2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Multi-Robot Mission Planning in Dynamic Semantic Environments

Samarth Kalluraya[1], George J. Pappas[2], Yiannis Kantaros[1]

Abstract— **This paper addresses a new semantic multi-robot planning problem in uncertain and dynamic environments. Particularly, the environment is occupied with mobile and uncertain semantic targets. These targets are governed by stochastic dynamics while their current and future positions as well as their semantic labels are uncertain. Our goal is to control mobile sensing robots so that they can accomplish collaborative semantic tasks defined over the uncertain current/future positions and semantic labels of these targets. We express these tasks using Linear Temporal Logic (LTL). We propose a sampling-based approach that explores the robot motion space, the mission specification space, as well as the future configurations of the semantic targets to design optimal paths. These paths are revised online to adapt to uncertain perceptual feedback. To the best of our knowledge, this is the first work that addresses semantic mission planning problems in uncertain and dynamic semantic environments. We provide extensive experiments that demonstrate the efficiency of the proposed method.**

I. Introduction

Robot navigation has received considerable research attention [1]–[3]. Typically, motion planning problems require generating trajectories that reach known goal regions while avoiding known/unknown, and possibly dynamic, obstacles. Recent advances in computer vision and semantic mapping offer a unique opportunity to transition from these well-studied *geometric* planning approaches to *semantic* mission planning problems requiring reasoning about both the geometric and semantic environmental structure [4]–[6].

In this paper we address a new semantic multi-robot planning problem in uncertain and dynamic environments. The environment is assumed to have known and static geometry (e.g., walls) but it is occupied with mobile and uncertain labeled targets of interest (e.g., pedestrians, drones, etc) that do not interact with our multi-robot system. Particularly, the targets move as per known dynamics but they are subject to exogenous disturbances (e.g., wind gusts) resulting in positional uncertainty (*metric* uncertainty). The labels of these targets are also initially unknown (*semantic* uncertainty). Instead, the robots have access to a probabilistic prior belief about the initial positions and the semantic labels of the targets. This prior belief may be user-specified or computed by existing semantic mapping methods [4], [5]. The goal of the robots is to accomplish collaborative *semantic tasks* defined over the uncertain positions and/or the semantic labels of the targets. These tasks are expressed using Linear Temporal Logic (LTL) [7]. To accomplish them, the robots are equipped with imperfect perception systems (e.g., cameras and learning-based object detectors) that allow them to reason about the semantic environmental structure by detecting, classifying, and localizing objects. The considered planning problem gives rise to an optimal control problem that generates open-loop control policies. To solve this new problem, building upon our previous work [8], we propose a sampling-based approach that explores the robot motion space, the mission specification space, as well as the uncertain future states of the mobile semantic targets. The offline designed control policies are updated online to adapt to uncertain perceptual feedback. Extensions to targets with fully unknown dynamics are also discussed.

Related works: Several motion planning algorithms have been proposed that assume known [9]–[14] or unknown [15]–[27] but static environments. Recently, these methods have been extended to dynamic environments for both reach-avoid [28]–[38] and temporal logic tasks [39]–[43]. Common in these works is that they consider environments with unknown geometry but with known semantic structure. As a result, they consider tasks requiring reaching perfectly known and static regions while avoiding dynamic known/unknown obstacles. To the contrary, here, we consider semantic tasks that require reasoning about both the uncertain current/future positions and the semantic labels of mobile targets. For instance, consider a semantic task requiring a drone to take a picture of an 'abandoned car' while avoiding 'patrolling drones' where the current/future locations and the semantic labels of these targets (i.e., cars and drones) are uncertain. Related are also the works on active sensing for target tracking that require to actively decrease uncertainty of all mobile targets; see e.g., [44]–[52]. Nevertheless, the control objective in our work is fundamentally different; our goal is to design informative paths that satisfy temporal logic missions defined over these uncertain labeled targets. To the best of our knowledge, the most relevant work to the one proposed here is the recent work by the authors [8], which, unlike this work, considers *static* semantic environments. In this paper, we extend [8] to dynamic environments.

Contributions: *First*, we formulate a new semantic mission planning problem in dynamic and uncertain semantic environments. *Second*, we present a sampling-based approach to design paths that satisfy semantic missions captured by LTL specifications. *Third*, we provide extensive experiments that demonstrate the efficiency of the proposed algorithm.

II. Problem Definition

A. Modeling Uncertain & Dynamic Semantic Environments

We consider semantic environments, denoted by Ω, with known geometric structure (e.g., walls or buildings). This

[1]Department of Electrical and Systems Engineering (ESE), Washington University in St. Louis, St. Louis, MO, 63130, USA. {k.samarth,ioannisk@wustl.edu}.[2]Department of ESE, University of Pennsylvania, Philadelphia, PA, 19104, USA. {pappasg@seas.upenn.edu}. This work was supported by the ARL grant DCIST CRA W911NF-17-2-0181.

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160344

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.

known obstacle-free space, denoted by $\Omega_{\text{free}} \subseteq \Omega$, is cluttered with $M > 0$ mobile labeled targets ℓ_i giving rise to a dynamic semantic map $\mathcal{M}(t) = \{\ell_1, \ell_2, \ldots, \ell_M\}$. Each target $\ell_i = \{\mathbf{x}_i(t), c_i, \mathbf{g}_i\} \in \mathcal{M}$ is defined by its state $\mathbf{x}_i(t)$ (e.g., position and orientation) at a time t, it's class $c_i \in \mathcal{C}$, where $\mathcal{C}$ is a finite set of classes (e.g., 'car', 'pedestrian', and 'drone'), and dynamics $\mathbf{g}_i$. Specifically, each target is governed by the following dynamics: $\mathbf{x}_i(t+1) = \mathbf{g}_i(\mathbf{x}_i(t), \boldsymbol{\mu}_i(t), \boldsymbol{\nu}_i(t))$, where $\boldsymbol{\mu}_i(t)$ is the control input selected by target i at time t and $\boldsymbol{\nu}_i(t)$ models noise and exogenous disturbances in the target dynamics. We assume that the noise follows a Gaussian distribution, i.e., $\boldsymbol{\nu}_i \sim \mathcal{N}(0, \mathbf{R}_i)$ with known covariance matrix $\mathbf{R}_i$. **Assumption (a):** Hereafter, we assume that $\mathbf{g}_i$ models linear system dynamics, i.e., $\mathbf{x}_i(t+1) = \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{B}_i\boldsymbol{\mu}_i(t) + \boldsymbol{\nu}_i$. We compactly denote the dynamics of all targets by:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\boldsymbol{\mu}(t) + \boldsymbol{\nu}, \tag{1}$$

where $\mathbf{x}$ is a vector that stacks the states of all targets. **Assumption (b):** We assume that the dynamics $\mathbf{g}_i$, i.e., the matrices $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{R}_i$ as well as $\boldsymbol{\mu}_i(t)$ are known for all t; for instance, target dynamics can be learned using existing data-driven methods. The control inputs $\boldsymbol{\mu}_i(t)$ are pre-determined (offline) and they are not affected by the multi-robot system introduced in Section II-B.

The true state $\mathbf{x}_i(t)$ of target i at time t is uncertain. **Assumption (c):** We assume that we have access to a prior Gaussian belief about the initial state of each target, i.e., $\mathbf{x}_i(0) \sim \mathcal{N}(\hat{\mathbf{x}}_i(0), \Sigma_i(0))$, where $\hat{\mathbf{x}}_i(0)$ and $\Sigma_i(0)$ denote the mean and covariance matrix for $\mathbf{x}_i(0)$. We note that assumptions (a)-(c) are quite common in the related literature [45], [46], [52]; in Section III-D, we discuss data-driven methods to relax these assumptions. As it will be discussed later, these assumptions allow us to model the uncertain future states $\mathbf{x}_i(t)$ as Gaussian distributions, using a Kalman filter (KF) approach. These Gaussian distributions model *metric* uncertainty in the environment. Also, the true labels/classes c_i of the targets ℓ_i are uncertain as well. **Assumption (d):** We assume that we have access to an arbitrary discrete distribution d_i for all targets ℓ_i that model the probability that the class for ℓ_i is $c_i \in \mathcal{C}$, i.e., $c_i \sim d_i$. The discrete distributions model *semantic* uncertainty in the environment. This prior information can be provided by semantic mapping methods [4], [5] or it can be user-specified.

B. Modeling Perception-based Robots

Consider N mobile robots governed by the following dynamics: $\mathbf{p}_j(t+1) = \mathbf{f}_j(\mathbf{p}_j(t), \mathbf{u}_j(t))$, for all $j \in \{1, \ldots, N\}$, where $\mathbf{p}_j(t) \in \mathbb{R}^n$ stands for the state (e.g., position and orientation) of robot j in the free space Ω_{free} at discrete time t, and $\mathbf{u}_j(t) \in U_j$ denotes a control input selected from a *finite* space of admissible controls U_j. For simplicity, we also denote $\mathbf{p}_j(t) \in \Omega$. Hereafter, we compactly denote the dynamics of all robots as

$$\mathbf{p}(t+1) = \mathbf{f}(\mathbf{p}(t), \mathbf{u}(t)), \tag{2}$$

where $\mathbf{p}(t) \in \Omega^N$, $\forall t \geq 0$, and $\mathbf{u}(t) \in U := U_1 \times \cdots \times U_N$.

The robots are equipped with sensors (e.g., cameras) to collect measurements associated with $\mathbf{x}(t)$. **Assumption (e):** We assume that these sensors can be modeled as per the following *linear observation model*: $\mathbf{y}_j(t) = \mathbf{M}_j(\mathbf{p}_j(t))\mathbf{x}(t) + \mathbf{v}_j(t)$, where $\mathbf{y}_j(t)$ is the measurement signal at discrete time t taken by robot j. Also, $\mathbf{v}_j(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_j)$ is Gaussian noise with known covariance $\mathbf{Q}_j$; similar sensor models are used e.g., in [53]. We compactly denote all observation models as

$$\mathbf{y}(t) = \mathbf{M}(\mathbf{p}(t))\mathbf{x} + \mathbf{v}(t), \;\; \mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{3}$$

where $\mathbf{y}(t)$ collects measurements taken at time t by all robots associated with any target. The robots are also equipped with object recognition systems allowing them to collect measurements associated with the target classes. These measurements typically consist of label measurements along with label probabilities [54]–[56]. The semantic measurement is generated by the following observation model: $[\mathbf{y}_j^c, \mathbf{s}_j^c] = \mathbf{g}_j(L_i)$ (see also [57]) where $\mathbf{y}_j^c$ and $\mathbf{s}_j^c$ represent a class/label measurement and the corresponding probability scores over all available classes, respectively, and L_i stands for the true class of the detected target ℓ_i. We compactly denote the object recognition model of all robots as

$$[\mathbf{y}^c(t), \mathbf{s}^c(t)] = \mathbf{g}(\mathbf{L}), \tag{4}$$

where $\mathbf{L}$ denote the true classes of all targets detected.

C. Kalman Filter for Offline Map Prediction

Our belief about the metric/semantic environmental structure can be updated online by using the observations generated by (3)-(4) using existing semantic mapping methods [4], [5]. Here we leverage a Kalman Filter (KF) to approximately predict offline (i.e., without observations) the future target states as well as their associated metric uncertainty. First, due to assumptions (b)-(c), the expected states of the targets at time $t+1$ can be predicted by applying recursively the KF prediction formula, i.e., $\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\boldsymbol{\mu}(t)$. Note that this is the *a-priori* state estimate of targets ℓ_i; the *a-posteriori* state estimates require observations that are not available offline. Due to assumptions (a)-(c) and (e), the associated covariance matrix $\Sigma(t+1)$, capturing metric uncertainty, can be computed using the KF Riccati equation $\Sigma(t+1) = \rho(\Sigma(t), \mathbf{p}(t+1))$. Note that this is the *a-posteriori* covariance matrix which can be computed optimally without the need for measurements [45]. Finally, the semantic uncertainty, captured by the discrete distribution d, cannot be updated offline as it requires field measurements (i.e., images). Hereafter, we compactly denote by $\hat{\mathcal{M}}(t) = (\hat{\mathbf{x}}(t), \Sigma(t), d(t))$, our offline estimate for all targets at t.

D. Semantic Mission & Safety Specifications using LTL

The goal of the robots is to accomplish a collaborative semantic task captured by a global co-safe Linear Temporal Logic (LTL) specification ϕ. Similar to [8], [58], [59], this specification is defined over probabilistic atomic predicates that depend on both the multi-robot state $\mathbf{p}(t)$ and the estimate $\hat{\mathcal{M}}(t)$. Specifically, we define perception-based

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.

predicates, defined as follows:

$$\pi_p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \Delta) = \begin{cases} \text{true}, & \text{if } p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \Delta) \geq 0 \\ \text{false}, & \text{otherwise} \end{cases} \tag{5}$$

In (5), Δ is a set of user-specified and case-specific parameters (e.g., probabilistic thresholds or robots indices) and $p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \Delta) : \Omega^N \times \hat{\mathcal{M}}(t) \times \Delta \to \mathbb{R}$. Hereafter, when it is clear from the context, we simply denote a perception-based predicate by π_p. First, we define the following function reasoning about the metric uncertainty:

$$p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{j, \ell_i, r, \delta\}) = \\ \mathbb{P}(||\mathbf{p}_j(t) - \mathbf{x}_i(t)|| \leq r) - (1 - \delta). \tag{6}$$

The predicate associated with (6) is true at time t if the probability of robot j being within distance less than r from target ℓ_i (regardless of its class) is greater than $1 - \delta$, after applying control actions $\mathbf{u}_{0:t}$, for some user-specified parameters $r, \delta > 0$. We also define the following function reasoning about both the metric and the semantic uncertainty:

$$p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{j, r, \delta, c\}) = \\ \max_{\ell_i} [\mathbb{P}(||\mathbf{p}_j(t) - \mathbf{x}_i(t)|| \leq r) d_i(c)] - (1 - \delta). \tag{7}$$

In words, the predicate associated with (7) is true at time t if the probability of robot j being within distance less than r from at least one target with class c is greater than $1 - \delta$.

The syntax of co-safe LTL over a sequence of multi-robot states $\mathbf{p}(t)$ and uncertain semantic estimates $\hat{\mathcal{M}}(t)$ is defined as $\phi ::= \text{true} \mid \pi_p \mid \neg\pi_p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \, \mathcal{U} \, \phi_2$, where (i) π_p is a perception-based predicate defined before and (ii) $\wedge$, $\vee$, $\neg$, and $\mathcal{U}$, denote the conjunction, disjunction, negation, and until operator, respectively. Using $\mathcal{U}$, the eventually operator $\Diamond$, can be defined as well [7].

E. Safe Planning over Uncertain Dynamic Semantic Maps

Given a task ϕ, the sensing model (3), the robot dynamics, and under assumptions (a)-(e), our goal is to select a stopping horizon H and a sequence $\mathbf{u}_{0:H}$ of control inputs $\mathbf{u}(t)$, for all $t \in \{0, \ldots, H\}$, that satisfy ϕ while minimizing a user-specified motion cost function. This gives rise to the following optimal control problem:

$$\min_{H, \mathbf{u}_{0:H}} \left[J(H, \mathbf{u}_{0:H}) = \sum_{t=0}^{H} c(\mathbf{p}(t), \mathbf{p}(t+1)) \right] \tag{8a}$$

$$[\mathbf{p}_{0:H}, \hat{\mathcal{M}}_{0:H}] \models \phi \tag{8b}$$

$$\mathbf{p}(t) \in \Omega^N_{\text{free}}, \tag{8c}$$

$$\mathbf{p}(t+1) = \mathbf{f}(\mathbf{p}(t), \mathbf{u}(t)) \tag{8d}$$

$$\hat{\mathbf{x}}(t+1) = A\hat{\mathbf{x}}(t) + B\boldsymbol{\mu}(t) \tag{8e}$$

$$\Sigma(t+1) = \rho(\Sigma(t), \mathbf{p}(t+1)) \tag{8f}$$

$$d(t+1) = d(0) \tag{8g}$$

where the constraints (8d)-(8g) hold for all $t \in [0, H]$. In (8a), any motion cost function $c(\mathbf{p}(t), \mathbf{p}(t+1))$ can be used associated with the transition cost from $\mathbf{p}(t)$ to $\mathbf{p}(t+1))$ as long as it is positive (e.g., traveled distance). The constraints (8b)-(8c) require the robots to accomplish the mission specification ϕ and always avoid the known obstacles/walls respectively. With slight abuse of notation, in (8b), $[\mathbf{p}_{0:H}, \hat{\mathcal{M}}_{0:H}]$ denotes a finite sequence of length/horizon H of multi-robot states and semantic estimates while $[\mathbf{p}_{0:H}, \hat{\mathcal{M}}_{0:H}] \models \phi$ means that the symbols generated along this finite sequence satisfy ϕ. The constraint (8d) requires the robots to move according to their known dynamics. Also, (8e)-(8g) capture the offline map prediction (Section II-C).

Problem 1: Under assumptions (a)-(e), and given an initial robot state $\mathbf{p}(0)$, a sensor model (3), and a task ϕ, compute a horizon H and control inputs $\mathbf{u}(t)$ for all $t \in \{0, \ldots, H\}$ as per (8).

Remark 2.1 (Online Re-planning): The object recognition method (4) is not required to solve (8) as, therein, the semantic uncertainty is not updated. In fact, (8) is an offline problem yielding open-loop/offline paths that are agnostic to (4). A sampling-based algorithm to solve (8) is presented in Sections III-A-III-B. In Section III-C we discuss how and when the offline paths may need to be revised online to adapt to perceptual feedback captured by (3)-(4).

III. Safe Planning In Dynamic Semantic Maps

In this section, we present an algorithm to solve the semantic planning problem defined in (8). To solve this new problem, first, in Section III-A, we convert (8) into a reachability problem that is defined over a hybrid state space. This state space consists of the multi-robot states, future states of the mobile semantic targets along with their corresponding quantified metric and semantic uncertainty (captured by Gaussian and discrete distributions), and a discrete automaton state space associated with the LTL task. To solve this reachability problem, we appropriately modify the sampling-based algorithm proposed in our previous work [8] to account for dynamic environments; see Section III-B. We note that the major difference with [8] lies in the structure of the state space that needs to be explored. Specifically, unlike [8], here, exploration of the future uncertain states of the dynamic targets is needed due to (8e). In Section III-C, we show how the proposed algorithm can be used online to to adapt to perceptual feedback (3)-(4). In Section III-D, we discuss how to relax assumptions (a)-(e) (see Section II-A).

A. Reachability in Uncertain Hybrid Spaces

First, we convert (8) into a reachability problem. This is achieved by converting ϕ into a Deterministic Finite state Automaton (DFA), defined as follows [7].

Definition 3.1 (DFA): A Deterministic Finite state Automaton (DFA) D over $\Sigma = 2^{\mathcal{AP}}$ is defined as a tuple $D = \left(\mathcal{Q}_D, q_D^0, \Sigma, \delta_D, q_F\right)$, where $\mathcal{Q}_D$ is the set of states, $q_D^0 \in \mathcal{Q}_D$ is the initial state, Σ is an alphabet, $\delta_D : \mathcal{Q}_D \times \Sigma \to \mathcal{Q}_D$ is a deterministic transition relation, and $q_F \in \mathcal{Q}_D$ is the accepting/final state.

We also define a labeling function $L : \Omega^N \times \hat{\mathcal{M}}(t) \to 2^{\mathcal{AP}}$ determining which atomic propositions are true given the current multi-robot state $\mathbf{p}(t)$ and the current map $\hat{\mathcal{M}}(t)$. Given a robot trajectory $\mathbf{p}_{0:H}$ and a corresponding sequence of maps $\hat{\mathcal{M}}_{0:H}$, we get the labeled sequence $L(\mathbf{p}_{0:H}, \hat{\mathcal{M}}_{0:H}) = L([\mathbf{p}(0), \hat{\mathcal{M}}(0)]) \ldots L([\mathbf{p}(H), \hat{\mathcal{M}}(H)])$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.

This labeled sequence satisfies the specification ϕ, if starting from the initial state q_D^0, each symbol/element in $L(\mathbf{p}_{0:H}, \hat{\mathcal{M}}_{0:H})$ yields a DFA transition so that eventually -after H DFA transitions- the final state q_F is reached [7]. As a result, we can equivalently re-write (8) as follows:

$$\min_{H, \mathbf{u}_{0:H}} \left[J(H, \mathbf{u}_{0:H}) = \sum_{t=0}^{H} c(\mathbf{p}(t), \mathbf{p}(t+1)) \right] \tag{9a}$$

$$q_D(t+1) = \delta_D(q_D(t), \sigma(t)), \tag{9b}$$

$$\mathbf{p}(t+1) = \mathbf{f}(\mathbf{p}(t), \mathbf{u}(t)) \tag{9c}$$

$$\mathbf{p}(t) \in \Omega^N_{\text{free}}, \tag{9d}$$

$$\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\boldsymbol{\mu}(t) \tag{9e}$$

$$\Sigma(t+1) = \rho(\Sigma(t), \mathbf{p}(t+1)) \tag{9f}$$

$$d(t+1) = d(0) \tag{9g}$$

$$q_D(H) = q_F \tag{9h}$$

where $q_D(0) = q_D^0$, $\sigma(t) = L([\mathbf{p}(t), \hat{\mathcal{M}}(t)])$, and $d(t)$. Note that the constraint in (9b) captures the automaton dynamics, i.e., the next DFA state that will be reached from the current DFA state under the observation/symbol $\sigma(t)$. In words, (9), is a reachability problem defined over a joint space consisting of the automaton state-space (see (9b)), the robot motion space (see (9c)-(9d)), and the future target states along with their corresponding metric and semantic uncertainty (see (9e)-(9g)) while the terminal constraint requires to reach the final automaton state (see (9h)).

B. Sampling-based Algorithm

In this section, we present a sampling-based algorithm to solve (9). The key idea in the proposed algorithm is to incrementally build a tree exploring the hybrid space over which (9) is defined in order to find the optimal path. We present a brief overview of it while a more detailed description can be found in [60].

First, the proposed algorithm builds a tree denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, J_{\mathcal{G}}\}$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. The set of nodes $\mathcal{V}$ contains states of the form $\mathbf{q}(t) = [\mathbf{p}(t), \hat{\mathcal{M}}(t), q_D(t)]$; when it is clear from the context, we drop the dependence of $\mathbf{q}(t)$ on t. The function $J_{\mathcal{G}} : \mathcal{V} \to \mathbb{R}_+$ assigns the cost of reaching node $\mathbf{q} \in \mathcal{V}$ from the root of the tree. The root of the tree, denoted by $\mathbf{q}(0)$, is constructed so that it matches the initial multi-robot state $\mathbf{p}(0)$, the initial estimate for the semantic targets $\hat{\mathcal{M}}(0)$ (determined by the prior information), and the initial DFA state, i.e., $\mathbf{q}(0) = [\mathbf{p}(0), \hat{\mathcal{M}}(0), q_D^0]$. By convention the cost of the root $\mathbf{q}(0)$ is $J_{\mathcal{G}}(\mathbf{q}(0)) = 0$, while the cost of a node $\mathbf{q}(t+1) \in \mathcal{V}$, given its parent node $\mathbf{q}(t) \in \mathcal{V}$, is computed as $J_{\mathcal{G}}(\mathbf{q}(t+1)) = J_{\mathcal{G}}(\mathbf{q}(t)) + c(\mathbf{p}(t), \mathbf{p}(t+1))$. Observe that by applying this equation recursively, we get the objective function in (8). The tree is initialized so that $\mathcal{V} = \{\mathbf{q}(0)\}$, $\mathcal{E} = \emptyset$, and $J_{\mathcal{G}}(\mathbf{q}(0)) = 0$. Also, the tree is built incrementally by adding new states $\mathbf{q}_{\text{new}}$ to $\mathcal{V}$ and corresponding edges to $\mathcal{E}$, at every iteration n, based on a *sampling* and *extending-the-tree* operation. Particularly, at every iteration, first we *sample* an existing node in the tree ($\mathbf{q}_{\text{rand}}(t) = [\mathbf{p}_{\text{rand}}, \hat{\mathcal{M}}_{\text{rand}}, q_D^{\text{rand}}]$) and a control input ($\mathbf{u}_{\text{new}} \in U$) that are used to construct a new node $\mathbf{q}_{\text{new}}(t+1) = [\mathbf{p}_{\text{new}}, \hat{\mathcal{M}}_{\text{new}}, q_D^{\text{new}}]$. To this end, first we compute $\mathbf{p}_{\text{new}} = \mathbf{f}(\mathbf{p}_{\text{rand}}, \mathbf{u}_{\text{new}})$, as in (9c). If $\mathbf{p}_{\text{new}}$ belongs to the obstacle-free space, as required by (9d), then we proceed with the *extend* operation defined as follows. Given $\hat{\mathcal{M}}_{\text{rand}}$, we compute $\hat{\mathcal{M}}_{\text{new}}$ as per (9e)-(9g), i.e., $\hat{\mathbf{x}}_{\text{new}}(t+1) = \mathbf{A}\hat{\mathbf{x}}_{\text{rand}}(t) + \mathbf{B}\boldsymbol{\mu}(t)$, $\Sigma_{\text{new}}(t+1) = \rho(\Sigma_{\text{rand}}(t), \mathbf{p}_{\text{new}}(t+1))$, and $d_{\text{new}}(t+1) = d_{\text{rand}}(t)$. Also, we compute the DFA state q_D^{new}, where $q_D^{\text{new}} = \delta_D(q_D^{\text{rand}}, L([\mathbf{p}_{\text{rand}}, \mathcal{M}_{\text{rand}}]))$ as required by (9b). Once the new sample $\mathbf{q}_{\text{new}}$ is constructed, it is added to the set $\mathcal{V}$ and an edge from $\mathbf{q}_{\text{rand}}$ to $\mathbf{q}_{\text{new}}$ is added to $\mathcal{E}$.

After taking $n_{\max} \geq 0$ samples, where $n_{\max}$ is user-specified, the algorithm terminates and returns a feasible solution to (9) (if it has been found), i.e., a terminal horizon H and a sequence of control inputs $\mathbf{u}_{0:H}$. To extract such a solution, we need first to define the set $\mathcal{X}_g \subseteq \mathcal{V}$ that collects all states $\mathbf{q}(t) \in \mathcal{V}$ of the tree that satisfy $q_D(t) = q_F$ capturing the terminal constraint (9h). Then, among all nodes $\mathcal{X}_g$, we select the node with the smallest cost $J_{\mathcal{G}}(\mathbf{q}(t))$, denoted by $\mathbf{q}(t_{\text{end}})$. Then, the terminal horizon is $H = t_{\text{end}}$, and the control inputs $\mathbf{u}_{0:H}$ are recovered by computing the path $\mathbf{q}_{0:t_{\text{end}}}$ in $\mathcal{G}$ that connects $\mathbf{q}(t_{\text{end}})$ to the root $\mathbf{q}(0)$, i.e., $\mathbf{q}_{0:t_{\text{end}}} = \mathbf{q}(0), \ldots, \mathbf{q}(t_{\text{end}})$. Observe that the computed path satisfies all constraints in (9) by construction of the tree.

The proposed algorithm is probabilistically complete (Theorem 3.2) and asymptotically optimal (Theorem 3.3). The proof of this result follows the same logic as in [8] and, therefore, it is omitted. The key idea is to show that the sampling-based algorithm exhaustively searches all possible sequences $\mathbf{u}_{0:H}$ and finite horizons H.

Theorem 3.2 (Probabilistic Completeness): The proposed sampling-based algorithm is probabilistically complete, i.e., ff there exists a solution to (8), then the probability of finding a feasible solution goes to 1 as $n \to \infty$.

Theorem 3.3 (Asymptotic Optimality): Assume that there exists an optimal solution to (8). Then, the proposed algorithm is asymptotically optimal, i.e., the optimal horizon H and the optimal sequence of control inputs $\mathbf{u}_{0:H}$ will be found with probability 1, as $n_{\max} \to \infty$.

C. Online Execution and Re-planning

The proposed algorithm generates an open-loop sequence $\mathbf{q}_{0:H} = \mathbf{q}(0), \mathbf{q}(1), \ldots, \mathbf{q}(H)$, where $\mathbf{q}(t) = [\mathbf{p}(t), \hat{\mathcal{M}}(t), q_D(t)]$, so that the resulting robot trajectory $\mathbf{p}_{0:H}$ and sequence of maps $\hat{\mathcal{M}}_{0:H}$ satisfy ϕ. While executing these paths, the robots take measurements (see (3)-(4)) to update the (i) a posteriori mean and covariance of the landmarks and (ii) the discrete distribution associated with the landmark classes yielding an updated semantic map denoted by $\hat{\mathcal{M}}_{\text{online}}(t)$ which may be different from the offline estimate of the map $\hat{\mathcal{M}}(t)$. This may require the robots to replan to adapt to the new map. Formally, at time t the robots replan if, for some $H - t > T > 0$, the DFA state $q_D(t+T)$ in $\mathbf{q}_{0:H}$ cannot be reached, given the remaining robot path $\mathbf{p}_{t:t+T}$ and the sequence of maps $\hat{\mathcal{M}}_{t:t+T}$, where $\hat{\mathcal{M}}(t) = \hat{\mathcal{M}}_{\text{online}}(t)$ and $\hat{\mathcal{M}}(t+k)$, for all $k \in \{t+1, \ldots, t+T\}$ is computed as in Section II-C. Notice that as T increases, the computational cost of reasoning

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.

whether replanning is needed increases as well. But, a larger T may prevent unnecessary replanning events. Re-planning is called every time when needed, till the robots complete the mission by reaching the final accepting state in the DFA.

D. Extensions & Future Work

We discuss how assumptions (a)-(e) can be relaxed; see Section II. Assumptions (a) and (e) require sensor and target linear models which may not hold in practice. This was required to compute offline the optimal a posteriori covariance matrices. This can be relaxed by linearizing them and applying an extended KF as e.g., in [46]. Assumption (b) requires knowledge of the target dynamics including their control inputs. This is required to predict the future target states. This assumption can be relaxed by leveraging recurrent neural networks (RNNs) along with estimates of their predictive uncertainty [61]. At test time, RNNs take as input the current trajectory of a target and predict its future waypoints along with corresponding confidence intervals [61]. These confidence intervals along with the discrete distribution d can be used to reason about task satisfaction. Assumptions (c) and (d) require prior information about the targets, so that initial paths can be designed. These can be relaxed by leveraging recently proposed exploration methods that can predict regions where semantic objects may be located in, based on the environmental context [62], [63]. This will also allow us to handle cases where any available prior information is utterly wrong. Our future work will focus on formally relaxing these assumptions.

IV. Experimental Validation

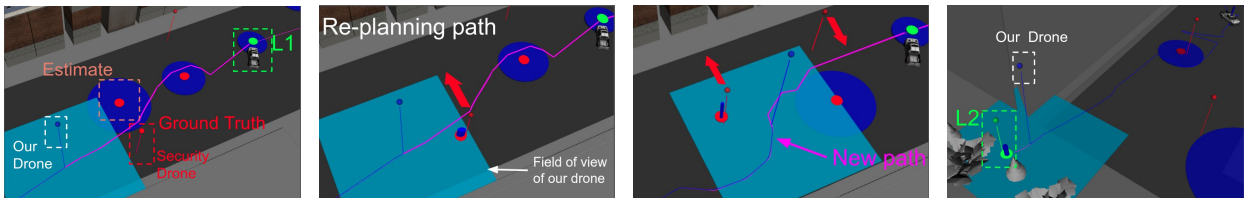
We present experiments to (i) demonstrate the performance of the proposed algorithm as the metric and semantic priors become more inaccurate and (ii) evaluate its scalability. To accelerate the construction of the trees, we employ the sampling functions developed in [8]. We conducted our experiments on Gazebo (ROS, python3) on a computer with Intel Core i5 - 8350U 1.7GHz and 16Gb RAM. Videos can be found in the supplemental material or in [64].

Experiment Setup: Our experiments involve AsTech Firefly Unmanned Aerial Vehicles (UAVs) being governed by first order linear dynamics. The UAV state includes the position, velocity, orientation, and biases in the measured angular velocities and acceleration; more details can be found in [8], [65]. The UAVs operate in a semantic city with dimensions $150\text{m} \times 150\text{m}$ consisting of M semantic targets. The set of available classes is $\mathcal{C} = \{$'Parked car', 'Person', 'Enemy recon drone', 'Enemy security drone'$\}$, unless otherwise specified. Targets with classes 'Parked car' and 'Person' are stationary (i.e., $\mathbf{x}_i(t+1) = \mathbf{x}_i(t)$), while the rest are moving in the environment to accomplish their own individual tasks. We assume that these mobile targets are also drones governed by linear dynamics as defined in [65], and they apply control inputs $\boldsymbol{\mu}_i(t)$ to either move back and forth along a straight line or track a circular trajectory. The covariance matrix $\mathbf{R}_i$ of the noise in the target dynamics is selected so that it has zero non-diagonal entries and diagonal entries equal to 0.2. Finally, we assume that the drones that we control are equipped with a downward facing sensor with square field of view with dimensions 24×24 m that can take noisy positional measurements of targets falling inside its field-of-view (see e.g., Fig. 1) as per the observation model in (3). In (3), we select $\mathbf{M}(\mathbf{p}(t))$ to be the identity matrix (as e.g., in [53]) and the covariance matrix of the measurement noise to have diagonal entries equal to 2. As for the object recognition method (4), we assume that our drones are equipped with a neural network that is capable of detecting objects and returning a discrete distribution over the available classes (see e.g. [56]). To this end, we have simulated a classifier with confusion matrix that has diagonal entries with values that range from 0.75 to 0.9; the remaining entries are generated randomly so that the sum of each row is equal to 1. Any other model for (4) can be used though.

Case Study I - Effect of Metric Uncertainty: First, we consider a single drone tasked with a surveillance mission in an environment with $M = 6$ semantic targets. Particularly, the drone must go into enemy territory and take photos of two specific targets, ℓ_1 (an abandoned car) and ℓ_2 (an enemy recon drone) in this order, while avoiding all targets belonging to 'Enemy security drone' class. All enemy drones fly at a fixed altitude of 8 meters, while the drone we control flies at at a fixed altitude of 16 meters. We assume that the enemy security drones are equipped with cameras allowing them to detect any object that lies within a 3D ball of radius of 9m. Thus, our drone has to always keep a distance of 9m in the xyz plane from them, or equivalently, 4m in the xy plane since all drones fly at fixed altitudes. This mission is captured by the following LTL formula: $\phi = \Diamond[\pi_1 \wedge \Diamond\pi_2] \wedge [\neg\pi_3 \mathcal{U} \pi_1] \wedge [\neg\pi_3 \mathcal{U} \pi_2]$, where $\pi_1 = \pi_p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{1, \ell_1, 2\text{m}, 0.25\})$, $\pi_2 = \pi_p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{1, \ell_2, 2\text{m}, , 0.25\})$ are defined as in (6) and $\pi_3 = \pi_p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{1, 4\text{m}, 0.9, \text{'Security'}\})$ is defined as in (7). All distances are computed based on the x, y coordinates. This LTL formula corresponds to a DFA with 4 states. In what follows, we illustrate the performance of the proposed algorithm for various prior Gaussian distributions. Specifically, we consider three configurations where in the first, second, and third setup, our estimated initial positions of all targets are off from their ground truth, by on an average 10m (see e.g., Fig. 1(a)), 4m, and 1m, respectively. The larger this deviation, the 'larger' the corresponding initial covariance matrices $\Sigma_i(0)$. For instance, the diagonal entries of $\Sigma_1(0)$ in the first, second, and third setup are $[4, 4]$, $[2, 2.1]$, and $[1, 0.8]$, respectively. Note that even if the initial covariances are large enough, the proposed algorithm designs informative paths that aim to actively decrease the metric uncertainty so that ϕ satisfied; this is a well-studied property in the related active sensing literature [45], [46]. After running each setup thrice, on average, replanning was triggered 5.33, 3, and 1.33 times in the first, second, and last setup respectively. Intuitively, better priors result in less frequent replanning. Snapshots of the UAV path for the first setup are shown in Fig. 1. The runtimes to design paths for the first run of each setup are shown in Fig. 2; observe that paths can be designed quite fast (< 1.5secs).

Case Study II - Effect of Semantic Uncertainty: Second,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.



(a) Planned path (b) New estimate of obstacle (c) New path generated (d) Mission accomplished

Fig. 1. Snapshots of the UAV trajectory for case study I. The targets ℓ_1 and ℓ_2 (car and recon drone) that should be visited are highlighted with green markers. The mobile enemy security drones that should always be avoided are highlighted with red markers. The blue ellipsoid denotes the covariance of the prior Gaussian distributions with means illustrated with red and green disks. The drone initially heads towards ℓ_1 ('car') (Fig. 1(a)). When the drone updates its belief about the position of the first security drone, it decides to revise its path as the current one, given the new estimates, violates the imposed probabilistic avoidance requirements; see Figs. 1(b)-1(c). Once ℓ_1 and ℓ_2 are approached the mission terminates; see Fig. 1(d).

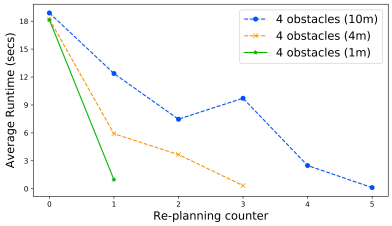


Fig. 2. Graphical depiction of re-planning frequency and average re-planning runtime for case study I. Note that run times when re-planning counter is 0 refers to the time needed to design the initial path. We note that these average runtimes are significantly smaller (< 1.5secs) when the algorithm is not executed in the Gazebo simulation environment.

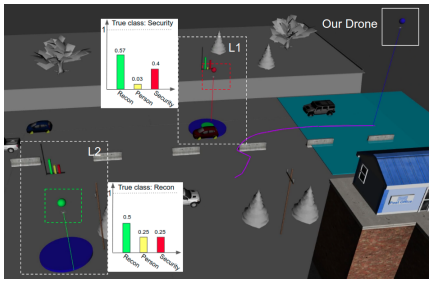


Fig. 3. Graphical depiction of the initial path (magenta) for case study II. Each target is illustrated by a red or green sphere depending on their actual label: red for 'security' and green for 'recon'.

we illustrate the performance of the proposed algorithm for various prior discrete distributions. To demonstrate this, we consider an environment with $M = 2$ targets and a set of classes defined as $\mathcal{C} = \{$'Enemy recon drone', 'Person', 'Enemy security drone'$\}$. Our goal is to control a single drone so that it achieves a task requiring to find and approach a target with class 'recon drone', while always avoiding all targets with class 'security drone'. This task is specified by the following LTL formula: $\phi = \Diamond\pi_1 \wedge [\neg\pi_2 \mathcal{U} \pi_1]$, where $\pi_1 = \pi_p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{1, 2\text{m}, 0.25, \text{'Recon'}\})$ and $\pi_2 = \pi_p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{1, 4\text{m}, 0.9, \text{'Security'}\})$ are defined as in (7). This LTL formula corresponds to a DFA with 3 states. The prior map is defined so that the discrete distribution for target ℓ_1 is 'incorrect'. Specifically, target ℓ_1 is a 'security drone' but its prior distribution is defined so that ℓ_1 is a 'recon drone' with probability 0.57 and a 'security drone' with probability 0.4; see Fig. 3. Also, target ℓ_2 is a 'recon drone' and its discrete distribution is defined so that ℓ_2 is a 'recon drone' with probability 0.5 and a 'security drone' with probability 0.25; see Fig. 3. Observe also that these semantic priors are not informative enough to satisfy the probabilistic requirements captured in the task. As a result, (9) is infeasible, and, therefore, the proposed algorithm cannot generate a path. To address this issue, in this case study, we aim to design least violating paths. To achieve this, when we build the tree to design the initial paths, instead of checking satisfaction of (7), we check satisfaction of a relaxed version of it. Specifically, we deterministically assign to each target the most likely class determined by d_i. Thus, in this example, both ℓ_1 (incorrectly) and ℓ_2 are considered 'recon drones'. Then, among all targets with a label c (e.g., 'recon drone'), we investigate satisfaction of the following predicate:$p(\mathbf{p}(t), \hat{\mathcal{M}}(t), \{j, r, \delta, c\}) = \max_{\ell_i,\ c_i=c}[\mathbb{P}(||\mathbf{p}_j(t) - \mathbf{x}_i(t)|| \leq r)] - (1 - \delta)$. Given this semantic prior, the first feasible path generated by our algorithm leads the drone towards ℓ_1; see Fig. 3. However, once the target is in sight, our robot updates d_1 and realizes that ℓ_1 is in fact a security drone. At this point, given this new semantic belief, the robot designs a new path that leads to ℓ_2, while avoiding ℓ_1. Note that, it is challenging to determine if the semantic priors are informative enough so that (9) is feasible. A possible approach to address this is to run the proposed algorithm with the original predicates for a large number of iterations. In parallel, the algorithm is run with the relaxed predicates in case the original problem is infeasible.

Case Study III - Multi-Robot Experiments: Third, we consider multi-robot teams in an environment with $M = 12$ targets. We consider the LTL task: $\phi = \Diamond(\xi_1 \wedge \Diamond\xi_2) \wedge \Diamond\xi_3 \wedge \Diamond\xi_4 \wedge \neg\xi_3 \mathcal{U} \xi_1 \wedge \neg\xi_4 \mathcal{U} \xi_2$, where each ξ_i is a Boolean formula defined over atomic predicates of the form (6) for various robots and landmarks. This LTL formula corresponds to a DFA with 14 states. For $N \in \{1, 2, 3, 5, 10\}$ robots, the time needed to design the initial feasible path is $\{0.01, 0.03, 0.21, 1.58, 7.35\}$ mins, respectively; these runtimes depend on the DFA size and the environmental structure. Notice that as N increases, these runtimes tend to increase. This may prevent application of the proposed algorithm for controlling large-scale multi-robot systems in rapidly-changing environments. A potential approach to mitigate this issue is to perform task decomposition, as e.g., in [26] which, however, may sacrifice completeness.

V. Conclusion

This paper addressed a multi-robot planning problem in uncertain and dynamic semantic environments. We proposed a sampling-based algorithm to design paths that are revised online based on perceptual feedback. We validated the proposed algorithm in complex semantic navigation tasks.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[2] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.

[3] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[4] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, June 2020, pp. 1689–1696.

[5] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *IEEE International Conference on Robotics and Automation*, Singapore, May-June 2017, pp. 1722–1729.

[6] V. Vasilopoulos, G. Pavlakos, S. L. Bowman, J. D. Caporale, K. Daniilidis, G. J. Pappas, and D. E. Koditschek, "Reactive semantic planning in unexplored semantic environments using deep perceptual feedback," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4455–4462, 2020.

[7] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press Cambridge, 2008, vol. 26202649.

[8] Y. Kantaros, S. Kalluraya, Q. Jin, and G. J. Pappas, "Perception-based temporal logic planning in uncertain semantic maps," *IEEE Transactions on Robotics*, 2022.

[9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[10] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[11] Y. Kantaros and M. M. Zavlanos, "Stylus*: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems," *International Journal of Robotics Research*, 2020.

[12] K. Leahy, D. Zhou, C.-I. Vasile, K. Oikonomopoulos, M. Schwager, and C. Belta, "Persistent surveillance for unmanned aerial vehicles subject to charging and temporal logic constraints," *Autonomous Robots*, vol. 40, no. 8, pp. 1363–1378, 2016.

[13] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005, pp. 2020–2025.

[14] M. Kloetzer and C. Belta, "Automatic deployment of distributed teams of robots from temporal logic motion specifications," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 48–61, 2010.

[15] V. Vasilopoulos, W. Vega-Brown, O. Arslan, N. Roy, and D. E. Koditschek, "Sensor-based reactive symbolic planning in partially known environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018, pp. 1–5.

[16] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," *arXiv preprint arXiv:1903.02531*, 2019.

[17] M. Ryll, J. Ware, J. Carter, and N. Roy, "Efficient trajectory planning for high speed flight in unknown environments," in *International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 732–738.

[18] H. Zhai, M. Egerstedt, and H. Zhou, "Path planning in unknown environments using optimal transport theory," *arXiv preprint arXiv:1909.11235*, 2019.

[19] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," *arXiv preprint arXiv:1910.08184*, 2019.

[20] 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 (ICRA)*, Karlsruhe, Germany, 2013, pp. 5025–5032.

[21] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local ltl specifications," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 218–235, 2015.

[22] M. R. Maly, M. Lahijanian, L. E. Kavraki, H. Kress-Gazit, and M. Y. Vardi, "Iterative temporal motion planning for hybrid systems in partially unknown environments," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*. ACM, 2013, pp. 353–362.

[23] M. Lahijanian, M. R. Maly, D. Fried, L. E. Kavraki, H. Kress-Gazit, and M. Y. Vardi, "Iterative temporal planning in uncertain environments with partial satisfaction guarantees," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 583–599, 2016.

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

[25] 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 *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4588–4595.

[26] Y. Kantaros, M. Malencia, V. Kumar, and G. J. Pappas, "Reactive temporal logic planning for multiple robots in unknown environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, June 2020, pp. 11 479–11 485.

[27] V. Vasilopoulos, Y. Kantaros, G. J. Pappas, and D. E. Koditschek, "Reactive planning for mobile manipulation tasks in unexplored semantic environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6385–6392.

[28] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. Di Caro, "Human-friendly robot navigation in dynamic environments," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 6-10, 2013, pp. 423–430.

[29] M. Otte and E. Frazzoli, "Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.

[30] A. Pierson, C.-I. Vasile, A. Gandhi, W. Schwarting, S. Karaman, and D. Rus, "Dynamic risk density for autonomous navigation in cluttered environments without object detection," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5807–5814.

[31] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[32] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 9590–9596.

[33] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 225–238.

[34] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *IEEE International Conference on Robotics and Automation*, Brisbane, Australia, May 2018, pp. 3399–3406.

[35] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic foundations of robotics X*. Springer, 2013, pp. 475–491.

[36] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[37] J. Tordesillas and J. P. How, "Panther: perception-aware trajectory planner in dynamic environments," *arXiv preprint arXiv:2103.06372*, 2022.

[38] M. Cleaveland, E. Yel, Y. Kantaros, I. Lee, and N. Bezzo, "Learning enabled fast planning and control in dynamic environments with intermittent information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2022. [Online]. Available: https://arxiv.org/abs/2209.04534

[39] A. Ulusoy and C. Belta, "Receding horizon temporal logic control in dynamic environments," *The International Journal of Robotics Research*, vol. 33, no. 12, pp. 1593–1607, 2014.

[40] B. Hoxha and G. Fainekos, "Planning in dynamic environments through temporal logic monitoring," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[41] D. Maity and J. S. Baras, "Motion planning in dynamic environments with bounded time temporal logic specifications," in *2015 23rd Mediterranean Conference on Control and Automation (MED)*. IEEE, 2015, pp. 940–946.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.

[42] Y. Li, E. M. Shahrivar, and J. Liu, "Safe linear temporal logic motion planning in dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 9818–9825.

[43] Z. Li, M. Cai, S. Xiao, and Z. Kan, "Online motion planning with soft metric interval temporal logic in unknown dynamic environment," *IEEE Control Systems Letters*, 2022.

[44] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8260881

[45] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *IEEE International Conference on Robotics and Automation*, Seattle, WA, 2015, pp. 4775–4782.

[46] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Sampling-based planning for non-myopic multi-robot information gathering," *Autonomous Robots*, vol. 45, no. 7, pp. 1029–1046, 2021.

[47] B. Charrow, V. Kumar, and N. Michael, "Approximate representations for multi-robot control policies that maximize mutual information," *Autonomous Robots*, vol. 37, no. 4, pp. 383–400, 2014. [Online]. Available: https://link.springer.com/article/10.1007/s10514-014-9411-2

[48] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[49] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/0278364914533443

[50] A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri, and R. Siegwart, "An incremental sampling-based approach to inspection planning: the rapidly exploring random tree of trees," *Robotica*, vol. 35, no. 6, pp. 1327–1340, 2017.

[51] M. Tzes, Y. Kantaros, and G. J. Pappas, "Distributed sampling-based planning for non-myopic active information gathering," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2021, pp. 5872–5877.

[52] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *arXiv preprint arXiv:2207.00254*, 2022.

[53] C. Freundlich, S. Lee, and M. M. Zavlanos, "Distributed active state estimation with user-specified accuracy," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 418–433, 2018.

[54] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[55] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[56] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[57] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Localization from semantic observations via the matrix permanent," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 73–99, 2016.

[58] A. Jones, M. Schwager, and C. Belta, "Distribution temporal logic: Combining correctness with quality of estimation," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 4719–4724.

[59] S. Haesaert, P. Nilsson, C. I. Vasile, R. Thakker, A.-a. Aghamohammadi, 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.

[60] "Multi-robot mission planning in dynamic semantic environments," https://arxiv.org/pdf/2209.06323.pdf.

[61] A. Alaa and M. Van Der Schaar, "Frequentist uncertainty in recurrent neural networks via blockwise influence functions," in *International Conference on Machine Learning*. PMLR, 2020, pp. 175–190.

[62] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis, "Learning to map for active semantic goal navigation," *arXiv preprint arXiv:2106.15648*, 2021.

[63] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis, "Uncertainty-driven planner for exploration and navigation," *arXiv preprint arXiv:2202.11907*, 2022.

[64] "Simulation videos: Multi-robot mission planning in dynamic semantic environments," https://vimeo.com/749441727.

[65] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:27 UTC from IEEE Xplore. Restrictions apply.