Planning Under Uncertainty in the Continuous Domain: a Generalized Belief Space Approach

Vadim Indelman, Luca Carlone, and Frank Dellaert

***Abstract*— This work investigates the problem of planning under uncertainty, with application to mobile robotics. We propose a probabilistic framework in which the robot bases its decisions on the *generalized belief*, which is a probabilistic description of its own state and of external variables of interest. The approach naturally leads to a dual-layer architecture: an *inner estimation layer*, which performs inference to predict the outcome of possible decisions, and an *outer decisional layer* which is in charge of deciding the best action to undertake. The approach does not discretize the state or control space, and allows planning in continuous domain. Moreover, it allows to relax the assumption of *maximum likelihood observations*: predicted measurements are treated as random variables and are not considered as *given*. Experimental results show that our planning approach produces smooth trajectories while maintaining uncertainty within reasonable bounds.**

I. Introduction

Autonomous navigation in complex unknown scenarios involves a deep intertwining of estimation and planning capabilities. A mobile robot is required to perform *inference*, from sensor measurements, in order to build a model of the surrounding environment and to estimate variables of interest. Moreover, it has to *plan* actions to accomplish a given goal. Clearly, if the scenario in which the robot operates is unknown, robot decisions need to be based on the estimates coming from the inference process. This problem is an instance of *a partially observable Markov decision process* (POMDP), which describes a decisional process in Markovian systems, in which the state is not directly observable. While POMDP are intractable in general [12], it is of practical interest to device problem-specific solutions or approximations (e.g., *locally* optimal plans) that trade-off optimality for computational efficiency.

The estimation problem arising in robot navigation has recently reached a good maturity. State-of-the-art techniques for localization and mapping (*SLAM*) allow fast solution of medium-large scenarios [16], [19], [21], [17], using efficient nonlinear optimization techniques, that exploit the structure of the underlying problem. On the other hand, planning under uncertainty is still attracting a consistent attention from the robotic community as, besides the amount of published papers, few approaches are able to deal with the complexity and time-constraints of real-world problems.

Related literature offers heterogeneous contributions on the topic. A recurrent idea is to restrict the state space or the control space to few possible values. While this *discretization* prevents obtaining an optimal solution, it allows to frame the "computation" of a plan into a "selection" of the best plan among few candidates. In this case, a consistent research effort has been devoted to design suitable metrics to quantify the quality of a candidate plan. The problem itself is not trivial as the metric depends on the representation we use for the environment and on the inference engine. Examples of this effort are the work of Stachniss et al. [31], [30], Blanco et al. [4], and Du et al. [8], in which particle filters are used as estimation engine. Martinez-Cantin et al. [23], [24] and Bryson and Sukkarieh [5] investigate planning techniques in conjunction with the use of EKF for estimation. Huang et al. [10] propose a model predictive control (MPC) strategy, associated with EKF-SLAM. Leung et al. [22] propose an approach in which the MPC strategy is associated with a heuristic based on global attractors. Sim and Roy [28] propose A-optimal strategies for solving the active SLAM problem. Carrillo et al. [6] provide an analysis of the uncertainty metrics used in EKF-based planning. Other examples are [18], [32] in which the belief is assumed to be a Gaussian over current and past poses of the robot.

We notice that, while all the previous approaches are applied to mobile robot navigation problems (the corresponding problem is usually referred to as *active Simultaneous Localization and Mapping*), similar strategies can be found with application to manipulation and computer vision (e.g., *next best view* problem [26]). A related problem is also the so-called *informative path planning* (although in these problems the estimation aspects are often neglected). A greedy strategy for informative path planning is proposed by Singh et al. [29] while a branch and bound approach is proposed by Binney et al. in [3]. More recently, Hollinger et. al [9] propose more efficient algorithms, based on rapidly-exploring random trees and probabilistic roadmaps. These approaches usually assume that the robot moves in a partially known environment; a remarkable property of these techniques is that they approach optimality when increasing the runtime (which is exponential in the size of the problem).

Recent literature on planning under uncertainty is trying to go beyond three limitations that are common to most of the works mentioned so far. The first limitation is *discretization*. Reasoning in terms of a discretized grid world may lead to suboptimal plans (Figure 1c) as the performance of the corresponding approaches heavily depends on grid resolution, while increasing grid resolution usually implies an increase in the computational cost of planning. Continuous models appear as more natural representations for real problems, in which robot states (e.g., poses) and controls (e.g., steering angles) are not constrained to few discrete values. In [1], Bai

V. Indelman, L. Carlone, and F. Dellaert are with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA, indelman@cc.gatech.edu, luca.carlone@gatech.edu, frank@gatech.edu.

et al. avoid discretization by using Monte Carlo sampling to update an initial policy. Platt et al. [11] apply linear quadratic regulation (LQR) to compute locally optimal policies. Kontitsis et al. [20] propose a sampling based approach for solving a constrained optimization problem in which the constraints correspond to state dynamics, while the objective function to optimize includes uncertainty and robot goal. A hierarchical goal regression for mobile manipulation is proposed by Kaelbling et al. in [13], [14], [15].

A second limitation regards the assumption that some *prior knowledge of the environment* is available and the belief typically represents only the robot trajectory [2], [27]. In particular, the Belief Roadmap [27] incorporates predicted uncertainty of future position estimates into planning, while assuming the environmental map to be given.

A third limitation is the assumption of *maximum likelihood observations*: since future observations are not given at planning time, the robot assumes that it will acquire the measurements that are most likely given the predicted belief. Van den Berg et al. [2] relax the maximum likelihood assumption, while assuming prior knowledge on the environment and on the state-dependent measurement noise.

This work addresses the three limitations mentioned in the literature review. We introduce the concept of *generalized belief space* (GBS): the robot keeps a joint belief over its own state and the state of external variables of interest. This allows relaxing the assumption that the environment is known or partially known, and enables applications in completely unknown and unstructured scenarios. Planning in GBS, similarly to planning in belief space [11], is done in a continuous domain and avoids the maximum likelihood assumption that characterizes earlier works. The concept of generalized belief space is introduced in Section II. Our planning strategy is then described in Section III. In Section IV, we present experimental results, comparing the proposed approach with a planning technique based on state space discretization. Conclusions are drawn in Section V.

II. Generalized Belief Space

A. Notation and Probabilistic Formulation

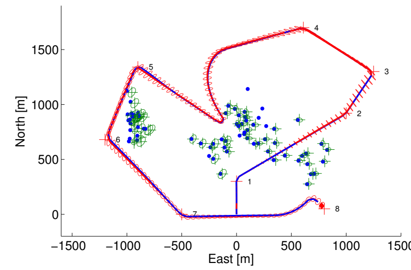
Let x_i and $\mathcal{W}_i$ denote the *robot state* and the *world state* at time t_i. For instance, in mobile robots navigation, x_i may describe robot pose at time t_i and $\mathcal{W}_i$ may describe the positions of landmarks in the environment observed by the robot by time t_i. In a manipulation problem, instead, x_i may represent the pose of the end effector of the manipulator, and $\mathcal{W}_i$ may describe the pose of an object to be grasped. The world state $\mathcal{W}_i$ is time-dependent in general (e.g., to account for possible variations in the environment, or to model the fact that the robot may only have observed a subset of the environment by time t_i) and for this reason we keep the index i in $\mathcal{W}_i$. Let Z_i denote the available observations at time t_i and u_i the control action applied at time t_i.

We define the joint state at time t_k as $X_k \doteq \{x_0, \ldots, x_k, \mathcal{W}_k\}$, and we write the probability distribution function (pdf) over the joint state as:

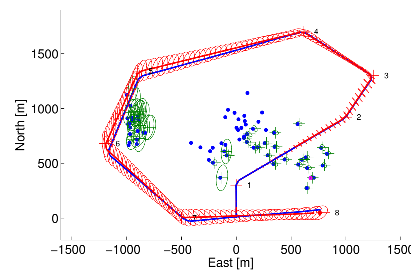
$$p\left(X_k | \mathcal{Z}_k, \mathcal{U}_{k-1}\right), \tag{1}$$

where $\mathcal{Z}_k \doteq \{Z_0, \ldots, Z_k\}$ represent all the available observations until time t_k, and $\mathcal{U}_{k-1} \doteq \{u_0, \cdots, u_{k-1}\}$ denotes all past controls. The probabilistic motion model given the control u_i and robot state x_i is

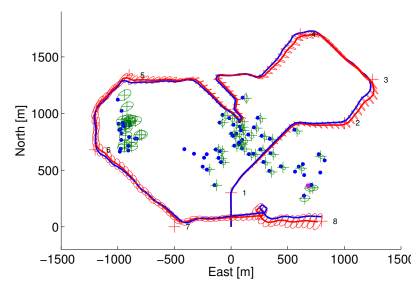
$$p\left(x_{i+1} | x_i, u_i\right). \tag{2}$$



(a)



(b)



(c)

Fig. 1: Planning under uncertainty. A mobile robot starts at the origin without prior knowledge of the environment and has to reach a sequence of goals, labeled from 1 to 8, while keeping bounded its uncertainty. The figure shows: trajectories, mapped landmarks, and uncertainties for different planning approaches. (a) Planning in the GBS; (b) Planning in the GBS without uncertainty terms in the objective function; (c) Discrete planning.

We consider a general observation model that involves at time t_i a subset of joint states $X_i^o \subseteq X_i$:

$$p\left(Z_i | X_i^o\right). \tag{3}$$

The basic observation model, commonly used in motion planning, e.g., [2], involves only the current robot state x_i at each time t_i and is a particular case of the above general model (3). In general Z_i may include several observations: in the next section we consider the case in which $Z_i = \{z_{i,1}, \ldots, z_{i,n_i}\}$, making explicit that at time t_i the robot acquires n_i measurements.

The joint pdf (1) at the current time t_k can be written according to motion and observation models (2) and (3):

$$p\left(X_k|\mathcal{Z}_k, \mathcal{U}_{k-1}\right) = priors \cdot \prod_{i=1}^{k} p\left(x_i|x_{i-1}, u_{i-1}\right) p\left(Z_i|X_i^o\right). \tag{4}$$

The $priors$ term includes $p\left(x_0\right)$ and any other available prior information.

B. Generalized Belief Space (GBS)

Our goal is to compute an optimal control action over L look-ahead steps. This control action will depend in general on the belief over the time horizon. The *belief* at the lth look-ahead step involves the distribution over all the states so far, accounting for all the commands and observations until that time. For example, the belief at the planning time t_k is exactly given by Eq. (4). However, for computing the control policy we also need to predict the belief at future time instances t_{k+1} until t_{k+L} whose general expression for the lth look-ahead step is given as follows:

$$gb\left(X_{k+l}\right) \doteq p\left(X_{k+l}|\mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right), \tag{5}$$

where we separated actions $\mathcal{U}_{k-1}$ and observations $\mathcal{Z}_k$ occurring until the planning time t_k from the actions $u_{k:k+l-1}$ and observations $Z_{k+1:k+l}$ from the first l look-ahead steps (t_{k+1} until t_{k+l}).

We consider the case of motion and observation models with additive Gaussian noise:

$$x_{i+1} = f\left(x_i, u_i\right) + w_i \quad , \quad w_i \sim N\left(0, \Omega_w\right) \tag{6}$$

$$z_{i,j} = h\left(X_{i,j}^o\right) + v_i \quad , \quad v_i \sim N\left(0, \Omega_v\right), \tag{7}$$

where $z_{i,j}$ represents the jth observation at time t_i (recall that $Z_i = \{z_{i,1}, \ldots, z_{i,n_i}\}$, where n_i is the total number of observations acquired at time t_i); observation model for the measurement $z_{i,j}$ involves the subset of states $X_{i,j}^o$. For instance, $z_{i,j}$ could represent observations of the jth landmark. For notational convenience, we use $\epsilon \sim N(\mu, \Omega)$ to denote a Gaussian random variable ϵ with mean μ and information matrix Ω (inverse of the covariance matrix). For simplicity we assume the same measurement model $h\left(\cdot\right)$ and noise v_i for all the observations at time t_i, although the above formulation can be easily generalized.

The complexity stems from the fact that it is unknown ahead of time whether or not an observation $z_{i,j}$ will be acquired. For example, if the robot is far away from the jth landmark there will be no measurement $z_{i,j}$ in practice. For this reason, we introduce a *binary* random variable $\gamma_{i,j}$ for each observation $z_{i,j}$ to represent the event of this measurement being acquired. Furthermore, even if the measurement $z_{i,j}$ is acquired, its actual value is unknown ahead of time, and therefore we also treat $z_{i,j}$ as random variable. Therefore, we can define a joint probability density over the random variables in our problem:

$$p\left(X_{k+l}, \Gamma_{k+1:k+l}, Z_{k+1:k+l}|\mathcal{Z}_k, \mathcal{U}_{k-1}, u_{k:k+l-1}\right), \tag{8}$$

where $\Gamma_i \doteq \{\gamma_{i,j}\}_{j=1}^{n_i}$, with n_i being the number of possible observations at time t_i. In our mobile robotics example n_i is simply the number of landmarks observed thus far.

We refer to (8) with the term *generalized belief* as now the probability distribution is defined over the robot and world state (included in the vector X_{k+l}), as well as over the random variables $\Gamma_{k+1:k+l}$, and $Z_{k+1:k+l}$. Using the chain rule and taking the standard assumption of uninformative priors yields

$$p\left(X_{k+l}, \Gamma_{k+1:k+l}, Z_{k+1:k+l}|\mathcal{Z}_k, \mathcal{U}_{k-1}, u_{k:k+l-1}\right) = p\left(X_{k+l}, \Gamma_{k+1:k+l}|\mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right).$$

Since the belief (5) is defined only over the states, we marginalize the latent variables $\Gamma_{k+1:k+l}$ and get

$$gb\left(X_{k+l}\right) = \sum_{\Gamma_{k+1:k+l}} p\left(X_{k+l}, \Gamma_{k+1:k+l}|\mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right). \tag{9}$$

We represent this belief by a Gaussian

$$gb\left(X_{k+l}\right) \sim N\left(X_{k+l}^*, I_{k+l}\right), \tag{10}$$

where the mean of the normal distribution is set to the maximum a posteriori (MAP) estimate

$$X_{k+l}^* = \arg\max_{X_{k+l}} gb\left(X_{k+l}\right) = \arg\min_{X_{k+l}} -\log gb\left(X_{k+l}\right). \tag{11}$$

While in this section we introduced the concept of generalized belief space, we did not provide computational procedures to compute the Gaussian approximation of the belief (10). A computationally efficient technique to compute the belief constitutes the inference layer of our planning approach and is detailed in Section III-B.

III. Planning in GBS

In our receding horizon approach, we want to compute an optimal control action over L look-ahead steps. The control action has to minimize the *general* objective function $J_k\left(u_{k:k+l-1}\right)$:

$$J_k\left(u_{k:k+L-1}\right) \doteq \mathop{\mathbb{E}}_{Z_{k+1:k+L}} \left\{ \sum_{l=0}^{L-1} c_l(gb\left(X_{k+l}\right), u_{k+l}) + c_L(gb\left(X_{k+L}\right)) \right\}, \tag{12}$$

where c_l is a general immediate cost for each of the L look-ahead steps. The expectation is taken to account for all the possible observations during the planning lag, since these are not given at planning time and are stochastic in nature. Since the expectation is a linear operator we rewrite (12) as

$$J_k\left(u_{k:k+L-1}\right) \doteq \sum_{l=0}^{L-1} \mathop{\mathbb{E}}_{Z_{k+1:k+l}} \left[c_l\left(gb\left(X_{k+l}\right), u_{k+l}\right)\right] + \mathop{\mathbb{E}}_{Z_{k+1:k+L}} \left[c_L\left(gb\left(X_{k+L}\right)\right)\right]. \tag{13}$$

The optimal control $u^*_{k:k+L-1} \doteq \left\{u^*_k, \dots, u^*_{k+L-1}\right\}$ is the control policy π:

$$u^*_{k:k+L-1} = \pi\left(gb\left(X_k\right)\right) = \underset{u_{k:k+L-1}}{\arg\min}\ J_k\left(u_{k:k+L-1}\right). \quad (14)$$

Calculating the optimal control policy (14) involves the optimization of the objective function $J_k\left(u_{k:k+L-1}\right)$. According to (12), the objective depends on the (known) GBS at planning time t_k, on the predicted GBS at time $t_{k+1}, \dots, t_{k+L}$, and on the future controls $u_{k:k+L-1}$. Since in general the immediate costs $c_l\left(gb\left(X_{k+l}\right), u_{k+l}\right)$ are nonlinear functions, $\underset{Z_{k+1:k+l}}{\mathbb{E}}\left[c_l\left(gb\left(X_{k+l}\right), u_{k+l}\right)\right] \neq c_l\left(\underset{Z_{k+1:k+l}}{\mathbb{E}}\left[gb\left(X_{k+l}\right)\right], u_{k+l}\right)$, and we have to preserve the dependence of the belief $gb\left(X_{k+l}\right)$ on the observations $Z_{k+1:k+l}$. As anticipated in Section II-B, these observations are treated as random variables.

A. Approach Overview

In order to optimize the objective function (13) we resort to an iterative optimization approach, starting from a given initial guess on the controls. The overall approach can be described as a *dual-layer inference*: the inner layer performs inference to calculate the GBS at each of the look-ahead steps, for a given $u_{k:k+L-1}$. The computation of the GBS is done via (11), by applying a computationally efficient technique, based on *expectation-maximization* and a Gauss-Newton method. The outer layer performs inference over the control $u_{k:k+L-1}$, minimizing the objective function (13). In the next sections we describe in detail each of these two inference processes, starting from the inner layer: inference in the *generalized belief space*.

B. Inner Layer: Inference in GBS

In this section we discuss the computation of the Gaussian approximation (10), whose mean value coincides with the MAP estimate (11). As this inference is performed as part of the higher-level optimization over the control (see Section III-A), the current values for $u_{k:k+l-1}$ are given in the inner inference layer. There are several complications arising when one tries to solve (11). First, in order to obtain the distribution $gb\left(X_{k+l}\right)$ one has to marginalize the latent variables $\Gamma_{k+1:k+l}$, according to (9). Second, contrarily to standard estimation problems (in which one would resort to numerical optimization techniques to solve (11)) the predicted belief $gb\left(X_{k+l}\right)$ is a function of $Z_{k+1:k+l}$, which are unknown at planning time. Therefore, rather than computing a single vector representing the MAP estimate, here the objective is to compute a vector-valued function of $Z_{k+1:k+l}$, which gives the MAP estimate for each possible value of $Z_{k+1:k+l}$. We solve these two complications exploiting two tools: *expectation-maximization* and a Gauss-Newton approach.

According to the *expectation-maximization* (EM) approach, instead of minimizing directly the probability in (11) (which requires marginalization of $\Gamma_{k+1:k+l}$), we minimize the following upper bound of $-\log gb\left(X_{k+l}\right)$:

$$X^*_{k+l} = \arg\min_{X_{k+l}} \quad (15)$$
$$\underset{\Gamma_{k+1:k+l}|\bar{X}_{k+l}}{\mathbb{E}}\left[-\log p\left(X_{k+l}, \Gamma_{k+1:k+l} | \mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right)\right],$$

where the expectation is computed assuming a given nominal state $\bar{X}_{k+l}$. EM guarantees convergence to a local minimum of the cost in (11), see e.g., [25] and the references therein. We now want to compute explicit expressions for the expectation in (15). The jont pdf over $X_{k+l}, \Gamma_{k+1:k+l}$ can be written as

$$p(X_k|\mathcal{Z}_k, \mathcal{U}_{k-1}) \prod_{i=1}^{l} p(x_{k+i}|x_{k+i-1}, u_{k+i-1}) p\left(Z_{k+i}, \Gamma_{k+i}|X^o_{k+i}\right) \quad (16)$$

and the measurement likelihood term $p\left(Z_{k+i}, \Gamma_{k+i}|X^o_{k+i}\right)$ at each look-ahead step i can be further expanded as

$$\prod_j p\left(z_{k+i,j}|X^o_{k+i,j}, \gamma_{k+i,j}\right) p\left(\gamma_{k+i,j}|X^o_{k+i,j}\right) \quad (17)$$

Plugging in Eqs. (16)-(17) into Eq. (15), recalling the Gaussian motion and observation models (6)-(7), and taking the expectation, results in

$$X^*_{k+l} = \underset{X_{k+l}}{\arg\min} \left\|X_k - \hat{X}^*_k\right\|^2_{I_k} + \sum_{i=1}^{l} \left\|x_{k+i} - f\left(x_{k+i-1}, u_{k+i-1}\right)\right\|^2_{\Omega_w} + \sum_{i=1}^{l}\sum_{j=1}^{n_i} p\left(\gamma_{k+i,j} = 1|\bar{X}_{k+l}\right) \left\|z_{k+i,j} - h\left(X^o_{k+i,j}\right)\right\|^2_{\Omega_v}, \quad (18)$$

where we used the standard notation $\|y-\mu\|^2_\Omega = (y-\mu)^T \Omega (y-\mu)$ for the Mahalanobis norm, with Ω being the information matrix. We also exploited the fact that the latent variable $\gamma_{k+i,j}$ is binary and, by definition, no observation is taken for $\gamma_{k+i,j} = 0$. The expression $p\left(\gamma_{k+i,j} = 1|\bar{X}_{k+l}\right)$ represents the probability of acquiring measurement j at time t_i assuming that the state is $\bar{X}_{k+l}$. This probability may depend on the sensor equipment of the robot, although there are grounded ways to compute it: for instance, one can assume that the probability of observing a landmark j decreases with the distance of the robot from the landmark (recall that $\bar{X}_{k+l}$ contains estimates of both robot and landmark positions) and becomes zero outside a given *sensing* radius. If we define $\bar{\Omega}^{ij}_v = p\left(\gamma_{k+i,j} = 1|\bar{X}_{k+l}\right)\Omega_v$, we can rewrite (19) as:

$$X^*_{k+l} = \underset{X_{k+l}}{\arg\min} \left\|X_k - \hat{X}^*_k\right\|^2_{I_k} + \sum_{i=1}^{l}\sum_{j=1}^{n_i} \left\|z_{k+i,j} - h\left(X^o_{k+i,j}\right)\right\|^2_{\bar{\Omega}^{ij}_v} + \sum_{i=1}^{l} \left\|x_{k+i} - f(x_{k+i-1}, u_{k+i-1})\right\|^2_{\Omega_w}, \quad (19)$$

which suggests the following interpretation: taking the expectation over the binary variables produces a *scaling* effect over the measurement information matrices Ω_v; in particular, according to $\bar{\Omega}_v^{ij} = p\left(\gamma_{k+i,j}=1|\bar{X}_{k+l}\right)\Omega_v$, a low probability $p\left(\gamma_{k+i,j}=1|\bar{X}_{k+l}\right)$ is naturally modeled in EM by decreasing the information content of the measurement. In the limit case $p\left(\gamma_{k+i,j}=1|\bar{X}_{k+l}\right) = 0$, the matrix $\bar{\Omega}_v^{ij}$ only contains zeros, and the term $\left\|z_{k+i,j}-h\left(X^o_{k+i,j}\right)\right\|^2_{\bar{\Omega}_v^{ij}}$ disappears from the sum, correctly modeling that if $p\left(\gamma_{k+i,j}=1|\bar{X}_{k+l}\right)=0$, then the measurement j is not actually acquired.

The EM framework allowed to transform the original problem (11) into (19). However, problem (19) is still difficult to solve, since, in general, $f(\cdot)$ and $h(\cdot)$ are nonlinear functions. In estimation a standard way to solve the minimization problem (19) is the Gauss-Newton method, where a single iteration involves linearizing the above equation about the current estimate $\bar{X}_{l+l}$, calculating the delta vector ΔX_{k+l} and updating the estimate $\bar{X}_{l+l} \leftarrow \bar{X}_{l+l}+\Delta X_{k+l}$. This process should be repeated until convergence. While this is standard practice in information fusion, what makes it challenging in the context of planning is that the observations $Z_{k+1:k+l}$ are unknown and considered as random variables. In order to perform a Gauss-Newton iteration on (19), we linearize the motion and observation models in Eqs. (6) and (7) about the linearization point $\bar{X}_{k+l}\left(u_{k:k+l-1}\right)$. The linearization point for the existing states at planning time is set to $\hat{X}_k$, while the future states are predicted via the motion model (7) using the current values of the controls $u_{k:k+l-1}$:

$$\bar{X}_{k+l}\left(u_{k:k+l-1}\right) \equiv \begin{pmatrix} \bar{X}_k \\ \bar{x}_{k+1} \\ \bar{x}_{k+2} \\ \vdots \\ \bar{x}_{k+l} \end{pmatrix} \doteq \begin{pmatrix} \hat{X}_k \\ f\left(\hat{x}_{k|k},u_k\right) \\ f\left(\bar{x}_{k+1},u_{k+1}\right) \\ \vdots \\ f\left(\bar{x}_{k+l-1},u_{k+l-1}\right) \end{pmatrix}. \tag{20}$$

Using this linearization point, Eq. (19) turns into:

$$\underset{\Delta X_{k+l}}{\arg\min} \|\Delta X_k\|^2_{I_k} + \sum_{i=1}^{l}\left\|\Delta x_{k+i}-F_i\Delta x_{k+i-1}-b_i^f\right\|^2_{\Omega_w} + \sum_{i=1}^{l}\sum_{j=1}^{n_i}\left\|H_i\Delta X^o_{k+i}-b^h_{i,j}\right\|^2_{\bar{\Omega}_v^{ij}}, \tag{21}$$

where the Jacobian matrices $F_i \doteq \nabla_x f$ and $H_i \doteq \nabla_x h$ are evaluated about $\bar{X}_{k+l}\left(u_{k:k+l-1}\right)$. The right hand side vectors b_i^f and $b^h_{i,j}$ are defined as

$$b_i^f \doteq f\left(\bar{x}_{k+i-1},u_{k+i-1}\right)-\bar{x}_{k+i}, \tag{22}$$

$$b^h_{i,j}\left(z_{k+i,j}\right) \doteq z_{k+i,j}-h\left(\bar{X}^o_{k+i,j}\right) \tag{23}$$

Note that $b^h_{i,j}$ is a function of the random variable $z_{k+i,j}$. Also note that under the maximum-likelihood assumption this terms would be nullified: assuming maximum likelihood measurements essentially means assuming zero *innovation*, and $b^h_{i,j}$ is exactly the innovation for measurement $z_{k+i,j}$. We instead keep, for now, the observation $z_{k+i,j}$ as a variable and we will compute the expectation over this random variable only when evaluating the objective function (13). In order to calculate the update vectors ΔX_k and $\Delta x_{k+1},\ldots,\Delta x_{k+l}$, it is convenient to write Eq. (21) in a matrix formulation, which can be compactly represented as:

$$\left\|\mathcal{A}_{k+l}(u_{k:k+l-1})\Delta X_{k+l}-\breve{b}_{k+l}(u_{k:k+l-1},Z_{k+1:k+l})\right\|^2, \tag{24}$$

where $\mathcal{A}_{k+l}$ is built by stacking the appropriate Jacobian matrices in (21), and $\breve{b}_{k+l}$ is obtained by stacking the right hand side vectors b_i^f and $b^h_{i,j}$. The update vector ΔX_{k+l}, that minimizes (24), is given by

$$\Delta X_{k+l}(u_{k:k+l-1},Z_{k+1:k+l}) \doteq \left(\mathcal{A}^T_{k+l}\mathcal{A}_{k+l}\right)^{-1}\mathcal{A}^T_{k+l}\breve{b}_{k+l}. \tag{25}$$

Using the vector ΔX_{k+l} in (25) we can update the nominal state $\bar{X}_{k+l}$:

$$\hat{X}_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right) = \bar{X}_{k+l}+\Delta X_{k+l} \tag{26}$$

We can also compute a local approximation of the information matrix of the estimate as:

$$I_{k+l}\left(u_{k:k+l-1}\right) \doteq \mathcal{A}^T_{k+l}\mathcal{A}_{k+l}, \tag{27}$$

The estimate $\hat{X}_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right)$ is the outcome of a single iteration of the nonlinear optimization (19).
We note that the (random) measurements $z_{k+l,j}$ only appear in $\breve{b}_{k+l}$. Moreover, since $\breve{b}_{k+l}$ is obtained by stacking vectors b_i^f and $b^h_{i,j}$, and observing that each $b^h_{i,j}$ is a linear function of the corresponding measurement $z_{k+l,j}$, it follows that each entry in $\breve{b}_{k+l}$ is a *linear* function of the measurements $Z_{k+1:k+l}$. This, in turn, implies that $\Delta X_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right)$ and $\hat{X}_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right)$ are also linear functions of the measurements. This fact greatly helps when taking the expectation over $Z_{k+1:k+l}$ of the immediate cost function (13). Considering more iterations would better capture the dependence of the estimate on the measurements; however, more iterations would make $\hat{X}_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right)$ a nonlinear function of the measurements, making it challenging to devise explicit expressions for (13). We currently assume a single iteration sufficiently captures the effect of the measurements for a certain control action on the GBS. Therefore,

$$\hat{X}^*_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right)=\hat{X}_{k+l}\left(u_{k:k+l-1},Z_{k+1:k+l}\right). \tag{28}$$

Thus, according to the derivation in this section we are now able to compute a predicted belief at the l look-ahead step, which is parametrized as a Gaussian with mean (28) and information matrix (27). In the next section we discuss how to use the predicted belief in the outer layer of our approach.

C. Outer Layer: Inference over the Control

Finding a locally-optimal control policy $u^*_{k:k+L-1}$ corresponds to minimizing the general objective function (13). The *outer* layer is an iterative optimization over the nonlinear function $J_k\left(u_{k:k+L-1}\right)$. In each iteration of this layer we are looking for the delta vector $\Delta u_{k:k+L-1}$ that is used to update the current values of the controls:

$$u^{(i+1)}_{k:k+L-1} \leftarrow u^{(i)}_{k:k+L-1}+\Delta u_{k:k+L-1}, \tag{29}$$

where i denotes the iteration number. Calculating $\Delta u_{k:k+L-1}$ involves computing the GBS of all the L look-ahead steps based on the current value of the controls $u^{(i)}_{k:k+L-1}$. This process of calculating the GBS is by itself a non-linear optimization and has been already described in Section III-B. The GBS $gb\left(X_{k+l}\right)$, given the current values of the controls $u_{k:k+L-1}$, is represented by the mean $\hat{X}^{*}_{k+l}\left(Z_{k+1:k+l}\right)$ and the information matrix I_{k+l}. The mean is a linear function in the unknown observations $Z_{k+1:k+l}$. The immediate cost function, in the general case, may involve both the mean and the information matrix, and is therefore also a function of $Z_{k+1:k+l}$. Taking the expectation over these random variables produces the expected cost that is only a function of $u_{k:k+L-1}$ and captures the effect of the current controls on the lth look-ahead step.

We conclude this section by noting that the control update (29) is performed in a *continuous* domain and can be realized using different optimization techniques (e.g., dynamic programming, gradient descent, Gauss-Newton).

IV. Experiments

We evaluate our approach in a simulated scenario where the robot has to navigate to different goals in an unknown environment, assuming no sources of absolute information are available (e.g., no GPS). Results were obtained on an Intel i7-2600 processor with a 3.40GHz clock rate and 16GB of RAM, using a single-threaded implementation.

The specific objective function considered in this study is (12) with the immediate cost functions defined as

$$\begin{aligned} c_{l}\left(gb\left(X_{k+l}\right), u_{k+l}\right) &\doteq tr\left(M_{\Sigma} I^{-1}_{k+l} M^{T}_{\Sigma}\right) + \left\| \zeta\left(u_{k+l}\right)\right\|_{M_{u}} \\ c_{L}(gb\left(X_{k+L}\right)) &\doteq \left\| E^{G}_{k+L}\hat{X}^{*}_{k+L} - X^{G}\right\|_{M_{x}} + tr\left(M_{\Sigma} I^{-1}_{k+L} M^{T}_{\Sigma}\right), \end{aligned} \tag{30}$$

where $tr(\cdot)$ returns the trace of a matrix. The immediate cost function c_l, with $l \in [0, L-1]$, involves both the robot uncertainty and control usage terms, while the terminal cost c_L penalizes over the distance between the robot to a desired (given) goal X^G as well as robot uncertainty. The matrices M_Σ, M_u and M_x are weight matrices; in our example M_Σ is constructed such that it extracts robot covariance at the appropriate time instant (i.e., robot uncertainty at each look ahead step) from the overall state covariance I^{-1}_{k+l}. Likewise, the matrix E^{G}_{k+L} selects the terminal position of the robot from $\hat{X}^{*}_{k+L}$. The function $\zeta\left(u\right)$ is some known function that, depending on the application, quantifies the usage of control u. For simplicity, we assume the robot can only control its heading angle while keeping the velocity constant, and therefore define the function $\zeta\left(u\right)$ as the change in the heading angle.

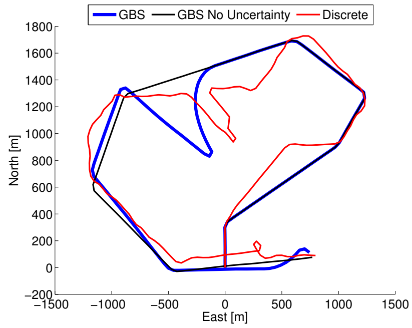
The matrix M_u has a very intuitive function - larger values induce conservative policies, penalizing for extensive use of control. The choice of the matrices M_Σ and M_x is instead less intuitive. A balance between these two matrices is crucial for letting the robot satisfy the concurrent tasks of reaching a goal and minimizing the estimation uncertainty. Instead of assuming the weight on the uncertainty and the goal terms in Eqs. (30) to be constant (i.e., determined by the matrices M_Σ and M_x), we express the tradeoff between these two aspects by introducing a parameter α that is adaptively computed, according to the robot uncertainty and a *user-specified uncertainty threshold* β, as $\alpha = \frac{tr\left(M_{\Sigma} I^{-1}_{k+L} M^{T}_{\Sigma}\right)}{\beta}$. The uncertainty terms in Eqs. (30) are then pre-multiplied by α and the goal term (in second equation) is pre-multiplied by $(1-\alpha)$. When robot uncertainty is close to β, α and $(1-\alpha)$ will be close to 1 and 0 respectively, prioritizing uncertainty reduction over reaching the goal. As we are not imposing a hard constraint on $tr\left(M_{\Sigma} I^{-1}_{k+L} M^{T}_{\Sigma}\right)$, this term may become larger than β and for that reason the final definition of α is changed into $\alpha = \min\left(\frac{tr\left(M_{\Sigma} I^{-1}_{k+L} M^{T}_{\Sigma}\right)}{\beta}, 1\right)$.

The control is found by minimizing the objective function (12), according to our dual-layer inference approach. We use a gradient descent method for optimizing the outer layer and the approach of Section III-B for calculating inference in the inner layer. The number of look-ahead steps (L) is set to 5.

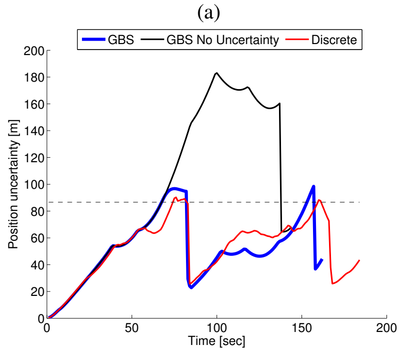
We assume onboard range and camera sensors that allow the robot to detect and measure the relative positions of nearby landmarks, and consider the measurements from these sensors to be corrupted by a Gaussian noise with standard deviation of 1 meter and 0.5 pixels, respectively. We use sensing radius of 1000 meters and consider observations of landmarks only within this radius. The motion model is represented by a zero-mean Gaussian with standard deviation of 0.05 meters in position and 0.5 degrees in orientation. The robot starts at the origin, without prior knowledge of the environment, and has to reach a sequence of goals (labeled from 1 to 8 in Figure 1a-c).

In this study we compare our method to a discrete planning approach and also demonstrate the importance of incorporating uncertainty into the planning by dropping the uncertainty terms from Eqs. (30). In discrete planning we closely follow Kim and Eustice [18], who recently considered a robotic coverage problem, where the planning generates trajectories to promising loop-closure candidates and the decision whether to continue exploration or to pursue one of these trajectories is made according to the robot current uncertainty and the expected uncertainty gain by performing a loop closure. The authors discretize the space and use a global A* algorithm [7], with the heuristic function weighted by local *saliency* of each of the nodes in the grid to prioritize trajectories going through high-saliency nodes. The reader is referred to [18] for further details. In our implementation of this discrete planning approach, we define saliency to be 1 within sensing radius of an observed landmark and 0 otherwise. Similarly to [18], to avoid treating each mapped landmark as loop closure candidate, we perform clustering of nearby landmarks and consider cluster centers as loop closure candidates.

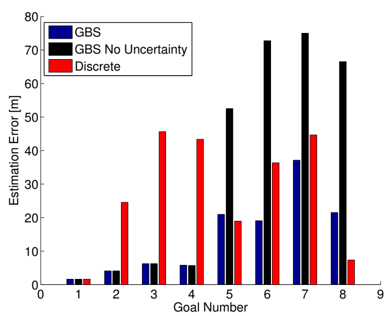
Figure 1 provides the obtained results in each of the three compared methods (*planning in the GBS*, *planning in the GBS without uncertainty*, and *discrete planning*). Each plot shows the actual and estimated robot trajectories, denoted respectively by blue and red lines, the robot uncertainties (at the end of the scenario), the 8 considered goals and the landmarks. The mapped landmarks and the associated uncertainty are denoted by green color; ground truth for *all* the landmarks is shown in blue.



(a)



(b)



(c)

Fig. 2: (a) Comparison between trajectories from Figure 1. Planning in the GBS produces continuous trajectories. (b) Position uncertainty for each of the compared methods. The *soft* uncertainty bound is also shown as a dashed line. (c) Estimation error when reaching the goals (*miss distance*).

As observed in Figure 1a, in the proposed method (planning in the GBS), the robot continues moving to the goal, while mapping nearby landmarks, as long as the uncertainty does not exceed a threshold. Once robot uncertainty is too high, the planner first guides the robot to previously observed landmarks to perform loop closure (on the way from goal 4 to goal 5, and from goal 7 to goal 8), and only then the robot proceeds to the appropriate goal. Uncertainty drops to lower values each time a loop closure is performed, as shown in Figure 2b. As we use a soft threshold on the uncertainty, performing loop closure is determined not only by the robot uncertainty levels but also by the weight matrices M_Σ, M_u and M_x in Eqs. (30). Therefore, the robot uncertainty can still exceed the uncertainty threshold by some amount.

Method	Trajectory length [m]	Duration [sec]
GBS	**8068**	**163**
GBS no uncertainty	7185	145
Discrete planning	9070	185

TABLE I: Trajectory length and duration.

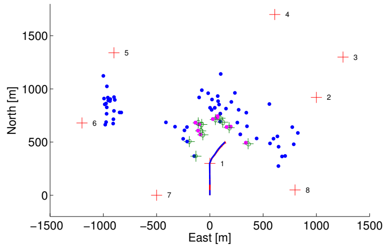
Figure 1b shows the results when planning is done without incorporating uncertainty. While the overall trajectory is shorter (Table I), the uncertainty of robot pose and of newly-observed landmarks constantly develops over time (see Figure 2b) leading to high estimation errors when reaching the goal (i.e. high *miss distance*), as shown in Figure 2c.

Discrete planning produces the result shown in Figure 1c. Observe the trajectories are no longer smooth as in the continuous case and look less natural; see also the trajectories overlay in Figure 2a. As an example, we elaborate in Figure 3 on the two-phase trajectory segment when traveling from goal 1 to goal 2. The robot position and observed landmarks at planning time are shown in Figure 3a, while Figure 3b illustrates the A*-generated trajectories. As discussed, from the robot current position, discrete planning engages A* to generate a trajectory for each of the landmark cluster centers and to the goal. In the current case, the objective function evaluated for the goal trajectory yields the lowest cost and is therefore selected (shown in green). However, instead of directly going to the goal (as in the continuous planning case), the generated trajectory involves two segments that seem unnecessary. Note that saliency is 1 in the entire region (according to sensing range) and thus does not have any effect, in this particular case, on the generated trajectories. We observed a similar underlying behavior also for other parts in the discrete trajectory that is shown in Figure 1c.

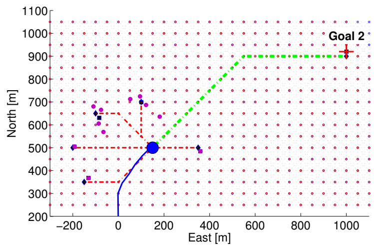
While discrete planning and our continuous planning approach result in similar levels of robot uncertainty (Figure 2b), in this example the former generates considerably longer trajectories (Table I). As to the estimation errors, although no general statements can be made at this point, in the considered example we observe them to be higher in the discrete case, as shown in Figure 2c. In this particular case, the high levels of estimation errors (e.g. goals 2 and 3) stem from the two-segment trajectory on the way from goal 1 to goal 2, that is longer than the trajectory generated by planning in the GBS and lead accordingly to further development of uncertainty.

V. Conclusions

This work investigates the problem of planning under uncertainty, and addresses several limitations of the state-of-the-art techniques, namely (i) *state or control discretization*, (ii) assumption of *maximum likelihood observations*, and (iii) assumption of *prior knowledge about the environment* in which the robot operates. We propose a planning approach that does not resort to discretization, but directly operates in a continuous domain. The approach is based on a dual-layer architecture, with an *inner inference layer*, that is in charge



(a)



(b)

Fig. 3: Discrete planning using A*. (a) Trajectory and mapped landmarks at planning time. (b) Generated trajectories to the goal and to each of the landmark clusters are shown in red. Trajectory corresponding to the lowest cost is shown in green. Robot position at planning time is denoted by filled blue circle, mapped landmarks are represented by purple circle marks. Grid resolution is 50 meters.

of predicting the belief on robot and world states, and *outer inference layer*, which has to compute an optimal control strategy using the predictions of the inner layer. In the inner inference layer, we treat future measurements as random variables, hence avoiding the assumption of maximum likelihood observations, common in related work. We also include random binary variables to model the fact that it is not known in advance whether a measurement is acquired or not. Therefore, we introduce the concept of *generalized belief,* that besides robot and world state, models also the joint probability distributions over future measurements and latent binary variables. We evaluate the approach in numerical experiments and compare it with a recent related work, based on discretization.

References

[1] H. Bai, H. David, and W.S. Lee. Integrated perception and planning in the continuous space: A POMDP approach. In *Robotics: Science and Systems (RSS)*, 2013.

[2] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.

[3] J. Binney and G. S. Sukhatme. Branch and bound for informative path planning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2147–2154, 2012.

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

[5] M. Bryson and S. Sukkarieh. Observability analysis and active control for airborne SLAM. *IEEE Trans. Aerosp. Electron. Syst.*, 44:261–280, 2008.

[6] H. Carrillo, I. Reid, and J.A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2080–2087, 2012.

[7] Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of a*. *Journal of the ACM (JACM)*, 32(3):505–536, 1985.

[8] J. Du, L. Carlone, M. Kaouk Ng, B. Bona, and M. Indri. A comparative study on active SLAM and autonomous exploration with particle filters. In *Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pages 916–923, 2011.

[9] G. Hollinger and G. Sukhatme. Stochastic motion planning for robotic information gathering. In *Robotics: Science and Systems (RSS)*, 2013.

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

[11] R. Platt Jr, R. Tedrake, L.P. Kaelbling, and T. Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, pages 587–593, 2010.

[12] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[13] L.P. Kaelbling and T. Lozano-Pérez. Pre-image backchaining in belief space for mobile manipulation. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2011.

[14] L.P. Kaelbling and T. Lozano-Pérez. Unifying perception, estimation and action for mobile manipulation via belief space planning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2952–2959, 2012.

[15] L.P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *Intl. J. of Robotics Research*, 2013.

[16] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.

[17] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[18] A. Kim and R.M. Eustice. Perception-driven navigation: Active visual SLAM for robotic area coverage. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.

[19] K. Konolige, G. Grisetti, R. Kuemmerle, W. Burgard, L. Benson, and R. Vincent. Efficient sparse pose adjustment for 2D mapping. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 22–29, Taipei, Taiwan, Oct 2010.

[20] M. Kontitsis, E.A. Theodorou, and E. Todorov. Multi-robot active slam with relative entropy optimization. In *American Control Conference*, 2013.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[22] C. Leung, S. Huang, and G. Dissanayake. Active SLAM using model predictive control and attractor based exploration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5026–5031, 2006.

[23] R. Martinez-Cantin, N. de Freitas, E. Brochu, J.A. Castellanos, and A. Doucet. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2):93–103, 2009.

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

[25] T.P. Minka. Expectation-Maximization as lower bound maximization. November 1998.

[26] C. Potthast and G. Sukhatme. Next best view estimation with eye in hand camera. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[27] Samuel Prentice and Nicholas Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 28(11-12):1448–1465, 2009.

[28] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 661–666, 2005.

[29] A. Singh, A. Krause, C. Guestrin, and W.J. Kaiser. Efficient informative sensing using multiple robots. *J. of Artificial Intelligence Research*, 34:707–755, 2009.

[30] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, pages 65–72, 2005.

[31] C. Stachniss, D. Haehnel, and W. Burgard. Exploration with active loop-closing for FastSLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[32] R. Valencia, M. Morta, J. Andrade-Cetto, and J.M. Porta. Planning reliable paths with pose SLAM. *IEEE Trans. Robotics*, 2013.