2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Uncertainty-Constrained Robot Exploration: A Mixed-Integer Linear Programming Approach

Luca Carlone and Daniel Lyons

***Abstract*— In this paper we consider the situation in which a robot is deployed in an unknown scenario and has to explore the entire environment without possibility of measuring its absolute position. The robot can take relative position measurements (from odometry and from place revisiting episodes) and can then estimate autonomously its trajectory. Therefore, the quality of the resulting estimate depends on the motion strategy adopted by the robot. The problem of *uncertainty-constrained exploration* is then to explore the environment while satisfying given bounds on the admissible uncertainty in the estimation process. We adopt a moving horizon strategy in which the robot plans its motion T steps ahead. Our formulation leads to a mixed-integer linear problem that has several desirable properties: (i) it guarantees that the robot motion is collision free, (ii) it guarantees that the uncertainty constraints are met, (iii) it enables the design of algorithms that efficiently solve moderately sized instances of the exploration problem. We elucidate on the proposed formulation with numerical experiments.**

I. INTRODUCTION

Numerous application endeavors, ranging from search and rescue to planetary exploration and surveillance, require the robot to explore an unknown environment. Exploration can be functional to provide the robot with the situational awareness needed to accomplish a given task (e.g., monitoring and surveillance) or can be useful to build a model of the environment (*map*) that supports human intervention (e.g., for search and rescue). In presence of absolute localization information (e.g., GPS), the exploration process reduces to choose the exploration targets that maximize the opportunity of visiting unknown areas, see, e.g., [18]. Related approaches can be found in [2], [8], while examples of generalizations to the multi robot case are [3], [9].

When no absolute localization service is available to the robot, the exploration task, which can be considered a decisional process, needs to be accompanied by the concurrent estimation of robot positions (and possibly a map of the environment). This estimation process is usually referred to as *Simultaneous Localization and Mapping* (SLAM). While the maturity of SLAM has been recognized by the robotic community, the problem of exploration under uncertainty still remains an open issue. In literature, the problem is sometimes referred to as *active SLAM and exploration* and the contributions are usually tailored to a particular technique used for position estimation (e.g., Extended Kalman Filter). An early contribution to active SLAM and exploration with EKF is proposed by Feder et al. [7]. A model predictive control (MPC) strategy, associated with EKF-SLAM, is introduced by Huang et al. [10]; an attractor-based heuristic is associated to MPC in [12]. Kollar and Roy propose the use of a reinforcement learning approach in [11]. Sim and Roy address the problem using ideas from A-optimal experimental design in [16]. Martinez-Cantin et al. use a simulation-based active policy learning algorithm [14]. In [13], instead, the problem is solved by applying a Bayesian method that allows reducing the number of simulations needed for planning. More recently, a lucid treatment of EKF-based active SLAM is provided in [5], which also discusses the use of different uncertainty metrics.

While it is common to most part of the mentioned papers to frame the planning problem in terms of mathematical optimization, several issues arise from the literature review. In most EKF-based approaches there is no explicit modelling of the obstacles, therefore, the feasibility of the planned trajectory is not guaranteed in general. Moreover, the objective function of the corresponding optimization problem is often intangible and can be only evaluated at few points, as it happens in simulation-based approaches, e.g., [1], [14]. Furthermore, the objective function is usually an additive cost comprising (at least) two summands [4]: a cost related to exploration and a cost related to uncertainty reduction; the weights assigned to these costs are scenario-dependant and difficult to assign in general. As a consequence, the solution is not guaranteed to be optimal in general, and solving the problem is computationally demanding.

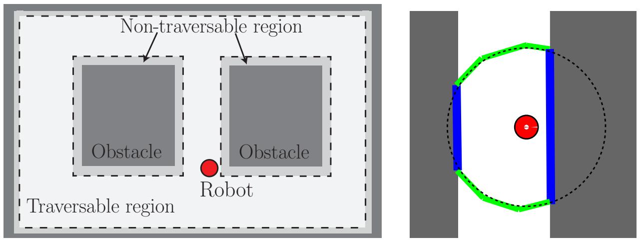
In this work we adopt a slightly different perspective on the estimation aspect of the problem: we consider a simple linear framework to estimate robot's positions; this enables a better understanding of the structure of the matrices describing the estimation error and allows quantifying the estimation uncertainty without recurring to simulations. Moreover, we explicitly model the requirement that robot motion has to be collision free, hence our planning strategy is guaranteed to produce feasible trajectories over a given planning horizon. Our formulation leads to a mixed-integer linear problem (MILP), that can be solved with reasonable computational effort in practice, and has been already demonstrated to be an effective solution for planning collision-free trajectories (an application to spacecrafts' navigation can be found in [15]). Although MILPs are NP-hard to solve in the worst case, they are known to be tractable for moderate problem sizes or by employing problem-specific heuristics.

The structure of the paper is as follows: uncertainty-constrained exploration is formalized in Section II, where we specify the requirements that guide autonomous exploration. In Section III we rewrite these requirements in mathematical terms. Finally, we present numerical results in Section IV.

II. PROBLEM FORMULATION

A disc-shaped mobile robot $\mathcal{R}$ is deployed at time $t = 0$ in an unknown bounded environment $\mathcal{S} \subset \mathbb{R}^d$ (*scenario*),

L. Carlone is with the College of Computing, Georgia Institute of Technology, USA. luca.carlone@gatech.edu
D. Lyons is with the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology, Germany. daniel.lyons@kit.edu



(a) (b)

Fig. 1. (a) An example of scenario $\mathcal{S}$ with the robot depicted as a red dot, the traversable region $\mathcal{T}$ in white, the obstacles $\mathbf{O}$ in dark grey, and the non traversable region $\mathcal{O}$ delimited by a dashed line. (b) Zoomed in view of the robot, its sensing radius $\mathbf{r}_{\mathrm{SR}}$ (dashed circle), and the local visited region with frontiers (in green) and obstacle boundaries (in blue).

with $d \in \{2,3\}$ (planar or three-dimensional scenario). Denoting with $x_t \in \mathbb{R}^d$ the position of the robot at time t, we will interpret $\mathcal{R}(x_t)$ as the set of points in $\mathbb{R}^d$ that belongs to the robot when it is in position x_t. In particular, if $\mathbf{r}_{\mathrm{R}}$ is the radius of the (holonomic) robotic platform, $\mathcal{R}(x_t) = \mathcal{B}(x_t, \mathbf{r}_{\mathrm{R}})$, where $\mathcal{B}(x_t, \mathbf{r}_{\mathrm{R}})$ is the Euclidean ball of radius $\mathbf{r}_{\mathrm{R}}$ centered on x_t. In the environment $\mathcal{S}$ we can distinguish a set of obstacles $\mathbf{O}$ and a connected obstacle-free region $\mathbf{T}$. Notice that the robot cannot assume all positions within $\mathbf{T}$: if the center x_t of the robot is closer than $\mathbf{r}_{\mathrm{R}}$ to an obstacle, then the robot is touching the obstacle. Therefore, it is convenient to define a (possibly non-connected) *non traversable region* $\mathcal{O}$ (points in the environment that lead the robot to touch an obstacle), and a *traversable region* $\mathcal{T}$, which is a connected set of collision-free positions. Clearly, it holds $\mathcal{T} = \mathbf{T} \ominus \mathcal{B}(\mathbf{r}_{\mathrm{R}})$ and $\mathcal{O} = \mathbf{O} \oplus \mathcal{B}(\mathbf{r}_{\mathrm{R}})$, where $\mathcal{B}(\mathbf{r}_{\mathrm{R}})$ is the ball $\mathcal{B}(\mathbf{0}_d, \mathbf{r}_{\mathrm{R}})$, and $\oplus$ and $\ominus$ are the Minkowski sum and difference among sets, respectively. Note that $\mathbf{T} \cup \mathbf{O} = \mathcal{T} \cup \mathcal{O} = \mathcal{S}$, see Fig. 1(a).

After travelling for t discrete time steps the robot describes the positions (discrete trajectory) $x_{0:t} = [x_0^\top \ \ldots \ x_t^\top]^\top$; at each time ($i = 0, \ldots, t-1$) the robot applies a motion *command* $\bar{z}_{i,i+1} \in \mathbb{R}^d$ to its locomotion system, which corresponds to the desired displacement between the current position and the next position. However, because of actuation noise, the actual motion of the robot is $z_{i,i+1} \doteq x_{i+1} - x_i \neq \bar{z}_{i,i+1}$; equivalently, we may consider $\bar{z}_{i,i+1}$ as a noisy measurement of the displacement between consecutive positions (*odometry*). By assumption the commands are limited (since the robot travels at limited speed), therefore, $\|\bar{z}_{i,i+1}\| \leq \mathbf{r}_{\mathrm{V}}$ for all i, with $\mathbf{r}_{\mathrm{V}}$ being a known bound (*maximum speed*).

While moving, the robot also acquires exteroceptive sensor measurements. Exploiting exteroceptive sensors, we assume that the robot is able to recognize an already visited place (*loop closing*): if at time t the robot revisits the same place observed from position x_i, $i < t$, it is able to take a noisy measurement of the relative position $z_{t,i} \doteq x_i - x_t$, by comparing the current perception with the sensor reading taken at time i. We call $\bar{z}_{t,i}$ the relative position measurement. Common techniques for measuring $\bar{z}_{t,i}$ are scan matching and vector registration. Clearly, in order to measure $\bar{z}_{t,i}$, the position x_t should be close enough to the previous position x_i. We assume that a loop closing is possible if $\|x_i - x_t\| \leq \mathbf{r}_{\mathrm{LC}}$, where $\mathbf{r}_{\mathrm{LC}}$ is a known *loop closing radius*. We take the following assumption on measurement noise.

Assumption 1 (Unknown-but-bounded (UBB) noise)**.** *A generic measurement $\bar{z}_{i,j}$ (being it a command or a loop closing) can be written as $\bar{z}_{i,j} = x_j - x_i + \epsilon_{i,j}$, with $\epsilon_{i,j}^\top (P_{i,j}^z)^{-1} \epsilon_{i,j} \leq 1$. Moreover, we assume that $P_{i,j}^z = (\sigma_{i,j}^2 \mathbf{I}_d)$, i.e., the noise is bounded within a ball of radius $\sigma_{i,j}$.* □

The previous assumption is taken for simplicity, although a similar derivation can be obtained assuming a probabilistic setup; for instance one may consider zero mean Gaussian noise and then consider an α-confidence ellipsoid that, with probability α, contains the measurement noise, leading to a (probabilistic) bounded-noise setup. Similarly, if we have a generic ellipsoidal set $\epsilon_{i,j}^\top (P_{i,j}^z)^{-1} \epsilon_{i,j} \leq 1$, we can compute a bounding ball containing the ellipsoidal set and then we can write $\epsilon_{i,j}^\top (\sigma_{i,j}^2 \mathbf{I}_d)^{-1} \epsilon_{i,j} \leq 1$ for a suitable $\sigma_{i,j}$.

When the robot is in position x_t, it can sense a surrounding region $\mathcal{V}(x_t)$ (*local visited region*), see Fig. 1(b). The shape of the set $\mathcal{V}(x_t)$ is sensor-dependent and, in general, satisfies the condition $\mathcal{V}(x_t) \subseteq \mathcal{B}(x_t, \mathbf{r}_{\mathrm{SR}})$, where $\mathcal{B}(x_t, \mathbf{r}_{\mathrm{SR}})$ is the Euclidean ball with radius $\mathbf{r}_{\mathrm{SR}} > 0$ (*sensing radius*), and center x_t. We will denote the overall *visited region* with $\mathcal{V}(x_{0:t}) = \cup_{i=0}^{t} \mathcal{V}(x_i)$.

Assumption 2. *For each time $t = 0, 1, \ldots$, the local visited region $\mathcal{V}(x_t)$ is a polygon (not necessarily the same at all times), whose vertices are exactly known with respect to the position x_t.* □

The polygonal representation of Assumption 2 essentially comes without loss of generality, since we do not specify any shape or number of edges (or convexity properties of the polygon). The hypothesis of accurate knowledge of the polygon w.r.t. the corresponding position x_t is only taken for simplicity and it may be relaxed; as we will see in a while in our setting are the positions themselves to be uncertain. As a consequence of Assumption 2, along the boundary of each local visited region, we can distinguish a collection of line segments, that can be classified as either *obstacle boundaries* or *frontiers*, see Fig. 1(b).

We now introduce few definitions to formally characterize the exploration problem.

Definition 1 (Collision-free positions)**.** *A set of positions $x_{0:t} = [x_0^\top \ \ldots \ x_t^\top]^\top$ is said to be collision-free if $x_i \in \mathcal{T}$, for all $i \in \{0, \ldots, t\}$.* □

For safe operation, the robot has to travel along collision-free positions. In our case, the robot has to satisfy this requirement while exploring an unknown scenario $\mathcal{S}$. We assume that the robot has no prior knowledge about the scenario. Then, after deployment, it has to apply suitable motion strategies to visit the entire environment. The exploration process ends when it is not possible to further expand the sensed areas.

Definition 2 (Exploration complete)**.** *The exploration process is said to be complete at time t, if $\mathcal{V}(x_{0:t}) \cup \mathcal{V}(\bar{x}) = \mathcal{V}(x_{0:t})$, for any position $\bar{x} \in \mathcal{T}$ (i.e., no new observation can enlarge the visited area).* □

Definitions 1 and 2 are common to other works in which the exploration process is carried out with accurate knowledge of the positions $x_{0:t}$ assumed by the robot. Now, instead, we characterize the concept of uncertainty-constrained exploration. In this work we assume that $x_{0:t}$ is unknown, and we only have an estimate $\hat{x}_{0:t}$ of the positions, together with a matrix describing the estimation error. How this estimate is obtained from the available measurements is discussed in Section II-A. We notice that the robot, at each time step t, knows the command $\bar{z}_{t,t+1}$. However, the possibility of acquiring loop closing measurements depends on the motion strategy of the robot (i.e., if it revisits known places or not). Therefore, the (position) estimation and the exploration problem are interdependent, contrarily to the case in which direct measurements of robot positions are taken (e.g. from GPS). For this reason, here we want to design an algorithm that allows the robot to complete the exploration while satisfying given bounds on the estimation error.

Problem 1 (Uncertainty-constrained exploration, `UCE`)**.** *The robot $\mathcal{R}$ has to completely explore the scenario $\mathcal{S}$, moving along collision-free positions, while satisfying a given constraint on the admissible uncertainty of position estimation.*

A. Preliminaries on trajectory estimation

As discussed in the previous section the available measurements come from commands $\bar{z}_{i,i+1}$, $i = 0, \ldots, t-1$, and loop closing measurements $\bar{z}_{j,i}$ between position j and position i; in both cases the measurements evaluate the difference between two positions. Graph formalism provides a natural model for the problem at hand: we consider a graph $\mathcal{G}_{0:t}$ (*position graph*), where the vertex (or node) set $\mathcal{N}_{0:t}$ is $\{0, 1, \ldots, t\}$, and the edge set $\mathcal{E}_{0:t}$ is the unordered set of pairs (i, j) such that if $(i, j) \in \mathcal{E}_{0:t}$, then a measurement in the form $\bar{z}_{i,j}$ was acquired in the time interval $[0, t]$. Therefore, the position estimation problem reduces to assign a position estimate $\hat{x}_i$ to each node $i \in \mathcal{N}_{0:t}$. Since, the robot is only able to measure relative positions between node pairs, nodes' position can be only defined up to an arbitrary roto-translation. For avoiding this ambiguity, the first node is set to the origin of the reference frame, i.e., x_0 is conventionally set to $x_0 = \mathbf{0}_d$. Therefore, the estimand is the vector $x_{1:t} \in \mathbb{R}^{dt}$. We label the available measurements from 1 to m and we define the *measurement vector* $\bar{z} = [\bar{z}_1^\top \ \ldots \ \bar{z}_m^\top]^\top \in \mathbb{R}^{dm}$. We also define the matrix $\bar{P}_z = \mathrm{diag}(\bar{P}_1^z, \ldots, \bar{P}_m^z) \in \mathbb{R}^{dm \times dm}$, and the corresponding inverse $\bar{\Omega}_z = \bar{P}_z^{-1}$. Then the available measurements can be rewritten as:

$$\bar{z} = \bar{A}^\top x_{1:t} + \epsilon \tag{1}$$

where ϵ is a bounded noise, satisfying $\epsilon^\top \bar{\Omega}_z \epsilon \leq 1$, while $\bar{A} = A \otimes \mathbf{I}_d$, where A is the *reduced incidence matrix* of graph $\mathcal{G}$, $\otimes$ is the Kronecker product, and $\mathbf{I}_d$ is the identity matrix of size d. The reduced incidence matrix is the graph incidence matrix without the first row, corresponding to the node that was set to $\mathbf{0}_d$. The effect of the Kronecker product is to substitute the entries -1 and $+1$ of the incidence matrix with $-\mathbf{I}_d$ and $+\mathbf{I}_d$, respectively. We assume that the BLUE (Best Linear Unbiased Estimator)[1] is used for estimating $x_{1:t}$ from (1):

$$\hat{x}_{1:t} = (\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}\bar{A}\bar{\Omega}_z\bar{z}. \tag{2}$$

From Assumption 1 we can write $\bar{\Omega}_z \doteq \mathrm{diag}(\frac{1}{\sigma_1^2}\mathbf{I}_d, \ldots, \frac{1}{\sigma_m^2}\mathbf{I}_d)$. Let us define $\Omega_z \doteq \mathrm{diag}(\frac{1}{\sigma_1^2}, \ldots, \frac{1}{\sigma_m^2})$, then it holds $\bar{\Omega}_z = \Omega_z \otimes \mathbf{I}_d$. Since $\bar{A} = A \otimes \mathbf{I}_d$, we may exploit the inverse and the mixed-product property of the Kronecker product to write:

$$(\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1} = (A\Omega_z A^\top)^{-1} \otimes \mathbf{I}_d \tag{3}$$

This equality will be useful later, since $(\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}$ and $(A\Omega_z A^\top)^{-1}$ play an important role in the estimation process, according to the following proposition.

Proposition 1. *The following facts hold true when using the BLUE for estimating robot trajectory in the UBB setup:*

1) *The trajectory estimation error $\epsilon_{1:t} \doteq \hat{x}_{1:t} - x_{1:t}$ satisfies $\epsilon_{1:t}^\top \bar{P}_{1:t}^{-1} \epsilon_{1:t} \leq 1$, with $\bar{P}_{1:t} \doteq (\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}$;*
2) *The error in the estimation of the i-th robot position $\epsilon_i \doteq \hat{x}_i - x_i$ satisfies $\|\epsilon_i\|^2 \leq \sigma_i^2$, where σ_i^2 is the i-th diagonal element of $P_{1:t} \doteq (A\Omega_z A^\top)^{-1}$.*

Proof: The first fact can be proven by substituting the measurement model (1) into the expression of the BLUE:

$$\hat{x}_{1:t} = (\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}\bar{A}\bar{\Omega}_z\bar{z} =$$
$$(\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}\bar{A}\bar{\Omega}_z(\bar{A}^\top x_{1:t} + \epsilon) = x_{1:t} + \epsilon_{1:t}.$$

with $\epsilon_{1:t} \doteq (\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}\bar{A}\bar{\Omega}_z\epsilon$. Now, recalling that $\epsilon^\top\bar{\Omega}_z\epsilon \leq 1$ it is easy to demonstrate that $\epsilon_{1:t}^\top \bar{P}^{-1}\epsilon_{1:t} \leq 1$ ($\epsilon_{1:t}$ is a linear combination of ϵ), with $\bar{P}_{1:t} \doteq (\bar{A}\bar{\Omega}_z\bar{A}^\top)^{-1}$, which proves the first claim. The second claim stems from a well known property of ellipsoidal sets: if $\epsilon_{1:t}^\top \bar{P}^{-1}\epsilon_{1:t} \leq 1$ and we consider a subvector of $\epsilon_{1:t}$, say ϵ_i, corresponding to the estimation error for the i-th position, then $\epsilon_i^\top \bar{P}_i^{-1}\epsilon_i \leq 1$ where $\bar{P}_i$ is the submatrix of $\bar{P}_{1:t}$ corresponding to the subvector ϵ_i; in a probabilistic setup this would correspond to marginalizing ϵ_i from $\epsilon_{1:t}$. Now, recalling (3), we also have $\bar{P}_i = \sigma_i^2\mathbf{I}_d$, where σ_i^2 is the i-th diagonal element of $P_{1:t}$. Therefore,

$$\epsilon_i^\top \bar{P}_i^{-1}\epsilon_i \leq 1 \iff \epsilon_i^\top(\sigma_i^2\mathbf{I}_d)^{-1}\epsilon_i \leq 1 \iff \epsilon_i^\top\epsilon_i \leq \sigma_i^2 \tag{4}$$

that concludes the proof. ∎

We call $\bar{P}_{1:t}$ the *uncertainty matrix* and $P_{1:t}$ the *reduced uncertainty matrix*. When convenient, we adopt the equivalent notation $\bar{\Omega}_{1:t} = \bar{P}_{1:t}^{-1}$ (*information matrix*) and $\Omega_{1:t} = P_{1:t}^{-1}$ (*reduced information matrix*). With the notation introduced so far we can take the last assumption required for our derivation.

Assumption 3. *Let $t = 0, 1, \ldots$, and define $\sigma_{\max}^2(t) \doteq \max_{i=\{1,\ldots,t\}} \sigma_i^2$ as the maximum diagonal entry of $P_{1:t}$, and σ_{cmd}^2 as the maximum actuation noise, i.e., $\sigma_{cmd}^2 \doteq \max_{i=\{0,1,\ldots\}} \sigma_{i,i+1}^2$. We assume that, given a planning horizon T, $\mathbf{r}_{\mathrm{SR}} > \sigma_{\max}(t) + \mathbf{r}_{\mathrm{R}} + \sqrt{\sigma_{\max}^2(t) + T\sigma_{cmd}^2}$, and $\mathbf{r}_{\mathrm{LC}} > \sqrt{2\sigma_{\max}^2(t) + T\sigma_{cmd}^2}$, for all t.*

The assumption requires that the sensing radius and the loop closing radius are bigger than the maximum accumulated estimation error. Also in this case the assumption is not strict since our objective is to bound the estimation error, which in practice remains smaller than the two radii.

B. Uncertainty metrics

In the rest of the paper we will be interested in constraining some metrics related to the uncertainty matrix $\bar{P}_{1:t}$. Examples of uncertainty metrics are: (i) the determinant $\det(\bar{P}_{1:t})$, (ii) the maximum eigenvalue $\lambda_{\max}(\bar{P}_{1:t})$, (iii) the trace $\mathrm{tr}(\bar{P}_{1:t})$, (iv) the maximum diagonal entry

[1] We postulate the use of the BLUE estimator, for two main reasons: (i) it enables extensions to the probabilistic setup, and (ii) it can be efficiently computed in practice, using both iterative and batch algorithms.

$\sigma^2_{\max}(\bar{P}_{1:t})$, (v) the i-th diagonal entry $\sigma^2_i(\bar{P}_{1:t})$. Exploiting relation (3) we notice that we can infer the previous metrics directly from the reduced uncertainty matrix $P_{1:t}$, since $\bar{P}_{1:t}$ is only an augmented version of $P_{1:t}$. In particular, the following equivalences hold (they are obtained from basic properties of the Kronecker product): (i) $\det(\bar{P}_{1:t}) = \det(P_{1:t})^d$, (ii) $\lambda_{\max}(\bar{P}_{1:t}) = \lambda_{\max}(P_{1:t})$, (iii) $\text{trace}(\bar{P}_{1:t}) = d \cdot \text{trace}(P_{1:t})$, (iv) $\sigma^2_{\max}(\bar{P}_{1:t}) = \sigma^2_{\max}(P_{1:t})$, (v) $\sigma^2_{d(i-1)+1}(\bar{P}_{1:t}) = \sigma^2_{d(i-1)+2}(\bar{P}_{1:t}) = \ldots = \sigma^2_{d(i-1)+d}(\bar{P}_{1:t}) = \sigma^2_i(P_{1:t})$. Therefore, we can bound the metric on $P_{1:t}$, instead of the corresponding metric applied to $\bar{P}_{1:t}$. The advantage is that $P_{1:t}$ has a simpler structure, as discussed in Section III-C. Moreover, according to Proposition 1, the diagonal entries of $P_{1:t}$ have a very intuitive meaning: the i-th diagonal entry is the square of the radius of the Euclidean ball bounding the estimation error for the i-th position.

III. Uncertainty-Constrained Exploration

In the following, we adopt a standard *receding horizon planning* strategy, in which the robot at time t (*decision time*), has to decide an optimal (in a sense discussed later) motion strategy over a finite time horizon consisting of T steps. A motion strategy is encoded in a collection of *waypoints* $\chi_{t+1}, \ldots, \chi_{t+T}$ the robot has to reach in order to maximize the explored areas, while satisfying the constraints. Notice that deciding over the variables $\chi_{t+1}, \ldots, \chi_{t+T}$ is the same as deciding over the commands $\bar{z}_{t,t+1}, \ldots, \bar{z}_{t+T-1,t+T}$: indeed $\chi_{t+k} = \hat{x}_t + \sum_{i=t}^{t+k-1} \bar{z}_{i,i+1}$, for $k \in \{1, \ldots, T\}$; we here prefer to adopt the former parametrization, since the waypoints have the more natural interpretation of "desired positions the robot wants to reach". Since the robot has no knowledge about the scenario outside of its visited region, his goal for exploration can only be to extend the visited region as quickly as possible. Therefore, an optimality criterion for a motion strategy is the following.

Definition 3 (Optimal motion strategy over the horizon T)**.** *A motion strategy $\chi_{t+1}, \ldots, \chi_{t+T}$ is optimal if it satisfies problem constraints (uncertainty bounds and collision-free trajectories) and leads the robot to a frontier of the visited regions in the smallest number of steps, i.e., if χ_{t+k} is on a frontier of the visited region there exists no other feasible plan that reaches a frontier at time $j < k$.* □

We now present a pseudo-optimization program that summarizes the planning problem the robot has to solve at time t:

P[`UCE`] :			
variables :	$\chi_{t+1}, \ldots, \chi_{t+T}$		
objective :	reach a frontier in the smallest number of time steps	(i) explore the environment	(5)
s.t. :	$\|\chi_{t+k} - \chi_{t+k-1}\| \leq \mathbf{r}_{\mathrm{V}}$, $k \in \{1, \ldots, T\}$,	(ii) respect speed limit	(6)
	$x_{t+k} \in \mathcal{T}$, $k \in \{1, \ldots, T\}$,	(iii) follow collision-free trajectories	(7)
	$\texttt{size}(P_{1:t+T}) < \bar{\mathbf{U}}$	(iv) satisfy uncertainty constraints	(8)

where $\texttt{size}(\cdot)$ is a function measuring the uncertainty of the trajectory estimate, chosen among the metrics proposed in Section II-B and $\bar{\mathbf{U}}$ is a given uncertainty bound. We set $\chi_t = \hat{x}_t$, i.e., the plan starts from the current position estimate.

The constraint (6) is convex and is already expressed in mathematical form, hence it does not need further manipulations. The constraints (7) and (8), and the objective (5) need further elaboration. For instance, constraint (7) requires the knowledge of the traversable region $\mathcal{T}$ and the future position of the robot x_{t+k}, that are unknown in practice. Similar complications emerge for constraint (8), which requires the knowledge of $P_{1:t+T}$, that is only known a-posteriori (i.e., after the motion steps $\chi_{t+1:t+T}$ are applied). The role of the following sections is to formalize the constraints (7) and (8), and the objective (5).

A. Constraint (7): Collision-free Trajectory

In this section we show how the robot, thus having uncertain knowledge of its own position and, hence, only an estimate of obstacles' position, can still plan collision-free trajectories.

1) Estimating a subset of the traversable region: Proposition 1 guarantees that the estimation error of robot position at time t, $\epsilon_t = \hat{x}_t - x_t$, satisfies $\|\epsilon_t\| \leq \sigma_t$. This means that the true position of the robot at time t lies within a radius σ_t around the estimated position. We also recall that, at time t, the robot senses the local visited region $\mathcal{V}$ and can construct the *estimated local visited region* $\mathcal{V}(\hat{x}_t)$ (which is built on the estimated position, since x_t is unknown).

Let us define the set of points in $\mathcal{V}(\hat{x}_t)$ that have a distance greater than or equal to σ from the boundary of $\mathcal{V}(\hat{x}_t)$:

$$\mathcal{V}(\hat{x}_t) \ominus \mathcal{B}(\sigma) \doteq \{x \in \mathcal{V}(\hat{x}_t) \colon \mathcal{B}(x, \sigma) \subset \mathcal{V}(\hat{x}_t)\} \ . \quad (9)$$

Lemma 1. *For $t = 0, 1, \ldots$, it holds that $\mathcal{V}(\hat{x}_t) \ominus \mathcal{B}(\sigma_t) \subset \mathcal{V}(x_t)$, i.e., when we "scale down" the estimated local visited region by the uncertainty in the position estimate, it will be contained in the true visited region.*

Proof: The error made from $\mathcal{V}(\hat{x}_t)$ to $\mathcal{V}(x_t)$ is a translational error, i.e., there exists a fixed vector z such that $\mathcal{V}(x_t) = \mathcal{V}(\hat{x}_t) \oplus z \doteq \{x + z \colon x \in \mathcal{V}(\hat{x}_t)\}$. Then it holds that there exists a z such that $x_t = \hat{x}_t + z$ with $\|z\| = \|x_t - \hat{x}_t\| \leq \sigma_t$ by Proposition 1. Let $x^* \in \mathcal{V}(\hat{x}_t) \ominus B(\sigma_t)$. For x^* it holds that for all $\tilde{z}$ with $\|\tilde{z}\| \leq \sigma_t$ $x^* + \tilde{z} \in \mathcal{V}(\hat{x}_t)$. Hence, also $x^* + z \in \mathcal{V}(\hat{x}_t)$ but it also holds that $x^* + z \in \mathcal{V}(\hat{x}_t) \oplus z = \mathcal{V}(x_t)$. ■

This is an important statement since, by definition, $\mathcal{V}(x_t) \subset \mathbf{T}$, i.e., the true local visited region is a subset of the obstacle-free region, and therefore $\mathcal{V}(x_t) \ominus B(\mathbf{r}_{\mathrm{R}}) \subset \mathcal{T}$; therefore, the lemma can guarantee that, even under estimation errors, a subset of the estimated local visited region is included in the traversable region. Therefore, a simple way to build a subset of the traversable region is provided by the following proposition.

Proposition 2. *Define the (estimated) traversable subregion:*

$$\hat{\mathcal{T}}(t) = \bigcup_{i=0:t} \mathcal{V}(\hat{x}_i) \ominus B(\sigma_i + \mathbf{r}_{\mathrm{R}}) \quad (10)$$

where σ_0 is conventionally set to 0 (no uncertainty of the position x_0, i.e., $\hat{x}_0 = x_0$). Then, it holds true that $\hat{\mathcal{T}}(t)$ is a non-empty polygon and $\hat{\mathcal{T}}(t) \subset \mathcal{T}$.

Proof: From Assumption 2, $\hat{\mathcal{T}}(t)$ is a polygon (each estimated local visited region $\mathcal{V}(\hat{x}_i)$ is a polygon, and remains a polygon after applying the Minkowski difference and the union). Moreover, Assumption 3 assures that each set $\mathcal{V}(\hat{x}_i) \ominus B(\sigma_i + \mathbf{r}_\mathrm{R})$ is non-empty, since $\mathbf{r}_\mathrm{SR} > \sigma_{\max}(t) + \mathbf{r}_\mathrm{R} + \sqrt{\sigma_{\max}^2(t) + T\sigma_{\mathrm{cmd}}^2} \geq \sigma_i + \mathbf{r}_\mathrm{R}$. The second claim follows from Lemma 1: for each $i = 0, \ldots, t$, it holds $\mathcal{V}(\hat{x}_i) \ominus B(\sigma_i) \subset \mathcal{V}(x_i)$, which implies $\mathcal{V}(\hat{x}_i) \ominus B(\sigma_i + \mathbf{r}_\mathrm{R}) \subset \mathcal{T}$; since $\hat{\mathcal{T}}(t)$ is the union of subsets of $\mathcal{T}$, it is itself a subset of $\mathcal{T}$. ∎

In the previous proposition we ensured that $\hat{\mathcal{T}}(t)$ is a subset of the traversable region $\mathcal{T}$. In the next proposition we show that a motion strategy for that the planned positions $\chi_{t+1:t+T}$ lie within a certain subset of $\hat{\mathcal{T}}(t)$ is guaranteed to produce collision-free positions $x_{t+1:t+T}$, satisfying constraint (7). We remark that the mismatch between $\chi_{t+1:t+T}$ and $x_{t+1:t+T}$ is a consequence of actuation noise.

Proposition 3. *Assume that the robot at time step t computes the plan $\chi_{t+1:t+T}$, and use the plan to apply the motion commands for the successive time steps $t, t+1, \ldots, t+T-1$. Then, it holds $\|\delta_{t+k}\| \doteq \|\chi_{t+k} - x_{t+k}\| \leq \sqrt{\sigma_t^2 + \sum_{i=t}^{t+k-1} \sigma_{i,i+1}^2}$. Moreover, if χ_{t+k}, $k \in \{1, \ldots, T\}$, is such that $\chi_{t+k} \in \hat{\mathcal{W}}(t+k) \doteq \hat{\mathcal{T}}(t) \ominus \mathcal{B}(\sqrt{\sigma_t^2 + \sum_{i=t}^{t+k-1} \sigma_{i,i+1}^2})$ then $x_{t+k} \in \mathcal{T}$, i.e., the actual robot positions at time $t+k$ is collision free.*

Proof: Consider a $k \in \{1, \ldots, T\}$. We observed at the beginning of Section III that we can write $\chi_{t+k} = \hat{x}_t + \sum_{i=t}^{t+k-1} \bar{z}_{i,i+1}$. Now we write the estimate and the commands in terms of actual quantities plus noise: $\chi_{t+k} = x_t + \epsilon_t + \sum_{i=t}^{t+k-1}(x_{i+1} - x_i + \epsilon_{i,i+1})$. The sum of the true (unknown) positions can be simplified as follows: $\chi_{t+k} = x_{t+k} + \epsilon_t + \sum_{i=t}^{t+k-1} \epsilon_{i,i+1}$, which demonstrates that the desired position χ_{t+k} and the true position x_{t+k} may only differ by $\delta_{t+k} \doteq \epsilon_t + \sum_{i=t}^{t+k-1} \epsilon_{i,i+1}$. Recalling Assumption 1 and Proposition 1 we have that the linear combination δ_{t+k} satisfies $\delta_{t+k}^\top \left((\sigma_t^2 + \sum_{i=t}^{t+k-1} \sigma_{i,i+1}^2)\mathbf{I}_d\right)^{-1} \delta_{t+k} \leq 1$, which implies $\|\chi_{t+k} - x_{t+k}\| = \|\delta_{t+k}\| \leq \sqrt{\sigma_t^2 + \sum_{i=t}^{t+k-1} \sigma_{i,i+1}^2}$. Therefore, considering a $\chi_{t+k} \in \hat{\mathcal{W}}(t+k)$ guarantees that the actual x_{t+k} lies within $\hat{\mathcal{T}}(t)$, which is a subset of $\mathcal{T}$ by Proposition 2, and then is collision free. ∎

2) Stay in the traversable subregion: In the previous section we demonstrated the implication $\chi_{t+k} \in \hat{\mathcal{W}}(t+k) \Longrightarrow x_{t+k} \in \mathcal{T}$, $k \in \{1, \ldots, T\}$. In this section, we formalize the constraint $\chi_{t+k} \in \hat{\mathcal{W}}$ (we omit the time index for simplicity). In general, $\hat{\mathcal{W}}$ is a nonconvex set, therefore, the resulting optimization problem (with nonconvex contraints) will be hard to solve. In order to address this issue we propose a mixed-integer linear formulation of the constraint. According to Proposition 2, $\hat{\mathcal{W}}$ is a polygon and we can divide it into $n_{\hat{\mathcal{W}}}$ convex polygons $\hat{\mathcal{W}}_i$, such that $\cup_{i=1}^{n_{\hat{\mathcal{W}}}} \hat{\mathcal{W}}_i = \hat{\mathcal{W}}$. Therefore, the condition $\chi_{t+k} \in \hat{\mathcal{W}}$ is equivalent to the condition that there exists an i, such that $\chi_{t+k} \in \hat{\mathcal{W}}_i$:

$$\chi_{t+k} \in \hat{\mathcal{W}} \Longleftrightarrow \chi_{t+k} \in \hat{\mathcal{W}}_1 \vee \cdots \vee \chi_{t+k} \in \hat{\mathcal{W}}_{n_{\hat{\mathcal{W}}}} \tag{11}$$

Since, for each $i = 1, \ldots, n_{\hat{\mathcal{W}}}$, $\hat{\mathcal{W}}_i$ is a convex polygon with n_i^f faces, the condition $\chi_{t+k} \in \hat{\mathcal{W}}_i$ can be written explicitly using n_i^f linear inequalities:

$$\chi_{t+k} \in \hat{\mathcal{W}}_i \Leftrightarrow \quad a_i^j \cdot \chi_{t+k} \leq b_i^j, \ j = 1, \ldots, n_i^f \ . \tag{12}$$

where $\cdot$ is the scalar product. In order to encode the "or" condition in (11) we employ the "big M" method [17], by replacing the "or" constraints in (11) with binary decision variables. The constraint $\chi_{t+k} \in \hat{\mathcal{W}}$ is then equivalent to

$$\forall i = 1, ..., n_{\hat{\mathcal{W}}}, \ j = 1, ..., n_i^f : \ a_i^j \cdot \chi_{t+k} \leq b_i^j + \beta_{t+k}^i M \tag{13}$$

$$\sum_{i=1}^{n_{\hat{\mathcal{W}}}} \beta_{t+k}^i \leq n_{\hat{\mathcal{W}}} - 1, \ \beta_{t+k}^i \in \{0, 1\} \ , \tag{14}$$

where β_{t+k}^i, $i = 1, \ldots, n_{\hat{\mathcal{W}}}$, are binary slack variables and M is some arbitrary big number. In (13), if $\beta_{t+k}^i = 0$, then the constraints $a_i^j \cdot \chi_{t+k} \leq b_i^j$, $j = 1, \ldots, n_i^f$ are enforced, and χ_{t+k} belongs to the i-th polygon at time $t+k$. If, for some i, χ_{t+k} does not belong to the polygon $\hat{\mathcal{W}}_i$, one of the corresponding linear inequalities will not be satisfied and β_{t+k}^i has to be set to 1. In (14) we then limit the number of relaxations, such that at least one of the β_{t+k}^i has to be set to zero, i.e., the robot has to be inside at least one region $\hat{\mathcal{W}}_i$, for some $i \in \{1, \ldots, n_{\hat{\mathcal{W}}}\}$. Since the planning is done with a receding horizon approach, we need to enforce these constraints for all χ_{t+k}, $k \in \{1, \ldots, T\}$. Constraints (13)–(14) constitute a sufficient condition to satisfy the "ideal" constraint (7).

B. The Objective (5): Frontier-based Exploration

The objective function of the optimization problem P[UCE] rewards the robot for extending the visited region. A sufficient condition for extending the visited region is to reach a frontier of $\mathcal{V}$. Now, Assumption 3 guarantees that the sensing radius is bigger than the maximum displacement of the frontiers caused by uncertainty (in Propositions 2 and 3, we scale down each estimated visited region by a quantity that is smaller than $\sigma_{\max}(t) + \mathbf{r}_\mathrm{R} + \sqrt{\sigma_{\max}^2(t) + T\sigma_{\mathrm{cmd}}^2}$); therefore, reaching a frontier of $\hat{\mathcal{W}}$ is sufficient for extending the visited region.

According to our problem setup, each frontier is essentially a line segment; call x_1^j and x_2^j the endpoints of the j-th frontier, and define the vector n_0^j, that is normal to the line that goes through x_1^j and x_2^j. Then, the robot position χ_{t+k} is on the line through x_1^j, x_2^j when: $n_0^j \cdot \chi_{t+k} = n_0^j \cdot x_1^j$. In order to enforce that the position is on the segment between x_1^j, x_2^j we introduce the vectors $n_1^j \doteq (x_2^j - x_1^j)$, $n_2^j \doteq (x_1^j - x_2^j)$. and the constraints $n_1^j \cdot \chi_{t+k} \leq n_1^j \cdot x_2^j$ and $n_2^j \cdot \chi_{t+k} \leq n_2^j \cdot x_1^j$. These constraint are again relaxed via the "big M" method to

$$n_0^j \cdot \chi_{t+k} \leq n_0^j \cdot x_1^j + \alpha_{t+k}^j M, \tag{15}$$

$$-n_0^j \cdot \chi_{t+k} \leq -n_0^j \cdot x_1^j + \alpha_{t+k}^j M \tag{16}$$

$$n_1^j \cdot \chi_{t+k} \leq n_1^j \cdot x_2^j + \alpha_{t+k}^j M, \tag{17}$$

$$n_2^j \cdot \chi_{t+k} \leq n_2^j \cdot x_1^j + \alpha_{t+k}^j M, \quad \alpha_{t+k}^j \in \{0, 1\}, \tag{18}$$

where if $\alpha_{t+k}^j = 0$, then χ_{t+k} is on the j-th frontier, while if $\alpha_{t+k}^j = 1$, χ_{t+k} is not on the frontier. Constraints (15)–(18) are imposed for all future time steps $k \in \{1, \ldots, T\}$; the sum of the binary variables α_{t+k}^j is added to the objective function in order not to force the robot to be on a frontier but to encourage it: the cost function (5) is then $\sum_{k,j} \alpha_{t+k}^j$ (with $k \in \{1, \ldots, T\}$ and j going from 1 to the number of frontiers). This follows since the sum is minimal if the robot is on as many frontiers as possible, since then as many α_{t+k}^j as possible are equal to zero.

C. Constraint (8): Bounding the Estimation Uncertainty

We now formalize the uncertainty constraints, exploiting basic insights from graph theory and linear algebra.

1) Estimation Uncertainty and Graph Structure: In order to understand how robot motion (and then, graph structure) relates with the uncertainty matrix $P_{1:t+T}$ we should spend some words on the structure of this matrix. We start our derivation from the inverse of $P_{1:t+T}$, namely $\Omega_{1:t+T}$. The matrix $\Omega_{1:t+T}$ has a very natural interpretation in terms of graph theory since it corresponds to the (weighted) graph Laplacian matrix without the first row and column (corresponding to the node that we set to $\mathbf{0}_d$). If we call $\mathcal{N}_{\mathrm{IN}}(i)$ the neighbors of node i in the graph, $\Omega_{1:t+T}$ has the following structure:

- the i-th diagonal element is equal to $\sum_{j\in\mathcal{N}_{\mathrm{IN}}(i)}\left(\frac{1}{\sigma^2_{i,j}}\right)$;
- the element in position (i,j) in the matrix is equal to $-1/\sigma^2_{i,j}$ if a measurement between node i and j is available, or is zero otherwise.

We recall that $\mathcal{E}_{0:t+T}$ is the set of edges added in the interval of time $[0, t+T]$, and that $\mathcal{E}_{1:t+T} = \mathcal{E}_{0:t+T}$ since no edge is added at time 0 (initially there is only one node in the graph). Then, it is easy to see that we can write the matrix $\Omega_{1:t+T}$ as:

$$\Omega_{1:t+T} = \sum_{(i,j)\in\mathcal{E}_{1:t+T}} \frac{1}{\sigma^2_{i,j}} a_{ij} a_{ij}^\top \tag{19}$$

where $a_{ij} \in \{-1, 0, +1\}^{t+T}$ is a vector modelling edge (i,j), that has at most two nonzero elements, one equal to -1 in position i and one equal to $+1$ in position j; the only case in which there is a single nonzero element is the case in which the edge connects a node i with the node corresponding to x_0: in this case the only nonzero element is in position i and is -1 (respectively $+1$) if the edge goes out (respectively is incoming) from node i. Note that if the decision time is t (i.e., we are planning the motion steps $\chi_{t+1}, \ldots, \chi_{t+T}$) then the edges added until time t cannot be modified by the plan; therefore we can write $\mathcal{E}_{1:t+T} = \mathcal{E}_{1:t} \cup \mathcal{E}_{t+1:t+T}$, where we distinguish from the edges $\mathcal{E}_{1:t}$, already present in the graph at time t, from the edges $\mathcal{E}_{t+1:t+T}$ introduced in the next T time steps. Furthermore, we recall that the commands are always available, independently on the chosen motion strategy, while what our plan can change is the possibility to revisit previous nodes, i.e., may eventually add loop closings. Therefore, we write $\mathcal{E}_{t+1:t+T} = \mathcal{E}^{\mathrm{cmd}}_{t+1:t+T} \cup \mathcal{E}^{\mathrm{lc}}_{t+1:t+T}$, where $\mathcal{E}^{\mathrm{lc}}_{t+1:t+T}$ groups the edges corresponding to loop closings, while $\mathcal{E}^{\mathrm{cmd}}_{t+1:t+T}$ contains the remaining edges. We can then rewrite (19) as:

$$\Omega_{1:t+T} = \sum_{(i,j)\in\mathcal{E}_{1:t}} \frac{1}{\sigma^2_{i,j}} a_{ij} a_{ij}^\top + \sum_{(i,j)\in\mathcal{E}^{\mathrm{cmd}}_{t+1:t+T}} \frac{1}{\sigma^2_{i,j}} a_{ij} a_{ij}^\top + \sum_{(i,j)\in\mathcal{E}^{\mathrm{lc}}_{t+1:t+T}} \frac{1}{\sigma^2_{i,j}} a_{ij} a_{ij}^\top = \text{const.} + \sum_{(i,j)\in\mathcal{E}^{\mathrm{lc}}_{t+1:t+T}} \frac{1}{\sigma^2_{i,j}} a_{ij} a_{ij}^\top \tag{20}$$

where we remarked that the first two sums cannot be modified by our plan (they give a constant matrix, which can be precomputed at time t). Equation (20) highlights the fundamental structure of our problem: the uncertainty reduction problem consists in deciding which loop closings the robot has to perform to meet given constraints on $P_{1:t+T}$. Now we notice that in (19) we already obtained explicit expressions for $\Omega_{1:t+T}$, and we may use these expressions to pose constraints on the uncertainty for the optimization problem. For instance, if we want to bound $\det(P_{1:t+T})$, $\lambda_{\max}(P_{1:t+T})$, or $\mathrm{tr}(P_{1:t+T})$ we simply recognize that $\det(P_{1:t+T}) = \det(\Omega^{-1}_{1:t+T})$, $\lambda_{\max}(P_{1:t+T}) = \frac{1}{\lambda_{\min}(\Omega_{1:t+T})}$ and $\mathrm{tr}(P_{1:t+T}) = \mathrm{tr}(\Omega^{-1}_{1:t+T})$; bounds on these quantities can be formulated as convex constraints [6] in the entries of $\Omega_{1:t+T}$, which is a desirable condition. However, the objective of this paper is to formulate a mixed-integer *linear* program, therefore we want to further work out the expressions to obtain linear constraints. For this purpose, let us study the effect of adding an edge on $P_{1:t+T}$.

Proposition 4. *If we call $P^{pre}_{1:t+T}$ the uncertainty matrix before the addition of a loop closing edge (i,j), and $P_{1:t+T}$ the uncertainty matrix after the addition of edge (i,j) it holds:*

$$P_{1:t+T} = P^{pre}_{1:t+T} - \frac{1}{\sigma^2_{i,j}} P^{pre}_{1:t+T} a_{ij} \left(1 + \frac{1}{\sigma^2_{i,j}} a_{ij}^\top P^{pre}_{1:t+T} a_{ij}\right)^{-1} a_{ij}^\top P^{pre}_{1:t+T}$$

Proof: Let us define the information matrix $\Omega^{pre}_{1:t+T}$ before the insertion of edge (i,j) and the information matrix $\Omega_{1:t+T}$ after edge (i,j) is included in the graph:

$$P_{1:t+T} = \Omega^{-1}_{1:t+T} = \left(\Omega^{pre}_{1:t+T} + \frac{1}{\sigma^2_{i,j}} a_{ij} a_{ij}^\top\right)^{-1} \tag{21}$$

then, recalling that $(\Omega^{pre}_{1:t+T})^{-1} = P^{pre}_{1:t+T}$, the result is a simple application of the matrix inversion lemma. ■

Corollary 1. *The addition of a loop closing edge can only reduce (i) the determinant, (ii) the maximum eigenvalue, (iii) the trace, (iv) each diagonal element of $P_{1:t+T}$.*

The proof of the corollary (omitted for brevity) stems from the fact that the matrix $G_{i,j} \doteq \frac{1}{\sigma^2_{i,j}} P^{pre}_{1:t+T} a_{ij} \left(1 + \frac{1}{\sigma^2_{i,j}} a_{ij}^\top P^{pre}_{1:t+T} a_{ij}\right)^{-1} a_{ij}^\top P^{pre}_{1:t+T}$ is positive definite, hence, "reduces" the original uncertainty $P^{pre}_{1:t+T}$.

The previous corollary remarks that the addition of edges may only help in meeting our uncertainty constraints. Moreover, the advantage, or the *information gain*, in adding a specific edge can be quantified in $G_{i,j}$, i.e., the larger is the matrix $G_{i,j}$ (w.r.t. one of the considered metrics), the larger is the uncertainty reduction in adding edge (i,j).

Remark 1. *While each edge has an additive effect on the matrix $\Omega_{1:t+T}$, it has a more complex impact on $P_{1:t+T}$. In fact, while adding a single edge (i,j) leads to $P_{1:t+T} = P^{pre}_{1:t+T} - G_{i,j}$, adding several edges leads to complex nonlinear expressions, because $G_{i,j}$ depends on $P^{pre}_{1:t+T}$; therefore, in general, it holds $P_{1:t+T} \neq P_{1:t} - \sum_{(i,j)\in\mathcal{E}_{t+1:t+T}} G_{i,j}$.*

2) A Bound on the Estimation Uncertainty: Now consider the situation in which we have to impose a constraint of the type $\mathtt{size}(P_{1:t+T}) \leq \bar{\mathbf{U}}$, where $\mathtt{size}$ is an uncertainty metric chosen among the ones mentioned in Section II-B, and $\bar{\mathbf{U}}$ is a desired upper bound. We exploit Corollary 1 to find a simple (linear) way of imposing the uncertainty constraint. The basic idea is the following: we can precompute a matrix $P^{pre}_{1:t+T}$ assuming that there are no loop closings in $[t+1, t+T]$; then, if we plan a motion strategy $\chi_{t+1}, \ldots, \chi_{t+T}$ that allows to include the loop closing edge

(i^*, j^*), we can guarantee that the actual uncertainty at time $t+T$ is no larger than $P^{pre}_{1:t+T} - G^*_{i,j}$, since, according to Corollary 1, the addition of further edges can only reduce the uncertainty. More formally, for each $(i^*, j^*) \in \mathcal{E}_{t+1:T}$, it holds: $P_{1:t+T} \leq P^{pre}_{1:t+T} - G^*_{i,j}$, then

$$\texttt{size}(P^{pre}_{1:t+T} - G^*_{i,j}) \leq \bar{\mathbf{U}} \Longrightarrow \texttt{size}(P_{1:t+T}) \leq \bar{\mathbf{U}} \qquad (22)$$

Therefore, we can simply plan the inclusion of a single edge (i^*, j^*) that satisfies $\texttt{size}(P^{pre}_{1:t+T} - G^*_{i,j}) \leq \bar{\mathbf{U}}$ and we can assure that the uncertainty bound is satisfied. Clearly, in this way we are actually imposing a stricter constraint on the uncertainty, but this carries an important advantage: both $P^{pre}_{1:t+T}$ and $G^*_{i,j}$, can be computed a-priori, i.e., before the execution of the plan, and the inequality $\texttt{size}(P^{pre}_{1:t+T} - G^*_{i,j}) \leq \bar{\mathbf{U}}$ can be formulated as a linear constraint as we will see in a while.

For deciding on the addition of an edge we use a collection of binary variables: γ^j_{t+k} with $k \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, t\}$; if $\gamma^j_{t+k} = 1$ we plan to add an edge between the planned position χ_{t+k} and the previous estimated position $\hat{x}_j$; if $\gamma^j_{t+k} = 0$ no edge is added between node $t+k$ and j. Therefore, the uncertainty bound is imposed through the following constraint:

$$\texttt{size}\Bigg(P^{pre}_{1:t+T} - \sum_{\substack{k \in \{1,\ldots,T\} \\ j \in \{1,\ldots,t\}}} \gamma^j_{t+k} G^*_{i,j}\Bigg) \leq \bar{\mathbf{U}} \qquad (23)$$

$$\sum_{\substack{k \in \{1,\ldots,T\} \\ j \in \{1,\ldots,t\}}} \gamma^j_{t+k} \leq 1, \qquad \gamma^j_{t+k} \in \{0,1\} \quad (24)$$

where the first constraint corresponds to the upper bound (22), while the second condition imposes that we can plan to add at most one edge (the bound (22) holds for a single edge).

We miss a last ingredient to model the uncertainty constraints. We recall from Section II that we cannot add any edge (i,j) arbitrarily: these edges may only connect two nodes whose distance is smaller or equal than the loop closing radius $\mathbf{r}_{\text{LC}}$. In order to impose this restriction, we have to force $\gamma^j_{t+k} = 0$ when $\|x_j - x_{t+k}\| \geq \mathbf{r}_{\text{LC}}$. Also in this case, however, we should cope with the uncertainty: we do not know x_j and x_{t+k}, but we know $\hat{x}_j$ and χ_{t+k}. We can then impose the more conservative condition that prevents to plan the addition of an edge when:

$$\|\hat{x}_j - \chi_{t+k}\| \geq \mathbf{r}_{\text{LC}} - \sqrt{2\sigma^2_{\max}(t) + T\sigma^2_{\text{cmd}}}, \qquad (25)$$

which is always a positive quantity by Assumption 3. It is possible to demonstrate that (25) is a necessary condition for $\|x_j - x_{t+k}\| \geq \mathbf{r}_{\text{LC}}$, i.e., that the plan does not enable loop closings that are not possible in the reality. The proof follows from Proposition 1 and Proposition 3 and is omitted here due to space restrictions.

We impose that $\gamma^j_{t+k} = 0$ as soon as inequality (25) is satisfied; we do this by introducing a constraint for each binary γ^j_{t+k} with $k \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, t\}$:

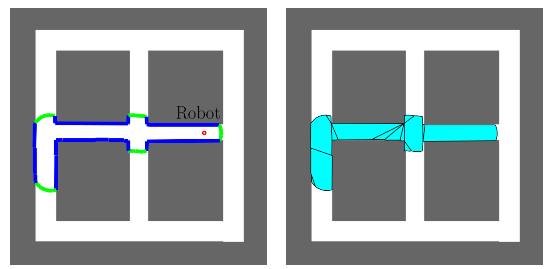
$$\|\hat{x}_j - \chi_{t+k}\| \leq \mathbf{r}_{\text{LC}} - \sqrt{2\sigma^2_{\max}(t) + T\sigma^2_{\text{cmd}}} + (1 - \gamma^j_{t+k})M \quad (26)$$

If $\|\hat{x}_j - \chi_{t+k}\| \geq \mathbf{r}_{\text{LC}} - \sqrt{2\sigma^2_{\max}(t) + T\sigma^2_{\text{cmd}}}$ the constraints forces $(1 - \gamma^j_{t+k})$ to be 1, then γ^j_{t+k} has to be zero, avoiding loops between far nodes. Constraints (23)–(26) constitute our formalization for the original uncertainty constraint (8).

Remark 2 (Final prescription)**.** *The norm constraints* (26) *and* (6) *are linear when using the ℓ_1 (Manhattan) norm for measuring distances. Recall that for any vector x, it holds $\|x\|_1 \geq \|x\|_2$, then the satisfaction of a constraint under the ℓ_1-norm is a sufficient condition for the satisfaction of the corresponding constraint under the ℓ_2-norm.*

IV. Numerical Experiments

We consider an experiment in which the robot has to explore the scenario in Fig. 2. The robot radius is $\mathbf{r}_{\text{R}} = 0.3$ m, the sensing radius is $\mathbf{r}_{\text{SR}} = 7$ m, and the loop closing radius is $\mathbf{r}_{\text{LC}} = 2$ m. The measurement noise if bounded in a ball of radius $\sigma_{i,j} = 0.15$ m, while the commands are bounded by $\mathbf{r}_{\text{V}} = 3$ m. We initialize the time horizon to 5, and we increase it in case it is not sufficient to let the robot reach a frontier (*replanning*). We use the maximum diagonal entry of the $P_{1:t+T}$ as uncertainty metric, i.e., $\texttt{size}(P_{1:t+T}) = \sigma^2_{\max}(P_{1:t+T})$, and we impose the bound $\sigma^2_{\max}(P_{1:t+T}) \leq 1$.

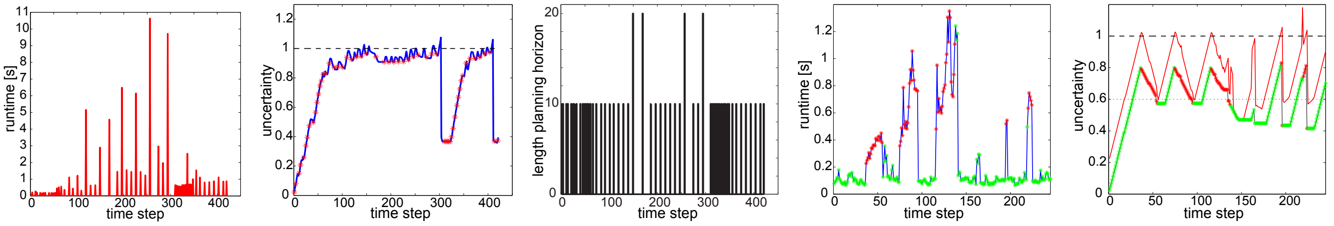


(a) (b)

Fig. 2. Scenario with obstacles in dark grey. In (a) the blue lines depict the obstacle boundaries and the green lines the frontiers. In (b) we show the convex polygons that together define the estimated visited region. For collision-free motion these are scaled down by the security margins (Proposition 3), giving the convex polygons $\hat{\mathcal{W}}_i$.

The results of the implementation of the proposed MILP are reported in Figures 3(a)–3(c), which describe the CPU time required for the optimization at each decision time and the trend of $\sigma^2_{\max}$. Also, we depict the length of the planning horizons for each decision time (they may vary when a replanning is needed). After planning we let the robot apply all commands in the plan, therefore, if at time t the robot plans a trajectory over an horizon T, the next planning time occurs at time $t+T$; this justifies the gaps in Fig. 3(c) that essentially correspond to time steps in which the robot is applying the previous plan (without solving any optimization problem).

We notice that in few time steps the current uncertainty $\sigma^2_{\max}(P_{1:t})$ is above the threshold. This is consistent with out formulation, since we require that $\sigma^2_{\max}(P_{1:t+T}) \leq 1$, i.e., the robot has to guarantee constraint satisfaction only after applying the entire plan; for instance the robot may plan a loop closing when reaching the last position in the plan χ_{t+T}, and the uncertainty reduction only occurs at time $t+T$. For this reason, the red markers in Fig. 3(b), corresponding to the decision times, are always below the desired uncertainty bound. Our formulation may also accommodate the stricter constraint that at each step the uncertainty should be smaller than the bound. We preferred not to consider this setup since it may easily lead to infeasible problem instances, while with the formulation P[UCE] the robot may always find a suitable



(a) (b) (c) (d) (e)

Fig. 3. (a) Runtime in seconds, (b) current uncertainty $\sigma^2_{\max}(P_{1:t})$, and (c) lengths of the planning horizon for the $\mathrm{P}[\mathtt{UCE}]$ algorithm. In (b) the red markers denote the decision times, i.e., the time steps in which the optimization problem is solved. (d) Runtime in seconds and (e) uncertainty metrics for each time step for $\mathrm{P}[\mathtt{UCE}]$ with thresholding. At the time steps marked in red the objective was geared towards uncertainty reduction and at the time steps marked in green the objective was focused on exploration. In (e) the solid red line denotes the upper bound $\sigma^2_{\max}(P^{pre}_{1:t+T})$, while the line with green and red markers corresponds to $\sigma^2_{\max}(P_{1:t})$.

time horizon that makes the uncertainty constraint feasible. We notice that for few decision times, the solution of $\mathrm{P}[\mathtt{UCE}]$ required several seconds, since the robot had to plan over a longer horizon. Although the reported times are still acceptable in real applications, this motivated us in developing a variant of the exploration algorithm, named $\mathrm{P}[\mathtt{UCE}]$ *with thresholding*. In this variant, when the uncertainty rises above a bound, the robot is required to reduce the uncertainty below a lower bound (0.6 in our experiments) before continuing the exploration. We implement this variant as follows: at the beginning the objective function is the one of Section III-B; when the quantity $\sigma^2_{\max}(P^{pre}_{1:t+T})$ (see Section III-C.1) rises above the uncertainty bound we substitute the exploration objective with one rewarding the addition of edges, exploiting the results of Section III-C.1. Note that $\sigma^2_{\max}(P^{pre}_{1:t+T})$ is the maximum position uncertainty assuming that no loop closing is done in the interval $[t+1, t+T]$, then it constitutes an upper bound for the actual uncertainty $\sigma^2_{\max}(P_{1:t+T})$. The exploration-driven objective is re-established as soon as $\sigma^2_{\max}(P^{pre}_{1:t+T})$ drops below 0.6. The results of this variant are reported in Fig. 3(d) and 3(e). The variant shows a relevant boost in the performance w.r.t. the original approach and it will be subject of future research[2].

V. Conclusion

We formulated a mixed-integer linear program that allows the robot to establish a suitable motion strategy to explore an unknown environment without any absolute position information. The formulation explicitly models the presence of obstacles in the environment, rewards the robot for expanding the visited regions, and includes several tools for bounding the uncertainty in position estimation. The paper enriches related literature by bridging between optimization-based planning techniques and estimation theory applied to localization and mapping. Future work includes an extensive numerical evaluation and the investigation of fundamental questions that still remain without answer. For instance, is it possible to prove that the complete environment is explored in finite time? How do the measurement error and the other parameters influence the time required for completing the exploration process?

[2] A video showing an example of uncertainty-constrained exploration with the algorithm $\mathrm{P}[\mathtt{UCE}]$ with thresholding can be found at `http://www.lucacarlone.com/index.php/resources/videos`.

References

[1] J.L. Blanco, J.A. Fernandez-Madrigal, and J. Gonzalez. A novel measure of uncertainty for mobile robot SLAM with Rao-Blackwellized particle filters. *Int. J. Robot. Res.*, 27(1):73–89, 2008.

[2] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proc. IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.

[3] P. Brass, F. Cabrera-Mora, A. Gasparri, and J. Xiao. Multirobot tree and graph exploration. *IEEE Transactions on Robotics*, 27(4):707–717, 2011.

[4] L. Carlone, J. Du, M. Kaouk Ng, B. Bona, and M. Indri. An application of Kullback-Leibler divergence to active SLAM and exploration with particle filters. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 287–293, 2010.

[5] H. Carrillo, I. Reid, and J.A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *Proc. of Int. Conf. on Robotics and Automation*, 2012.

[6] C. Durieu, E. Walter, and B.T. Polyak. Multi-input multi-output ellipsoidal state bounding. *J. Opt. Theory Appl.*, 111(2):273–303, 2001.

[7] H.J.S. Feder, J.J. Leonard, and C.M. Smith. Adaptive mobile robot navigation and mapping. *Int. Journal of Robotics Research*, 18(7):650–668, 1999.

[8] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed multirobot exploration and mapping. *Proc. of the IEEE*, 94(7):1325–1339, 2006.

[9] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Transactions on Mechatronics*, 14(2):163–175, 2009.

[10] S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang. Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In *Proc. of Int. Conf. on Robotics and Automation*, pages 1091–1096, 2005.

[11] T. Kollar and N. Roy. Trajectory optimization using reinforcement learning for map exploration. *Int. Journal of Robotics Research*, 27(2):175–196, 2008.

[12] C. Leung, S. Huang, and G. Dissanayake. Active SLAM using model predictive control and attractor based exploration. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5026–5031, 2006.

[13] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27:93103, 2009.

[14] R. Martinez-Cantin, N. De Freitas, A. Doucet, and J.A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Proc. of Robotics: Science and Systems*, 2007.

[15] A. Richards, T. Schouwenaars, J.P. How, and E. Feron. Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-integer Linear Programming. *Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002.

[16] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *Proc. of Int. Conf. on Robotics and Automation*, pages 661–666, 2005.

[17] H.A. Taha. *Operations research: an introduction*, volume 8. Prentice Hall Upper Saddle River, NJ, 1997.

[18] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. of CIRA 17*, 1997.