min_{\tau,\mu_0,\mu_1,\ldots,\mu_\tau} \quad & \mathbb{E}\big[\sum_{t=0}^{\tau} c(x_t, x_{t+1})\big] + \kappa \mathbb{P}(q_{\tau+1} \notin F) \\
\textit{s.t.} \quad & x_{t+1} = f(x_t, \mu_t(x_t, q_t)), \\
& q_{t+1} = T(q_t, L(x_{t+1}, \mathcal{M})),\ \forall\, 0 \le t < \tau,
\end{aligned}$$

where $\mathbb{P}(q_t \notin F)$ is the probability (induced by $\mathcal{P}$) that the automaton state q at time t does not belong to the set F, c is a positive-definite motion cost function that satisfies the triangle inequality, and $\kappa \ge 0$ determines the relative importance of satisfying the specification versus the total motion cost.

Remark: The optimal cost of Problem 1 is bounded below by 0 due to the assumptions on c, κ and above by κ, obtained by stopping immediately ($\tau = 0$) without satisfying φ.

III. Planning to be Probably Correct

The map uncertainty in Problem 1 leads to uncertainty in the evaluation of the atomic propositions and hence to uncertainty in the robot trajectory labeling. In turn, the automaton state is unobservable. Rather than solving the resulting optimal control problem with partial observability, we propose an alternative solution that generates a near-optimal plan with a probabilistic correctness guarantee for the temporal logic constraints. The main idea is to convert the original semantic map distribution to a high-confidence deterministic representation and solve a deterministic optimal control problem with this new representation. The advantage is that we can solve the deterministic problem very efficiently and still provide a correctness guarantee. This avoids the need for sampling-based methods in the continuous space of map distributions, which become computationally expensive when planning in large environments. To this end, we use a confidence region around the mean $\bar{\mathcal{M}} := \{(\bar{l}_i^p, l_i^c)\}_{i=1}^M$ of the semantic map distribution $\mathcal{P}$ to extend the definition of the labeling function.

Definition 2 (δ-Confident labeling function)**.** *Given a robot state $x \in X$, a map distribution $\mathcal{P}$, and a parameter $\delta \in (0, 1)$, a δ-confident labeling function is defined as follows:*

$$L^\delta(x, \mathcal{P}) := \begin{cases} L(x, \bar{\mathcal{M}}) & \textit{if } L(x, \bar{\mathcal{M}}) = L(x, m) \textit{ for all maps} \\ & m \textit{ in the } \delta\textit{-confidence region of } \mathcal{P}, \\ \emptyset & \textit{otherwise}, \end{cases}$$

where the δ-confidence region of $\mathcal{P}$ corresponds to the δ-confidence region of the joint landmark pose distribution specified by $\mathcal{P}$.

We now explain the intuition for defining the δ-confident labeling function as in Def. 2. For a given robot trajectory $x_{0:t}$, rather than maintaining a distribution over the possible label sequences, the robot keeps only a sequence of labels that, with probability δ, is a subsequence[3] of the label sequence $L(x_{0:t}, \mathcal{M})$ in the true environment. This statement is made precise in the following proposition.

Proposition 1. *Given a robot trajectory $x_{0:t}$ and a map distribution $\mathcal{P}$, $L^\delta(x_{0:t}, \mathcal{P})$ is a subsequence of $L(x_{0:t}, \mathcal{M})$ with probability δ.*

Proof. See Appendix A $\square$

Intuitively, a label $L(x_k, \mathcal{M})$ is preserved at the k-th position of $L^\delta(x_{0:t}, \mathcal{P})$ if for any two sample maps m, m' in the δ-confidence region of $\mathcal{P}$, $L(x_i, m) = L(x_i, m')$. Otherwise, it is replaced by $\emptyset$. Next, we show that when the LTL formula φ satisfies a particular property, if $L^\delta(x_{0,t}, \mathcal{P})$ is accepted by the DFA $\mathcal{A}_\varphi$, then with probability δ, $x_{0:t}$ satisfies the LTL specification φ. The required property is that the formal language characterization of the logic formula translates to a *simple polynomial* [27]. An ω-regular language $\mathcal{L}$ over an alphabet A is *simple monomial* if and only if it is of the form

$$A^* a_1 A^* a_2 A^* \ldots A^* a_k A^* (A^* b_1 A^* b_2 A^* \ldots A^* b_\ell A^*)^\omega$$

where $a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_\ell \in A$, $k \ge 0$, and $\ell \ge 0$. A finite union of simple monomials is called a *simple polynomial*.

Theorem 1. *If the the language $\mathcal{L}(\mathcal{A}_\varphi)$ of $\mathcal{A}_\varphi$ is a simple polynomial, then $L^\delta(x_{0:\tau}, \mathcal{P}) \in \mathcal{L}(\mathcal{A}_\varphi)$ implies that $\mathbb{P}(L(x_{0:\tau}, \mathcal{M}) \in \mathcal{L}(\mathcal{A}_\varphi)) = \delta$.*

Proof. See Appendix B. $\square$

The significance of Thm. 1 is that it allows us to reduce the stochastic control problem with an uncertain map (Problem 1) to a deterministic shortest path problem. We introduce the following product system to facilitate the conversion.

Definition 3 (δ-Probably correct product system)**.** *Given the robot system in* (1)*, the map distribution $\mathcal{P}$, the automaton $\mathcal{A}_\varphi$, and a parameter $\delta \in (0, 1)$, a* δ-probably correct product system *is a tuple $\mathcal{G}^\delta = \langle S, U, \Delta, s_0, S_F \rangle$ defined as follows.*

- *$S = X \times Q$ is the product state space.*
- *$\Delta : S \times U \to S$ is a transition function such that $\Delta((x, q), u) = (x', q')$ where $x' = f(x, u)$ and $q' = T(q, L^\delta(x', \mathcal{P}))$. It is assumed that $T(q, \emptyset) = q$, $\forall q \in Q$.*
- *$s_0 = (x_0, q_0)$ is the initial state.*
- *$S_F = X \times F$ is the set of final states.*

For the subclass of LTL co-safe formulas whose languages are simple polynomials, Thm. 1 guarantees that the projection on X of any trajectory $s_{0:t}$ of $\mathcal{G}^\delta$ that reaches S_F in the δ-confidence region of $\mathcal{P}$ has probability δ of satisfying the specification in the true map. The implications are explored in Sec. IV.

Before we proceed, however, it is important to know to what extent the expressiveness of LTL is limited by restricting

[3] For a word $u \in A^\omega$, u is a *subsequence* of w if u can be obtained from w by replacing symbols with the empty string λ.

it to the subclass of simple polynomials. In Appendix C, we show that such LTL formulas can express reachability and sequencing properties. Moreover, with a slight modification of Def. 3, we can also ensure the correctness of plans with respect to safety constraints.

Consider safety constraints in the following form $\square\phi_{safe}$ with ϕ_{safe} being a propositional logic formula over $\mathcal{AP}$. For example, an obstacle avoidance requirement is given by $\square(d(x, x_o) \geq r)$ where x_o are the coordinates of an obstacle. When the LTL formula includes such safety constraints, we need to modify the transition function in Def. 3 in the following way. For any state $s \in S$ and any input $u \in U$, let $s' = (x', q') = \Delta(s, u)$. Then, if there exists at least one m in the δ-confidence region of $\mathcal{P}$ such that the propositional logic formula corresponding to $L(x', m)$ implies[4] $\neg\phi_{safe}$, let $\Delta(s, u) = \mathsf{sink}$, where sink is a non-accepting sink state that satisfies $\Delta(\mathsf{sink}, u) = \mathsf{sink}$ for any $u \in U$. Thus, the state sink will not be visited by any trajectory of $\mathcal{G}^\delta$ that reaches S_F, which means the safety constraint will be satisfied with probability δ in the true environment. The following toy example illustrates the concepts.

Example 1. *In Fig. 1, a mobile robot is tasked with visiting at least one landmark in an uncertain environment. Formally, the LTL specification of the task is $\varphi := \lozenge p$ where $p := \vee_i (d(x, l_i^p) \leq 1)$. Given the robot trajectory $x_{0:t}$ represented by the dashed line in the figure, when the robot traverses the 95%-confidence region of l_1's pose distribution, it cannot confidently (with confidence level $\delta = 0.95$) decide the value of p in the true map because for some map realizations, $d(x, l_1^p) > 1$. On the other hand, when the robot is near l_2, p evaluates true because a unit ball around the robot covers the entire 95%-confidence region of l_2's pose distribution. Let $\mathcal{B}_r(l)$ a ball centered at l with radius r. The label sequence of $x_{0:t}$ in the true environment is $L(x_{0:t}, \mathcal{M}) = \emptyset^{k_1}\{p\}^{k_2}\emptyset^{k_3}\{p\} = \{p\}^{k_2+1}$ where k_1, k_2, k_3 are the numbers of steps before reaching $\mathcal{B}_1(l_1^p)$, in $\mathcal{B}_1(l_1^p)$, and after leaving $\mathcal{B}_1(l_1^p)$ but before reaching $\mathcal{B}_2(l_2^p)$. The label sequence $L^{0.95}(x_{0:t}, \mathcal{P}) = \emptyset^{k_1+k_2+k_3}\{p\} = \{p\}$. Clearly, $L^{0.95}(x_{0:t}, \mathcal{P})$ is a subsequence of $L(x_{0:t}, \mathcal{M})$. Moreover, the trajectory satisfies the LTL specification which is a reachability constraint.*

In the case of a safety constraint, e.g., $\varphi_{safe} = \square p$ where $p := d(x, l_i^p) > 1$ for any landmark i, once the robot gets close to l_1 the δ-probably correct product system will transition to the non-accepting sink because there exists a sample map m such that the safety constraint is violated. Thus, in any run that is safe in $\mathcal{G}^\delta$, the robot is able to safely avoid both l_1 and l_2 with probability δ.

IV. Reduction to Deterministic Shortest Path

For a fixed confidence δ, due to Thm. 1, we can convert Problem 1 to a deterministic shortest path problem within the probably correct product system $\mathcal{G}^\delta$. In this case, $\mathbb{P}(q_{\tau+1} \notin F) \in \{0, 1\}$ and the optimal solution to Problem 1 is either

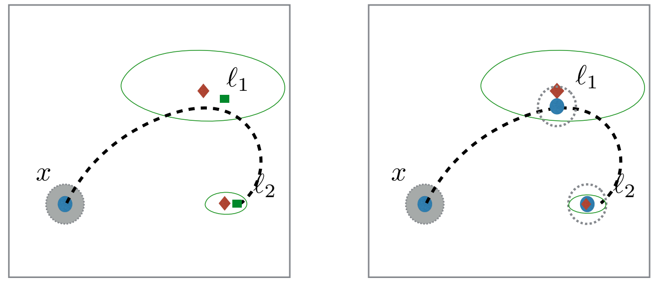


Fig. 1: Example: robot with a reachability objective. The red diamonds are the means of the landmarks' pose distributions and the ellipsoids are the associated 95%-confidence regions. The robot is represented by the blue dot. The green squares represent the true locations of landmarks l_1 and l_2.

to stop immediately ($\tau = 0$), incurring cost κ, or to find a robot trajectory satisfying φ with cumulative motion cost less than $\kappa\delta$. The latter corresponds to the following deterministic problem.

Problem 2. *Given an initial robot state $x_0 \in X$, a semantic map distribution $\mathcal{P}$, a confidence $\delta \in (0, 1)$, and an LTL co-safe formula φ represented by a DFA $\mathcal{A}_\varphi$, choose a stopping time τ and a sequence of control inputs $u_t \in U$ for $t = 0, 1, \ldots$ that minimize the motion cost of a trajectory that satisfies φ:*

$$\begin{aligned}
\min_{u_0, u_1, \ldots, u_\tau} \quad & \sum_{t=0}^{\tau} c(x_t, x_{t+1}) \\
\text{s.t.} \quad & x_{t+1} = f(x_t, u_t), \\
& q_{t+1} = T(q_t, L^\delta(x_t, \mathcal{P})),\ \forall\, 0 \leq t < \tau, \\
& q_{\tau+1} \in F, \\
& \sum_{t=0}^{\tau} c(x_t, x_{t+1}) \leq \kappa\delta.
\end{aligned}$$

If Problem 2 is infeasible, it is best in Problem 1 to stop immediately ($\tau = 0$), incurring cost κ; otherwise, the robot should follow the control sequence $u^*_{0:\tau}$ computed above and the corresponding trajectory $x^*_{0:\tau+1}$ to incur cost:

$$\sum_{t=0}^{\tau} c(x^*_t, x^*_{t+1}) + \kappa(1 - \delta) \leq \kappa$$

in the original Problem 1. Since Problem 2 is a deterministic shortest path problem, we can use any of the traditional motion planning algorithms, such as RRT [20], RRT* [14, 13] or A* [21] to solve it. We choose A* due to its completeness guarantees[5] [26] and because the automaton $\mathcal{A}_\varphi$ can be used to guide the search as we show next.

A. Admissible Heuristic

The efficiency of A* can be increased dramatically by designing an appropriate heuristic function to guide the search. Given a state $s := (x, q)$ in the product system (Def. 3), a heuristic function $h : S \to \mathbb{R}$ provides an estimate

[4]Given a label $L(x', m) \subseteq \mathcal{AP}$, the corresponding propositional logic formula is $\bigwedge_{\alpha_i \in L(x', m)} \alpha_i \wedge \bigwedge_{\alpha_j \in \mathcal{AP} \setminus L(x', m)} \neg\alpha_j$.

[5]To guarantee completeness of A* for Problem 2, the robot state space X needs to be assumed bounded and compact and needs to discretized.

of the optimal cost $h^*(s)$ from s to the goal set S_F. If the heuristic function is *admissible*, i.e., never overestimates the cost-to-go ($h(s) \leq h^*(s),\ \forall s \in S$), then A* is optimal [26].

Lacerda et al. [18] propose a distance metric to evaluate the progression of an automaton state with respect to an LTL co-safe formula. We use a similar idea to design an admissible heuristic function. We partition the state space Q of $\mathcal{A}_\varphi$ into level sets as follows. Let $Q_0 := F$ and for $i \geq 0$ construct $Q_{i+1} := \{q \in Q \setminus \bigcup_{k=0}^{i} Q_k \mid \exists q' \in Q_i, a \in 2^{\mathcal{AP}},$ such that $T(q,a) = q'\}$. The generation of level sets stops when $Q_i = \emptyset$ for some i. Further, we denote the set of all sink states by Q_∞. Thus, given $q \in Q$ one can find a unique level set Q_i such that $q \in Q_i$. We say that i is the level of q and denote it by $\mathsf{Level}(q) = i$.

Proposition 2. *Let $s_{0:t}$ be a trajectory of the product system $\mathcal{G}^\delta$ that reaches S_F, i.e., $s_t \in S_F$. Then, for any $0 \leq k < t$, given $s_k = (x_k, q_k)$ and $s_{k+1} = (x_{k+1}, q_{k+1})$, it holds that $\mathsf{Level}(q_k) \leq \mathsf{Level}(q_{k+1}) + 1$.*

Proof. Since $T(q_k, L^\delta(x_{k+1}, \mathcal{P})) = q_{k+1}$, if $q_{k+1} \in Q_i$ for some level i, then, by construction of the level sets, either $q_k \in Q_{i+1}$ or $q_k \in \bigcup_{j=0}^{i} Q_j$. □

By construction of the level sets, the automaton states $q_{0:t}$, associated with any trajectory $s_{0:t}$ of the product system that reaches a goal state ($s_t \in S_F$), have to pass through the level sets sequentially. In other words, if $\mathsf{Level}(q_0) = i$, then there exists a subsequence $q'_{0:i}$ of $q_{0:t}$ such that $\mathsf{Level}(q'_1) = i-1$, $\mathsf{Level}(q'_2) = i-2, \ldots, \mathsf{Level}(q'_i) = 0$. Thus, we can construct a heuristic function that underestimates the cost-to-go from some state $s := (x,q) \in S$ with $\mathsf{Level}(q) = i$ by computing the minimum cost to to reach a state $s' := (x', q')$ such that $\mathsf{Level}(q') \in \{i-1, i\}$ and $q \neq q'$. To do so, we determine all the labels that trigger a transition from q to q' in $\mathcal{A}_\varphi$ and then find all the robot states $\mathcal{B}$ that produce those labels. Then, $h(x,q)$ is the minimum distance from x to the set $\mathcal{B}$. The details of this construction and other functions needed for A* search with LTL specifications, are summarized in Alg. 1.

Proposition 3. *The heuristic function in Alg. 1 is admissible.*

Proof. See Appendix D. □

Prop. 3 guarantees that A* will either find the optimal solution to Problem 2 or will report that Problem 2 is infeasible. In the latter case, the robot cannot satisfy the logic specification with confidence δ and it should either reduce δ or stop planning.

Note that while the heuristic function is admissible, it is not guaranteed that it is also consistent[6]. Consider two arbitrary states (x,q) and (x',q') with $\mathsf{Level}(q) = n$ and $\mathsf{Level}(q') = n+1$. It is possible that the cost to get from (x,q) to a place in the environment, where a transition to level $n-1$ occurs, is very large, i.e., $h(x,q)$ is large, but it might be very cheap to get from (x',q') to (x,q) and vice versa. In other words, it is possible that the following inequalities hold:

$$c(x,x') + h(x',q') \leq c(x,x') + c(x',x) < h(x,q),$$

[6]A consistent heuristic h satisfies $h(s) = 0$ for $s \in S_F$ and $h(s) \leq c(s,s') + h(s')$ for any $s \in S$ and any successor s' of s.

Algorithm 1 Functions needed for LTL-constrained A* search

function COMPUTEHEURISTIC(x,q)
    if q is sink **then**
        return ∞
    if $q \in F$ **then**
        return 0
    $Q_0 \leftarrow$ current level set of q in DFA
    $Q_1 \leftarrow$ next level set of q in DFA
    $\mathcal{A} \leftarrow \{a \subseteq \mathcal{AP} \mid T(q,a) \in Q_1 \cup Q_0 \setminus \{q\}\}$
        ▷ a set of labels to reach $Q_1 \cup Q_0 \setminus \{q\}$
    $\mathcal{B} \leftarrow \{x' \in X \mid L^\delta(x', \mathcal{P}) \in \mathcal{A}\}$
    return $\min_{x' \in \mathcal{B}} d(x,x')$

function ISGOAL(x,q)
    if $q \in F$ **then return true**
    else return false

function GETSUCCESSORS(x,q)
    $Succ \leftarrow \emptyset$, $Cost \leftarrow \emptyset$
    for each motion primitive $\sigma^{[T]}$ **do**
        $x_0 \leftarrow x$, $q_0 \leftarrow q$
        $x_t \leftarrow f(x_{t-1}, \sigma_{t-1}^{[T]}), t = 1, \ldots, T$ ▷ Robot state sequence
        $\alpha_t \leftarrow L^\delta(x_t, \mathcal{P}), t = 1, \ldots, T$ ▷ Atomic propositions
        $q_t \leftarrow T(q_{t-1}, \alpha_t), t = 1, \ldots, T$ ▷ Automaton state sequence
        if any q_t is sink **or** x_t hits obstacle **then**
            continue
        $Succ \leftarrow Succ \cup \{x_T\}$, $Cost \leftarrow Cost + \left\{\sum_{t=1}^{T} c(x_{t-1}, x_t)\right\}$
    return $Succ, Cost$

which makes the heuristic in Alg. 1 inconsistent. We emphasize that, even with an inconsistent heuristic, A^* can be very efficient if a technique such as bi-directional pathmax is employed to propagate heuristics between neighboring states [32].

B. Summary

We formulate temporal logic planning in a probabilistic semantic map as a stochastic optimal control problem (Problem 1). Since Problem 1 is intractable, we reduce it to a deterministic shortest path problem (Problem 2) with probabilistic correctness guaranteed by Thm 1. We can solve Problem 2 optimally using A* because the heuristic function proposed in Alg. 1 is admissible (Prop. 3). The obtained solution is partial with respect to Problem 1 because, rather than a controller that trades off the probability of satisfying the specification and the total motion cost, it provides the optimal controller in the subspace of deterministic controllers that guarantee that the probability of satisfying the specification is δ.

V. EXAMPLES

We demonstrate LTL-constrained motion planning on a differential-drive robot with state $\mathrm{x}_t := (x_t, y_t, \theta_t)^T \in SE(2)$, where (x_t, y_t) and θ_t are the 2-D position and orientation of the robot, respectively. The kinematics of the robot are discretized using a sampling period τ as follows:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{cases} \begin{pmatrix} \tau\nu\cos(\theta_t + \tau\omega/2) \\ \tau\nu\sin(\theta_t + \tau\omega/2) \\ \tau\omega \end{pmatrix}, & |\tau\omega| < 0.001, \\ \begin{pmatrix} \frac{\nu}{\omega}(\sin(\theta_t + \tau\omega) - \sin\theta_t) \\ \frac{\nu}{\omega}(\cos\theta_t) - \cos(\theta_t + \tau\omega)) \\ \tau\omega \end{pmatrix}, & \text{else.} \end{cases}$$

We accelerate the A* search by using motion primitives for the robot in order to construct a lattice-based graph [28, 6]. The advantage of such a construction is that the underlying graph is sparse and composed of dynamically-feasible robot trajectories that can incorporate a variety of constraints. A motion primitive is similar to the notion of macro-action [5, 8] and consists of a collection of control inputs $\sigma^{[T]} := (u_0, u_2, \ldots, u_{T-1})$ that are applied sequentially to a robot state x_t so that:

$$x_{t+1+k} = f(x_{t+k}, \sigma^{[T]}(k)),\ k = 0, \ldots, T-1.$$

Instead of using the original control set U, we can plan with a set $\bar{U} := \{\sigma_j^{[T_j]}\}$ of motion primitives (see GetSuccessors in Alg. 1).

In our experiments, the mobile robot is controlled via motion primitives, whose segments are specified by $\nu = 1\ m/$s, $\tau = 2$ s, and $\omega \in [-3, 3]$ rad/s (See Fig. 2). Twenty locations with outward facing orientations were chosen on the perimeter of a circle of radius 10 m. A differential-drive controller was used to to generate a control sequence of length 5 that would lead a robot at the origin to each of the selected locations. Note that the trajectories generated with motion primitives are wavy because the controller tries to follow a straight line using a discrete set of velocity and angular-velocity inputs.

The LTL constraints were specified over the two types of atomic propositions in (2) for object classes $\mathcal{C} = \{■, ●, ⬣, ⧫, ▲\}$. Proposition $\alpha_i^c(y)$ means the class of i-th landmark is y for $y \in \mathcal{C}$. Proposition $\alpha_i^p(r):\ d(x^p - l_i^p) \leq r$ means the robot is r-close to landmark l_i. The following LTL specification was given to the robot:

$$\varphi:\ (\Diamond(\phi_1 \wedge \Diamond(\phi_2 \wedge \Diamond\phi_3)) \wedge \Diamond\phi_4) \wedge \Box\phi_{safe}$$

where ϕ_i, $i = 1, \ldots, 4$ are the following propositional logic formulas:

$$\phi_1 := \vee_{i\in\{1,\ldots,M\}} \left(\alpha_i^p(1) \wedge \alpha_i^c(▲)\right),$$
$$\phi_2 := \vee_{i,j\in\{1,\ldots,M\}} \left(\alpha_i^p(2) \wedge \alpha_i^c(⧫) \wedge \alpha_j^p(2) \wedge \alpha_j^c(●)\right),$$
$$\phi_3 := \vee_{i,j\in\{1,\ldots,M\}} \left(\alpha_i^p(2) \wedge \alpha_i^c(■) \wedge \alpha_j^p(2) \wedge \alpha_j^c(●)\right),$$
$$\phi_4 := \vee_{i\in\{1,\ldots,M\}} \left(\alpha_i^p(1) \wedge \alpha_i^c(⬣)\right),$$

and for $i, j \in \{1, \ldots, M\}$ the safety constraint is:

$$\phi_{safe} := \wedge_{i,j\in\{1,\ldots,M\}} \left(\neg(\alpha_i^p(2) \wedge \alpha_i^c(■) \wedge \alpha_j^p(2) \wedge \alpha_j^c(■))\right).$$

In other words, the robot needs to first visit a triangle, then go to a region where it is close to both a circle and a diamond, and finally visit a region where it is close to both a circle and a square, while visiting a hexagon at some point and avoiding getting stuck between any two squares.

Several case studies were carried out using a simulated semantic map distribution. Robot trajectories with least cost that satisfy the LTL specification with confidence $\delta = 0.95$ for different initial conditions were computed with A* and are shown in Fig. 3. Optimal paths with the same initial conditions but different confidence parameters $\delta = 0.95$ and 0.5 are shown in Fig. 4. As expected, we observe a trade-off between the probability of satisfying the LTL formula and the total cost of the path. With a lower confidence ($\delta = 0.5$), the total cost for satisfying the LTL formula is also lower than that of a path which satisfies the formula with a high confidence ($\delta = 0.95$). Particularly, the uncertainty in the pose of triangle l_1 with mean $\bar{l}_1^p = (2.5, 1.19)$ is the main reason for the difference in the planned trajectories. With $\delta = 0.95$, even though the robot can reach the vicinity of triangle l_1, it does not have enough confidence to ensure that the triangle would be visited. Instead, it plans to visit another triangle l_2 with mean position $\bar{l}_2^p = (-5.18, 12.04)$ for which the uncertainty in the pose distribution is smaller. Reducing the confidence requirements allows the robot to plan a path that visits triangle l_1 and has a lower total cost compared to that of visiting triangle l_2.

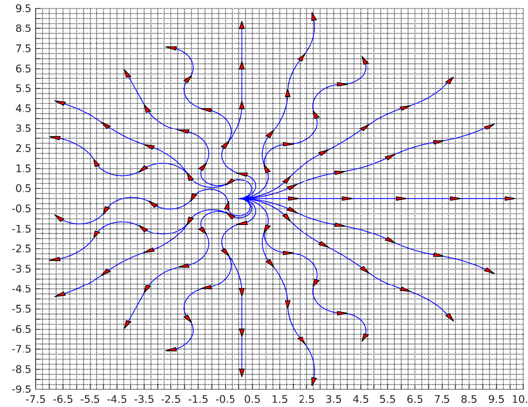
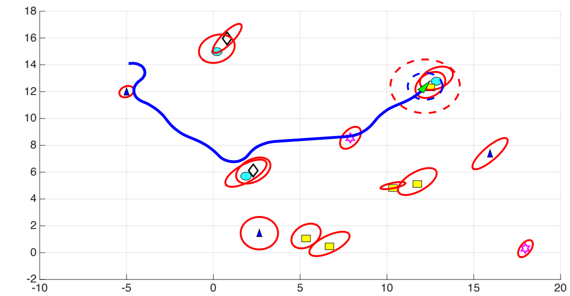


Fig. 2: The set of motion primitives used for LTL-constrained A* planning. Each segment contains 5 robot poses indicated by the red triangles. Each robot pose represents a different motion primitive.

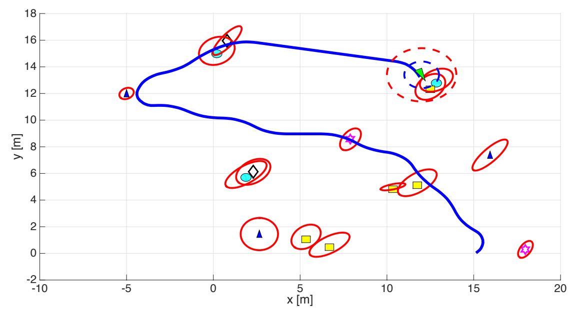
VI. Conclusion

This paper proposes an approach for planning optimal robot trajectories that probabilistically satisfy temporal logic specifications in uncertain semantic environments. By introducing a δ-confident labeling function, we show that the original stochastic optimal control problem in the continuous space of semantic map distributions can be reduced to a deterministic optimal control problem in the δ-confidence region of the map distribution. Guided by the automaton representation of the LTL co-safe specification, we develop an admissible A* algorithm to solve the deterministic problem. The advantage of our approach is that the deterministic problem can be solved very efficiently and yet the planned robot trajectory is guaranteed to have minimum cost and to satisfy the logic specification with probability δ.

This work takes an initial step towards integration of semantic SLAM and motion planning under temporal logic constraints. In future work, we plan to extend this method to handle the following: 1) landmark class uncertainty, 2) robot motion uncertainty, 3) a more general class of LTL specifications, 4) map distributions that are changing online. Our goal is to develop a coherent approach for planning autonomous robot behaviors that accomplish high-level temporal logic tasks in uncertain semantic environments.

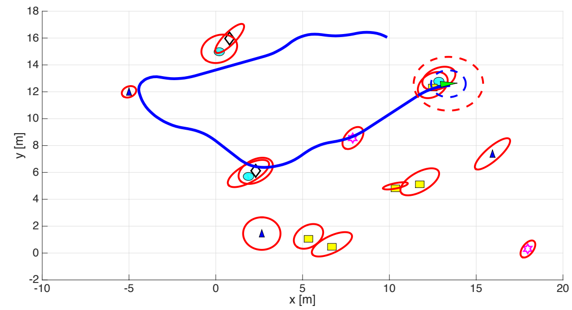


(a) $x_0 = [-5, 14, \pi/6]^T$, total motion cost 24 m.

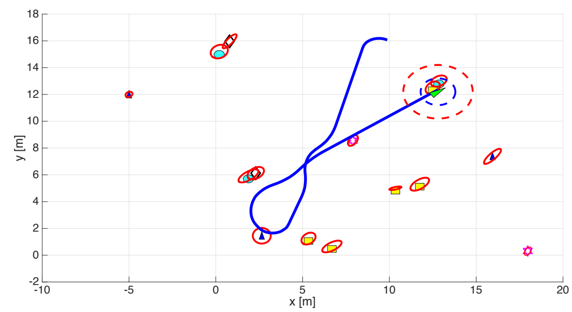


(b) $x_0 = [15, 0, \pi/6]^T$, total motion cost 44 m.

Fig. 3: Robot trajectories obtained by the planning algorithm with different initial conditions. The green triangle represents the mobile robot. The blue dash circle indicates the unit ball and the red dash circle indicates a ball of radius 2 m. The red sold ellipses represent the 0.95-confidence region of the landmark pose distributions. The landmark coordinates are drawn according to the mean of the pose distribution.



(a) The optimal plan with 0.95-confidence and motion cost 38 m.



(b) The optimal plan with 0.5-confidence and motion cost 34 m.

Fig. 4: Robot trajectories obtained by the planning algorithm with different confidence parameters. The initial state is $x_0 = [10, 16, 0.77\pi]^T$.

APPENDIX

A. Proof of Prop. 1

Let x_k be the state at time k. If for all samples m in the δ-confidence region of $\mathcal{P}$, $L(x_k, m) = L(x_k, \bar{\mathcal{M}})$, then $L^\delta(x_k, \mathcal{P}) = L(x_k, \bar{\mathcal{M}})$, which equals $L(x_k, \mathcal{M})$ with probability δ. Otherwise, $L^\delta(x_k, \mathcal{P}) = \lambda$ (empty string). Since k is arbitrary, we can conclude that $L^\delta(x_{0:t}, \mathcal{P})$ is a subsequence of $L(x_{0:t}, \mathcal{M})$ with probability δ. $\square$

B. Proof of Thm. 1

Since the language $\mathcal{L}(\mathcal{A}_\varphi)$ is a simple polynomial, the following *upward closure* [27] property holds: For any word $uv \in \mathcal{L}(\mathcal{A}_\varphi)$ and any $a \in 2^{\mathcal{AP}}$, it holds that $uav \in \mathcal{L}(\mathcal{A}_\varphi)$. In other words, if any empty string in a word from a simple polynomial language is replaced by a symbol in the alphabet, then the resulting word is still in the language.

For a given robot trajectory $x_{0:\tau}$, let $L^\delta(x_{0:\tau}, \mathcal{P}) = b_0 b_1 \dots b_\tau \in \mathcal{L}(\mathcal{A}_\varphi)$ be the δ-confident label sequence and $L(x_{0:\tau}, \mathcal{M}) = a_0 a_1 \dots a_\tau$ be the true label sequence. According to Prop. 1, for each $0 \le i \le \tau$, either $a_i = b_i$ or $a_i \neq b_i$ and $b_i = \emptyset$. Thus, if $b_0 b_1 \dots b_\tau$ belongs to $\mathcal{L}(\mathcal{A}_\varphi)$, $a_0 a_1 \dots a_\tau$ must be in $\mathcal{L}(\mathcal{A}_\varphi)$ because it is obtained by replacing each empty string in $b_0 b_1 \dots b_\tau$ with some symbol in $2^{\mathcal{AP}}$ and the language $\mathcal{L}(\mathcal{A}_\varphi)$ is upward closed. $\square$

C. Characterization of LTL co-safe formulas that translate to simple polynomials

Formally, the subset of LTL formulas is defined by the grammar

$$\varphi := \varphi_{reach} \mid \varphi_{seq} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi, \tag{3}$$

where $\varphi_{reach} = \Diamond \phi \mid \varphi_{reach} \wedge \varphi_{reach} \mid \varphi_{reach} \vee \varphi_{reach}$ represents the reachability, $\varphi_{seq} := \phi \mid \Diamond \varphi_{seq} \mid \Diamond(\varphi_{seq} \wedge \Diamond \varphi_{seq})$ is a set of formulas describing sequencing constraints. Here, ϕ is a propositional logic formula.

D. Proof of Prop. 3

We proceed by induction on the levels in $\mathcal{A}_\varphi$. In the base case, $q \in Q_0 \equiv F$ and $h(x, q) = 0$ for any $x \in X$. Suppose that the proposition is true for level n and let (x, q) be some state with $x \in X$ and $\mathsf{Level}(q) = n + 1$. As before, let $h^*(x, q)$ be the optimal cost-to-go. Due to Prop. 2, there are only three possibilities for the next state (x', q') along the optimal path starting from (x, q):

- $\mathsf{Level}(q') = n$: By construction of h:

$$h^*(x, q) = c(x, x') + h^*(x', q') \ge h(x, q) + 0.$$

- $\mathsf{Level}(q') = n + 1$: Same conclusion as above.
- $\mathsf{Level}(q') = k > n + 1$: In this case, there exists another state (x'', q'') later along the optimal path such that $\mathsf{Level}(q'') = n + 1$ (otherwise the optimal path cannot

reach the goal set). Then, by the triangle inequality for the motion cost c:

$$\begin{aligned} h^*(x,q) &= c(x,x') + c(x',x'') + h^*(x'',q'') \\ &\geq c(x,x'') + 0 \geq h(x,q) \end{aligned}$$

Thus, we conclude that $h^*(x,q) \geq h(x,q) \geq 0$ for all $x \in X$ and $q \in Q$. □

References

[1] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Semantic Localization Via the Matrix Permanent. In *Robotics: Science and Systems*, 2014.

[2] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy. Estimation, planning, and mapping for autonomous flight using an rgb-d camera in GPS-denied environments. *The International Journal of Robotics Research*, 31(11):1320–1343, 2012.

[3] S. Bao and S. Savarese. Semantic Structure from Motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2025–2032, 2011.

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

[5] A. Botea, M. Enzenberger, M. Muller, and J. Schaeffer. Macro-FF: Improving AI planning with automatically learned macro-operators. *Journal of Artificial Intelligence Research*, 24:581–621, 2005.

[6] B. Cohen, G. Subramanian, S. Chitta, and M. Likhachev. Planning for Manipulation with Adaptive Motion Primitives. In *IEEE International Conference on Robotics and Automation*, pages 5478–5485, 2011.

[7] X. Ding, S. Smith, C. Belta, and D. Rus. Optimal control of Markov decision processes with linear temporal logic constraints. *IEEE Transactions on Automatic Control*, 59(5): 1244–1257, May 2014.

[8] R. Fikes and N. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3):189–208, 1971.

[9] J. Fu and U. Topcu. Probably approximately correct mdp learning and control with temporal logic constraints. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[10] A. Guez and J. Pineau. Multi-tasking SLAM. In *IEEE International Conference on Robotics and Automation*, pages 377–384, 2010.

[11] M. Guo, K. H. Johansson, and D. V. Dimarogonas. Revising motion planning under linear temporal logic specifications in partially known workspaces. In *IEEE International Conference on Robotics and Automation*, pages 5025–5032, 2013.

[12] B. Johnson and H. Kress-Gazit. Analyzing and revising synthesized controllers for robots with sensing and actuation errors. *The International Journal of Robotics Research*, 34(6): 816–832, 2015.

[13] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning with deterministic μ-calculus specifications. In *American Control Conference*, pages 735–742, 2012.

[14] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

[15] 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.

[16] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

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

[18] B. Lacerda, D. Parker, and N. Hawes. Optimal policy generation for partially satisfiable co-safe LTL specifications. In *International Joint Conference on Artificial Intelligence*, pages 1587–1593, 2015.

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

[20] S. LaValle. Rapidly-Exploring Random Trees A New Tool for Path Planning. Technical Report Computer Science Dept., TR 98-11, Iowa State University, 1998.

[21] M. Likhachev, G. Gordon, and S. Thrun. ARA* : Anytime A* with Provable Bounds on Sub-Optimality. In *Advances in Neural Information Processing Systems*, pages 767–774. 2004.

[22] S. C. Livingston, R. M. Murray, and J. W. Burdick. Backtracking temporal logic synthesis for uncertain environments. In *IEEE International Conference on Robotics and Automation*, pages 5163–5170, 2012.

[23] S. C. Livingston, P. Prabhakar, A. B. Jose, and R. M. Murray. Patching task-level robot controllers based on a local μ-calculus formula. In *IEEE International Conference on Robotics and Automation*, pages 4588–4595, 2013.

[24] P. E. Missiuro and N. Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *IEEE International Conference on Robotics and Automation*, pages 1261–1267, 2006.

[25] A. Nüchter and J. Hertzberg. Towards Semantic Maps for Mobile Robots. *Robotics and Autonomous Systems*, 56(11): 915–926, 2008.

[26] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[27] J.-É. Pin. Mathematical foundations of automata theory. *Lecture notes LIAFA, Université Paris*, 7, 2010.

[28] M. Pivtoraiko and A. Kelly. Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3231–3237, 2005.

[29] A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13(1):45 – 60, 1981.

[30] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013.

[31] E. M. Wolff, U. Topcu, and R. M. Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *IEEE Annual Conference on Decision and Control*, pages 3372–3379, 2012.

[32] Z. Zhang, N. Sturtevant, R. Holte, J. Schaeffer, and A. Felner. A* search with inconsistent heuristics. In *International Joint Conference on Artificial Intelligence*, pages 634–639, 2009.