FIRM: Sampling-based Feedback Motion Planning Under Motion Uncertainty and Imperfect Measurements

Ali-akbar Agha-mohammadi, Suman Chakravorty, Nancy M. Amato

Abstract

In this paper we present FIRM (Feedback-based Information RoadMap), a multi-query approach for planning under uncertainty, that is a belief-space variant of Probabilistic Roadmap Methods (PRMs). The crucial feature of FIRM is that the costs associated with the edges are independent of each other, and in this sense it is the first method that generates a graph in belief space that preserves the optimal substructure property. From a practical point of view, FIRM is a robust and reliable planning framework. It is robust since the solution is a feedback and there is no need for expensive replanning. It is reliable because accurate collision probabilities can be computed along the edges. In addition, FIRM is a scalable framework, where the complexity of the planning with FIRM is a constant multiplier of the complexity of planning with PRM. In this paper, FIRM is introduced as an abstract framework. As a concrete instantiation of FIRM, we adopt Stationary Linear Quadratic Gaussian (SLQG) controllers as belief stabilizers and introduce the so-called SLQG-FIRM. In SLQG-FIRM we focus on kinematic systems and then extend to dynamical systems by sampling in the equilibrium space. We investigate the performance of SLQG-FIRM in different scenarios.

I. Introduction

Decision making under uncertainty is a crucial ability for most robotic systems. In the presence of uncertainty in a robot's motion and uncertainty in its sensory readings, the true robot state is not available for decision-making purposes. In such cases, a state estimation module can provide a probability distribution over all possible states, referred to as *information-state* or *belief*. Therefore, decision-making under motion and sensing uncertainties needs to be performed in the information space (belief space). In its most general form, this decision-making can be formulated as a Partially Observable Markov Decision Process (POMDP) problem [Astrom, 1965], [Smallwood and Sondik, 1973], [Kaelbling et al., 1998]. However, only a very small class of problems formulated using POMDP can be solved exactly due to its computational complexity [Papadimitriou and Tsitsiklis, 1987], [Madani et al., 1999]. In particular, planning (i.e., solving POMDPs) over continuous state, control and observation spaces is a big challenge.

On the other hand, in the absence of uncertainty, sampling-based path planning algorithms including graph-based methods such as Probabilistic Roadmap Methods (PRM) [Kavraki et al., 1996] and their variants (e.g., [Amato et al., 1998]) and tree-based methods such as Rapidly exploring Randomized Trees (RRT) [Lavalle and Kuffner, 2001], expansive space tree [Hsu, 2000], and their variants (e.g., [Karaman and Frazzoli, 2011]) have shown great success in solving robot motion planning problems. Nevertheless, direct transformation of the roadmap-based methods to planning under uncertainty (in belief space) is a challenge for two main reasons. The first issue is ensuring that the roadmap nodes are reachable. The second challenge is that the incurred costs on different edges of the roadmap depend on each other, which violates a basic assumption in roadmap-based methods that each roadmap edge represents an independent planning problem.

In this paper, we generalize the PRM framework to obtain the Feedback-based Information RoadMap (FIRM) framework that takes into account both motion and sensing uncertainties. FIRM is constructed as a roadmap (graph) in the belief space, where graph nodes are beliefs (rigorously speaking, small subsets of the belief space) and edges are local controllers in belief space. FIRM is an abstract generic framework that relies on the existence of an appropriate belief node sampler and connector (local controller). We also construct a Stationary Linear Quadratic Gaussian controller-based (SLQG-based) instantiation of this generic framework, called SLQG-FIRM, where we provide a specific node sampler and connector. In SLQG-FIRM we first focus on the kinematic systems and then extend it to dynamical systems by restricting sampling space to the equilibrium space. The SLQG-FIRM is the first method that generalizes the PRM to the belief space such that the incurred costs on different edges of the roadmap are independent of each other, while providing a straightforward approach to sample reachable belief nodes. This property is a direct consequence of utilizing feedback controllers in the construction of FIRM. Based on this property, the FIRM framework breaks the curse of history in POMDPs [Pineau et al., 2003], and provides the optimal *feedback policy* over the roadmap instead of returning a single nominal path.

Figure 1 illustrates the problem of edge dependence in the direct transformation of PRM to stochastic domains. It also shows the approach of FIRM in generating a graph in belief space with independent edges. Figure 1(a) depicts a simple PRM in the state space with twelve nodes $\mathcal{V} = \{\mathbf{v}^0, \cdots, \mathbf{v}^{11}\}$. Figure 1(b) shows the belief evolution on the underlying PRM. Assuming

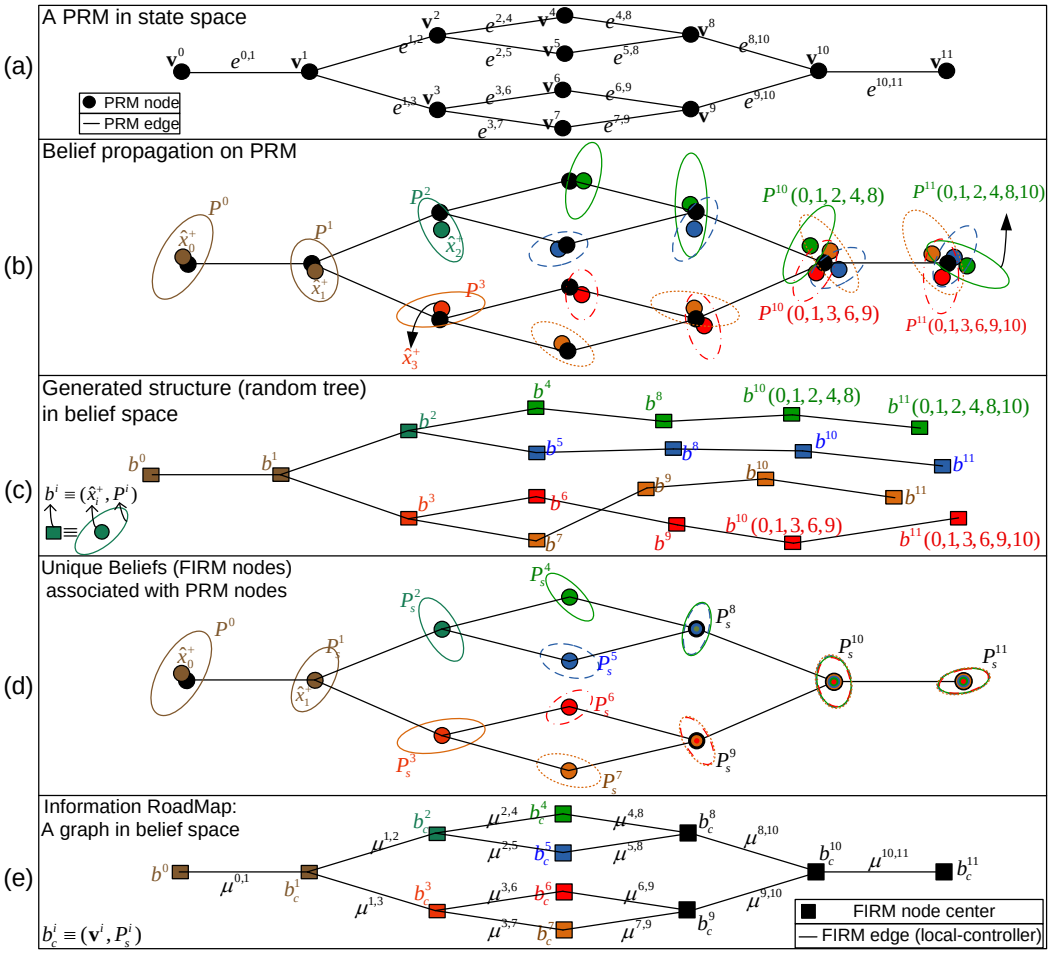


Fig. 1. (a) A simple PRM in state-space. (b) Assuming Gaussian belief space, belief snapshots along different paths starting from $\mathbf{v}^0$ ending at $\mathbf{v}^{11}$ are shown. As it is seen, the obtained belief depends on the traveled path by robot. For example $P^{11}(0,1,3,6,9,10)$ denotes the estimation covariance at node $\mathbf{v}^{11}$, when the robot has traversed a path through nodes $(0,1,3,6,9,10)$ prior to node 11. (c) Corresponding belief paths in the belief space. Belief at each node depends on the initial belief, taken actions (edges), and obtained observations (random). Therefore, the generated structure in the belief space is not a graph but is a random tree. (d) Unique beliefs assigned to each PRM node. Using stabilizers, regardless of the action and observation history, the belief at each node stops at these predefined beliefs. (e) The FIRM corresponding to the given PRM. b_c^i's are graph nodes in the belief space and μ^{ij}'s are local planners (graph edges).

the belief is Gaussian in this example, we represent a point in belief space using a mean $\widehat{x}^+$ and a covariance P, i.e., a belief b is characterized by the pair $b \equiv (\widehat{x}^+, P)$. In Fig. 1(b), mean values are shown by small filled circles, and covariance matrices are shown by their corresponding 3σ ellipse centered at the mean. We drive the system from $\mathbf{v}^0$ toward the node $\mathbf{v}^{11}$. The initial belief at node $\mathbf{v}^0$ is $b^0 \equiv (\widehat{x}_0^+, P^0)$. The belief propagation from left to right starting from b_0 is shown in Fig. 1(b).

Although there exists a single edge $e^{(10,11)}$ between nodes $\mathbf{v}^{10}$ and $\mathbf{v}^{11}$ in PRM (cf. Fig. 1(a)), the belief evolution along $e^{(10,11)}$ is not unique (cf. Fig. 1(b-c)) since it depends on *(i)* the initial belief, *(ii)* obtained observations (observation history), and *(iii)* taken path (action history) that has led to $\mathbf{v}^{10}$. Figure 1(c) shows the corresponding belief propagation in the belief space, where each rectangle encodes a mean and covariance. As seen in Fig. 1(c), the belief paths do not form a graph; rather, they form a random tree in belief space. Hence, in practice, where observations are random, not only does the number of possible beliefs grow exponentially, but the belief also evolves randomly. Therefore, to predict edge costs, full knowledge of the belief at the start of the edge is required. In turn, this requires full knowledge of the history of observations and actions leading up to the start of the edge. Even if future observations were assumed to be deterministic for the purpose of planning, the generated structure would still be a tree that grew exponentially in the size of the underlying PRM graph.

In FIRM, we use local feedback planners to drive the belief process toward the predefined unique beliefs associated with PRM nodes (cf. Fig. 1(d)). As a result, the evolution of belief after a FIRM node is reached is independent of the evolution of belief before that node is reached. This breaks the curse of history, allowing us to construct a PRM-like roadmap in the belief space with independent edge costs. Therefore, in contrast to the main body of the literature in motion planning under uncertainty, FIRM can be re-used for future queries and does not need to reconstruct the roadmap every time a new query is submitted.

From an algorithmic perspective, this edge independence is an example of the optimal substructure property. A problem has an optimal substructure only if the optimal solution can be obtained from a combination of optimal solutions to its subproblems

[Cormen et al., 2001]. To solve a problem using Dynamic Programming (DP) or its successive approximation schemes such as Dijkstra's algorithm, the optimal substructure assumption has to hold [Sniedovich, 2006], i.e., the cost of any subpath has to be independent of what precedes it and what succeeds it. As mentioned, the direct transformation of sampling-based methods to belief space breaks this assumption, while FIRM preserves it. Furthermore, edge independence allows the challenging task of computing collision probabilities to be done offline, for each edge separately, without performing costly computations repeatedly and without any simplifying assumption.

The current paper draws on earlier work published in conference papers [Agha-mohammadi et al., 2011], [Agha-mohammadi et al., 2012b [Agha-mohammadi et al., 2013b]. In [Agha-mohammadi et al., 2011], we presented the FIRM framework in a somewhat heuristic fashion, while in this paper, we construct the FIRM framework more rigorously by detailing the procedure of transforming the POMDP problem to the belief SMDP (Semi-Markov Decision Process) problem, and then, to the FIRM MDP (FIRM Markov Decision Process) problem, where the policy on the graph and overall hybrid policy generated by FIRM are distinguished clearly. Also, in this paper we provide a clearer distinction between the abstract FIRM framework and its instantiations, and we provide more rigorous explanation and proofs on SLQG-FIRM. Further, we append the proofs of the probabilistic completeness of FIRM to this paper, which completes the work in [Agha-mohammadi et al., 2012b]. We also present new unpublished results on the performance of SLQG-FIRM in more difficult environments, and demonstrate its real-time planning capabilities. Further, we provide a complexity analysis of the method and compare it to the state-of-art methods.

Outline: In the next section, we review the most relevant related work. Section III provides an overview of the method and its contributions. In Section IV we describe the general problem of feedback motion planning under uncertainty, present notation, and formulate the POMDP problem. In Section V, we present the SLQG-based instantiation of the abstract FIRM framework by providing concrete belief samplers and connectors (local planners). In Section VI, assuming the existence of belief samplers and connectors, we introduce the abstract FIRM framework and detail the process of transforming POMDP to a FIRM MDP. In Section VII, aiming at evaluating the quality of the FIRM solution, we extend the concepts of success and probabilistic completeness to the stochastic setting and prove the probabilistic completeness of the FIRM framework. Experimental results are presented in Section VIII. In Section IX, we discuss limitations of the framework, future work, and open issues. Section X concludes the paper.

II. Related Work

In this section we review the related work and place our work into context. First, we review the related work on planning algorithms under uncertainty and then, we consider the work concerning probabilistic completeness.

A. Planning Algorithms

Uncertainty in robotic systems usually can stem from three sources: *i)* Motion uncertainty, which results from the noise that affects system dynamics; *ii)* Sensing uncertainty, caused by noisy sensory measurements, which is also referred to as imperfect state information; *iii)* Uncertainty in environment map, such as uncertain obstacle locations or uncertain location of features (information sources) in the environment.

Methods such as [Nakhaei and Lamiraux, 2008], [Guibas et al., 2008], [Missiuro and Roy, 2006] deal with map uncertainty. However, we do not scrutinize these methods, since we assume there is no uncertainty in the environment map. Methods such as [Alterovitz et al., 2007], [Chakravorty and Kumar, 2009], [Melchior and Simmons, 2007], [Chakravorty and Kumar, 2011] exploit sampling-based motion planning ideas to deal with motion uncertainty. However, methods that are most related to FIRM consider both motion and sensing uncertainties in planning, where the ultimate goal is to solve a POMDP problem, i.e., to find the best policy that generates optimal actions as a function of belief. However, due to the intractability of the POMDP solution, the practical results using these methods are usually limited to problems with small set of discrete states [Kaelbling et al., 1998]. Point-based POMDP solvers such as [Porta et al., 2006], [Kurniawati et al., 2008], [Ong et al., 2010], [Bai et al., 2010] have increased the size of problems that can be solved by POMDPs. However, they do not handle continuous state, control, and observation spaces. For the Gaussian belief case [van den Berg et al., 2011], [van den Berg et al., 2012] handle continuous spaces locally around a given trajectory in belief space. [Platt et al., 2011] generalize the local approaches to non-Gaussian beliefs.

In continuous state, control, and observation space, the main body of methods does not follow the POMDP framework due to its extreme complexity. Instead, they return a nominal path as the solution of the planning problem, which is fixed regardless of the process and sensor noise in the execution phase. [Censi et al., 2008] propose a planning algorithm based on graph search and constraint propagation on a grid-based representation of the space. [Platt et al., 2010] plan in continuous space by finding the best nominal path using nonlinear optimization methods. In the LQG-MP method [van den Berg et al., 2010], among the finite number of RRT paths, the best path is found by simulating the performance of LQG on all RRT paths. [Bry and Roy, 2011] propose a tree-based approach, in which the underlying nominal trajectory is optimized using RRT*. Vitus and Tomlin [Vitus and Tomlin, 2011] also propose an approach to optimize the underlying trajectory by formulating the problem as a chance constrained optimal control problem. In [van den Berg et al., 2011], the authors also extend the LQG-MP

to roadmaps. [Prentice and Roy, 2009] and [Huynh and Roy, 2009] also utilize roadmap-based methods based on the PRM approach, where the best path is found through breadth-first search on the Belief Roadmap (BRM). However, in all these roadmap-based methods, the optimal substructure assumption is violated, i.e., costs of different edges on the graph depend on each other. The point-based POMDP planner in [Kurniawati et al., 2012] takes into account motion, observation, and map uncertainties and advances the previous point-based methods by introducing guided cluster sampling. It starts with a roadmap in the configuration space, and grows a single-query tree in the belief space, rooted in the initial belief.

Since these methods return a nominal path instead of a feedback policy, the path needs to be recomputed (i.e., replanning has to be performed) in the case of large deviations or when starting from a new point. However, unless the planning domain is small (e.g., [Platt et al., 2010]), replanning using these methods is computationally very expensive. The reason for this is the constructed planning tree depends on the starting belief, and therefore all computations needed to construct the tree (including predicting future costs) have to be reproduced from the new starting belief. BRM ameliorates this expensive computation using covariance factorization techniques, but it still does not satisfy the optimal substructure assumption. Thus, for a new query from a new initial point, BRM needs to perform the search algorithm again. In the presence of obstacles, recomputing the collision probabilities is also needed, which makes replanning even more expensive. In other words, these methods are single-query, in the sense that the edge costs are computed for a given query.

Since these methods are single-query, online replanning can be done only if the planning domain is small (e.g., [Platt et al., 2010]) or if the planning horizon is short, such as Receding Horizon Control-based (RHC-based) approaches (e.g., [Chakravorty and Erwin, 2011]). The method proposed in [Toit and Burdick, 2010], is an RHC-based method, where the nominal path is updated dynamically over an N-step horizon. The PUMA framework proposed in [He et al., 2011] is also an RHC-based framework, where instead of a single action, a sequence of actions (macro-action) is selected at every decision stage. However, these methods entail repeatedly solving open loop optimal control problems at every time step, which is computationally very expensive as the previous computations cannot be reused for the queries from the new initial point. In FIRM, however, a feedback policy, i.e., a mapping from belief space to actions, is computed offline. Thus in replanning from a new initial point, the computations need not be reproduced. Thus for a fixed goal, the algorithm is robust to changes in the start point of the query. It is also robust to changes in the goal point, because graph feedback can be computed (see Eq. (32)) online, which results in a multi-query roadmap in the belief space.

In the methods that account for sensing uncertainty, the state has to be estimated based on measurements. To handle unknown future measurements in the planning stage, methods in [Censi et al., 2008],[Platt et al., 2010], [Prentice and Roy, 2009], [Huynh and Roy, 2009], [Toit and Burdick, 2010] consider the maximum likelihood (ML) observation sequence to predict the estimation performance. In contrast, FIRM takes all possible future observations into account in the planning stage. Methods in [van den Berg et al., 2010] and [van den Berg et al., 2011] also consider all possible future observation.

In the presence of obstacles, due to the dependency of collision events in different time steps, it is a burdensome task to include the collision probabilities in planning. Thus, the methods in [Censi et al., 2008], [van den Berg et al., 2010], [Toit and Burdick, 2010], [van den Berg et al., 2011] design some safety measures to account for obstacles in planning. A problem with some of these collision probability measures is that they are built on the assumption that the collision probabilities at different stages along the path are independent of each other, which is not true in general and may lead to very conservative plans (cf. Fig. 2). As a result, different methods (e.g. [Patil et al., 2012]) aim at providing more accurate and faster ways of computing collisions probabilities. In FIRM, however, collision probabilities can be computed and seamlessly incorporated into the planning stage without making any simplifying assumptions.

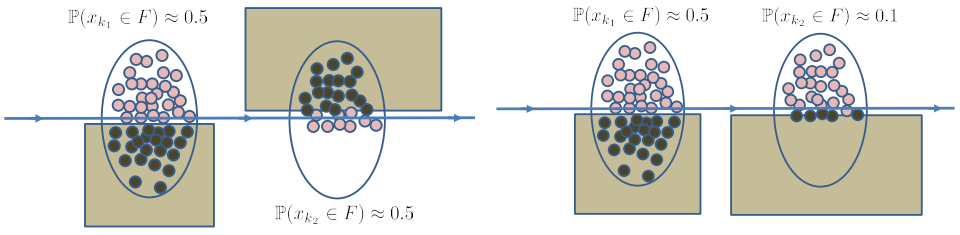


Fig. 2. This figure shows the dependence of the collision probability in time step k_1 and k_2, i.e., $\mathbb{P}(x_{k_1} \in F)$ and $\mathbb{P}(x_{k_2} \in F)$, where x_k is the robot state at time step k and F is the obstacle set (shown by rectangles). Drawn ellipses are 3σ ellipses of Gaussian distributions obtained by Kalman filtering. Also, the samples in Monte-carlo simulation are shown by small circles. The dark ones have collided with obstacles and do not get propagated, and the light ones are the safe samples. Although the overall collision probability in Fig. (a) is much more than the collision probability Fig. (b), simplified safety measures based on ellipse-obstacle intersection area lead to the same safety measure in Fig. (a) and (b), and are unable to capture this dependency.

B. Probabilistic Completeness

Due to the success of sampling-based methods in many practical planning problems, researchers have investigated the theoretical basis for this success. However, almost all of these investigations have been performed for algorithms that are

designed for planning in the absence of uncertainty. The literature in this direction falls into two categories: path isolation-based methods and space covering-based methods.

Path isolation-based analysis: In this approach, one path is chosen, and it is tiled with some sets such as ϵ-balls [Kavraki et al., 1998] or sets with arbitrary shapes but strictly positive measures [Ladd and Kavraki, 2004]. Then the success probability is analyzed by investigating the probability of sampling in each of the sets that tile the given path in the obstacle-free space. Methods in [Kavraki et al., 1998], [Ladd and Kavraki, 2004], [Švestka and Overmars, 1997], and [Bohlin, 2002] are among those that perform path isolation-based analysis of the planning algorithm.

Space Covering-based analysis: In space covering-based analysis, an adequate number of sampled points needed to find a successful path is expressed in terms of a parameter ϵ, which is a property of the environment. A space is ϵ-good if every point in the state space can be connected to at least an ϵ fraction of the space using local planners. Methods in [Hsu, 2000] and [Kavraki et al., 1995] are among these.

These methods were developed for the situation where the desired result from the planning algorithm is a path. However, in the presence of uncertainty, the concept of "successful path" is no longer meaningful, because on a given path, different policies may result in different success probabilities, where some are interpreted as successful and some are not. Thus, since the planning algorithm returns a policy instead of a path, the success has to be defined for a policy. This paper extends these concepts to probabilistic spaces, i.e., to sampling-based methods concerning planning under uncertainty. In Section VII, we define and formulate the concepts of successful policy and probabilistic completeness under uncertainty.

III. Method overview and contributions

The highlights and contributions of this paper can be divided into theoretical and practical parts. The theoretical highlights can be summarized as follows:

- *Abstract frameworks:* We introduce the abstract Information RoadMap (IRM) framework as a graph in the belief space, where the graph nodes are beliefs (rigorously speaking, small subsets of the belief space) and edges are local controllers. The abstract FIRM framework is defined as an IRM where local controllers are feedback controllers. These abstract frameworks rely on the existence of an appropriate belief node sampler and connector (local controller) and are general enough to capture any form of belief. Discussing the concept of belief reachability under feedback controllers, we detail the reduction of a POMDP to a tractable MDP on the FIRM graph, which is referred to as the FIRM MDP.
- *SLQG-FIRM*: To instantiate a FIRM, we need concrete belief samplers and connectors. A concrete example of these components based on SLQG controllers is given in Section V. Basically, it is shown that under an SLQG controller the belief can be driven into the ϵ-neighborhood of the sampled Gaussian beliefs in finite time, and thus node reachability is achieved. In this fashion, SLQG-FIRM addresses the hard task of sampling in reachable belief space that is required in belief space planning [Pineau et al., 2006], [Spaan and Vlassis, 2005], [Kurniawati et al., 2010]. The focus of SLQG-FIRM is on kinematic systems. However, we also extend it to dynamical systems by restricting the nodes to the equilibrium space.
- *Graph (multi-query roadmap) in belief space*: FIRM is the first framework that generates a *graph* in the belief space with independent edges. In other words, it is a multi-query roadmap, which distinguishes it from other methods in the belief space.
- *Breaking the curse of history*: A fundamental contribution of FIRM is that the optimal action at a given node does not depend on the traversed nodes, actions, and observations prior to this node, i.e., it is independent of the history of the information process (cf. Fig. 1). This is a direct consequence of inducing reachable belief nodes using feedback controllers, which breaks the curse of history in POMDPs. In addition, the sampling-based nature of the method, borrowed from PRM, allows us to ameliorate the curse of dimensionality.
- *Probabilistic completeness*: Finally, we generalize the conventional concept of "probabilistic completeness" (which is defined for motion planning methods in deterministic environments) to the concept of "probabilistic completeness under uncertainty" (which is defined for the planners in the presence of uncertainty). According to this definition, we prove that FIRM is a probabilistically complete algorithm. Moreover, we perform an analysis on the absorption probability of the local planners in the belief space, which provides useful general tools that can be used in analyzing planning methods under uncertainty.

More importantly, FIRM offers a set of practical contributions, which we believe provides an important step toward utilizing POMDPs as a practical tool for robot motion planning under uncertainty. The main practical highlights can be summarized as follows:

- *Efficient planning*: The construction of FIRM is offline and thus online planning (and replanning) is feasible and almost instantaneous.
- *Robustness*: The optimal feedback policy, instead of a nominal path, is computed offline. It is obtained by solving the dynamic programming problem associated with the FIRM MDP on the belief graph. As a result, no replanning is needed even in the case of large deviations (or just local real-time replanning is sufficient), and the feedback over the belief space can take care of deviations. Therefore, the method is robust to large deviations. It is also less sensitive to linearization

errors, since if the system goes out of a linearization region of a controller, it falls into the valid linearization region of some other controller (assuming a sufficient number of FIRM nodes) that can take the belief and drive it to the goal.

- *Reliability (Incorporating obstacles in planning)*: In the FIRM framework, collision probabilities can be computed, which leads to more accurate plans, as opposed to simplified collision measures that may lead to conservative plans (cf. Fig. 2). The obstacles add a *failure node* to the FIRM graph into which the robot can be absorbed. Further, due to the offline construction of FIRM, the heavy computational burden of estimating collision probabilities can be done offline.
- *Scalability*: Belief space planners usually have an exponential planning complexity either in the number of nodes (if they are sampling-based methods) or in the size of grid (if they rely on discretizing the environment). However, the complexity of the FIRM construction is a constant multiplier of the complexity of the PRM construction. Moreover, the complexity of planning (or replanning) with FIRM is a constant, which is independent of the size of the underlying graph.

IV. Problem Formulation

The main sources of uncertainty in motion planning are the lack of exact knowledge of the robot's motion model, the robot's sensing model, and the environment model, which are referred to as motion uncertainty, sensing uncertainty, and map uncertainty, respectively. In this paper, we focus on motion and sensing uncertainty, but some of the concepts are extensible to problems with map uncertainty. The Markov Decision Process (MDP) problem and the Partially Observable MDP (POMDP) problem are the most general formulations, respectively, for planning problems under motion uncertainty and for planning problems under both motion and sensing uncertainty.

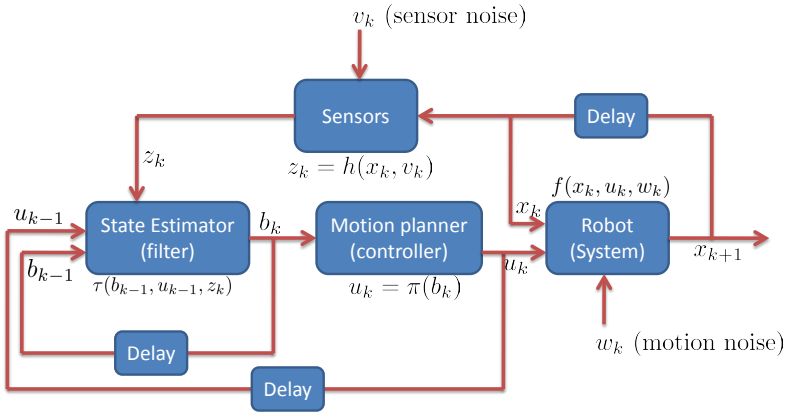


Fig. 3. Block diagram corresponding to the problem of planning under motion and sensing uncertainty

While in the deterministic setting, we seek an optimal path as the solution of the motion planning problem, in the stochastic setting, we seek an optimal feedback (mapping) π as the solution of the motion planning problem. In the case of an MDP, π is a mapping from the state space to the control space, while in the case of POMDP, π is a mapping from the belief space to the control space (see Fig. 3). In the rest of paper, we focus on POMDPs.

A. Preliminaries and Notation

As mentioned, the POMDP formulation is the most general formulation for the planning problem under process (motion) uncertainty and imperfect state information (sensing uncertainty). POMDPs are introduced in [Astrom, 1965], [Smallwood and Sondik, 1973], [Kaelbling et al., 1998]. In the following, we first explain different elements in the POMDP problem, and then present a form of the POMDP formulation, which is known as the belief MDP problem [Bertsekas, 2007], [Thrun et al., 2005].

State, control, and Observation: Let $x_k \in \mathbb{X}$, $u_k \in \mathbb{U}$, and $z_k \in \mathbb{Z}$ denote the system state, control, and observation at time step k, respectively, where $\mathbb{X} \subseteq \mathbb{R}^{d_x}$, $\mathbb{U} \subseteq \mathbb{R}^{d_u}$, and $\mathbb{Z} \subseteq \mathbb{R}^{d_z}$ are the state, control, and observation spaces. Scalars d_x, d_u, and d_z are the state, control, and observation dimensions and $\mathbb{R}^d$ denotes the d-dimensional Euclidean space.

Basically, system state x encodes all information needed for decision-making at a specific time instant. It is worth noting that the state space in our problem is continuous. Control space $\mathbb{U}$, which contains all possible control inputs (or actions), can also be continuous. $u_{0:k} := \{u_0, u_1, \cdots, u_k\}$ denotes the control history up to step k. Similarly, the observation space $\mathbb{Z}$ that contains all possible observations (sensor measurements) can also be continuous. $z_{0:k} := \{z_0, z_1, \cdots, z_k\}$ is the observation history up to step k.

State Evolution Model: The process model (or the motion model) $x_{k+1} = f(x_k, u_k, w_k)$ describes how the system state evolves as a function of the applied control u_k and the process (motion) noise w_k, which is distributed according to the (known) pdf $p(w_k|x_k, u_k)$. An equivalent representation of this evolution model is through the transition pdf $p(x'|x, u) : \mathbb{X} \times \mathbb{U} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$, which encodes the probability density of the transition from state x to state x' under the control u.

Observation (sensor) model: Although x_k is sufficient information to make the decision (generate control u_k), in partially-observable systems, the system state is *unknown* and the only available data for decision-making are the imperfect measurements of the state made by the sensors. The observation model $z_k = h(x_k, v_k)$ encodes the relation between system state x_k and its measurements z_k, where v_k is the observation noise at time step k, which is distributed according to the (known) pdf $p(v_k|x_k)$. An equivalent representation of this observation model is through the likelihood pdf $p(z|x) : \mathbb{X} \times \mathbb{Z} \to \mathbb{R}_{\geq 0}$.

Information-state (belief): In partially-observable environments, the available data for decision-making in time step k is the history of observations we have made, $z_{0:k}$, and the history of actions we have taken, $u_{0:k-1}$. Let us denote this data history by $\mathcal{H}_k = \{z_{0:k}, u_{0:k-1}\}$. This data history can be compressed to a conditional probability distribution over all possible states, i.e., $b_k = p(x_k|z_{0:k}; u_{0:k-1})$. The pdf $b_k : \mathbb{X} \times \mathbb{Z}^k \times \mathbb{U}^{k-1} \to \mathbb{R}_{\geq 0}$ is called the information-state or belief at the k-th step. $\mathbb{B}$ denotes the belief space of the problem, containing all possible beliefs $b \in \mathbb{B}$.

Belief Evolution Model (Filter Model): In recursive state estimation techniques, belief can be computed recursively. The belief evolution model (or belief dynamics) introduced by this recursion is shown by function $\tau : \mathbb{B} \times \mathbb{U} \times \mathbb{Z} \to \mathbb{B}$, which computes the next belief based on the last action and current observation $b_{k+1} = \tau(b_k, u_k, z_{k+1})$. This belief evolution model can be derived using Bayes rule and the law of total probability [Bertsekas, 2007],[Thrun et al., 2005] as follows:

$$b_{k+1} = p(z_{k+1}|\mathcal{H}_k, u_k)^{-1} p(z_{k+1}|x_{k+1}) \int_{\mathbb{X}} p(x_{k+1}|x_k, u_k) b_k dx_k =: \tau(b_k, u_k, z_{k+1}). \tag{1}$$

An equivalent representation of the belief evolution model is through the transition pdf $p(b'|b, u) : \mathbb{B} \times \mathbb{U} \times \mathbb{B} \to \mathbb{R}_{\geq 0}$ that encodes the probability density of the transition from belief b to belief b' under the control u.

Policy: In a partially-observable system, the planner π (also called the policy or feedback controller) has to be a function that returns an action u_k given the available data $\mathcal{H}_k$. However, it can be shown that the compression of data $\mathcal{H}_k$ to belief b_k preserves all the information needed for decision-making [Kumar and Varaiya, 1986]. Therefore, a policy $\pi(\cdot)$ has to be a function that returns an action u_k given the belief b_k, i.e., $\pi(\cdot) : \mathbb{B} \to \mathbb{U}$.

$$u_k = \pi(b_k), \quad \forall b_k \in \mathbb{B}. \tag{2}$$

The space of all possible $\pi(\cdot)$ is denoted by Π.

Cost-to-go: To choose an optimal policy, we need to have a cost function, which is a task-dependent quantity. But, let us in general denote the one step cost of taking action u at belief b by $c(b, u) : \mathbb{B} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$. Therefore, we can define the cost-to-go function $J^\pi(\cdot) : \mathbb{B} \to \mathbb{R}_{\geq 0}$ from a belief b_0 under the policy π as:

$$\begin{aligned} J^\pi(b_0) &:= \sum_{k=0}^{\infty} \mathbb{E}[c(b_k, \pi(b_k))] \\ s.t. \quad & b_{k+1} = \tau(b_k, \pi(b_k), z_{k+1}), \qquad z_k \sim p(z_k|x_k) \end{aligned} \tag{3}$$

where $\mathbb{E}[\cdot]$ is the expectation operator. Consider a goal region $B^{goal} \subset \mathbb{B}$ such that, for all u, we have $c(b \in B^{goal}, u) = 0$; i.e., the goal region is cost absorbing. Then, the above cost-to-go would be finite for a policy that can drive the state to the goal region in finite time.

B. POMDP problem

Given the motion model f, observation model h, and the cost-to-go J^π, the POMDP problem seeks the best policy that minimizes the cost-to-go function from every belief in the belief space. Formally, if we denote the *optimal cost-to-go* function by $J(\cdot)$, we can define *optimal policy* $\pi(\cdot) : \mathbb{B} \to \mathbb{U}$, which is the solution of POMDP as follows:

$$J(\cdot) := \min_{\Pi} J^\pi(\cdot) \tag{4}$$

$$\pi = \arg\min_{\Pi} J^\pi(\cdot) \tag{5}$$

This formulation of the POMDP problem is also known as the *belief-MDP* problem [Bertsekas, 2007], [Thrun et al., 2005], because it is an MDP over the belief space.

Dynamic Programming (DP): It is well known that the optimal cost-to-go is obtained by solving the following stationary Dynamic Programming (DP) equation on the belief space $\mathbb{B}$ [Bertsekas, 2007], [Thrun et al., 2005]. Subsequently, the solution of POMDP (i.e., π) can be computed as a function that returns the argument of this minimization, i.e., returns the optimal action at every belief.

$$J(b) = \min_u \{c(b, u) + \int_{\mathbb{B}} p(b'|b, u) J(b') db'\}, \quad \forall b \in \mathbb{B} \tag{6a}$$

$$\pi(b) = \arg\min_u \{c(b, u) + \int_{\mathbb{B}} p(b'|b, u) J(b') db'\}, \quad \forall b \in \mathbb{B} \tag{6b}$$

However, it is well known that this DP equation is exceedingly difficult to solve since it is defined over the entire belief space and suffers from the curse of history [Pineau et al., 2003] and the curse of dimensionality.

Constrained POMDP problems: The presence of constraints makes this problem even more difficult. We denote the constraint set (or the failure set) in the state and control space by $F \subset \mathbb{X} \times \mathbb{U}$, which needs to be avoided by the system, i.e., $(x_k, u_k) \notin F$, for all k.

C. Problem description

We aim at constructing a sampling-based solution to the belief MDP problem. The main goals of this paper are as follows:

SLQG-based FIRM: First, we construct a roadmap in belief space using a sampled roadmap of local controllers where stationary LQG controllers are utilized as belief stabilizers. We perform this construction for a certain class of systems, and show that the belief reachability condition is guaranteed. In designing SLQG-FIRM, we first focus on kinematic systems (satisfying $x = f(x, 0, 0)$). Then, using the notion of equilibrium space and restricting the sampling to this space, we apply the method to dynamical systems as well.

General FIRM framework: After studying the concrete SLQG-FIRM example, we consider the more general case, where, for a general system, assuming that there exists a controller under which belief reachability is guaranteed, *(i)* we construct a graph in the belief space encoding the failure probabilities on its edges, *(ii)* reduce the intractable belief MDP in Eq. (4) into a tractable MDP problem on this graph, and *(iii)* compute a feedback solution on this graph.

V. SLQG-FIRM

In this section, we discuss a particular instance of the FIRM framework in which belief reachability is accomplished by Stationary LQG controllers. In Section VI, we propose the general FIRM framework.

We start this section by restricting our attention to the class of systems that SLQG-FIRM can handle. Then, we present a brief review of LQG controllers and address how we can define nodes in the belief space to satisfy the reachability using SLQG controllers. Next we explain the procedure of constructing local controllers (i.e., FIRM edges) and the SLQG-based FIRM graph. Finally, we compute transition probabilities and costs associated with each graph edge and compute the graph feedback.

A. Preliminaries on SLQG

In this section, we assume the noise is Gaussian, and we start by defining the notation needed in dealing with Gaussian beliefs.

Gaussian belief space: We denote the random estimation vector by x^+, whose distribution is $b_k = p(x_k^+) = p(x_k | z_{0:k}, u_{0:k-1})$, and denote the mean and covariance of x^+ by $\widehat{x}^+ = \mathbb{E}[x^+]$ and $P = \mathbb{E}[(x^+ - \widehat{x}^+)(x^+ - \widehat{x}^+)^T]$, respectively. Denoting the Gaussian belief space by $\mathbb{GB}$, every function $b(\cdot) \in \mathbb{GB}$, can be characterized by a mean-covariance pair $(\widehat{x}^+, P)$. Abusing the notation, we also show this pair by $b \equiv (\widehat{x}^+, P) \in \mathbb{R}^n \times \mathbb{S}_+^n$, where the mean vector belongs to the n-dimensional Euclidean space $\mathbb{R}^n$ and the covariance matrix belongs to the space of all positive semi-definite $n \times n$ matrices $\mathbb{S}_+^n$.

LQG controllers: An LQG controller is composed of a Kalman filter as the state estimator and an LQR controller (see Fig. 3). Thus, the belief dynamic $b_{k+1} = \tau(b_k, u_k, z_{k+1})$ is known and comes from the Kalman filtering equations, and the controller $u_k = \mu(b_k)$ that acts on the belief comes from the LQR equations. Considering a quadratic cost for state error and control error, LQG is an optimal controller for linear systems with Gaussian noise [Bertsekas, 2007]. However, it is also often used for stabilization of nonlinear systems around a given trajectory or around a given point.

Stationary and time-varying LQG: Time-varying LQG is designed to track a given trajectory, in which at every time step, a different feedback policy is utilized. Stationary LQG is a time-invariant policy, in which LQG is designed around a given point, say $\mathbf{v}$, to steer the state of the system to $\mathbf{v}$ [Bertsekas, 2007]. In Appendices B and C we review these controllers in detail.

Equilibrium space: Let us denote a configuration of a robotic system [Lozano-Perez, 1983] by q. Kinematic models are specified in terms of the configuration variable q, while dynamical models are specified by the state $x = (q, \dot{q})$, where $\dot{q}$ denotes the corresponding velocities. In SLQG-FIRM, we sample the underlying PRM nodes (stabilizer parameters) from the configuration space. Thus, for dynamical systems, we impose the condition $\dot{q} = 0$ on the samples, i.e., we sample from the equilibrium space of the system, which is denoted by X in this paper.

Remark 1. *FIRM can be generalized to cases that do not need to sample in equilibrium space. For example, in systems such as fixed-wing aircraft, the system cannot reach the zero velocity $\dot{q} = 0$. In such cases, SLQG is not a suitable choice and one needs to design more appropriate controllers, such as periodic controllers as detailed in [Agha-mohammadi et al., 2012c], [Agha-mohammadi et al., 2013a]. In such a case, we sample periodic maneuvers as FIRM nodes. In other words, we go from periodic trajectory to periodic trajectory instead of going from point to point [Agha-mohammadi et al., 2012c], [Agha-mohammadi et al., 2013*

B. Belief Stabilizers

In SLQG-FIRM nodes, we use Stationary LQG (SLQG) controllers as belief stabilizers, i.e., as a tool to reach (stabilize to) a predefined belief (FIRM node). To explain how SLQG works as a belief stabilizer, consider a fixed point $\mathbf{v} \in \mathsf{X}$ in the state space and consider the following linear (linearized) system about $\mathbf{v}$:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{G}w_k, \quad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{7a}$$

$$z_k = \mathbf{H}x_k + v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{7b}$$

SLQG controller: The goal of the SLQG controller designed about $\mathbf{v}$ is to keep the state as close as possible to the desired point $\mathbf{v}$ and also keep the consumed energy at a reasonable level. More rigorously, SLQG minimizes the following quadratic cost:

$$J = \mathbb{E}\{\sum_{k\geq 0}(x_k - \mathbf{v})^T \mathbf{W}_x (x_k - \mathbf{v}) + u_k^T \mathbf{W}_u u_k\}, \tag{8}$$

where $\mathbf{W}_x$ and $\mathbf{W}_u$ are positive definite weight matrices that are defined by the user. In Appendix C, the SLQG controller minimizing the above cost is discussed in detail. However, in brief, the belief propagation and control generation is carried out as follows:

$$b_{k+1} \equiv \begin{bmatrix} \widehat{x}_{k+1}^+ \\ P_{k+1}^+ \end{bmatrix} = \begin{bmatrix} \mathbf{A}\widehat{x}_k^+ + \mathbf{B}u_k + \mathbf{K}_{k+1}(z_{k+1} - \mathbf{H}(\mathbf{A}\widehat{x}_k^+ + \mathbf{B}u_k)) \\ (I - \mathbf{K}_{k+1}\mathbf{H})(\mathbf{A}P_k^+\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T) \end{bmatrix} \equiv \tau(b_k, u_k, z_{k+1}), \tag{9}$$

where $\mathbf{K}_k$ is called the Kalman gain at the k-th time step and is computed as follows:

$$\mathbf{K}_{k+1} = (\mathbf{A}P_k^+\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T)\mathbf{H}^T(\mathbf{H}(\mathbf{A}P_k^+\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T)\mathbf{H}^T + \mathbf{M}\mathbf{R}\mathbf{M}^T)^{-1}. \tag{10}$$

Control signal is generated using a stationary feedback gain $\mathbf{L}_s$:

$$u_k = -\mathbf{L}_s(\widehat{x}_k^+ - \mathbf{v}) =: \mu(b_k), \qquad \mathbf{L}_s = (\mathbf{B}_s^T S_s \mathbf{B}_s + \mathbf{W}_u)^{-1}\mathbf{B}_s^T S_s \mathbf{A}_s, \tag{11}$$

where, S_s is the solution of the following Discrete Algebraic Riccati Equation (DARE):

$$S_s = \mathbf{W}_x + \mathbf{A}_s^T S_s \mathbf{A}_s - \mathbf{A}_s^T S_s \mathbf{B}_s (\mathbf{B}_s^T S_s \mathbf{B}_s + \mathbf{W}_u)^{-1}\mathbf{B}_s^T S_s \mathbf{A}_s. \tag{12}$$

Controllable and Observable pairs: Consider an $n \times n$ matrix $\mathbf{A}$. A pair of matrices $(\mathbf{A}, \mathbf{B})$ is called a controllable pair if the controllability matrix $\mathfrak{C} = [\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \cdots, \mathbf{A}^{n-1}\mathbf{B}]$ has rank n [Bertsekas, 2007]. A pair of matrices $(\mathbf{A}, \mathbf{H})$ is called observable if the pair $(\mathbf{A}^T, \mathbf{H}^T)$ is controllable [Bertsekas, 2007].

Controllable and Observable systems: Let us also define the matrices $\check{\mathbf{Q}}$ and $\check{\mathbf{W}}_x$ such that $\mathbf{G}\mathbf{Q}\mathbf{G}^T = \check{\mathbf{Q}}\check{\mathbf{Q}}^T$, $\mathbf{W}_x = \check{\mathbf{W}}_x^T\check{\mathbf{W}}_x$. We next consider a class of linear systems and quadratic cost weights that satisfy the following property:

Property 1. *Pairs $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}, \check{\mathbf{Q}})$ are controllable pairs, and pairs $(\mathbf{A}, \mathbf{H})$ and $(\mathbf{A}, \check{\mathbf{W}})$ are observable pairs.*

In the following, we present three lemmas, through which we can construct reachable SLQG-FIRM nodes for the systems that satisfy Property 1. However, approaches such as periodic LQG-based FIRM [Agha-mohammadi et al., 2012c] or dynamic feedback linearization-based FIRM [Agha-mohammadi et al., 2012a] extend this class of systems by excluding the controllability part in Property 1, and thus consider a broader class of systems.

Lemma 1. *Consider the SLQG controller designed to drive the state of the system in Eq. (7) to a point $\mathbf{v} \in \mathsf{X}$. Given that Property 1 is satisfied, in the absence of a stopping region, the belief b_k under SLQG controller converges to a unique stationary belief b_s, in distribution (i.d.). In other words, the distribution over belief converges to a unique distribution. That is,*

$$b_k \xrightarrow{i.d.} b_s \sim \mathcal{N}(b_c, \mathtt{C}). \tag{13}$$

Note that b_k is a random belief that converges to another random belief b_s. In the Gaussian setting, the distribution over the random belief b_s is $\mathcal{N}(b_c, \mathtt{C})$, where, $b_c = \mathbb{E}[b_s] = (\mathbf{v}, P_s)$. The stationary estimation covariance matrix P_s is characterized in Lemma 3, and the covariance $\mathtt{C}$ is characterized in the Appendix C.

Proof: In Appendix C, we review the stationary LQG and prove Lemma 1. ■

Lemma 2. *Given Property 1, the following Algebraic Riccati equation (DARE) has a unique symmetric positive definite solution [Bertsekas, 2007], denoted by P_s^-:*

$$P_s^- = \mathbf{G}\mathbf{Q}\mathbf{G}^T + \mathbf{A}(P_s^- - P_s^-\mathbf{H}^T(\mathbf{H}P_s^-\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}P_s^-)\mathbf{A}^T. \tag{14}$$

Moreover, the stationary covariance matrix P_s introduced in Lemma 1 is computed as:

$$P_s = P_s^- - P_s^-\mathbf{H}^T(\mathbf{H}P_s^-\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}P_s^-. \tag{15}$$

Proof: See Appendix C or [Bertsekas, 2007]. ■

Now we state the main result, through which we can construct *reachable* FIRM nodes under SLQG-based belief stabilizers:

Lemma 3. *Consider the SLQG controller designed to drive the state of the system in Eq.* (7) *to a point* $\mathbf{v} \in \mathsf{X}$. *Suppose matrix* $\mathbf{H}$ *is full rank and Property 1 is satisfied. Then, any set* $B \subset \mathbb{B}$, *whose interior contains* $b_c = (\mathbf{v}, P_s)$, *is reachable under the designed SLQG controller starting from any Gaussian distribution. Moreover, the estimation covariance* P_k *converges to the unique deterministic stationary covariance* P_s.

Proof: See Appendix D. ■

Therefore, based on Lemma 3, SLQG can accomplish the belief reachability for appropriately chosen region B. In the next subsection we explicitly characterize regions B.

C. Designing SLQG-FIRM Nodes

Underlying PRM: As mentioned, to construct a FIRM we first construct an underlying PRM [Kavraki et al., 1996]. In the SLQG-FIRM, nodes of the underlying PRM, denoted by $\{\mathbf{v}^j\}_{j=1}^{N_v}$, are sampled from the obstacle-free space. Considering linear systems or nonlinear systems that are locally well approximated by linearization, we linearize the system about every PRM node. Let us denote the linear (linearized) system about $\mathbf{v}^j$ as follows:

$$x_{k+1} = \mathbf{A}^j x_k + \mathbf{B}^j u_k + \mathbf{G}^j w_k, \quad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^j) \tag{16a}$$

$$z_k = \mathbf{H}^j x_k + v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^j). \tag{16b}$$

where w_k and v_k are motion and measurement noise, respectively, drawn from zero-mean Gaussian distributions with covariances $\mathbf{Q}^j$ and $\mathbf{R}^j$.

FIRM nodes: To design the j-th FIRM node B^j, we first design the SLQG controller μ_s^j (see Eq. (11)) corresponding to the system in Eq. (16). The controller μ_s^j is called the j-th node controller or the j-th belief stabilizer. Given Property 1, based on Lemma 1, the limiting random belief $b_s^j = (\widehat{x}_s^{+^j}, P_s^j)$ exists. $\widehat{x}_s^{+^j}$ and P_s^j are the stationary estimation mean and covariance, respectively. Note that under SLQG, $\widehat{x}_s^{+^j}$ is a random variable and P_s^j is a deterministic matrix. Moreover, in Lemma 1, it is shown that $b_c^j = \mathbb{E}[b_s^j] = (\mathbf{v}^j, P_s^j)$, where P_s^j is shown to be unique and computed in Lemma 2. Thus, we can characterize the j-th node center:

$$b_c^j = (\mathbf{v}^j, P_s^j). \tag{17}$$

As a result, considering B^j as a ball with an arbitrary radius $\epsilon > 0$ centered at b_c^j, the pair (B^j, μ_s^j) is a *proper pair*, based on Lemma 3; i.e., B^j is reachable under μ_s^j. Thus, one can define the j-th FIRM node as $B^j = \{b \ : \ \|b - b_c^j\|_b < \delta\}$, where $\|\cdot\|_b$ denotes a suitable norm in belief space and δ defines the FIRM node size. A typical example of such a FIRM node in Gaussian belief space can be defined by considering mean and covariance separately:

$$B^j = \{b = (x, P) \ : \ \|x - \mathbf{v}^j\| < \delta_1, \|P - P_s^j\|_m < \delta_2\} \tag{18}$$

where δ_1 and δ_2 are suitably small thresholds that determine the size of FIRM node B^j. $\|\cdot\|$ is a suitable vector norm and $\|\cdot\|_m$ is a suitable matrix norm. We denote the set of all SLQG-FIRM nodes as $\mathbb{V} = \{B^i\}$.

D. Designing SLQG-FIRM Edges

A FIRM edge is actually a local planner (local feedback controller). In SLQG-based FIRM, the local controller representing the (i, j)-th edge is denoted by μ^{ij}. The role of μ^{ij} is to drive the belief from the node B^i to the node B^j. Based on Lemma 3, for a linear system, if we choose $\mu^{ij} = \mu_s^j$, as has been done in [Agha-mohammadi et al., 2011], the node B^j is reachable under μ^{ij}. However, to better cope with nonlinearities, we construct the local controller μ^{ij} by preceding the node-controller with a time-varying LQG controller $\overline{\mu}_k^{ij}$, which is called an *edge-controller* here. Time-varying LQG controllers are described in detail in Appendix B.

PRM edge: To design edge-controllers, first the underlying PRM edges, denoted by $\mathcal{E} = \{\mathbf{e}^{ij}\}$, have to be constructed. For kinematics-based models there are many different methods in the PRM literature to construct such edges. For dynamical models, there are fewer choices. A few examples are [van den Berg and Overmars, 2007] or [Agha-mohammadi et al., 2012c].

Edge-controllers: An edge-controller $\overline{\mu}_k^{ij}$ in SLQG-FIRM is built by linearizing the system along the (i, j)-th PRM edge $\mathbf{e}^{ij}$ and designing a time-varying LQG controller to track it (see Appendix B). The edge-controller has two major roles. First it tries to track the PRM edge and thus exploits the available information on the PRM edges, such as some clearance from the obstacles. Second, in the case that the neighboring PRM nodes are not close to each other, it takes the belief into the valid linearization region of the j-th belief stabilizer, where it hands over the system to the belief stabilizer, and the belief stabilizer in turn takes the system to the j-th FIRM node.

Local controllers: Thus, overall, the (i, j)-th local controller μ^{ij} is the concatenation of the (i, j)-th edge controller $\overline{\mu}_k^{ij}$ and j-th node-controller (belief stabilizer) μ_s^j. We denote the set of all SLQG-FIRM edges by $\mathbb{M} = \{\mu^{ij}\}$ and the set of all SLQG-FIRM edges originating from B^i by $\mathbb{M}(i)$.

SLQG-FIRM: Formally, we define SLQG-FIRM as a graph with the set of nodes $\mathbb{V} = \{B^i\}$ and the set of edges (or local controllers) $\mathbb{M} = \{\mu^{ij}\}$. The set of controllers originating from B^i is denoted by $\mathbb{M}(i) \subset \mathbb{M}$.

E. Transition probabilities and edge costs

To find a feedback on a FIRM graph, we need to compute the cost associated with the graph edges. Moreover, we include the constraint set F into the planning with FIRM by computing the probability of violating the constraint $(x, u) \notin F$ along the graph edges. Let us denote the cost of taking controller μ^{ij} at node B^i by $C^g(B^i, \mu^{ij})$. Superscript g refers to the "global" (or "graph-level") quantities, as these quantities are used to find the global policy (or policy on the graph). Similarly, let $\mathbb{P}^g(B^j|B^i, \mu^{ij})$ and $\mathbb{P}^g(F|B^i, \mu^{ij})$ denote the probability of the transition to B^j and F under μ^{ij}, respectively. These quantities are rigorously defined in Section VI and their connection with the original POMDP is established. However, in this subsection, we give just an example of how such costs and transition probabilities can be computed.

Transition probabilities: Computing transition probabilities $\mathbb{P}^g(\cdot|B^i, \mu^{ij})$ in general can be computationally expensive. Here, we utilize particle-based methods to approximate the distributions and thus compute the collision probabilities. Basically, we can approximate the failure and reachability probabilities based on the number of particles that violate the constraints (hit the set F) and based on the number of particles that can reach the target node (hit the set B^j). The method is described in more detail with the experiments in Section VIII-A4. The dependency of collision events in different time steps, which is ignored in most collision probability computation methods in the POMDP literature, can be taken into account rigorously in particle-based methods. Owing to the offline construction of FIRM, the high computational burden of particle-based approaches can be tolerated. However, any other method for computing transition probabilities can also be adopted, such as [Patil et al., 2012].

Edge costs: The FIRM edge costs in general and their derivation based on the one-step costs of the original POMPD problem is defined in Section VI. However, roughly speaking, we can define the cost $C^g(B^i, \mu^{ij})$ as the sum of all one-step costs along the edge until the system reaches the target node B^j or hits the failure set F. Depending on the application, one can define a variety of cost functions. Here, we form a cost function based on a linear combination of the estimation accuracy and edge traverse time. This cost function aims to find paths for which the estimator (and hence the controller) can perform well and also to find faster paths. An indicator of estimation error is the trace of estimation covariance. Thus, we define $\Phi^{ij} = \mathbb{E}[\sum_{k=1}^{\mathcal{T}} \mathrm{tr}(P_k^{ij})]$ along the edge. In stationary LQG, the covariance matrix evolves deterministically and thus the expectation operator can be omitted. However, if the filter of choice in the edge-controller is the Extended Kalman Filter (EKF), the covariance matrix evolution is also stochastic, and this measure can take into account its stochasticity. Let us denote the mean stopping time under controller μ^{ij} as $\widehat{\mathcal{T}}^{ij}$. Then, the total edge cost is considered as a linear combination of estimation accuracy and expected stopping time, with suitable coefficients α_1 and α_2.

$$C^g(B^i, \mu^{ij}) = \alpha_1 \Phi^{ij} + \alpha_2 \widehat{\mathcal{T}}^{ij}. \tag{19}$$

F. Graph Feedback on SLQG-FIRM

Graph Policy: Graph policy $\pi^g : \mathbb{V} \to \mathbb{M}$ is a function that returns an edge (local controller) for any given node of the graph. We denote the space of all graph policies by Π^g. To choose the best graph policy in Π^g we define the optimal graph cost-to-go J^g from every graph node.

Graph Cost-to-go: The cost-to-go from a given node B^i is equal to the cost of the next taken controller, i.e., $C^g(B^i, \pi^g(B^i))$, plus the expected cost-to-go from the next node or from the failure set. In other words, the dynamic programming equations for this graph are:

$$\begin{aligned} J^g(B^i) = \min_{\mathbb{M}(i)} & C^g(B^i, \mu^{ij}) + J^g(F)\mathbb{P}^g(F|B^i, \mu^{ij}) \\ & + J^g(B^j)\mathbb{P}^g(B^j|B^i, \mu^{ij}), \end{aligned} \tag{20a}$$

$$\begin{aligned} \pi^g(B^i) = \arg\min_{\mathbb{M}(i)} & C^g(B^i, \mu^{ij}) + J^g(F)\mathbb{P}^g(F|B^i, \mu^{ij}) \\ & + J^g(B^j)\mathbb{P}^g(B^j|B^i, \mu^{ij}). \end{aligned} \tag{20b}$$

in which, $J(F)$ is a suitably high user-defined cost-to-go for hitting the obstacles. The cost-to-go from goal node B^{goal} is defined to be zero, i.e., $J^g(B^{goal}) = 0$.

Solving SLQG-FIRM DP: The DP in Eq. (20) is a tractable DP as it is defined on a finite number of graph nodes. Computing the transition costs and probabilities offline, this DP can be solved online using standard techniques, such as value/policy iteration methods, for any submitted query. As a result, FIRM is indeed a multi-query roadmap in belief space. Moreover, if the goal node is fixed and only the starting point of the query changes, then this DP can be solved offline and π^g can be stored as a look-up table.

Offline Construction of SLQG-FIRM: Algorithm 1 details the construction of SLQG-FIRM with a given goal node.

Algorithm 1: Offline Construction of SLQG-FIRM

input : Free space map, X_{free}
output : FIRM graph $\mathcal{G}$
Sample PRM nodes $\mathcal{V} = \{\mathbf{v}^j\}_{j=1}^{N_v}$ and construct its edges $\mathcal{E} = \{\mathbf{e}^{ij}\}$;
forall the *PRM nodes* $\mathbf{v}^j \in \mathcal{V}$ **do**
    Design the node controller (stationary LQG) μ_s^j about the node $\mathbf{v}^i$ using Eq. (11);
    Compute associated b_c^j using Eq. (17);
    Construct FIRM node B^j using Eq. (18);
Construct $\mathbb{V} = \{B^i\}$;
forall the *PRM edges* $\mathbf{e}^{ij} \in \mathcal{E}$ **do**
    Design the edge controller (time-varying LQG) $\overline{\mu}_k^{ij}$ along the edge $\mathbf{e}^{ij}$ (detailed in Appendix B);
    Construct the local controller μ^{ij} by concatenating edge controller $\overline{\mu}_k^{ij}$ and node controller μ_s^j;
    Set $b_0 = b_c^i$;
    Generate sample belief paths $b_{0:\mathcal{T}}$ and ground truth paths $x_{0:\mathcal{T}}$ induced by controller μ^{ij} invoked at B^i;
    Compute the transition probabilities $\mathbb{P}^g(F|B^i, \mu^{ij})$ and $\mathbb{P}^g(B^j|B^i, \mu^{ij})$ and transition cost $C^g(B^i, \mu^{ij})$;
Construct $\mathbb{M} = \{\mu^{ij}\}$;
Compute the cost-to-go J^g and feedback π^g over the FIRM nodes by solving the DP in Eq. (20);
$\mathcal{G} = (\mathbb{V}, \mathbb{M}, J^g, \pi^g)$;
return $\mathcal{G}$;

G. Planning with SLQG-FIRM (Query-phase)

Given that the FIRM graph is computed offline, the online phase of planning (and replanning) on the roadmap becomes very efficient, and thus feasible in real time. In this section, we assume that the goal node is fixed and we just input the start point as the query. However, as discussed in the previous subsection, one can easily submit queries with different goal locations by solving DP online. If the initial belief b_0 of the submitted query does not belong to any B^i, we create a singleton set $B_0 = \{b_0\}$ as the initial FIRM node. To connect B_0 to the FIRM graph, we go back into the state space, where the underlying PRM is constructed. There, we add a new PRM node to the graph $\mathbf{v}_0$, which is the expected value of the robot state, i.e., $\mathbf{v}_0 = \mathbb{E}[x_0]$. Then, we connect $\mathbf{v}_0$ to the underlying PRM graph based on the connecting function of the adopted PRM. We denote the set of newly added edges originating from $\mathbf{v}_0$ by $\mathcal{E}(0)$. Then, corresponding to each edge in $\mathcal{E}(0)$, we design a local controller and call the set of them $\mathbb{M}(0)$. Finally, we choose the best initial controller among the local controllers in $\mathbb{M}(0)$ using:

$$\mu_0^*(\cdot) = \arg\min_{\mu \in \mathbb{M}(0)} \{C^g(B_0, \mu) + \mathbb{P}^g(B(\mu)|B_0, \mu) J^g(B(\mu)) + \mathbb{P}^g(F|B_0, \mu) J^g(F)\}, \tag{21}$$

where $B(\mu)$ is the target node of the controller μ. Under the controller μ_0^*, belief evolves and enters one of FIRM nodes, if no collision occurs. From this FIRM node, a combination of the global graph policy π^g and the local edge policies $\{\mu^{ij}\}$ can take the belief to the goal node, as explained below.

Merging Global and Local Feedbacks: After computing a global graph feedback π^g and local edge feedbacks $\{\mu^{ij}\}$, we can construct a full feedback π. Actually, at every time instance, π is equal to one of the local feedbacks, which is chosen by the global feedback in the last visited node. In other words, given the current FIRM node, we use policy π^g defined on FIRM nodes to find μ^* and pick μ^* to move the robot into $B(\mu^*)$. This process is continued until the system reaches the goal region or hits the failure set. Algorithm 2 illustrates this procedure.

Kidnapped robot problem: "In robotics, the kidnapped robot problem commonly refers to a situation where an autonomous robot in operation is carried to an arbitrary location," [Choset et al., 2005]. Consider a kidnapped robot problem in a known environment. Just after the robot is kidnapped, it would be risky to apply any control, because the robot may be close to an obstacle. Thus, in such a scenario, we first initialize the system belief with a Gaussian with large covariance and go into an "information gathering" mode, where we do not apply any control signal and only gather measurements, until the covariance shrinks to a reasonable covariance or it remains unchanged for a significant amount of time (i.e., when there is no additional information to reduce the uncertainty). Afterwards, we connect the resulting belief to the FIRM nodes and continue applying the FIRM policy to move the robot towards goal region.

VI. General FIRM Framework

The goal of this section is to construct a general FIRM framework, assuming that there exists a mechanism to guarantee belief reachability. As a result, if for a certain class of systems, one comes up with a controller that can accomplish belief reachability, a graph in belief space directly follows according to this general framework.

Algorithm 2: Online Phase Algorithm (Planning or Replanning with SLQG-FIRM)

input : Initial belief b_0, FIRM graph $\mathcal{G}$
if $\exists B^m \in \mathbb{V}$ *such that* $b_0 \in B^m$ **then**
  Set $i = m$ and compute $\mu^* = \pi^g(B^m)$;
else
  Compute $\mathbf{v}_0 = \mathbb{E}[x_0]$ based on b_0, and connect $\mathbf{v}_0$ to the PRM. Let $\mathcal{E}(0)$ denote the set of outgoing edges from $\mathbf{v}_0$;
  Set $B_0 = \{b_0\}$; Design local controllers associated with edges in $\mathcal{E}(0)$. Call the set of these local controllers $\mathbb{M}(0)$;
  forall the $\mu \in \mathbb{M}(0)$ **do**
    Generate sample belief paths $b_{0:\mathcal{T}}$ and ground truth paths $x_{0:\mathcal{T}}$ induced by controller μ invoked at b_0;
    Compute the transition probabilities $\mathbb{P}^g(F|B_0, \mu)$ and $\mathbb{P}^g(B(\mu)|B_0, \mu)$ and transition costs $C^g(B_0, \mu)$;
  Set $i = 0$ and choose the best initial local controller μ^{ij} within the set $\mathbb{M}(0)$ using Eq. (21);
while $B^i \neq B^{goal}$ **do**
  while $b_k \notin B^j$ *and "no collision"* **do**
    Apply the control $u_k = \mu^{ij}(b_k)$ to the system;
    Get the measurement z_{k+1} from sensors;
    if *Collision happens* **then return** Collision;
    Update belief as $b_{k+1} = \tau(b_k, \mu^{ij}(b_k), z_{k+1})$;
  Set $B^i = B^j$, then compute $\mu^{ij} = \pi^g(B^i)$;

To construct the general FIRM, we start by defining elements and assumptions needed in the FIRM construction. Accordingly, we transform the original intractable POMDP problem into an SMDP problem in the belief space, inspired by sampling-based methods. Then, we construct an arbitrarily good approximation to the solution of this belief SMDP over finite subsets of belief space (FIRM nodes). Doing so, we end up with a tractable MDP, the so called FIRM MDP. We discuss this derivation first for the obstacle-free case and then we add the obstacles to the planning framework. We characterize the quality of the solution obtained by FIRM via its success probability and provide a generic algorithm for planning with FIRM.

A. Feedback Controllers and Reachability

Belief transition probability: As discussed in Section IV, in partially observable environments, the available data for decision-making at time step k can be compressed as the information-state or belief b_k. As discussed, using dynamic estimation schemes, belief can be propagated as $b_{k+1} = \tau(b_k, u_k, z_{k+1})$ (See Eq. (1)), which can be presented as a one-step transition pdf $p(b_{k+1}|b_k, u_k)$ or a one-step transition probability $\mathbb{P}(B|b_k; u_k) = \int_B p(b_{k+1}|b_k, u_k)$, where $B \subset \mathbb{B}$.

Feedback controllers and induced transition probability: In partially observable environments, at each stage, the decision-making process is performed based on the belief at that stage. Therefore, a controller is a mapping from the belief space to the control space, i.e., $\mu(\cdot) : \mathbb{B} \to \mathbb{U}$. Accordingly, a controller μ induces a Markov chain with the one-step transition probability $\mathbb{P}(B|b; \mu) := \mathbb{P}(B|b; \mu(b))$ over the belief space.

Hitting time: Let $\mathcal{T}(D|b, \mu) \in [0, \infty]$ denote the hitting time on the set $D \subset \mathbb{B}$, under the controller μ starting from belief b. Formally it is defined as:

$$\mathcal{T}(D|b, \mu) := \min\{k \geq 0, b_k \in D|b_0 = b, \mu\} \tag{22}$$

Stopping region: We call region $B \subset \mathbb{B}$ a stopping region of the controller μ if we force the controller to stop executing as the state reaches the region B, i.e., for all $b \in B$, we impose $\mathbb{P}_1(B|b, \mu) = 1$.

n-step transition probability: We define the n-step transition probability as the probability of landing in the stopping region B in *at most n steps*:

$$\mathbb{P}_n(B|b, \mu) := \Pr(\mathcal{T}(B|b, \mu) \leq n) \tag{23}$$

Stationary Transition Probability: Consider the controller μ that starts executing from belief b and stops executing when the state enters region B. Thus, we can define $\mathbb{P}(B|b, \mu)$ as the transition probability from b to B induced by μ, when the controller stops executing, i.e., $\mathbb{P}(B|b, \mu)$ would be the probability of landing in the stopping region B in a finite time:

$$\mathbb{P}(B|b, \mu) := \Pr(\mathcal{T}(B|b, \mu) < \infty) \tag{24}$$

Reachability and Accessibility: The stopping region B is called reachable under a controller μ from b if $\mathbb{P}(B|b, \mu) = 1$. The stopping region B is called accessible under a controller μ from b, if $\mathbb{P}(B|b, \mu) > 0$.

αT-reachability: The stopping region B is called αT-reachable under a controller μ from b if $\mathbb{P}_T(B|b, \mu) = \Pr(\mathcal{T}(B|b, \mu) \leq T) > \alpha$, i.e., the controller can drive the system into B in fewer than T steps with a probability greater than α.

Reachability Basin: The reachability basin $\breve{B}$ associated with the pair (μ, B) is the set of all states from which B is reachable under μ in the absence of constraints. The reachability (and αT-reachability) basins are thus defined as follows, respectively:

$$\breve{B} = \{b \in \mathbb{B} \ : \ \mathbb{P}(B|b, \mu) = 1\}, \tag{25}$$

$$\breve{B}(\alpha, T) = \{b \in \mathbb{B} \ : \ \mathbb{P}_T(B|b, \mu) \geq \alpha\}, \tag{26}$$

Clearly, $B \subset \breve{B}$, and in practical cases, B is much smaller than $\breve{B}$.

B. FIRM Graph

In this section, we assume that there are no constraints (i.e., $F = \emptyset$), and we reduce planning over the entire belief space to planning over a representative graph constructed within the belief space. Doing so, we can reduce the MDP problem in (4) over the continuous space into a tractable MDP problem defined over the graph nodes.

Stabilizer sampling: The first step in the construction of the proposed framework is to sample a set of stabilizers $\{\mu^j\}$, where each stabilizer $\mu(\cdot)$ is a mapping from the belief space to the control space. Typically, every stabilizer is characterized by a d_v-vector of parameters $\mathbf{v}^i \in \mathbb{R}^{d_v}$, i.e., we can denote the j-th stabilizer more rigorously as $\mu^i(\cdot; \mathbf{v}^i) : \mathbb{B} \to \mathbb{U}$. As a result, we can sample the parameters $\mathcal{V} = \{\mathbf{v}^j\}$ and then construct a stabilizer corresponding to each parameter. One can view the set $\mathcal{V}$ as a set of underlying PRM nodes in the parameter space.

Sampling FIRM nodes: Basically, FIRM nodes $\{B_j\}$ are disjoint sets in the belief space, where the j-th node has to be chosen such that it is reachable under the j-stabilizer, i.e., $\mathbb{P}(B^j|b, \mu^j) = 1$, with a sufficiently large $\breve{B}$. We discuss the size of $\breve{B}$ further below. Note that, for practical purposes, the reachability condition can be replaced by αT-reachability if needed.

Connecting samples: Consider a set of N samples $\{(\mu^i, B^i)\}_{i=1}^N$, where the reachability basin of the i-th sample is denoted by $\breve{B}^i$. Now, consider $\{B^i\}_{i=1}^N$ as the nodes of a graph. The node B^i is connected to the node B^j if, starting from any $b \in B^i$, we can reach B^j using μ^j. In other words B^i is connected to the node B^j if $B^i \subset \breve{B}^j$. Again, the reachability condition can be replaced by the αT-reachability condition.

Checking connection condition: For simple systems (linear with Gaussian noise) and some controllers (such as SLQG), the connection condition can be checked analytically. However, in general, checking this connection condition analytically may be very difficult. In such cases, the Markov chain induced by the controller can be simulated numerically (e.g., using particle-based methods). Accordingly, we can approximate the reachability (or αT-reachability) probability and check if the condition is true or not. Since this process is done offline, the computational burden can be tolerated. However, as we will see further below, in many cases, designing suitable *edge controllers* in practice increases the reachability probability such that practically one can assume the reachability is satisfied and so there is no need to propagate the probability distribution.

Stopping region: By definition, the graph node B associated with the controller μ acts as the stopping region of the controller. However, if the process under the stabilizer hits another graph node before its corresponding graph node, we can stop the controller and pick the best controller from this intermediate node. Therefore, we can extend the stopping region for all controllers to the union of all nodes $\Psi := \cup_{l=1}^N B^l$. As a result, we will not necessarily have $\mathbb{P}(B^i|b, \mu^i) = 1$ since the process may hit some other node before B^i. However, we will have $\mathbb{P}(\Psi|b, \mu^i) = 1$ for all i in the absence of constraints.

Local controllers (Simplified connecting strategy): To ease the connection step, and to have more distant nodes, we can precede each stabilizer by a time-varying controller (referred to as the edge-controller). To illustrate this idea, consider two nodes B^i and B^j, where $B^i \nsubseteq \breve{B}^j$, i.e., B^i cannot be connected to B^j through μ^j. In this case, we can connect the underlying state nodes $\mathbf{v}^i$ and $\mathbf{v}^j$ in the state space by a finite trajectory e^{ij} (say with length ι) and then design a time varying controller $\overline{\mu}_k^{ij}$, for $k = 0, 1, \cdots, \iota$ to track this finite trajectory. Therefore, if the node B^i is in the basin of reachability of the pair $(\overline{\mu}_k^{ij}, \breve{B}^j)$, then obviously B^i would be in the basin of reachability of the controller $\mu^{ij} = \{\overline{\mu}_{0:\iota}^{ij}, \mu^j\}$. We call μ^{ij} the (i, j)-th local controller, as it connects the node B^i to the node B^j.

Graph: Formally, we define the constructed graph with the set of nodes $\mathbb{V} = \{B^i\}_{i=1}^N$ and the set of edges (or local controllers) $\mathbb{M} = \{\mu^{ij}\}$. The set of controllers available at B^i is denoted by $\mathbb{M}(i)$, i.e., the set of edges starting from B^i. Similar to PRM, in which the path (final solution) is constructed as a concatenation of edges on the roadmap, in FIRM, the policy is constructed by the concatenation of the local policies. However, it is worth noting that by this construction we still perform planning in a continuous space and do not discretize the control space.

Local controllers versus Macro-actions: By the term "macro-action", we mean a sequence of controls (actions) [He et al., 2010], [He et al., 2011]. In other words, a macro-action is a sequence of open-loop policies. It is important to note that a local controller is *not* a macro-action, but rather a sequence of policies (macro-policy), each of which is a mapping from belief space to the continuous control space. Using macro-actions results in an open-loop policy, which cannot compensate for the belief state deviation from the planned path. However, under local-controllers (macro-policies), the effect of noise can be compensated for, due to the feedback nature of the controllers, and thus, the belief can be steered towards a stopping region.

C. Belief SMDP

In this section, we reduce planning over the entire belief space into planning over a subset of belief space, which is actually the union of the belief FIRM graph nodes, i.e., $\Psi = \cup_j B^j$.

SMDP transition costs: First, we generalize the concept of one-step cost $c(b,u):\mathbb{B}\times\mathbb{U}\to\mathbb{R}_{\geq 0}$ to the one-step SMDP cost $C^s(b,\mu):\mathbb{B}\times\mathbb{M}\to\mathbb{R}_{\geq 0}$, which represents the cost of invoking the local controller $\mu(\cdot)$ at the belief state b, i.e.,

$$C^s(b,\mu):=\sum_{t=0}^{\mathcal{T}} c(b_t,\mu(b_t)|b_0=b), \tag{27}$$

where $\mathcal{T}:=\mathcal{T}(\Psi|b,\mu)$.

Belief SMDP: According to the above definitions, the original POMDP, formulated using DP in Eq. (6), can be reduced to a Semi-Markov Decision Process (SMDP) [Sutton et al., 1999] in the belief space, referred to as a *belief SMDP*:

$$J^s(b)=\min_{\mu\in\mathbb{M}(i)} C^s(b,\mu)+\int_{\Psi} p(b'|b,\mu)J^s(b')db',\ \forall b\in B^i,\ \forall i. \tag{28}$$

The integration over the entire belief space in Eq. (6) is reduced to integration over the sampled nodes, i.e., Ψ, in Eq. (28) as μ stops executing.

D. FIRM MDP

Graph transitions: The DP in (28), though computationally more tractable than the original POMDP, is defined on the continuous neighborhoods B^i and thus is still formidable to solve. However, for sufficiently small B^i's, and sufficiently smooth cost functions, the cost-to-go of all beliefs in B^i, are approximately equal. Thus, we can define the graph-level transition cost and probabilities $C^g:\mathbb{V}\times\mathbb{M}\to\mathbb{R}$ and $\mathbb{P}^g:\mathbb{V}\times\mathbb{V}\times\mathbb{M}\to[0,1]$ on the FIRM graph, i.e., over the finite space $\mathbb{V}$, such that $\mathbb{P}^g(B^j|B^i,\mu)$ is the transition probability from B^i to B^j under the local planner μ. Similarly, $C^g(B^i,\mu)$ denotes the cost of invoking local planner μ at the FIRM node B^i. Accordingly, $J^g:\mathbb{V}\to\mathbb{R}$ is the cost-to-go function over the FIRM nodes. These roadmap level quantities are defined using the following "piecewise constant approximation", which is an arbitrarily good approximation for smooth enough functions and sufficiently small B^i's:

$$\forall b\in B^i,\forall i\quad \begin{cases} J^g(B^i):=J^s(b^i_c)\approx J^s(b),\\ C^g(B^i,\mu):=C^s(b^i_c,\mu)\approx C^s(b,\mu),\\ \mathbb{P}^g(\cdot|B^i,\mu):=\mathbb{P}(\cdot|b^i_c,\mu)\approx\mathbb{P}(\cdot|b,\mu),\end{cases} \tag{29}$$

where b^i_c is a representative point in B^i. For example, if B^i is a ball, the typical value for b^i_c is the center of B^i. This approximation essentially states that any belief in the region B^i is represented by b^i_c for the purpose of decision-making.

Obstacle-free FIRM MDP: Given the approximation in Eq. (29), the DP equation in Eq. (28) becomes:

$$\begin{aligned} J^g(B^i)=J^s(b^i_c)&=\min_{\mu\in\mathbb{M}(i)} C^s(b^i_c,\mu)+\int_{\Psi} p(b'|b^i_c,\mu)J^s(b')db'\\ &=\min_{\mu\in\mathbb{M}(i)} C^s(b^i_c,\mu)+\sum_j\int_{B^j} p(b'|b^i_c,\mu)J^s(b')db'\\ &\approx\min_{\mu\in\mathbb{M}(i)} C^g(B^i,\mu)+\sum_j\int_{B^j} p(b'|b^i_c,\mu)J^g(B^j)db'\\ &=\min_{\mu\in\mathbb{M}(i)} C^g(B^i,\mu)+\sum_j J^g(B^j)\mathbb{P}(B^j|b^i_c,\mu)\\ &=\min_{\mu\in\mathbb{M}(i)} C^g(B^i,\mu)+\sum_j J^g(B^j)\mathbb{P}^g(B^j|B^i,\mu),\ \forall i \end{aligned} \tag{30}$$

The approximation essentially states that any belief in the region B^i is represented by b^i_c for the purpose of decision-making. In other words, we can get the graph feedback $\pi^g:\mathbb{V}\to\mathbb{M}$ through the following DP:

$$J^g(B^i)=\min_{\mu\in\mathbb{M}(i)} C^g(B^i,\mu)+\sum_j\mathbb{P}^g(B^j|B^i,\mu)J^g(B^j),\ \forall i \tag{31a}$$

$$\pi^g(B^i)=\arg\min_{\mu\in\mathbb{M}(i)} C^g(B^i,\mu)+\sum_j\mathbb{P}^g(B^j|B^i,\mu)J^g(B^j),\ \forall i \tag{31b}$$

Thus, the original POMDP over the entire belief space, becomes a finite N_v-state MDP in Eq. (31) defined on the finite set of FIRM nodes $\mathbb{V}=\{B^i\}_{i=1}^{N_v}$. We call the MDP in Eq. (31) the FIRM MDP in the absence of obstacles. It is worth noting that $J^g(\cdot):\mathbb{V}\to\mathbb{R}$ is the cost-to-go function over the FIRM nodes, which assigns a cost-to-go for every FIRM node B^i and the mapping $\pi^g(\cdot):\mathbb{V}\to\mathbb{M}$ is a mapping over the FIRM graph, from FIRM nodes into the set of local controllers that returns the optimal local controller that has to be taken at any FIRM node. Given $C^g(B,\mu)$ for all (B,μ) pairs, the DP equation in Eq. (31) can be solved *offline* using standard DP techniques such as the value/policy iteration to yield a feedback policy π^g over FIRM nodes B^i.

E. Incorporating Obstacles into FIRM MDP

In the presence of obstacles (i.e., state or control constraints), we may not assure that the local controller $\mu^{ij}(\cdot)$ can drive any $b \in B^i$ into B^j with probability one. Instead, we have to specify the failure probabilities that the robot collides with an obstacle (hits the failure set F).

Let us generalize the transition probabilities by defining $\mathbb{P}(F|b,\mu)$ as the probability of hitting failure set F before hitting stopping region Ψ under μ starting from b. Similarly, we generalize $\mathbb{P}^g$ such that $\mathbb{P}^g(F|B^i,\mu) := \mathbb{P}(F|b^i_c,\mu)$. Finally, we generalize the cost-to-go function by adding F to its input set, i.e., $J^g : \{\mathbb{V}, F\} \to \mathbb{R}_{\geq 0}$, such that $J^g(F)$ is a user-defined suitably high cost for hitting obstacles. Note that the cost-to-go from the goal node is zero, i.e., $J^g(B^{goal}) = 0$. Therefore, we can modify Eq. (31) to incorporate constraints, by repeating the procedure in the previous subsection to get the FIRM MDP in the presence of obstacles:

$$J^g(B^i) = \min_{\mu \in \mathbb{M}(i)} C^g(B^i,\mu) + J^g(F)\mathbb{P}^g(F|B^i,\mu) + \sum_j J^g(B^j)\mathbb{P}^g(B^j|B^i,\mu), \tag{32a}$$

$$\pi^g(B^i) = \arg\min_{\mu \in \mathbb{M}(i)} C^g(B^i,\mu) + J^g(F)\mathbb{P}^g(F|B^i,\mu) + \sum_j J^g(B^j)\mathbb{P}^g(B^j|B^i,\mu). \tag{32b}$$

All that is required to solve the above DP equation are the values of the costs $C^g(B^i,\mu)$ and the transition probability functions $\mathbb{P}^g(\cdot|B^i,\mu)$. Thus, the main difference from the obstacle free case is the addition of a "failure" state to the FIRM MDP along with associated probabilities of failure from various nodes B^i.

F. Overall policy π

The overall feedback $\pi : \mathbb{B} \to \mathbb{U}$ is generated by combining the global policy π^g on the graph and local policies $\{\mu^{ij}\}$. Suppose at the k-th time step the active local controller is shown by μ^*_k. It remains unchanged $\mu^*_{k+1} = \mu^*_k$, and keeps generating control signals based on the belief b_k at each time step, until the belief reaches the corresponding stopping region, Ψ. Once the belief enters the stopping region $\Psi = \cup_j B^j$, it is in a graph node, say $B^k_* \in \mathbb{V}$. Accordingly, the global policy π^g chooses the next local controller, i.e., $\mu^*_{k+1} = \pi^g(B^k_*)$. Thus, this hybrid policy is stated as follows:

$$u_k = \pi(b_k) = \begin{cases} \mu^*_k(b_k), \ \mu^*_k = \pi^g(B^*_{k-1}), & \text{if } b_k \in B^*_{k-1} \\ \mu^*_k(b_k), \ \mu^*_k = \mu^*_{k-1}, & \text{if } b_k \notin \Psi \end{cases} \tag{33}$$

Initial controller: Given the initial belief is b_0, if b_0 is in one of the graph nodes, then we just choose the best local controller using π^g. However, if b_0 does not belong to any of the graph nodes, we first make a singleton set $B^0 = \{b_0\}$ and connect it to the graph nodes based on the connect methods discussed in Section VI-B. Denoting the outgoing edges (local controllers) from B^0 by $\mathbb{M}(0)$, we compute the transition cost $C^g(B^0,\mu)$, the transition probabilities $\mathbb{P}^g(B^j|B^0,\mu)$ for all j, and failure probability $\mathbb{P}(F|B^0,\mu)$ for invoking local controllers $\mu \in \mathbb{M}(0)$ at B^0. Then, we choose the best initial controller μ^0_* as:

$$\mu^0_* = \begin{cases} \arg\min_{\mu \in \mathbb{M}(0)} \{C^g(B^0,\mu) + \mathbb{P}^g(F|B^0,\mu)J^g(F) + \sum_j \mathbb{P}^g(B^j|B^0,\mu)J^g(B^j)\}, & \text{if } \nexists r, \text{ s.t. } b_0 \in B^r \\ \pi^g(B^r), & \text{if } \exists r, \text{s.t. } b_0 \in B^r \end{cases} \tag{34}$$

It is worth noting that computing μ^0_* is the only part of the computation that depends on the initial belief b_0 and that has to be performed online, i.e., if a large deviation occurs, μ^0_* is the only part that needs to be reproduced for the new initial point. After μ^0_* drives the system to a graph node, from thereon the optimal policy is already known. Moreover, computing μ^0_*, in case of large deviations, is feasible in real-time as $\mathbb{M}(0)$ contains a limited number of finite length edges.

G. Success probability

We would also like to quantify the quality of the solution π in the presence of obstacles. To this end, we require the probability of success of the policy π^g at the higher level Markov chain on B^i's given by Eq. (32b). Without loss of generality let us assume that the first node B^1 is the goal node B^{goal}. The DP in Eq. (32b) has $N+1$ states $\{F, B^{goal}, B^2, \cdots, B^N\}$ that can be decomposed into three disjoint classes: the failure class F, the goal class B^{goal}, and the transient class $\{B^2, B^3, \cdots, B^{N+1}\}$. The goal and failure classes are absorbing recurrent classes of this Markov chain. As a result, the transition probability matrix of this higher level $N+1$ state Markov chain can be decomposed as follows [Norris, 1997]:

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_f & 0 & 0 \\ 0 & \mathcal{P}_{goal} & 0 \\ \mathcal{R}_f & \mathcal{R}_{goal} & \mathcal{Q} \end{bmatrix}. \tag{35}$$

where, $\mathcal{P}_{goal} = \mathbb{P}^g(B^1|B^1,\cdot) = 1$ and $\mathcal{P}_f = \mathbb{P}^g(F|F,\cdot) = 1$, since goal and failure classes are the absorbing recurrent classes, i.e., the system stops once it reaches the goal or it fails. $\mathcal{Q}$ is a matrix that represents the transition probabilities between transient nodes in the transient class, whose (i,j)-th element is $\mathcal{Q}[i,j] = \mathbb{P}^g(B^{i+1}|B^{j+1}, \pi^g(B^{j+1}))$. Vectors $\mathcal{R}_{goal}$ and $\mathcal{R}_f$ are $(N-1)\times 1$ vectors that represent the probability of transient nodes $\mathbb{V} \setminus B^{goal}$ getting absorbed into the goal and failure node, respectively, i.e., $\mathcal{R}_{goal}[j] = \mathbb{P}^g(B^1|B^{j+1}, \pi^g(B^{j+1}))$ and $\mathcal{R}_f[j] = \mathbb{P}^g(F|B^{j+1}, \pi^g(B^{j+1}))$. Then, it can be shown that the success probability from any desired node $B^i \in \mathbb{V} \setminus B^{goal}$ is given as follows [Norris, 1997]:

$$\begin{aligned}\mathbb{P}(\text{success}|B^i,\pi^g) &:= \mathbb{P}(B^{goal}|B^i,\pi^g)\\ &= \Gamma_{i-1}^T (I-\mathcal{Q})^{-1}\mathcal{R}_{goal}, \quad \forall i \geq 2,\end{aligned} \tag{36}$$

where Γ_i is a column vector with all elements equal to zero except the i-th element which is set to one. Note that the vector $\mathcal{P}^s = (I-\mathcal{Q})^{-1}\mathcal{R}_{goal}$ includes the success probability from every graph node.

In the next section, we will discuss the success probability in more detail in the context of probabilistic completeness. However, according to the computed $\mathbb{P}(\text{success}|B^i,\pi^g)$, one can compute the success probability from any given initial belief b_0 as

$$\mathbb{P}(\text{success}|b_0,\pi) = \sum_j \mathbb{P}(B^j|b_0,\mu_0^*)\mathbb{P}(\text{success}|B^j,\pi^g), \tag{37}$$

where μ_0^* is given by Eq. (34). Then, this success probability is compared with a minimum acceptable success probability, denoted by p_{min}. If the condition $\mathbb{P}(\text{success}|b_0,\pi) > p_{min}$ is not satisfied, then the number of nodes in the graph has to be increased until the condition is satisfied. If, from the initial point b_0, a successful policy in the class of admissible policies exists, then this procedure will eventually find a successful policy by increasing the number of nodes, due to the probabilistic completeness of the method, which is discussed in Section VII-A.

H. Generic FIRM Algorithms

The generic algorithms for the offline construction of FIRM and online planning with FIRM are presented in Algorithms 3 and 4, respectively. Concrete instantiations of these algorithms for SLQG-FIRM are given in Algorithms 1 and 2, respectively.

Algorithm 3: Generic Construction of the FIRM graph (Offline)

1. Sample a set of stabilizer parameters $\mathcal{V} = \{\mathbf{v}^i\}$ and construct stabilizers $\mathbb{M} = \{\mu^i\}$ accordingly;
2. Sample set of belief nodes $\mathbb{V} = \{B^i\}$ such that they satisfy the reachability condition;
3. Connect the belief nodes using local controllers μ^{ij};
4. For each B^i and $\mu \in \mathbb{M}(i)$, compute the transition cost $C^g(B^i,\mu)$, and transition probabilities $\mathbb{P}^g(B^j|B^i,\mu)$ and $\mathbb{P}^g(F|B^i,\mu)$ associated with invoking μ at B^i;
5. Solve the graph DP in Eq. (32) to compute feedback π^g over graph nodes, and compute the π accordingly;

Algorithm 4: Generic planning (or replanning) on FIRM (Online)

1. Given an initial belief b_0, invoke the controller $\mu_0(\cdot)$ in Eq. (34), to take the robot into some FIRM node B;
2. **while** $B \neq B^{goal}$ **do**
3. Given the system is in FIRM node B, invoke the global feedback policy π^g to choose the local feedback policy $\mu(\cdot) = \pi^g(B)$;
4. Let the local controller $\mu(\cdot)$ execute until the robot is absorbed into a FIRM node B' or until it hits the failure set;
5. **if** *Collision happens* **then return** Collision;
6. Update current node $B \leftarrow B'$;

Single-query versus multi-query: As mentioned earlier, most approaches for planning in belief space in continuous state, action, and observation spaces result in query-dependent plans. However, one of the contributions of FIRM is that its construction does not depend on the query. In Algorithms 3 and 4, it is assumed that the goal is fixed for all queries; in this case in the planning phase we are only robust to changes in the starting point of the query. However, to make the algorithms also robust to changes in the goal belief, one can just move the last line of Algorithm 3 to the first line of Algorithm 4. Note that the computationally expensive part of Algorithm 3 is the computation of edge costs, which is independent of the start and goal location of the submitted query.

I. Discussion

In summary, in FIRM we aim to transform the original POMDP into a belief SMDP and solve it on a subset of the belief space. Given the smoothness of the cost function and transition probabilities, the solution of the FIRM MDP is arbitrarily close to the solution of the belief SMDP over FIRM nodes. The important characteristic of FIRM is that it is solved offline and thus performing the online phase of planning (or replanning) is computationally feasible in real-time. To exploit the generic FIRM framework, one has to find (B, μ) pairs, where B is reachable (or αT-reachable) under μ, as FIRM nodes and edges. Also, transition costs and probabilities need to be computed. Finally, the corresponding FIRM MDP needs to be solved, which provides a global feedback policy on the graph that can be used in planning, as detailed in Algorithm 4. SLQG-FIRM, presented in Section V, is an instant of FIRM, in which the design of local controllers μ^{ij} and FIRM nodes B^i is based on the properties of SLQG controllers.

VII. Probabilistic Completeness Under Uncertainty

In this section, we extend the concept of probabilistic completeness of planning algorithms for deterministic systems to the concept of probabilistic completeness of planning algorithms under uncertainty based on [Agha-mohammadi et al., 2012b]. Accordingly, in the next subsection, we discuss the probabilistic completeness of the FIRM-based algorithms. We start by reviewing the definition of success and probabilistic completeness in the deterministic case, and then we extend these definitions to the stochastic case.

Success in the deterministic case: In the deterministic case, such as conventional PRM, the outcome of the planning algorithm is a path. Thus, success is defined for paths: For a given initial and goal point, a successful path is a path connecting the start point to the goal point, which entirely lies in the obstacle-free space.

Probabilistic completeness in the deterministic case: In the absence of uncertainty, a sampling-based motion planning algorithm is probabilistically complete if by increasing the number of samples, the probability of finding a successful path, if one exists, asymptotically approaches to one.

A difference between the deterministic and the probabilistic case: In the presence of uncertainty, success cannot be defined for a path and it has to be defined for a policy. Indeed, on a given path, different policies may result in different success probabilities. Moreover, under uncertainty, one can only assign a probability for reaching goal. Thus, to define success for a policy we consider a threshold $p_{min} \in [0, 1]$ and decide about success or failure accordingly.

Successful policy: In the presence of uncertainty, the solution of the planning algorithm is a function, called a closed-loop policy or feedback. Therefore, success is defined for policies: For a given initial belief b_0 and goal region B^{goal}, a successful policy is a policy under which the probability of reaching the goal from the given initial point is greater than some predefined threshold p_{min}. In other words, π is successful for a given b_0 if $\mathbb{P}(\text{success}|b_0, \pi) := \mathbb{P}(B^{goal}|b_0, \pi) > p_{min}$.

Policy in sampling-based methods: In sampling-based methods, a policy is parametrized by a set of samples. These samples can be in the state or belief space, based on the algorithm. Let us denote these samples in a generic space by $\{\gamma_1, \gamma_2, \cdots, \gamma_N\}$. Thus, we can highlight the dependency of the sampling-based policy on the samples by the notation $\pi(\cdot; \{\gamma_1, \gamma_2, \cdots, \gamma_N\})$. The number of samples is denoted by N.

Strong Probabilistic Completeness Under Uncertainty (SPCUU): Suppose there exists a successful policy $\breve{\pi}$. Then a sampling-based motion planning algorithm is SPCUU if increasing the number of samples without bound causes the probability of finding a successful policy to approach one. In other words, if there exists a successful policy $\breve{\pi}$, then we have the following property for the sampling-based policy π:

$$\lim_{N \to \infty} \mathbb{P}(B^{goal}|b_0, \pi) > p_{min}, \tag{38}$$

where N is the number of samples in the sampling-based method.

Achieving an algorithm that is SPCUU requires searching in the entire space of policies, which is a computationally intractable task. Usually, in solving POMDPs the space of admissible policies is restricted to a sufficiently rich subset of policy space, denoted by Π, within which the method searches for the best policy. Restricting the successful policy to the set Π, we define a weaker notion of probabilistic completeness under uncertainty:

Probabilistic completeness under uncertainty (PCUU): Suppose there exists a successful policy $\breve{\pi} \in \Pi$. Then, a sampling-based motion planning algorithm is probabilistically complete under uncertainty (PCUU), if increasing the number of samples without bound, the probability of finding a successful policy approaches one. In other words, if there exists a successful policy $\breve{\pi} \in \Pi$, then for the sampling-based policy π, we have $\lim_{N \to \infty} \mathbb{P}(B^{goal}|b_0, \pi) > p_{min}$.

As discussed in Section VI, in FIRM, inspired by the sampling-based PRM framework, this reduction from the entire function space to the restricted set of policies Π is performed by sampling feedback local planners and concatenating them. Therefore, the structure of local planners defines the set Π. Each local planner μ^{ij} is parametrized by its corresponding parameter $\mathbf{v}^j$. However, as mentioned in Section VI-B, we can consider the set $\mathcal{V} = \{\mathbf{v}^i\}$ as the set of an underlying PRM nodes. Thus, any policy $\pi \in \Pi$ is parametrized by the set of underlying PRM nodes $\mathcal{V} = \{\mathbf{v}^i\}_{i=1}^{N_v}$. We highlight this dependency explicitly

through the notation $\pi(\cdot;\mathcal{V})$. Therefore, the PCUU condition for FIRM can be written more explicitly as:

$$\lim_{N_v\to\infty} \mathbb{P}(B^{goal}|b_0,\pi(\cdot;\mathcal{V})) > p_{min}. \tag{39}$$

For a concrete instantiation of FIRM, we can explicitly characterize the set Π. For example, in SLQG-FIRM, Π is the set of all possible policies that can be generated by concatenating LQG controllers.

A. Probabilistic Completeness of FIRM

Obviously, FIRM-based methods are not SPCUU algorithms. However, in this section, we show that under mild practical conditions, FIRM-based methods are PCUU algorithms. We first provide an analysis of the local planners in belief space, and then state the assumptions more rigorously.

Notation: The norm $\|\cdot\|$ is the supremum norm, when it is applied to functions. The norm $\|\cdot\|_{op}$ is applied on operators and it stands for the operator norm [Keener, 2000]. It is worth noting that in this section, by the word "continuous" we mean "Lipschitz continuous." Finally, we assume that $\mathbb{X}_{free}$ is a compact set.

Hyper-state: $\mathcal{X}=(x,b)\in\mathbb{X}_h$ is referred to as hyper-state (or h-state), which is a state-belief pair. The space of all h-states is called hyper-state space (h-state space) $\mathbb{X}_h=\mathbb{X}\times\mathbb{B}$. The $p^{\mu}(\mathcal{X}'|\mathcal{X})$ denotes the one-step transition pdf induced by the local controller μ, over the h-state space. Also, let $\mathbb{P}_n(S|\mathcal{X},\mu)$ denote the transition probability from h-state $\mathcal{X}$ into the set $S\subset\mathbb{X}_h$ in at most n steps.

Local planner and extended stopping region: The role of the (i,j)-th local planner or local controller is to drive the belief from the region B^i to its stopping region B^j in the belief space (for the ease of notation, we ignore the case that the controller can stop in any FIRM node, and we restrict its stopping region to B^j). In the presence of obstacles, we extend the concept of stopping region to include obstacles also. The stopping regions $\{B^j\}$ in the belief space and the stopping region F in the state space, both can be extended to the h-state space, respectively denoted by $\{\mathcal{B}^j\}$ and $\mathcal{F}$, where $\mathcal{B}^j\subset\mathbb{X}_h$ and $\mathcal{F}\subset\mathbb{X}_h$ are defined as:

$$\mathcal{B}^j := \{(X,b)|X\in\mathbb{X}_{free}, b\in B^j\}, \tag{40}$$

$$\mathcal{F} := \{(X,b)|X\in F, b\in\mathbb{B}\}, \tag{41}$$

$$\mathcal{S}^j := \mathcal{B}^j\cup\mathcal{F}, \quad \overline{\mathcal{S}}^j := \mathbb{X}_h\setminus\mathcal{S}^j \tag{42}$$

where $\mathcal{S}^j$ and $\overline{\mathcal{S}}^j$, respectively, denote the entire stopping region and transient region under the local controller μ^{ij}.

Absorption probability of local planners: If, under the dynamics induced by the local planner, the system reaches the target node $\mathcal{B}^j$, the local planner is considered to be successful, and if the system hits an obstacle, the local planner is considered to be failed. The success probability of local planners, i.e., the absorption probability into FIRM nodes, is computed through solving the following integral equation that results from the law of total probability:

$$\begin{aligned}\mathbb{P}(\mathcal{B}^j|\mathcal{X},\mu^{ij}) &= \int_{\mathbb{X}_h} p^{\mu^{ij}}(\mathcal{X}'|\mathcal{X})\mathbb{P}(\mathcal{B}^j|\mathcal{X}',\mu^{ij})d\mathcal{X}' \\ &= \int_{\mathcal{B}^j} p^{\mu^{ij}}(\mathcal{X}'|\mathcal{X})d\mathcal{X}' + \int_{\overline{\mathcal{S}^j}} p^{\mu^{ij}}(\mathcal{X}'|\mathcal{X})\mathbb{P}(\mathcal{B}^j|\mathcal{X}',\mu^{ij})d\mathcal{X}'.\end{aligned} \tag{43}$$

where the second equality in Eq. (43) follows from substituting the following conditions, inherited from FIRM construction, into the first integral:

$$\mathbb{P}(\mathcal{B}^j|\mathcal{X},\mu^{ij}) = \begin{cases}1, & \text{if } \mathcal{X}\in\mathcal{B}^j\\ 0, & \text{if } \mathcal{X}\in\mathcal{F}\end{cases}. \tag{44}$$

Henceforth, we drop indices i and j to unclutter expressions. Thus, we can write:

$$\begin{aligned}\mathbb{P}(\mathcal{B}|\mathcal{X},\mu) &= \int_{\mathcal{B}} p^{\mu}(\mathcal{X}'|\mathcal{X})d\mathcal{X}' + \int_{\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X})\mathbb{P}(\mathcal{B}|\mathcal{X}',\mu)d\mathcal{X}' \\ &= R(\mathcal{X}) + \mathbf{T}_{\mathcal{S}}\left[\mathbb{P}(\mathcal{B}|\cdot,\mu)\right](\mathcal{X}),\end{aligned} \tag{45}$$

where the operator $\mathbf{T}_{\mathcal{S}}$ and the function $R(\mathcal{X})$ are defined as:

$$\mathbf{T}_{\mathcal{S}}\left[f(\cdot)\right](\mathcal{X}) := \int_{\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X})f(\mathcal{X}')d\mathcal{X}', \quad R(\mathcal{X}) := \int_{\mathcal{B}} p^{\mu}(\mathcal{X}'|\mathcal{X})d\mathcal{X}'. \tag{46}$$

The solution of the integral equation in Eq. (45) is expressed in the following as a Liouville-Neumann series [Keener, 2000], similar to the solution of the inhomogeneous Fredholm equation of second type [Keener, 2000].

$$\mathbb{P}(\mathcal{B}|\mathcal{X},\mu) = \sum_{n=1}^{\infty} \mathbf{T}_{\mathcal{S}}^{n}\left[R(\cdot)\right](\mathcal{X}). \tag{47}$$

We show that the series in Eq. (47) is a convergent series by resorting to the following assumption, which is a weaker version of the aforementioned FIRM condition on the design of nodes and local controllers.

Assumption 1. *We assume that there exists some time step N, at which the controller stops with a positive probability. Mathematically, there exists an $N < \infty$ and $\beta > 0$ such that $\mathbb{P}_N(\mathcal{S}^j|\mathcal{X},\mu^{ij}) \geq \beta > 0$, for all $\mathcal{X}$.*

This assumption is almost always true, as it rephrases the role of controller in driving the system toward the target region. For example, if we have Gaussian noise (as is the case in the SLQG-FIRM), the assumption is true in $N = 1$ regardless of the utilized controller.

Lemma 4. *Given Assumption 1, we have:*

$$\begin{cases} \|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} \leq 1, & n < N \\ \|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} \leq 1-\beta < 1, & n \geq N \\ \sum_{n=0}^{\infty} \|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} \leq c < \infty. \end{cases} \tag{48}$$

Proof: See Appendix E. ∎

Corollary 1. *The series $\sum_{n=0}^{\infty} \mathbf{T}_{\mathcal{S}}^{n}[R]$ is a convergent series, and therefore we can define the resolvent operator $(I - \mathbf{T}_{\mathcal{S}})^{-1}[R] = \sum_{n=0}^{\infty} \mathbf{T}_{\mathcal{S}}^{n}[R]$, where $\|(I-\mathbf{T}_{\mathcal{S}})^{-1}\|_{op} \leq c < \infty$.*

Proof: See Appendix F. ∎

According to Corollary 1, the success probability of the local controller μ can be written using the defined resolvent operator as:

$$\mathbb{P}(\mathcal{B}|\mathcal{X},\mu) = (I-\mathbf{T}_{\mathcal{S}})^{-1}[R(\cdot)](\mathcal{X}). \tag{49}$$

As the first result of this section (Proposition 1), we aim to show that this absorption probability varies continuously with respect to changes in parameters of the local planner. However, we will first state two more assumptions.

Assumption 2. *We assume the local planning law and induced transition probabilities are smooth, i.e.,*

- *Local control laws are continuous in their parameters, i.e., for the (i,j)-th local controller, mapping $\mu^{ij}(\cdot;\mathbf{v}^j) : \mathbb{B} \to \mathbb{U}$ is a continuous function in its parameter $\mathbf{v}^j$.*
- *The transition pdf on h-state, i.e., $p(\mathcal{X}'|\mathcal{X},u)$ is a continuous function of the control u, i.e., there exists a $c_1 < \infty$, such that $\|p(\mathcal{X}'|\mathcal{X},u) - p(\mathcal{X}'|\mathcal{X},\check{u})\| \leq c_1\|u-\check{u}\|$.*

Finally, we state the following assumption, in which we emphasize the fact that, as $\mathbf{v} \to \check{\mathbf{v}}$, the transition probability induced by the local controller $\mu(\cdot;\mathbf{v})$ into the sets $\mathcal{B}$ and $\check{\mathcal{B}}$ have to converge also, which is a reasonable assumption for a smooth control law.

Assumption 3. *Consider the controllers $\mu(\cdot;\mathbf{v})$, and $\check{\mu}(\cdot;\check{\mathbf{v}})$, whose corresponding extended absorption regions are denoted by $\mathcal{B}$ and $\check{\mathcal{B}}$, respectively. We assume that there exist real numbers $r > 0$ and $c' < \infty$, such that for $\|\mathbf{v}-\check{\mathbf{v}}\| \leq r$, we have:*

$$\|\mathbb{P}_1(\mathcal{B} \ominus \check{\mathcal{B}}|\mathcal{X},\mu)\| \leq c'\|\mathbf{v}-\check{\mathbf{v}}\| \tag{50}$$

where $\ominus$ is the symmetric difference operator, i.e., $\mathcal{B} \ominus \check{\mathcal{B}} = (\mathcal{B} \setminus \check{\mathcal{B}}) \cup (\check{\mathcal{B}} \setminus \mathcal{B})$.

Now we state the following proposition on the continuity of the success probability of local planners:

Proposition 1. (Continuity of absorption probabilities): Given Assumptions 1, 2, and 3, the absorption probability $\mathbb{P}(B^j|b,\mu^{ij})$ is continuous in parameter $\mathbf{v}^j$ for all $i, j,$ and b.

Proof: See Appendix G. ∎

Now we present the main result regarding the probabilistic completeness of FIRM-based methods:

Theorem 1. *Given Assumptions 1, 2, and 3, any planning algorithm under uncertainty that is generated based on the FIRM framework (i.e., guarantees belief node reachability and induces a roadmap in the belief space with independent edge costs) is probabilistically complete under uncertainty (PCUU).*

Proof: See Appendix H. ∎

The basic idea of probabilistic completeness under uncertainty stems from an idea similar to the one in the path isolation-based analysis for planners in deterministic systems. Roughly speaking, in the path isolation argument for sampling-based planners in the absence of uncertainty, if there is a successful path and a non-zero neighborhood of this path, in which every path is successful, we can eventually find a path in this neighborhood, by increasing the number of samples, unboundedly. Similarly, in the presence of uncertainty, if there is a successful policy, it is parametrized by some parameters (set of PRM nodes, in FIRM). Thus, if there exists a non-zero measure neighborhood of these parameters, within which selected parameters lead to a successful policy, we can eventually reach a successful policy by increasing the number of samples unboundedly and choosing samples in the target neighborhoods.

VIII. Experimental Results

In this section, we first illustrate theoretical results from the previous sections on a planar robot in a small three-dimensional planning domain. Then, we present planning results on a larger three-dimensional state space. Finally, we report the results of the method on a 8-DOF manipulator. This section is followed by a brief comparison with other state-of-art methods in Section IX.

A. Planar 3D Omnidirectional Robot: Illustrating Steps in Construction and Planning with SLQG-FIRM

In this subsection, we focus on an omni-directional robot. Its state is composed of its 2D position in the plane and its heading angle. The goal in this section is to illustrate the steps of constructing SLQG-FIRM and planning with it.

1) Motion Model: A 3-wheel omnidirectional mobile robot is used in experiments with the nonlinear kinematic model given in [Kalmár-Nagy et al., 2004]. The state vector is composed of a 2D location and heading angle $x = [{}^1x, {}^2x, \theta]^T$ in a global world frame. $u = [{}^1u, {}^2u, {}^3u]^T$ is the vector of controls, where ${}^iu$ is the linear velocity of the i-th wheel. w is the motion noise, which is drawn from a zero-mean Gaussian distribution. The motion dynamics for this robot, in its original continuous form is [Kalmár-Nagy et al., 2004]:

$$\dot{x} = \mathbf{f}_c(x, u, w) = T(x)u + w, \tag{51}$$

where

$$T(x) = \begin{pmatrix} -\frac{2}{3}\sin(\theta) & -\frac{2}{3}\sin(\frac{\pi}{3} - \theta) & \frac{2}{3}\sin(\frac{\pi}{3} + \theta) \\ \frac{2}{3}\cos(\theta) & -\frac{2}{3}\cos(\frac{\pi}{3} - \theta) & -\frac{2}{3}\cos(\frac{\pi}{3} + \theta) \\ \frac{1}{3r} & \frac{1}{3r} & \frac{1}{3r} \end{pmatrix}, \tag{52}$$

where r is the distance of the wheels from the robot's center of mass. The discrete motion dynamics is shown by:

$$x_k = \mathbf{f}(x_{k-1}, u_{k-1}, w_{k-1}). \tag{53}$$

$w_k \sim \mathcal{N}(0, \mathbf{Q})$ is the motion noise at the k-th time step, which is drawn from a zero-mean Gaussian distribution with covariance matrix $\mathbf{Q}$. It can be shown that if we linearize this system, the linearized motion model satisfies the controllability condition in Property 1.

2) Observation Model: In experiments, the robot is equipped with exteroceptive sensors that provide range and bearing measurements from existing landmarks (radio beacons) in the environment. The 2D location of the j-th landmark is denoted by L_j. Measuring L_j can be modeled as follows:

$${}^jz = {}^jh(x, {}^jv) = [\|{}^j\mathbf{d}\|, \text{atan2}({}^jd_2, {}^jd_1) - \theta]^T + {}^jv, \quad {}^jv \sim \mathcal{N}(\mathbf{0}, {}^j\mathbf{R}),$$

where ${}^j\mathbf{d} = [{}^jd_1, {}^jd_2]^T := [{}^1x, {}^2x]^T - L_j$. The vector ${}^jv$ is a state-dependent observation noise, with covariance

$${}^j\mathbf{R} = \text{diag}((\eta_r\|{}^j\mathbf{d}\| + \sigma_b^r)^2, (\eta_\theta\|{}^j\mathbf{d}\| + \sigma_b^\theta)^2). \tag{54}$$

In other words, the uncertainty (standard deviation) of the sensor reading increases as the robot gets farther from the landmarks. $\eta_r = \eta_\theta = 0.3$ determines this dependence, and $\sigma_b^r = 0.01$ meter and $\sigma_b^\theta = 0.5$ degrees are the bias standard deviations. A similar model for range sensing is used in [Prentice and Roy, 2009]. We assume the robot observes all N_L landmarks at all times and their observation noises are independent. Thus, the total measurement vector is denoted by $z = [{}^1z^T, {}^2z^T, \cdots, {}^{N_L}z^T]^T$, and, due to the independence of measurements of different landmarks, the observation model for all landmarks can be written as:

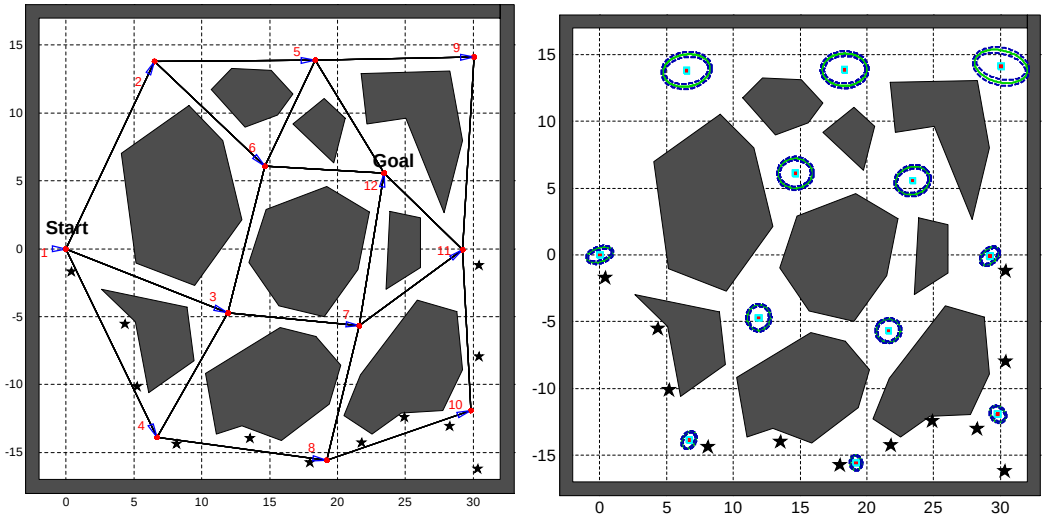
$$z = h(x) + v, \quad v \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \quad \mathbf{R} = \text{diag}({}^1\mathbf{R}, \cdots, {}^{N_L}\mathbf{R}). \tag{55}$$

It is straightforward to show that the linearized version of this observation model satisfies the observability condition in Property 1. Therefore, this entire system model (motion and sensing models) satisfies Property 1 and thus the SLQG-FIRM can be used for planning.

3) Construction of SLQG-FIRM Nodes and Edges: Figure 4(a) shows a sample environment, including obstacles, landmarks, and enumerated nodes in $({}^1x, {}^2x, \theta)$ space. Nodes are shown by blue triangles, which encode the position $({}^1x, {}^2x)$ and heading angle θ of the robot. Landmarks are shown by black stars. The corresponding FIRM nodes are computed and shown in Fig. 4(b). All elements in Fig. 4(b) are defined in $({}^1x, {}^2x, \theta)$ space but only the $({}^1x, {}^2x)$ portion of them is shown here. Each $b_c^j = (\mathbf{v}^j, P_s^j)$ is illustrated by a red dot representing $\mathbf{v}^j$ and a green ellipse, representing 3σ ellipse of covariance P_s^j. Each FIRM node B^j is a neighborhood around b_c^j. In the experiments, we define the node region using the component-wise version of Eq. (18), to handle the error scale difference in position and orientation variables:

$$B^j = \{b = (x, P) | \ \ |x - \mathbf{v}^j| \dot{<} \epsilon, |P - P_s^j| \dot{<} \Delta\}, \tag{56}$$

where $|\cdot|$ and $\dot{<}$ stand for the absolute value and component-wise comparison operators, respectively. We also set $\epsilon = [0.07(meter), 0.07(meter), 1(degree)]^T$ and $\Delta = \epsilon\epsilon^T$ to quantify B^j's. The projection of B^j onto the space of estimation mean, i.e., $B_x^j = \{\widehat{x}^+ \ : \ |\widehat{x}^+ - \mathbf{v}^j| \dot{<} \epsilon\}$ is a neighborhood around $\mathbf{v}^j$, which is shown by a cyan rectangle centered at $\mathbf{v}^j$. Projection of B^j onto the space of estimation covariances, i.e., $B_P^j = \{P \ : \ |P - P_s^j| \dot{<} \Delta\}$ is a neighborhood around P_s^j. However, in a 2D plot B_P^j cannot be shown due to its high dimension. Thus, we partially illustrate it only by two dashed green ellipses that represent 3σ covariances of $P_s^j - \Delta_d$ and $P_s^j + \Delta_d$, where Δ_d is the matrix Δ, whose off-diagonal elements are set to zero. For illustration purposes, both of these neighborhoods, i.e., B_x^j and B_P^j, are five times magnified in Fig. 4(b).

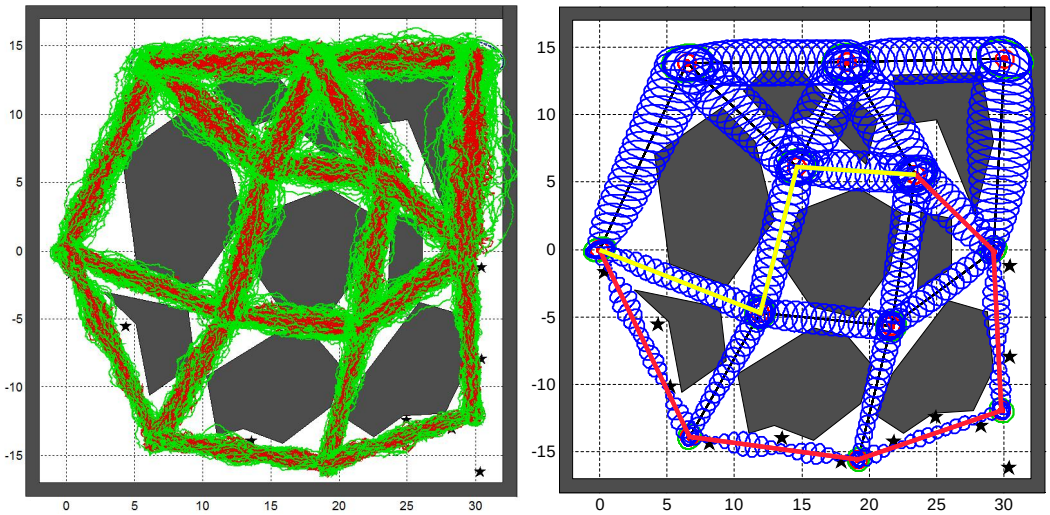


(a) (b)

Fig. 4. (a) Figure depicts the underlying PRM graph. Gray polygons are the obstacles and black stars represent the landmarks' locations. (b) FIRM nodes corresponding to PRM nodes.

4) Transition Costs and Probabilities: After designing FIRM nodes and local controllers, the transition costs and probabilities have to be computed. Based on the given task and needed accuracy, different approaches can be taken. Here, we use a particle-based approximation of the distribution to compute these quantities, and we use $M = 100$ particles. In other words, for every (B, μ) pair, we perform 100 runs. At every run, a sample path of state x, a sample path of estimation mean $\widehat{x}^+$, and a sample path of estimation covariance P is generated. If the filter of choice in the edge-controller is the Linearized Kalman Filter (LKF) [Crassidis and Junkins, 2004], [Simon, 2006], the covariance evolution is deterministic and there is no need to generate 100 different sample covariance paths. However, if the filter of choice in the edge-controller is the Extended Kalman Filter (EKF) [Crassidis and Junkins, 2004], [Simon, 2006], then we have to generate the sample covariance paths too, to take into account the stochasticity of the covariance matrix. Figure 5(a) depicts sample paths of the true state x and estimation mean $\widehat{x}^+$ in green and dark red, respectively, for $M = 100$ particles. Note that when a true state path (green path) collides with an obstacle, the process stops and failure happens. However, in this figure, for illustration purposes, we continue the process and ignore the obstacles to better show the uncertainty tube and information availability at different parts of the space. As seen in Fig. 5(a), the behavior of the true state on the edges, which have access to more accurate observations, is remarkably closer to the planned behavior. In contrast, on the edges that get less informative observations, the controller cannot effectively compensate for deviations of the ground truth from the nominal path, which can lead to collision with obstacles.

To avoid clutter, Fig. 5(b) depicts sample estimation covariance evolution only for a single particle. In this figure, we let the process and observation noise be zero, to keep the centers of ellipses (i.e., estimation mean) on the planned points. However, note that, in general, estimation mean is affected by the noise (as it is seen in Fig. 5(a)). Indeed, Fig. 5(b) can be seen as the maximum-likelihood estimation uncertainty tube over the roadmap.



(a) (b)

Fig. 5. Sample paths induced by controllers invoked at different nodes. (a) For $M = 100$ particles, sample ground truth paths and sample estimation mean paths are shown in green and dark red, respectively. (b) The most likely path under the optimal policy and shortest path are shown in red and yellow respectively. The 3σ ML estimation uncertainty tube is drawn in blue.

Let us denote the q-th sample path for the true state by $x^{(q)}_{0:\mathcal{T}^q}$, for the estimation mean by $\widehat{x}^{+(q)}_{0:\mathcal{T}^q}$, and for the estimation covariance by $P^{(q)}_{0:\mathcal{T}^q}$, where $\mathcal{T}^q$ is the stopping time of the q-th particle in executing μ at B. Moreover, one can assign a weight to each run q based on its probability of occurrence. There are different ways proposed to compute these weights in the Sequential Monte Carlo literature [Doucet et al., 2001]. However, the main condition is that they have to sum to one, i.e., $\sum_{q=1}^{M} w^{(q)} = 1$. Here we simply consider $w^{(q)} = M^{-1}$. Note that if we run μ^{ij} at B^i, all these quantities also have to have a ij superscript. Now, having these sample paths, we can compute the transition costs and probabilities associated with invoking the μ^{ij} at B^i. For the collision probability, we have:

$$\mathbb{P}^g(F|B^i, \mu^{ij}) = \mathbb{E}[\mathbb{I}_F|B^i, \mu^{ij}] \approx \sum_{q=1}^{M} w^{(q)} \mathbb{I}_F(x^{(q)}_{0:\mathcal{T}^q}) \tag{57}$$

$$\mathbb{P}^g(B^j|B^i, \mu^{ij}) = 1 - \mathbb{P}^g(F|B^i, \mu^{ij}) \tag{58}$$

where $\mathbb{I}_F$ is the failure indicator. It is one if there exists a time step $k \leq \mathcal{T}^{(q)}$, such that $x_k \in F$. Otherwise it is zero. $\mathcal{T}^q$, or more rigorously $\mathcal{T}^{ij^{(q)}}$, is the stopping time of the q-th particle in executing μ^{ij} at B^i. To compute $\mathcal{T}^{ij^{(q)}}$, we only need to check the condition $b \in B^j$ at every time step and find the first time step that belief b enters the stopping region B^j. Thus, we can compute the mean stopping time as

$$\widehat{\mathcal{T}}^{ij} = \mathbb{E}[\mathcal{T}^{ij}] \approx \sum_{q=1}^{M} w^{(q)} \mathcal{T}^{ij^{(q)}}. \tag{59}$$

To compute the filtering cost defined in Section V-E, again we use the particle-based representation of belief:

$$\Phi^{ij} = \mathbb{E}[\sum_{k=1}^{\mathcal{T}^{ij}} \text{tr}(P_k)|B^i, \mu^{ij}] \approx \sum_{q=1}^{M} \sum_{k=1}^{\mathcal{T}^q} w^{(q)} \text{tr}(P^{(q)}_k), \tag{60}$$

where $P^{(q)}_k$ is the estimation covariance at the k-th time step of q-th particle. Finally, the cost of taking μ^{ij} at B^j is as follows:

$$C^g(B^i, \mu^{ij}) = \alpha_1 \Phi^{ij} + \alpha_2 \widehat{\mathcal{T}}^{ij}$$

where we used the coefficients $\alpha_1 = 0.95$ and $\alpha_2 = 0.05$. Table I shows these quantities for several (B^i, μ^{ij}) pairs in corresponding to Fig. 5.

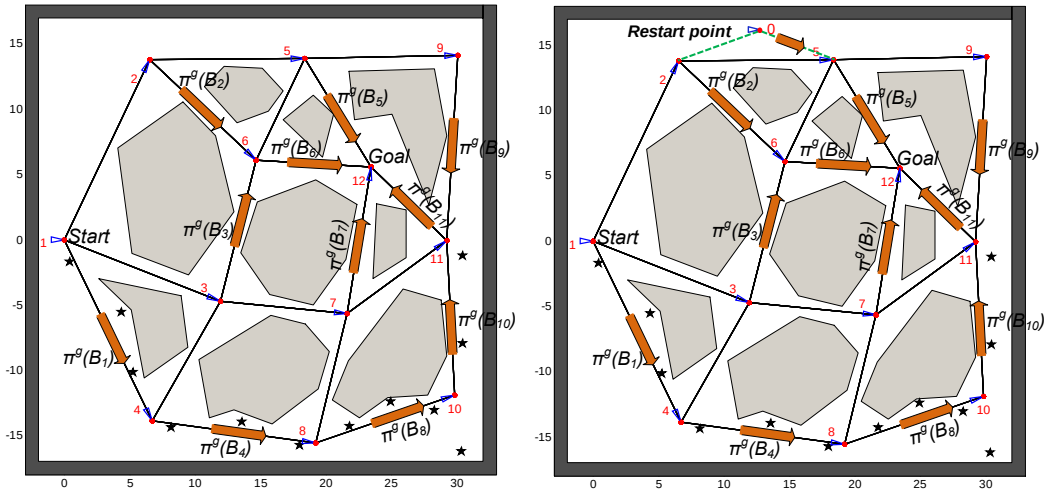
5) Planning and Replanning on FIRM: Plugging the computed transition costs and probabilities into Eq. (31), we can solve the DP and compute the graph policy π^g. This process is performed once offline if the goal location is fixed. Fig. 6(a) shows the policy π^g on the constructed FIRM in this example. Indeed, at every FIRM node B^i, the policy π^g decides which local controller has to be taken, which in turn aims to take the robot to the next FIRM node. Thus, the online part of the planning is significantly efficient and only reduces to executing the controller and generating the control signal, which is almost an instantaneous computation.

TABLE I
COMPUTED COSTS FOR SEVERAL NODE-CONTROLLER PAIRS IN FIRM USING 100 PARTICLES

(B^i,μ^{ij}) pair	$B^1,\mu^{1,4}$	$B^4,\mu^{4,8}$	$B^8,\mu^{8,10}$	$B^{10},\mu^{10,11}$	$B^{11},\mu^{11,12}$	$B^1,\mu^{1,3}$	$B^3,\mu^{3,6}$	$B^6,\mu^{6,12}$
$\mathbb{P}^g(B^j\|B^i,\mu^{ij})$	%97	%95	%99	%77	%79	%87	%55	%79
Φ^{ij}	18.5967	11.2393	6.8229	15.1148	26.2942	23.6183	48.8189	43.6207
$\mathbb{E}[\mathcal{T}^{ij}]$	238.2	193.0	150.0	209.6	170.8	200.3	242.4	219.2
$\sigma[\mathcal{T}^{ij}]$	21.8	28.7	15.1	24.5	22.6	22.7	30.1	26.7

Replanning: An important consequence of this framework is that replanning can be performed using FIRM efficiently. Suppose due to some unmodeled large disturbance, the robot's belief deviates significantly from the planned path, i.e., for some appropriate norm $\|\cdot\|$ on belief space we have $\|b_k - \mathbb{E}[b_k^p]\| > \varrho$, where b_k^p is the planned belief at k-th time step, and ϱ is the threshold for deciding if replanning is needed or not. In such cases, replanning occurs and based on Algorithm 2. In Fig. 6(b), we illustrate a simple replanning process. In this figure it is assumed that an unmodeled large disturbance affects the system, such that the estimation mean significantly deviates from the planned path. The deviated mean is shown on the figure as the "restart point". Thus, based on Algorithm 2, we connect this point to the PRM. In Fig. 6(b) the newly added PRM edges, i.e., $\mathcal{E}(0)$, are shown by dashed green lines. Then, for every edge in $\mathcal{E}(0)$, we design a local controller. Call the set of newly constructed local controllers $\mathbb{M}(0)$. For every $\mu \in \mathbb{M}(0)$ compute corresponding transition costs and probabilities. Finally, according to Bellman's principle of optimality, we use the precomputed cost-to-go's $J^g(\cdot)$ to decide which controller has to be taken at the "restart point" using Eq. (34). Taking this controller, the belief state returns to the FIRM nodes, and from there again we can use the precomputed π^g to control the robot toward the goal region.

We show the most likely path under π^g in red in Fig. 5(b). The shortest path is also illustrated in Fig. 5(b) in yellow. It can be seen that the "most likely path under the best policy" detours from the shortest path to a path along which the filtering uncertainty is smaller and it is easier for the controller to avoid the collisions.



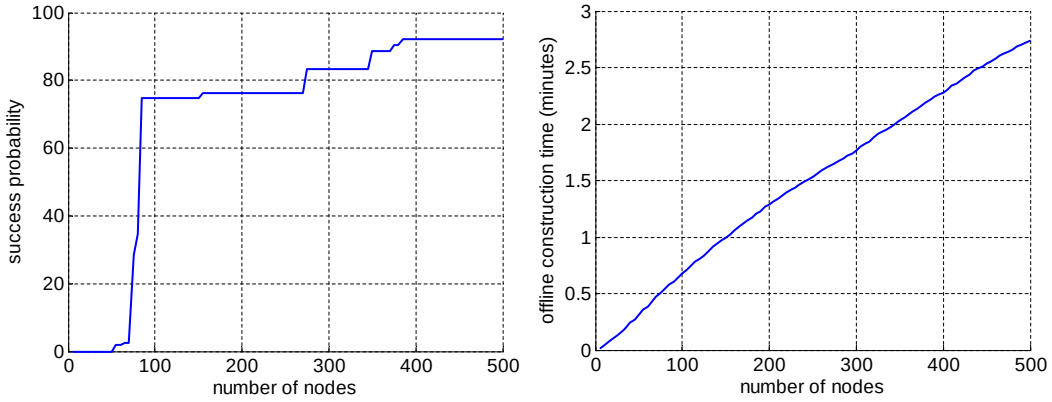
(a) (b)

Fig. 6. Planning and replanning on FIRM. (a) Policy π^g resulted from solving DP in Eq. (31) is shown by red arrows. Indeed for every FIRM node, the policy π^g tells that which controller has to be taken. (b) In this figure it is assumed that an unmodeled large disturbance affects the system, such that the estimation mean significantly deviates from the planned path. The deviated mean is denoted by "restart point" on the figure.

B. Larger Environment

In this section, we consider the same omnidirectional robot with the same observation model, and we perform planning in a larger environment (shown in Fig. 8), whose size is almost 10,000 square meters. Every grid square is a 10 by 10 area. The standard deviation of the process noise is assumed to be 1 meter for the positional degrees of freedom and 7 degrees for the angular degree of freedom. We start with a 5-node FIRM and at every step we randomly sample five more nodes until we reach 500 nodes. Thus, overall, we construct 100 FIRM graphs in this environment, for each of which we measure the construction time (cumulative) and compute the success probability. Plots in Fig. 8 show these quantities as a function of the number of nodes for a sample run on an Intel i5 dual-core 1.7 GHz machine with 4GB memory. 50 particles are used for collision checking, and every node in the underlying PRM is connected to its 3 nearest neighbors.

Basically, FIRM construction is an anytime algorithm in the sense that one can increase the number of nodes and stop enlarging the graph when a termination condition is satisfied such as: *(i)* achieving a desirable success probability or a



(a) (b)

Fig. 7. This figure shows (for a sample run) the success probability of the generated plan versus the number of nodes, as well as the construction time (offline) for the plan.

desirable cost-to-go, *(ii)* no change is observed in the success probability or in the cost-to-go for a significant time, or *(iii)* exceeding the maximum allowed time for offline computation.

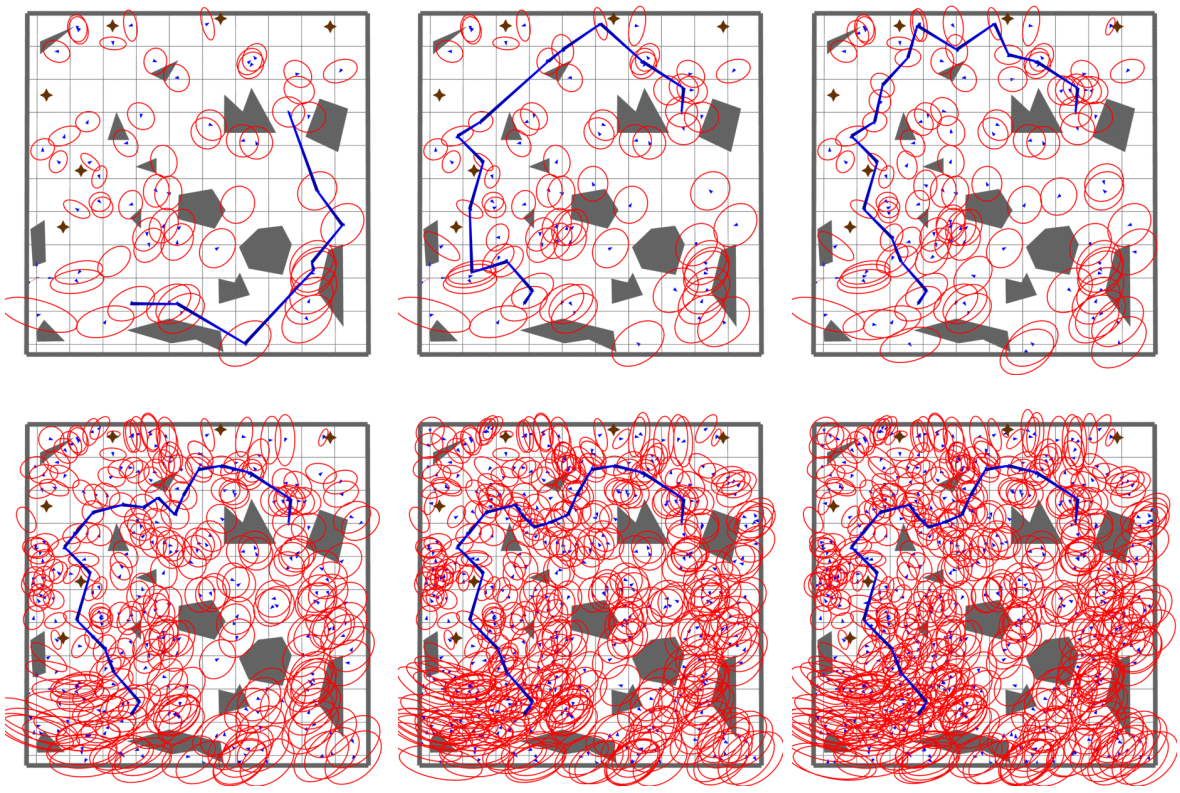


Fig. 8. This figure shows different snapshots of the roadmap for 50, 75, 105, 275, 425, and 500 nodes, respectively. The most likely path under the optimal plan is also shown in different snapshots. Stars show the landmarks. Mean and covariance of the FIRM node centers are shown by small blue triangles and their associated red ellipses, respectively. Also, see Extensions 1 and 2 in Table III, which are regarding the video of planning with FIRM in this environment.

Again, as it is seen from Fig. 8, the highest likelihood path under the optimal policy detours from the shortest path towards the more informative regions in the environment. As a result, it reduces the collision probability and at the same time increases the estimation accuracy and controller efficiency. For the video of executing this plan (with less number of nodes to unclutter the video) see Extension 1 in Table III.

We also conducted a simulation to illustrate the robustness of the method to large deviations. In this simulation, the robot is pushed away from the roadmap several times by some large disturbances, and replanning is performed online based on Algorithm 2. The video of this simulation is also available (see Table III).

C. 8-arm manipulator

On a given graph, the number of paths between two given points grows exponentially with the size of graph. Thus, in the direct propagation of uncertainty on a roadmap, the number of edge costs and transition probabilities that we have to compute is exponential in the number of underlying PRM nodes (see Section IX for a detailed analysis). As a result, when we deal with high dimensional state spaces, where PRM needs to have many edges and nodes, it is not feasible to use the methods that perform direct uncertainty propagation. However, using FIRM, we only need to compute the costs and transition probabilities for as many edges as the underlying PRM has. Thus, we can easily increase the dimension to the level that PRM can handle, and the complexity of the algorithm is increased only by a constant factor (involving computation of costs and transition probabilities of a single edge). In the following experiment, we verify the effectiveness of FIRM in handling high-dimensional systems through a simple example of an 8-arm manipulator. To the best of our knowledge, this is the first belief space planner that can provide a plan over an entire roadmap for an eight dimensional system, while incorporating expensive costs in planning such as computing collision probabilities. This experiment shows that FIRM can be used as a practical tool in many real-world problems.

1) Motion Model: We consider an 8-arm manipulator with eight revolute joints in the plane. The state of the system is described by the angles of joints and their velocities $x = (\theta_1, \cdots, \theta_8, \dot{\theta}_1, \cdots, \dot{\theta}_8)^T$, and the available control is considered to be the angular acceleration (or torque) of joints $u = (\alpha_1, \alpha_2, \cdots, \alpha_8)$. The process noise $w = (w_1, w_2, \cdots, w_8)$ is modelled as a zero-mean Gaussian noise on angular accelerations. Therefore, the continuous motion model for every link is $\ddot{\theta}_i = \alpha_i + w_i$, whose discrete version for entire state can be written as:

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{61}$$

where

$$A = \begin{pmatrix} I_8 & I_8\delta t \\ 0_8 & I_8 \end{pmatrix}, \quad B = \begin{pmatrix} 0_8 \\ I_8\delta t \end{pmatrix}, \quad G = \begin{pmatrix} 0_8 \\ I_8\sqrt{\delta t} \end{pmatrix}. \tag{62}$$

δt is the time interval between two consecutive time steps, and I_n and 0_n are the identity matrix and square zero matrix of dimension n, respectively.

2) Observation Model: We use the light-dark environment setting as the observation model, which is also used in [Platt et al., 2010], [Platt et al., 2011]. In the light-dark environment, the accuracy of sensory readings is encoded by a gray level, in which the regions that have access to more accurate sensory readings are lighter than the regions that do not have access to such informative sensory readings. In this experiment, we assume that we measure the state of the system, but this measurement is more accurate as we get closer to the left wall on which our sensor is mounted. (This model is adopted from [Platt et al., 2010].) Thus, we have $z = h(x) = [z^1, \cdots, z^8]^T$, where

$$z^i = \theta_i + v^i, \qquad v^i \sim \mathcal{N}(0, (\eta|\mathtt{x}^i - l| + \sigma_b)^2) \tag{63}$$

where $\mathtt{x}_i$ is the x coordinate of the i-th joint location, and l is the location of the vertical wall. η defines the dependency of the noise standard deviation on the distance from wall, and σ_b is the bias standard deviation. Figure 9 shows an example of such an environment, in which $l = -1.5$, $\eta = .1$, and $\sigma_b = 10^{-4}$. The full observation model can be written as:

$$z_k = h(x_k) = Hx_k + Mv_k \tag{64}$$

where, $H = [I_8, 0_8]$ and $M = I_8$.

3) Sampling stabilizer parameters: It is easy to show the described system is a controllable and observable system, and thus we adopt the SLQG controller as the stabilizing controller. Therefore, the parameters of the controller are points in the equilibrium space, as explained in Section V. In other words, to generate sample nodes in the state space, we need to sample the configuration space $(\theta_1, \cdots, \theta_8)$ and append zero angular velocities to it. To connect these samples in the state space we design simple trajectories between nodes, along which we accelerate the joints (angles) by a constant acceleration until half way to the next node and thereafter we decelerate the joints until reaching the next node.

4) Construction of the SLQG-FIRM and Planning with it: First, corresponding to sampled nodes in the state space, we compute corresponding FIRM nodes and then design local controllers according to Algorithm 1. In a similar procedure to the one in the previous experiment, we compute the transition costs and probabilities.

To solve the DP, we need to characterize the goal nodes. In Fig. 9, the goal region for the tip location of the manipulator is shown by a purple circle. We mark all PRM samples whose tip locations are within the goal region, as goal nodes. Setting the cost-to-go to zero for all goal nodes, we solve the DP and compute the optimal feedback on the graph according to Algorithm 1. Finally, we execute the plan based on Algorithm 2 and we illustrate the propagation of the covariance of the manipulator tip in Fig. 9 in red. As it can be seen in Fig. 9, there are two passages among the obstacles to reach the goal region. Although the right passage is closer to the initial configuration of the manipulator, the manipulator detours to a longer path through the left passage, because there is more accurate sensory information available in the left passage than the right one. As is seen in this example, the feedback plan minimizes the collision probability and picks the safest path, while it is robust to deviations. In other words, if for any reason the manipulator deviates significantly from the underlying PRM, the feedback plan connects the deviated belief to the best neighboring FIRM node, in real-time, and continues the pre-computed plan from this node.

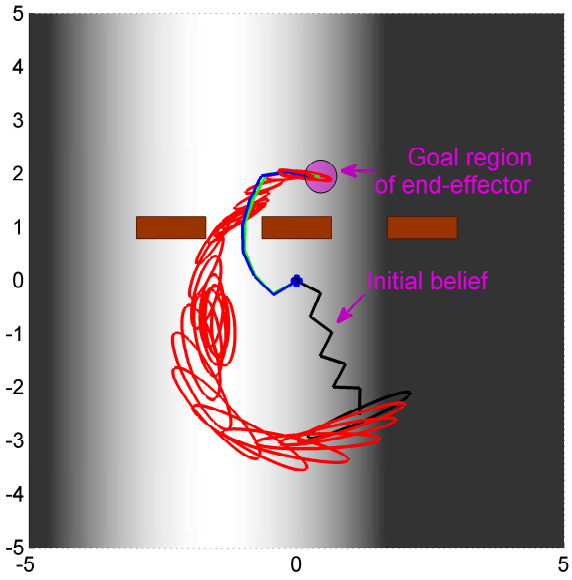


Fig. 9. This figure shows a result of executing the FIRM plan for an 8-arm manipulator in a light-dark (sensing) environment. The manipulator is attached to the origin $(0, 0)$ and the purple region is the goal region for the manipulator tip. To unclutter the figure, we only show the uncertainty of the manipulator tip (end-effector). The initial mean and covariance is shown by black, and the evolution of the tip covariance during the plan execution is shown in red. The final estimation mean and the true configuration of the manipulator are shown in blue and green, respectively. The obstacles are shown in brown. The manipulator follows a longer but safer path to the goal region through the left passage, compared to the shorter but risky (with high collision probability) path through the right passage.

IX. Comparison and Limitations

In this section, we perform a short comparison of SLQG-FIRM against the two most related methods in the literature: BRM [Prentice and Roy, 2009] and LQG-MP on roadmaps [van den Berg et al., 2011]. Both methods are belief space planners that exploit roadmap-based ideas. We compare the methods in terms of the offline construction and online planning complexity, and also in terms of some other properties listed in Table II. In the following, we go over the complexity analysis that leads to the entries in this table. Afterwards, we discuss limitations of the SLQG-FIRM.

Offline construction complexity: In a general graph, the number of paths between two given nodes is exponential in the number of nodes N. For example, if each node in a graph is connected to k-nearest neighbor nodes on the graph, for a search depth of d edges on the graph, the corresponding search tree contains k^d paths. Notice that each of these paths has d edges on it. Thus, if we directly (without using belief stabilizers) propagate the uncertainty on a roadmap for a depth of d, we have to evaluate the cost on dk^d edges. So, the asymptotic complexity of the overall problem is of the order $\mathcal{O}(Nk^N)$. Therefore, if we directly propagate the uncertainty on a roadmap, we have to evaluate the cost on $O(Nk^N)$ edges. Now, if computing the cost and transition probabilities associated with each edge under uncertainty is a constant multiplier $\mathcal{O}(c)$ of computing its cost in deterministic case, the overall complexity of the methods based on direct belief propagation is $\mathcal{O}(cNk^N)$. On the other hand, in any variant of FIRM, due to the edge independence, only the cost of $\mathcal{O}(Nk)$ edges needs to be constructed as in PRM, and thus the overall complexity of offline construction of FIRM is $\mathcal{O}(cNk)$.

Online planning (replanning) complexity: If the system deviates from the valid region of the plan, in direct propagation methods, edge costs need to be recomputed for all edges. So, in BRM and LQG-MP on roadmaps, the replanning complexity will be of the order $\mathcal{O}(Nk^N)$. If the cost of each edge is defined in such a way that it only depends on the belief at the start and end of edge (i.e., does not depend on the belief along the edge), BRM can reduce the computation complexity to $\mathcal{O}(c\frac{N}{l}k^N)$ through covariance factorization techniques, where l is assumed to be the length (number of steps) of each edge. In FIRM, in the case of replanning (submitting a query with new starting point), it is only required to connect the deviated belief to k neighboring FIRM nodes. Thus, only for the k new edges we need to compute the cost. It is worth noting that if the underlying PRM is dense enough such that the valid region of the local controllers covers the space, edge cost computation in the replanning phase reduces to zero, because if the system deviates out of a valid region of a local planner, it will fall into the valid region of some other planner.

To reduce the complexity of the search algorithm in BRM and LQG-MP on roadmaps, it is assumed that the costs on different edges of the roadmap are independent. This heuristic can reduce the complexity of the algorithm, but still it may be significantly high compared to the PRM or FIRM. Moreover, this heuristic (edge independent assumption) is not true without having belief stabilizers, and thus search algorithms relying on such a heuristic may result in solutions far different from the true solution of the search algorithm. Assuming that no such heuristic is used in the search algorithm, Table II summarizes the complexity of these algorithms.

TABLE II
BELIEF SPACE ROADMAP-BASED METHOD COMPARISON (WITHOUT USING HEURISTIC IN SEARCH ALGORITHMS)

Algorithm	offline construction complexity (no heuristic)	replanning (online planning) complexity	future observations	System requirement	valid region of plan	Collision probabilities
Generic PRM	$\mathcal{O}(Nk)$	$\mathcal{O}(k)$	———	assumes a controller exists to drive the system from node-to-node	on the graph only	———
BRM	$\mathcal{O}(cNk^N)$	$\mathcal{O}(c\frac{N}{l}k^N)$ or $\mathcal{O}(cNk^N)$	maximum likelihood observation	well-linearizable systems	vicinity of nominal path	not considered
LQG-MP on Roadmaps	$\mathcal{O}(cNk^N)$	$\mathcal{O}(cNk^N)$	All observations	well-linearizable systems	vicinity of nominal path	simplified measures are used
Generic FIRM	$\mathcal{O}(cNk)$	$\mathcal{O}(ck)$	———	assumes a controller exists to drive the system from node-to-node	union of convergence regions of local controllers	———
SLQG-FIRM	$\mathcal{O}(cNk)$	$\mathcal{O}(ck)$ or $\mathcal{O}(1)$	All observations	well-linearizable, and linear controllable and observable systems	vicinity of whole PRM (entire space for a dense PRM)	computed

The huge reduction in the computational complexity of the planning algorithm (in particular, the online phase), opens many possibilities in utilizing POMDP solvers in real-world applications. Moreover, due to its sampling-based nature, it ameliorates the curse of dimensionality just as PRM does in the deterministic case. In other words, if the dimension of the system increases, we need a greater number of nodes N in the underlying PRM to capture the free space connectivity, in which case we cannot use direct methods due to their complexity. However, FIRM can tolerate the increase in the dimension since its complexity is only a constant multiplier of the PRM complexity.

A. Limitations of the SLQG-FIRM and Future directions

In this section, we discuss limitations of the proposed method. It is important to distinguish which limitation is associated with the generic FIRM framework, and which limitation is associated with the particular presented instantiation of the FIRM, i.e., the SLQG-FIRM. In some cases, we also propose ways to remedy these limitations as future research directions.

Stabilization time: The FIRM framework introduces the usage of belief stabilizers. However, the time needed for the belief stabilization procedure is added to the overall execution time. If the number of time steps along the nominal path is l, and the number of time steps needed for stabilization is τ, usually, the extra time τ is negligible compared to l. However, τ can increase as the sensing uncertainty increases. In such a situation, one can consider two cases: if obstacles are close to the robot, it is indeed unsafe to move with a poor estimate, and it is indeed better to lose some time to gain more information, and then start moving. On the other hand, if there is no obstacle close to the robot, the one can increase the size of the corresponding FIRM node, and thus, decrease the extra stopping time. Moreover, efficient sampling-based methods, which are aware of available information at different locations of the environment, and thus aware of the mean stabilization time, can be used to efficiently sample the nodes in the locations with lower mean stabilization time. These issues open up new directions for future research. However, if an application is very sensitive to the extra time, FIRM may not be a good choice for it, and methods such as BRM or LQG-MP can result in better guarantees on execution time.

Controllability and observability: As mentioned in Section V, SLQG-FIRM works for systems that satisfy Property 1, which basically requires the linearized system about the PRM nodes to be controllable and observable. Although this includes a large class of systems, it excludes some important systems, such as non-holonomic systems that are not linearly controllable about any point. It is worth noting that this is not a limitation of the generic FIRM framework, but is a limitation of the SLQG-FIRM. More recent instantiations of FIRM, such as PLQG-based FIRM [Agha-mohammadi et al., 2012c], or DFL-based FIRM [Agha-mohammadi et al., 2012a] aim to relax the controllability requirements in Property 1 and thus, can include non-holonomic systems as well. However, relaxing the observability assumption is still an open problem.

Gaussian beliefs: The reachability argument in the SLQG-FIRM is restricted to Gaussian beliefs. In other words, we cannot guarantee reachability to some pre-defined non-Gaussian beliefs with stationary LQG controllers. This issue is a subject of future research.

Increasing the uncertainty: Although it may rarely happen in practice, it is possible to have a situation that leads to an uncertainty growth during the belief stabilization process. However, this issue can be addressed easily. Notice that FIRM nodes are known a priori. Thus, at the beginning of each stabilization procedure, we can compare the current belief with the stationary belief of the stabilizer. If the current belief has more information than the stationary belief (e.g., if all eigenvalues of the estimation covariance are strictly less than the corresponding eigenvalues of the stationary estimation covariance), we replan from the current belief based on Algorithm 2. Therefore, uncertainty will not be increased during the stabilization procedure.

Locally linearizable systems: If a linear representation of the system of interest cannot be obtained – for example, if the system state lives in a discrete set of states – the class of methods that use the linearized system as a local approximation of

the true system will not work. In this case, another class of methods can be adopted, which can handle these systems much better, such as [Kurniawati et al., 2008], [Kurniawati et al., 2011], and [Smith and Simmons, 2005]. It is also a future research direction to come up with belief stabilizers that work in discrete state space settings to design a discrete-state variant of FIRM.

Velocity reduction in dynamical systems: To apply SLQG-FIRM to dynamical systems, the underlying PRM samples need to be selected from the equilibrium space, i.e., they need to have zero velocity. As a result a reduction in the system's velocity is expected while trying to reach the FIRM nodes. However, in many applications it may be a useful tradeoff to reduce the speed at nodes to gain the robustness, reliability, and scalability offered by FIRM. Nevertheless, this reduction in speed may not be desirable for some applications where the system cannot (or should not) decrease its velocity. For such systems, [Agha-mohammadi et al., 2013a] propose a FIRM variant based on periodic controllers that does not require a reduction in the system's velocity. However, designing more efficient variants of FIRM that can sample points with non-zero velocities without introducing periodicity in the system's motion is an interesting future research direction.

X. Conclusion

In this paper, we have proposed the Feedback-based Information RoadMap (FIRM) framework for solving the motion planning problem under motion and sensing uncertainties. This problem is originally a POMDP, whose solution is computationally intractable. Exploiting feedback controllers, we reduced it to a tractable FIRM MDP that can be solved by standard DP techniques. FIRM utilizes feedback controllers to create reachable node regions in belief space. An important consequence is that FIRM preserves the optimal substructure property on the roadmap and thus overcomes the curse of history in the original POMDP problem. Finally, by computing the collision probabilities, obstacles are also appropriately taken into account in planning on FIRM. We showed an instantiation of the abstract FIRM framework using SLQG controllers and illustrated the construction and planning results on it. By extending the probabilistic completeness concept to planners under uncertainty, we also showed that FIRM is probabilistically complete under uncertainty. We believe that FIRM provides an important step toward solving POMDPs and utilizing them as a practical tool for robot motion planning under uncertainty.

Acknowledgements

The authors are grateful to the anonymous reviewers for their helpful suggestions as well as Aditya Mahadevan and Daniel Tomkins for many fruitful discussions and their help with experiments. This work is supported in part by NSF award RI-1217991. Additionally, the work of Agha-mohammadi and Chakravorty is supported in part by AFOSR Grant FA9550-08-1-0038 and the work of Agha-mohammadi and Amato is supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-0830753, IIS-0917266, IIS-0916053, EFRI-1240483, by NSF/DNDO award 2008-DN-077-ARI018-02, by NIH NCI R25 CA090301-11, by DOE awards DE-FC52-08NA28616, DE-AC02-06CH11357, B575363, B575366, by THECB NHARP award 000512-0097-2009, by Samsung, Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

References

[Agha-mohammadi et al., 2011] Agha-mohammadi, A., Chakravorty, S., and Amato, N. (2011). FIRM: Feedback controller-based Information-state RoadMap -a framework for motion planning under uncertainty-. In *International Conference on Intelligent Robots and Systems (IROS)*.

[Agha-mohammadi et al., 2012a] Agha-mohammadi, A., Chakravorty, S., and Amato, N. (2012a). Nonholonomic motion planning in belief space via dynamic feedback linearization-based FIRM. In *International Conference on Intelligent Robots and Systems (IROS)*.

[Agha-mohammadi et al., 2012b] Agha-mohammadi, A., Chakravorty, S., and Amato, N. (2012b). On the probabilistic completeness of the sampling-based feedback motion planners in belief space. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[Agha-mohammadi et al., 2012c] Agha-mohammadi, A., Chakravorty, S., and Amato, N. (2012c). Periodic-feedback motion planning in belief space for non-holonomic and/or non-stoppable robots. *Technical Report: TR12-003, Parasol Lab., CSE Dept., Texas A&M University*.

[Agha-mohammadi et al., 2013a] Agha-mohammadi, A., Chakravorty, S., and Amato, N. (2013a). Online replanning in belief space for dynamical systems: Towards handling discrete changes of goal location. In *IEEE International Conference On Robotics and Automation (ICRA): Workshop on Combining Task and Motion Planning*.

[Agha-mohammadi et al., 2013b] Agha-mohammadi, A., Chakravorty, S., and Amato, N. (2013b). Sampling-based stochastic control with constraints: A unified approach in state and information spaces. In *the American Control Conference (ACC)*.

[Alterovitz et al., 2007] Alterovitz, R., Siméon, T., and Goldberg, K. (2007). The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. In *Proceedings of Robotics: Science and Systems (RSS)*.

[Amato et al., 1998] Amato, N., Bayazit, B., Dale, L., Jones, C., and Vallejo, D. (1998). OBPRM: An Obstacle-Basaed PRM for 3D workspaces. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 155–168.

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

[Bai et al., 2010] Bai, H., Hsu, D., Lee, W. S., and Ngo, V. A. (2010). Monte carlo value iteration for continuous-state pomdps. In *WAFR*, volume 68 of *Springer Tracts in Advanced Robotics*, pages 175–191. Springer.

[Bertsekas, 1976] Bertsekas, D. (1976). *Dynamic Programming and Stochastic Control*. Academic Press.

[Bertsekas, 2007] Bertsekas, D. (2007). *Dynamic Programming and Optimal Control: 3rd Ed.* Athena Scientific.

[Bohlin, 2002] Bohlin, R. (2002). *Robot Path Planning*. PhD thesis, Chalmers University of Technology, Goteborg, Sweden.

[Bry and Roy, 2011] Bry, A. and Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In *ICRA*, pages 723–730.

[Censi et al., 2008] Censi, A., Calisi, D., Luca, A. D., and Oriolo, G. (2008). A Bayesian framework for optimal motion planning with uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA.

[Chakravorty and Erwin, 2011] Chakravorty, S. and Erwin, R. S. (2011). Information space receding horizon control. In *IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning (ADPRL)*.

[Chakravorty and Kumar, 2009] Chakravorty, S. and Kumar, S. (2009). Generalized sampling based motion planners with application to nonholonomic systems. In *Proceedings of the IEEE Internation Conference on Systems, Man, and Cybernetics*, San Antonio, TX.

[Chakravorty and Kumar, 2011] Chakravorty, S. and Kumar, S. (2011). Generalized sampling-based motion planners. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 41(3):855–866.

[Choset et al., 2005] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of robot motion: theory, algorithms, and implementations*. MIT Press.

[Cormen et al., 2001] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms, Second Edition*. MIT Press.

[Crassidis and Junkins, 2004] Crassidis, J. and Junkins, J. (2004). *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC.

[Doucet et al., 2001] Doucet, A., de Freitas, J., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer.

[Guibas et al., 2008] Guibas, L., Hsu, D., Kurniawati, H., and Rehman, E. (2008). Bounded uncertainty roadmaps for path planning. In *International Workshop on Algorithmic Foundations of Robotics*.

[He et al., 2010] He, R., Brunskill, E., and Roy, N. (2010). PUMA: Planning under uncertainty with macro-actions. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI)*, Atlanta, GA.

[He et al., 2011] He, R., Brunskill, E., and Roy, N. (2011). Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, 40:523–570.

[Hsu, 2000] Hsu, D. (2000). *Randomized single-query motion planning in expansive spaces*. PhD thesis, Department of Computer Science, Stanford University, Stanford, CA.

[Huynh and Roy, 2009] Huynh, V. and Roy, N. (2009). icLQG: combining local and global optimization for control in information space. In *IEEE International Conference on Robotics and Automation (ICRA)*.

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

[Kalmár-Nagy et al., 2004] Kalmár-Nagy, T., D'Andrea, R., and Ganguly, P. (2004). Near-optimal dynamics trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, 46(1):47–64.

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

[Kavraki et al., 1998] Kavraki, L., Kolountzakis, M., and Latombe, J. (1998). Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14:166–171.

[Kavraki et al., 1995] Kavraki, L., Latombe, J., Motwani, R., and Raghavan, P. (1995). Randomized query processing in robot motion planning. In *Proc. ACM Symp. Theory of Computing*, pages 353–362.

[Kavraki et al., 1996] Kavraki, L., Švestka, P., Latombe, J., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.

[Keener, 2000] Keener, J. P. (2000). *Principles of Applied Mathematics: Transformation and Approximation, 2nd Edition*. Westview Press.

[Kumar and Varaiya, 1986] Kumar, P. R. and Varaiya, P. P. (1986). *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice-Hall, Englewood Cliffs, NJ.

[Kurniawati et al., 2012] Kurniawati, H., Bandyopadhyay, T., and Patrikalakis, N. (2012). Global motion planning under uncertain motion, sensing, and environment map. *Autonomous Robots*, pages 1–18.

[Kurniawati et al., 2010] Kurniawati, H., Du, Y., Hsu, D., and Lee, W. S. (2010). Motion planning under uncertainty for robotic tasks with long time horizons. *International Journal of Robotics Research*, 30:308–323.

[Kurniawati et al., 2011] Kurniawati, H., Du, Y., Hsu, D., and Lee, W. S. (2011). Motion planning under uncertainty for robotic tasks with long time horizons. *The International Journal of Robotics Research*, 30(3):308–323.

[Kurniawati et al., 2008] Kurniawati, H., Hsu, D., and Lee, W. (2008). SARSOP: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proceedings of Robotics: Science and Systems*.

[Ladd and Kavraki, 2004] Ladd, A. and Kavraki, L. (2004). Measure theoretic analysis of probabilistic path planning. *IEEE Transactions on Robotics and Automation*, 20(2):229–242.

[Lavalle and Kuffner, 2001] Lavalle, S. and Kuffner, J. (2001). Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(378-400).

[Lozano-Perez, 1983] Lozano-Perez, T. (1983). Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120.

[Madani et al., 1999] Madani, O., Hanks, S., and Condon, A. (1999). On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *Proceedings of the Sixteen Conference on Artificial Intelligence (AAAI)*, pages 541–548.

[Melchior and Simmons, 2007] Melchior, N. and Simmons, R. (2007). Particle RRT for path planning with uncertainty. In *International Conference on Intelligent Robots and Systems (IROS)*.

[Meyn and Tweedie, 2009] Meyn, S. and Tweedie, R. L. (2009). *Markov Chains and Stochastic Stability: 2nd Edition*. Cambridge University Press.

[Missiuro and Roy, 2006] Missiuro, P. and Roy, N. (2006). Adapting probabilistic roadmaps to handle uncertain maps. In *ICRA*.

[Nakhaei and Lamiraux, 2008] Nakhaei, A. and Lamiraux, F. (2008). A framework for planning motions in stochastic maps. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[Norris, 1997] Norris, J. R. (1997). *Markov Chains*. Cambridge Univeristy Press.

[Ong et al., 2010] Ong, S. C. W., Png, S. W., Hsu, D., and Lee, W. S. (2010). Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, 29(8):1053–1068.

[Papadimitriou and Tsitsiklis, 1987] Papadimitriou, C. and Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450.

[Patil et al., 2012] Patil, S., van den Berg, J., and Alterovitz, R. (2012). Estimating probability of collision for safe planning under gaussian motion and sensing uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[Pineau et al., 2003] Pineau, J., Gordon, G., and Thrun, S. (2003). Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence*, pages 1025–1032.

[Pineau et al., 2006] Pineau, J., Gordon, G., and Thrun, S. (2006). Anytime point based approximations for large pomdps. *Journal of Artificial Intelligence Research*, 27:335–380.

[Platt et al., 2011] Platt, R., Kaelbling, L., Lozano-Perez, T., , and Tedrake, R. (2011). Efficient planning in non-gaussian belief spaces and its application to robot grasping. In *Proc. of International Symposium of Robotics Research, (ISRR)*.

[Platt et al., 2010] Platt, R., Tedrake, R., Kaelbling, L., and Lozano-Perez, T. (2010). Belief space planning assuming maximum likelihood observatoins. In *Proceedings of Robotics: Science and Systems (RSS)*.

[Porta et al., 2006] Porta, J. M., Vlassis, N., Spaan, M. T. J., and Poupart, P. (2006). Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research*, 7:2329–2367.

[Prentice and Roy, 2009] Prentice, S. and Roy, N. (2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *International Journal of Robotics Research*, 28(11-12).

[Simon, 2006] Simon, D. (2006). *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley and Sons, Inc.

[Smallwood and Sondik, 1973] Smallwood, R. D. and Sondik, E. J. (1973). The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088.

[Smith and Simmons, 2005] Smith, T. and Simmons, R. (2005). Point-based pomdp algorithms: Improved analysis and implementation. In *Proceedings of Uncertainty in Artificial Intelligence*.
[Sniedovich, 2006] Sniedovich, M. (2006). Dijkstra's algorithm revisited: the dynamic programming connexion. *Control and Cybernetics*, 35(3):599–620.
[Spaan and Vlassis, 2005] Spaan, M. and Vlassis, N. (2005). Perseus: Randomized point-based vallue iteration for pomdps. *Journal of Artificial Intelligence Research*, 24:195–220.
[Sutton et al., 1999] Sutton, R., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211.
[Švestka and Overmars, 1997] Švestka, P. and Overmars, M. (1997). Motion planning for car-like robots using a probabilistic learning approach. *International Journal of Robotics Research*, 16(2):119–143.
[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
[Toit and Burdick, 2010] Toit, N. D. and Burdick, J. W. (2010). Robotic motion planning in dynamic, cluttered, uncertain environments. In *ICRA*.
[van den Berg et al., 2010] van den Berg, J., Abbeel, P., and Goldberg, K. (2010). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. In *Proceedings of Robotics: Science and Systems (RSS)*.
[van den Berg et al., 2011] van den Berg, J., Abbeel, P., and Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *IJRR*, 30(7):895–913.
[van den Berg and Overmars, 2007] van den Berg, J. and Overmars, M. (2007). Kinodynamic motion planning on roadmaps in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, pages 4253–4258. IEEE.
[van den Berg et al., 2011] van den Berg, J., Patil, S., and Alterovitz, R. (2011). Motion planning under uncertainty using differential dynamic programming in belief space. In *Proc. of International Symposium of Robotics Research, (ISRR)*.
[van den Berg et al., 2012] van den Berg, J., Patil, S., and Alterovitz, R. (2012). Efficient approximate value iteration for continuous gaussian POMDPs. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.
[Vitus and Tomlin, 2011] Vitus, M. P. and Tomlin, C. J. (2011). Closed-loop belief space planning for linear, Gaussian systems. In *ICRA*, pages 2152–2159.

APPENDIX A
INDEX TO MULTIMEDIA EXTENSION

The multimedia extensions to this article can be found on-line by following the hyperlinks from http://www.ijrr.org.

TABLE III

Extension	Media type	Description
1	video	Executing the FIRM plan in the environment shown in Fig.8
2	video	Real-time replanning with FIRM, which shows the robustness of the method to large disturbances.

APPENDIX B
TIME-VARYING LQG CONTROLLER

The time-varying Linear Quadratic Gaussian (LQG) controller is often used to track a pre-planned trajectory (also called nominal, desired, or open-loop trajectory) in the presence of process and observation noise. In principal it is designed (and optimal) for linear systems with Gaussian noise, but it can also be utilized for stabilizing nonlinear systems locally around the planned trajectory. An LQG controller is composed of a Kalman Filter (KF) as an estimator and a Linear Quadratic Regulator (LQR) as a controller. At every time step k, the KF provides the *a posteriori* distribution (belief) b_k over the system state, and LQR generates control u_k based on b_k.

In this section, we first discuss the system linearization and planned nominal trajectory, and then discuss the KF, LQR and LQG corresponding with this nominal trajectory. Consider the nonlinear partially-observable state-space equations of the system as follows:

$$x_{k+1} = f(x_k, u_k, w_k), \quad w_k \sim \mathcal{N}(0, Q_k) \tag{65a}$$
$$z_k = h(x_k, v_k), \quad v_k \sim \mathcal{N}(0, R_k) \tag{65b}$$

A planned nominal trajectory for the robot is a sequence of planned states $(x_k^p)_{k\geq 0}$ and planned controls $(u_k^p)_{k\geq 0}$, such that it is consistent with the noiseless dynamics model, i.e., we have:

$$x_{k+1}^p = f(x_k^p, u_k^p, 0) \tag{66}$$

The planned trajectory can be a finite sequence of some length N. Then, x_0 and x_N are called the start and final states of the robot. The role of a closed-loop stochastic controller, during the trajectory tracking, is to compensate for the robot's deviations from the planned trajectory and to keep the robot close to the planned trajectory in the sense of minimizing the following quadratic cost:

$$J = \mathbb{E}\left[\sum_{k\geq 0}(x_k - x_k^p)^T W_x (x_k - x_k^p) + (u_k - u_k^p)^T W_u (u_k - u_k^p)\right] \tag{67}$$

where W_x and W_u are positive definite weight matrices for the state and control costs, respectively.

Since the state space is not fully observable and it is only partially observable, we do not have access to the perfect value of the state x_k, and thus, we provide the estimate x_k^+ of the state x_k based on the available partial observations $z_{0:k}$ from the beginning up to the current time step. Then, based on the separation principle [Kumar and Varaiya, 1986], [Bertsekas, 2007], it can be shown that in a linear system with Gaussian noise, the above minimization in terms of the error $x_k - x_k^p$ is equivalent to performing two separate minimizations based on the estimation error $x_k - \widehat{x}_k^+$ and the controller error $\widehat{x}_k^+ - x_k^p$, whose summation is the same as the original main error $x_k - x_k^p = (x_k - \widehat{x}_k^+) + (\widehat{x}_k^+ - x_k^p)$, where $\widehat{x}_k^+ = \mathbb{E}[x_k^+] = \mathbb{E}[x_k|z_{0:k}]$. As a major consequence, the design of the stochastic controller with a partially-observable state space (LQG), reduces to designing a controller with fully-observable state (LQR) and designing an estimator (KF), separately. In the following, we first discuss the linearization of a nonlinear model. Then we discuss how a KF and an LQR can be designed for this linearized system. Finally we combine them to construct a time-varying LQG controller.

Model linearization: Given a nominal trajectory $(x_k^p, u_k^p)_{k\geq 0}$, we linearize the dynamics and observation model in Eq. (65), as follows:

$$x_{k+1} = f(x_k^p, u_k^p, 0) + A_k(x_k - x_k^p) + B_k(u_k - u_k^p) + G_k w_k, \quad w_k \sim \mathcal{N}(0, Q_k) \tag{68a}$$

$$z_k = h(x_k^p, 0) + H_k(x_k - x_k^p) + M_k v_k, \quad v_k \sim \mathcal{N}(0, R_k) \tag{68b}$$

where

$$A_k = \frac{\partial f}{\partial x}(x_k^p, u_k^p, 0), \quad B_k = \frac{\partial f}{\partial u}(x_k^p, u_k^p, 0), \quad G_k = \frac{\partial f}{\partial w}(x_k^p, u_k^p, 0),$$
$$H_k = \frac{\partial h}{\partial x}(x_k^p, 0), \quad M_k = \frac{\partial h}{\partial v}(x_k^p, 0) \tag{69}$$

Now, let us define the following errors:

- LQG error (main error): $e_k = x_k - x_k^p$
- KF error (estimation error): $\widetilde{e}_k = x_k - \widehat{x}_k^+$
- LQR error (estimation of LQG error): $\widehat{e}_k^+ = \widehat{x}_k^+ - x_k^p$

Note that these errors are linearly dependent: $e_k = \widehat{e}_k^+ + \widetilde{e}_k$. Also, defining $\delta u_k = u_k - u_k^p$ and $\delta z_k = z_k - z_k^p := z_k - h(x_k^p, 0)$, we can rewrite the above linearized models as follows:

$$e_{k+1} = A_k e_k + B_k \delta u_k + G_k w_k \tag{70a}$$

$$\delta z_k = H_k e_k + M_k v_k \tag{70b}$$

Kalman Filter: In Kalman filtering, we aim to provide an estimate of the system's state based on the available partial information we have obtained until time k, i.e., $z_{0:k}$. The state estimate is a random vector denoted by x_k^+, whose distribution is the conditional distribution of the state on the obtained observations so far, which is called belief and is denoted by b_k:

$$b_k = p(x_k^+) = p(x_k|z_{0:k}) \tag{71}$$

$$\widehat{x}_k^+ = \mathbb{E}[x_k|z_{0:k}] \tag{72}$$

$$P_k = \mathbb{C}[x_k|z_{0:k}] \tag{73}$$

where $\mathbb{E}[\cdot|\cdot]$ and $\mathbb{C}[\cdot|\cdot]$ are the conditional expectation and conditional covariance operators, respectively. In the Gaussian case, we have $b_k = \mathcal{N}(\widehat{x}_k^+, P_k)$, i.e., the belief can be characterized only by its mean and covariance, i.e., $b_k \equiv (\widehat{x}_k^+, P_k)$.

Kalman filtering consists of two steps at every time stage: a prediction step and an update step. In the prediction step, the mean and covariance of prior x_k^- are computed. For the system in Eq. (70) prediction step is:

$$\widehat{e}_{k+1}^- = A_k \widehat{e}_k^+ + B_k \delta u_k \tag{74}$$

$$P_{k+1}^- = A_k P_k^+ A_k^T + G_k Q_k G_k^T \tag{75}$$

In the update step, the mean and covariance of posterior x_k^+ are computed. For the system in Eq. (70), the update step is:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \tag{76}$$

$$\widehat{e}_{k+1}^+ = \widehat{e}_{k+1}^- + K_{k+1}(\delta z_{k+1} - H_{k+1}\widehat{e}_{k+1}^-) \tag{77}$$

$$P_{k+1}^+ = (I - K_{k+1}H_{k+1})P_{k+1}^- \tag{78}$$

Note that

$$\widehat{x}_k^+ = \mathbb{E}[x_k|z_{0:k}] = x_k^p + \widehat{e}_k^+ = x_k^p + \mathbb{E}[e_k|z_{0:k}] \tag{79}$$

$$P_k = \mathbb{C}[x_k|z_{0:k}] = P_k^+ = \mathbb{C}[e_k|z_{0:k}] \tag{80}$$

LQR controller: Once we obtain the belief from the filter, a controller can generate an optimal control signal accordingly. In other words, we have a time-varying mapping μ_k from the belief space into the control space that generates an optimal

control based on the given belief $u_k = \mu_k(b_k)$ at every time step k. The LQR controller is of this kind and it is optimal in the sense of minimizing the following cost:

$$J_{LQR} = \mathbb{E}\left[\sum_{k\geq 0}(\widehat{x}_k^+ - x_k^p)^T W_x(\widehat{x}_k^+ - x_k^p) + (u_k - u_k^p)^T W_u(u_k - u_k^p)\right]$$
$$= \mathbb{E}\left[\sum_{k\geq 0}(\widehat{e}_k^+)^T W_x(\widehat{e}_k^+) + (\delta u_k)^T W_u(\delta u_k)\right] \tag{81}$$

The linear control law that minimizes this cost function for a linear system is of the form:

$$\delta u_k = -L_k \widehat{e}_k^+ \tag{82}$$

where the time-varying feedback gains L_k can be computed recursively as follows:

$$L_k = (B_k^T S_{k+1} B_k + W_u)^{-1} B_k^T S_{k+1} A_k \tag{83}$$
$$S_k = W_x + A_k^T S_{k+1} A_k - A_k^T S_{k+1} B_k L_k \tag{84}$$

If the nominal path is of length N, then $S_N = W_x$ is the initial condition of above recursion, which is solved backwards in time. Note that the full control is $u_k = u_k^p + \delta u_k$.

LQG controller: Plugging the obtained control law in LQR into the Kalman filtering equations, we obtain the following error dynamics, for the defined errors:

$$\begin{pmatrix} e_{k+1} \\ \widetilde{e}_{k+1} \end{pmatrix} = \begin{pmatrix} A_k - B_k L_k & B_k L_k \\ 0 & A_k - K_{k+1} H_{k+1} A_k \end{pmatrix} \begin{pmatrix} e_k \\ \widetilde{e}_k \end{pmatrix} + \begin{pmatrix} G_k & 0 \\ G_k - K_{k+1} H_{k+1} G_k & -K_{k+1} M_{k+1} \end{pmatrix} \begin{pmatrix} w_k \\ v_{k+1} \end{pmatrix} \tag{85}$$

or equivalently,

$$\begin{pmatrix} e_{k+1} \\ \widehat{e}_{k+1}^+ \end{pmatrix} = \begin{pmatrix} A_k & -B_k L_k \\ K_{k+1} H_{k+1} A_k & A_k - B_k L_k - K_{k+1} H_{k+1} A_k \end{pmatrix} \begin{pmatrix} e_k \\ \widehat{e}_k^+ \end{pmatrix} + \begin{pmatrix} G_k & 0 \\ K_{k+1} H_{k+1} G_k & K_{k+1} M_{k+1} \end{pmatrix} \begin{pmatrix} w_k \\ v_{k+1} \end{pmatrix} \tag{86}$$

Defining $\zeta_k := (e_k, \widehat{e}_k^+)^T$ and $q_k := (w_k, v_{k+1})^T$, we can rewrite Eq. (87) in a more compact form as

$$\zeta_{k+1} = \overline{F}_k \zeta_k + \overline{G}_k q_k, \quad q_k \sim \mathcal{N}(0, \overline{Q}_k), \quad \overline{Q}_k = \begin{pmatrix} Q_k & 0 \\ 0 & R_{k+1} \end{pmatrix} \tag{87}$$

with appropriate definitions for $\overline{F}_k$ and $\overline{G}_k$.

The above equation along with the equation on estimation covariance propagation,

$$P_{k+1} = (I - K_{k+1} H_{k+1})(A_k P_k A_k^T + G_k Q_k G_k^T), \tag{88}$$

characterize the evolution of state x_k and belief $b_k \equiv (\widehat{x}_k^+, P_k)$ under the time-varying LQG controller.

APPENDIX C
STATIONARY LQG CONTROLLER

The stationary Linear Quadratic Gaussian (SLQG) controller is often used to regulate (or stabilize) the system state to a pre-planned point (also called the set-point, nominal, or desired point) in the presence of process and observation noise. In principal it is designed (and optimal) for linear systems with Gaussian noise, but it can also be utilized for stabilizing nonlinear systems locally around the planned point. The SLQG controller is composed of a Stationary Kalman Filter (SKF) as an estimator and a Stationary Linear Quadratic Regulator (SLQR) as a controller. At every time step k, the SKF provides the *a posteriori* distribution (belief) b_k over the system state, and SLQR generates control u_k based on b_k.

In this section, we first discuss the system linearization around the planned point, and then discuss the SKF, SLQR and SLQG corresponding to this nominal point. Consider the nonlinear partially-observable state-space equations of the system as follows:

$$x_{k+1} = f(x_k, u_k, w_k), \quad w_k \sim \mathcal{N}(0, Q_k) \tag{89a}$$
$$z_k = h(x_k, v_k), \quad v_k \sim \mathcal{N}(0, R_k) \tag{89b}$$

and consider a planned state point x^p, to whose vicinity the controller has to drive the system. If the system state reaches the point x^p, it is assumed that the system remains there with zero control, $u^p = 0$, i.e.,

$$x^p = f(x^p, 0, 0) \tag{90}$$

The role of a closed-loop stochastic controller, during the state regulation, is to compensate for robot deviations from the desired point due to noise effects and to drive the robot close to the desired point in the sense of minimizing the following quadratic cost:

$$J = \mathbb{E}\left[\sum_{k\geq 0}(x_k - x^p)^T W_x (x_k - x^p) + (u_k)^T W_u (u_k)\right] \tag{91}$$

where W_x and W_u are positive definite weight matrices for the state and control cost, respectively.

Again, similar to the time-varying case, since we only have imperfect information of the state x_k, we have to make the estimate x_k^+ of the state based on the available partial observations $z_{0:k}$. Accordingly, the controller generates the control signal based on the estimated value of the state, i.e., belief. Based on the separation principle [Bertsekas, 2007], in a linear system with Gaussian noise, minimization of the cost in Eq.(91) is equivalent to performing two separate minimizations that lead to the separate design of the SKF and SLQR. In the following, we first discuss the linearization of a nonlinear model and then we discuss how the SKF and the SLQR can be designed for this linearized system and finally combine them to construct an SLQG controller.

Model linearization: Given a desired point x^p, we linearize the dynamics and observation model in Eq. (89), as follows:

$$x_{k+1} = f(x^p, 0, 0) + A_s(x_k - x^p) + B_s(u_k - 0) + G_s w_k, \quad w_k \sim \mathcal{N}(0, Q_s) \tag{92a}$$

$$z_k = h(x^p, 0) + H_s(x_k - x^p) + M_s v_k, \quad v_k \sim \mathcal{N}(0, R_s) \tag{92b}$$

where

$$A_s = \frac{\partial f}{\partial x}(x^p, 0, 0), \quad B_s = \frac{\partial f}{\partial u}(x^p, 0, 0), \quad G_s = \frac{\partial f}{\partial w}(x^p, 0, 0),$$
$$H_s = \frac{\partial h}{\partial x}(x^p, 0), \quad M_s = \frac{\partial h}{\partial v}(x^p, 0) \tag{93}$$

Now, let us define the following errors:

- SLQG error (main error): $e_k = x_k - x^p$.
- SKF error (estimation error): $\widetilde{e}_k = x_k - \widehat{x}_k^+$, where $\widehat{x}_k^+ = \mathbb{E}[x_k^+] = \mathbb{E}[x_k | z_{0:k}]$.
- SLQR error (estimation of SLQG error): $\widehat{e}_k^+ = \widehat{x}_k^+ - x^p$.

Note that these errors are linearly dependent: $e_k = \widehat{e}_k^+ + \widetilde{e}_k$. Also, defining $\delta u_k = u_k$ and $\delta z_k = z_k - z^p := z_k - h(x^p, 0)$, we can rewrite the above linearized models as follows:

$$e_{k+1} = A_s e_k + B_s \delta u_k + G_s w_k \tag{94a}$$

$$\delta z_k = H_s e_k + M_s v_k \tag{94b}$$

Stationary Kalman Filter: In SKF, we aim to provide an estimate of the system's state based on the available partial information we have obtained until time k, i.e., $z_{0:k}$. The state estimate is a random vector denoted by x_k^+, whose distribution is the conditional distribution of the state on the obtained observations so far, which is called belief and is denoted by $b_k = p(x_k^+) = p(x_k | z_{0:k})$. In the Gaussian case, the belief can only be characterized by its mean and covariance, i.e., $b_k \equiv (\widehat{x}_k^+, P_k)$. Thus, in the Gaussian case, we can write:

$$b_k = p(x_k^+) = p(x_k | z_{0:k}) = \mathcal{N}(\widehat{x}_k^+, P_k) \Leftrightarrow b_k \equiv (\widehat{x}_k^+, P_k) \tag{95}$$

$$\widehat{x}_k^+ = \mathbb{E}[x_k | z_{0:k}], \qquad P_k = \mathbb{C}[x_k | z_{0:k}] \tag{96}$$

where $\mathbb{E}[\cdot|\cdot]$ and $\mathbb{C}[\cdot|\cdot]$ are the conditional expectation and conditional covariance operators, respectively.

SKF consists of two steps at every time stage: a prediction step and an update step. In the prediction step, the mean and covariance of prior x_k^- are computed. For the system in Eq. (94) prediction step is:

$$\widehat{e}_{k+1}^- = A_s \widehat{e}_k^+ + B_s \delta u_k \tag{97}$$

$$P_{k+1}^- = A_s P_k^+ A_s^T + G_s Q_s G_s^T \tag{98}$$

In the update step, the mean and covariance of posterior x_k^+ are computed. For the error system in Eq. (94), the update step is:

$$K_k = P_k^- H_s^T (H_s P_k^- H_s^T + M_s R_s M_s^T)^{-1} \tag{99}$$

$$\widehat{e}_{k+1}^+ = \widehat{e}_{k+1}^- + K_{k+1}(\delta z_{k+1} - H_s \widehat{e}_{k+1}^-) \tag{100}$$

$$P_{k+1}^+ = (I - K_{k+1} H_s) P_{k+1}^- \tag{101}$$

Note that

$$\widehat{x}_k^+ = x^p + \widehat{e}_k^+, \quad P_k = P_k^+ \tag{102}$$

In SKF, if (A_s, H_s) is an observable pair and $(A_s, \check{Q}_s)$ is a controllable pair, where $G_s Q_s G_s^T = \check{Q}_s \check{Q}_s^T$ then the prior and posterior covariances P_k^- and P_k and the filter gain K_k, all converge to their stationary values, denoted by P_s^-, P_s, and K_s, respectively [Bertsekas, 2007]. The P_s^- can be computed by solving the following DARE (Discrete Algebraic Riccati Equation). Having P_s^-, stationary gain K_s and estimation covariance P_s is computed as follows:

$$P_s^- = G_s Q_s G_s^T + A_s(P_s^- - P_s^- H_s^T (H_s P_s^- H_s^T + M_s R_s M_s^T)^{-1} H_s P_s^-) A_s^T, \tag{103}$$

$$K_s = P_s^- H_s^T (H_s P_s^- H_s^T + M_s R_s M_s^T)^{-1}, \tag{104}$$

$$P_s = (I - K_s H_s) P_s^- \tag{105}$$

Stationary LQR controller: In Stationary LQR (SLQR) we have a stationary mapping μ_s from the belief space to the control space that generates an optimal control based on the given belief $u_k = \mu_s(b_k)$ at every time step k. The SLQR controller is optimal in the sense of minimizing the following cost:

$$\begin{aligned} J_{SLQR} &= \mathbb{E}\left[\sum_{k\geq 0} (\widehat{x}_k^+ - x^p)^T W_x (\widehat{x}_k^+ - x^p) + (u_k)^T W_u (u_k)\right] \\ &= \mathbb{E}\left[\sum_{k\geq 0} (\widehat{e}_k^+)^T W_x (\widehat{e}_k^+) + (\delta u_k)^T W_u (\delta u_k)\right] \end{aligned} \tag{106}$$

If the (A_s, B_s) is a controllable pair and $(A_s, \check{W}_x)$ is an observable pair, where $\check{W}_x{}^T \check{W}_x = W_x$, then, the stationary linear control law that minimizes the cost function J_{SLQR} for a linear system is of the form:

$$\delta u_k = -L_s \widehat{e}_k^+ \tag{107}$$

where the stationary feedback gains L_s can be computed as follows:

$$L_s = (B_s^T S_s B_s + W_u)^{-1} B_s^T S_s A_s \tag{108}$$

$$S_s = W_x + A_s^T S_s A_s - A_s^T S_s B_s L_s \tag{109}$$

where the second equation is indeed a DARE that can be efficiently solved for S_s. Plugging S_s into Eq. (108), we get the feedback gain L_s.

Stationary LQG controller: Plugging the obtained control law of SLQR into the SKF equations, we can get the following stationary dynamics for the defined errors:

$$\begin{aligned} \begin{pmatrix} e_{k+1} \\ \widetilde{e}_{k+1} \end{pmatrix} &= \begin{pmatrix} A_s - B_s L_s & B_s L_s \\ 0 & A_s - K_s H_s A_s \end{pmatrix} \begin{pmatrix} e_k \\ \widetilde{e}_k \end{pmatrix} \\ &+ \begin{pmatrix} G_s & 0 \\ G_s - K_s H_s G_s & -K_s M_s \end{pmatrix} \begin{pmatrix} w_k \\ v_{k+1} \end{pmatrix} \end{aligned} \tag{110}$$

or equivalently,

$$\begin{aligned} \begin{pmatrix} e_{k+1} \\ \widehat{e}_{k+1}^+ \end{pmatrix} &= \begin{pmatrix} A_s & -B_s L_s \\ K_s H_s A_s & A_s - B_s L_s - K_s H_s A_s \end{pmatrix} \begin{pmatrix} e_k \\ \widehat{e}_k^+ \end{pmatrix} \\ &+ \begin{pmatrix} G_s & 0 \\ K_s H_s G_s & K_s M_s \end{pmatrix} \begin{pmatrix} w_k \\ v_{k+1} \end{pmatrix} \end{aligned} \tag{111}$$

Defining $\zeta_k := (e_k, \widehat{e}_k^+)^T$ and $q_k := (w_k, v_{k+1})^T$, we can rewrite Eq. (111) in a more compact form as

$$\zeta_{k+1} = \overline{F}_s \zeta_k + \overline{G}_s q_k, \quad q_k \sim \mathcal{N}(0, \overline{Q}_s), \quad \overline{Q}_s = \begin{pmatrix} Q_s & 0 \\ 0 & R_s \end{pmatrix} \tag{112}$$

with appropriate definitions for $\overline{F}_s$ and $\overline{G}_s$.

It can be shown that if $\overline{F}_s$ is a stable matrix, i.e. $\lim_{\kappa\to\infty} (\overline{F}_s)^\kappa = 0$, the ζ_k converges in distribution to $\zeta_s \sim \mathcal{N}(0, \mathcal{P}_s)$. Stationary covariance $\mathcal{P}_s$ is the solution of the following Lyapunov equation:

$$\mathcal{P}_s = \overline{F}_s \mathcal{P}_s \overline{F}_s^T + \overline{G}_s \overline{Q}_s \overline{G}_s^T \tag{113}$$

Note that $\mathcal{P}_s$ can be decomposed to four blocks

$$\mathcal{P}_s = \begin{pmatrix} \mathcal{P}_{s,11} & \mathcal{P}_{s,12} \\ \mathcal{P}_{s,21} & \mathcal{P}_{s,22} \end{pmatrix} \tag{114}$$

such that $\mathcal{P}_{s,11} = \lim_{k\to\infty} \mathbb{C}[e_k]$ and $\mathcal{P}_{s,22} = \lim_{k\to\infty} \mathbb{C}[\widehat{e}_k^+]$. Therefore, since $\widehat{x}_k^+ = x^p + \widehat{e}_k^+$, the estimation mean is also converging to a stationary random variable, denoted by $\widehat{x}_s^+$:

$$\widehat{x}_s^+ := \lim_{k\to\infty} \widehat{x}_k^+ \sim \mathcal{N}(x^p, \mathcal{P}_{s,22}) \tag{115}$$

Due to the linear relation $e_k = \widehat{e}_k^+ + \widetilde{e}_k$, we can also conclude $\lim_{k\to\infty} \mathbb{C}[\widetilde{e}_k] = \mathcal{P}_{s,11} + \mathcal{P}_{s,22} - 2\mathcal{P}_{s,12}$. It can be proven that in stationary LQG, the stability of matrix $\overline{\mathbf{F}}_s$ is a direct consequence of the controllability of pair (A_s, B_s) and the observability of pair (A_s, H_s) [Bertsekas, 2007],[Bertsekas, 1976].

Thus, collecting all the conditions, if (A_s, B_s) and $(A_s, \check{Q}_s)$ are controllable pairs, where $G_s Q_s G_s^T = \check{Q}_s \check{Q}_s^T$, and if (A_s, H_s) and $(A_s, \check{W}_x)$ are observable pairs, where $W_x = \check{W}_x^T \check{W}_x$, then the belief b_k converges to a stationary belief under the stationary LQG:

$$b_s := \lim_{k\to\infty} b_k = \mathcal{N}(\widehat{x}_s^+, P_s^+) \tag{116}$$

where P_s^+ is a deterministic quantity and we can characterize the distribution over the stationary belief as:

$$b_s \equiv (\widehat{x}_s^+, P_s^+) \sim \mathcal{N}\left(\begin{pmatrix} x^p \\ P_s^+ \end{pmatrix}, \begin{pmatrix} \mathcal{P}_{s,22} & 0 \\ 0 & 0 \end{pmatrix}\right) \tag{117}$$

APPENDIX D
PROOF OF LEMMA 3

Proof: Let us consider the state space model of the linear system of interest as follows:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{G}w_k, \quad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{118a}$$

$$z_k = \mathbf{H}x_k + v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}). \tag{118b}$$

Based on Lemma 1, if $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}, \check{\mathbf{Q}})$ are controllable pairs, where $\mathbf{G}\mathbf{Q}\mathbf{G}^T = \check{\mathbf{Q}}\check{\mathbf{Q}}^T$, and if $(\mathbf{A}, \mathbf{H})$ and $(\mathbf{A}, \check{\mathbf{W}}_x)$ are observable pairs, where $\mathbf{W}_x = \check{\mathbf{W}}_x^T \check{\mathbf{W}}_x$, then the estimation covariance deterministically tends to a stationary covariance P_s. Therefore, for any $\epsilon > 0$, after a deterministic finite time, P_k enters the ϵ-neighborhood of the stationary covariance, denoted by P_s.

The estimation mean dynamics, however, is stochastic and is as follows for the system in Eq. (118):

$$\begin{aligned}
\widehat{x}_{k+1}^+ &= \mathbf{v} + (\mathbf{A} - \mathbf{B}\mathbf{L} - \mathbf{K}_{k+1}\mathbf{H}\mathbf{A})(\widehat{x}_k^+ - \mathbf{v}) \\
&\quad + \mathbf{K}_{k+1}\mathbf{H}\mathbf{A}(x_k - \mathbf{v}) + \mathbf{K}_{k+1}\mathbf{H}\mathbf{G}w_k + \mathbf{K}_{k+1}v_{k+1} \\
&= \mathbf{v} - (\mathbf{A} - \mathbf{B}\mathbf{L})\mathbf{v} + (\mathbf{A} - \mathbf{B}\mathbf{L} - \mathbf{K}_{k+1}\mathbf{H}\mathbf{A})\widehat{x}_k^+ \\
&\quad + \mathbf{K}_{k+1}\mathbf{H}\mathbf{A}x_k + \mathbf{K}_{k+1}\mathbf{H}\mathbf{G}w_k + \mathbf{K}_{k+1}v_{k+1}
\end{aligned} \tag{119}$$

where the Kalman gain $\mathbf{K}_k$ is:

$$\mathbf{K}_k = P_k^- \mathbf{H}^T (\mathbf{H} P_k^- \mathbf{H}^T + \mathbf{R})^{-1} \tag{120}$$

Since $\mathbf{K}$ is full rank (due to the condition on the rank of $\mathbf{H}$), and since v and w represent Gaussian noise, Eq. (119) induces an irreducible Markov process over the state space [Meyn and Tweedie, 2009]. Thus, if we have a stopping region for the estimation mean with size $\epsilon > 0$, the estimation mean process will hit this stopping region in finite time [Meyn and Tweedie, 2009], with probability one.

Based on the estimation mean dynamics in Eq. (119) and the state dynamics in Appendix C, in the absence of a stopping region, if the estimation mean process and state process start from $\widehat{x}_0^+$ and x_0, respectively, such that $\mathbb{E}[\widehat{x}_0^+] = \mathbf{v}$ and $\mathbb{E}[x_0] = \mathbf{v}$ (which indeed is the case in FIRM due to the usage of edge-controllers), "the mean of estimation mean" remains on the $\mathbf{v}$, i.e., $\mathbb{E}[\widehat{x}_k^+] = \mathbf{v}$, for all k. As a result, if we center the stopping region for the estimation mean at $\mathbf{v}$, the probability of hitting the stopping region is maximized and the stopping time is minimized.

Combining the results for estimation covariance and estimation mean, if we define the region B as a set in the Gaussian belief space with a non-empty interior centered at $(\mathbf{v}, P_s)$, the belief $b_k = (\widehat{x}_k^+, P_k)$ enters region B in finite time with probability one. Thus, the pair (B, μ) is a proper pair over $\mathbb{GB}$; i.e., B is reachable under μ starting from any Gaussian distribution. ■

APPENDIX E
PROOF OF LEMMA 4

Before proving Lemma 4, we state and prove the following lemma:

Lemma 5. *Consider the bounded function $0 \leq f(\mathcal{X}) \leq 1$, and kernel $k(\mathcal{X}', \mathcal{X}) \geq 0$. Then, for any set $\mathcal{A}$, we have:*

$$\| \int_{\mathcal{A}} k(\mathcal{X}', \mathcal{X}) f(\mathcal{X}') d\mathcal{X}' \| \leq \| \int_{\mathcal{A}} k(\mathcal{X}', \mathcal{X}) d\mathcal{X}' \|. \tag{121}$$

Proof: Given the properties of $f(\cdot)$ and $k(\cdot,\cdot)$, we have $k(\mathcal{X}',\mathcal{X})f(\mathcal{X}') \leq k(\mathcal{X}',\mathcal{X})$, for all $\mathcal{X}$ and $\mathcal{X}'$. Taking the integral from both sides with respect to $\mathcal{X}'$ and then taking the supremum norm with respect to $\mathcal{X}$, the result follows. ■

Now we prove Lemma 4.

Proof: If we denote the domain of operator $\mathbf{T}_{\mathcal{S}}$ by $\mathcal{D}$, we know that for all $f \in \mathcal{D}$, we have $0 \leq f(\mathcal{X}) \leq 1$, because $f(\mathcal{X})$ is the probability of some given set $\mathcal{S}$ under some given controller invoked at point $\mathcal{X}$. Thus, it cannot be negative or greater than one and based on Lemma 5, we have:

$$\begin{aligned}\|\mathbf{T}_{\mathcal{S}}\,[f]\,\| &= \|\int_{\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X})f(\mathcal{X}')d\mathcal{X}'\| \leq \|\int_{\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X})d\mathcal{X}'\| \\ &= \|\mathbb{P}_1(\overline{\mathcal{S}}|\mathcal{X},\mu)\| \leq 1. \end{aligned} \tag{122}$$

Therefore, based on the definition of operator norm, we have:

$$\|\mathbf{T}_{\mathcal{S}}\|_{op} = \sup_{f(\cdot)}\{\|\mathbf{T}_{\mathcal{S}}[f]\| \;:\; \forall f \in \mathcal{D}, \|f\| \leq 1\} \leq 1. \tag{123}$$

According to Assumption 1, there exists a finite number N, such that:

$$\inf_{\mathcal{X}} \mathbb{P}_n(\mathcal{S}|\mathcal{X},\mu) = \beta > 0 \quad \forall n > N, \tag{124}$$

where "inf" and "sup" denote the infimum and supremum, respectively. Thus, we have

$$\begin{aligned}\|\mathbb{P}_n(\overline{\mathcal{S}}|\mathcal{X},\mu)\| &= \sup_{\mathcal{X}}(1-\mathbb{P}_n(\mathcal{S}|\mathcal{X},\mu)) = 1 - \inf_{\mathcal{X}}\mathbb{P}_n(\mathcal{S}|\mathcal{X},\mu) \\ &= 1-\beta < 1 \quad \forall n > N. \end{aligned} \tag{125}$$

Let us denote the n-th iterated kernel of $\mathbf{T}_{\mathcal{S}}$ as $p_n(\mathcal{X}'|\mathcal{X},\mu)$. Since this iterated kernel is a pdf, we have $p_n(\mathcal{X}'|\mathcal{X},\mu) \geq 0,\ \forall\mathcal{X},\forall\mathcal{X}',\forall n$. We can write:

$$\begin{aligned}\|\mathbf{T}_{\mathcal{S}}^{N}\,[f]\,\| &= \|\int_{\overline{\mathcal{S}}} p_N(\mathcal{X}'|\mathcal{X},\mu)f(\mathcal{X}')d\mathcal{X}'\| \\ &\leq \|\int_{\overline{\mathcal{S}}} p_N(\mathcal{X}'|\mathcal{X},\mu)d\mathcal{X}'\| = \|\mathbb{P}_N(\overline{\mathcal{S}}|\mathcal{X},\mu)\| \leq \alpha < 1, \end{aligned} \tag{126}$$

where $\alpha = 1-\beta$, and similar to Eq. (123), we get $\|\mathbf{T}_{\mathcal{S}}^{N}\|_{op} \leq \alpha < 1$. From the operator norm properties, we have:

$$\|\mathbf{T}_{\mathcal{S}}^{N+1}\|_{op} \leq \|\mathbf{T}_{\mathcal{S}}^{N}\|_{op}\|\mathbf{T}_{\mathcal{S}}\|_{op} \leq \alpha < 1$$

and similarly for all $n \geq N$, we have:

$$\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} \leq \alpha < 1 \quad \forall n \geq N.$$

Now, consider the series: $\sum_{i=1}^{\infty}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op}$. We can split the sum to smaller pieces as follows:

$$\sum_{n=1}^{\infty}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} = \sum_{n=1}^{N}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} + \sum_{i=1}^{\infty}\sum_{n=iN+1}^{(i+1)N}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op}.$$

But because $\|\mathbf{T}_{\mathcal{S}}^{n+1}\|_{op} \leq \|\mathbf{T}_{\mathcal{S}}^{n}\|_{op}$ for all $n \geq N$, we have

$$\sum_{n=iN+1}^{(i+1)N}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} \leq N\|\mathbf{T}_{\mathcal{S}}^{iN}\|_{op}.$$

Also, we know

$$\|\mathbf{T}_{\mathcal{S}}^{iN}\|_{op} \leq \|\mathbf{T}_{\mathcal{S}}^{N}\|_{op}^{i} \leq \alpha^{i}$$

and thus, we have:

$$\begin{aligned}\sum_{n=1}^{\infty}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} &= \underbrace{\sum_{n=1}^{N}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op}}_{\leq N} + \sum_{i=1}^{\infty}\sum_{n=iN+1}^{(i+1)N}\|\mathbf{T}_{\mathcal{S}}^{n}\|_{op} \\ &\leq N + \sum_{i=1}^{\infty}N\alpha^{i} = N + \frac{N}{1-\alpha} = c < \infty. \end{aligned}$$

■

APPENDIX F
PROOF OF COROLLARY 1

Proof: We know $\|R\| \leq 1$, and thus we can write:

$$\|\sum_{n=0}^{\infty} \mathbf{T}_{\mathcal{S}}^n[R]\| \leq \sum_{n=0}^{\infty} \|\mathbf{T}_{\mathcal{S}}^n\|_{op}\|R\| \leq \sum_{n=0}^{\infty} \|\mathbf{T}_{\mathcal{S}}^n\|_{op} \leq c < \infty.$$

Thus, series $\sum_{n=0}^{\infty} \mathbf{T}_{\mathcal{S}}^n[R]$ is a convergent series and we can define the operator $(I - \mathbf{T}_{\mathcal{S}})^{-1}[R] = \sum_{n=0}^{\infty} \mathbf{T}_{\mathcal{S}}^n[R]$. We have

$$\|(I - \mathbf{T}_{\mathcal{S}})^{-1}\|_{op} = \|\sum_{n=0}^{\infty} \mathbf{T}_{\mathcal{S}}^n\|_{op} \leq c < \infty. \tag{127}$$

■

APPENDIX G
PROOF OF PROPOSITION 1

We first state the following lemma on the continuity of the transition probability in the local controller parameter.

Lemma 6. *Given Assumption 2, there exists a $c_2 < \infty$ such that*

$$\|p(\mathcal{X}'|\mathcal{X}, \mu(b; \mathbf{v})) - p(\mathcal{X}'|\mathcal{X}, \check{\mu}(b; \check{\mathbf{v}}))\| \leq c_2\|\mathbf{v} - \check{\mathbf{v}}\|. \tag{128}$$

Proof: The result directly follows by combining two parts of Assumption 2. ■

Now we are ready to prove Proposition 1.

Proof: To show $\mathbb{P}(\mathcal{B}|\mathcal{X}, \mu)$ is continuous in $\mathbf{v}$, we perturb $\mathbf{v}$ to some $\check{\mathbf{v}}$, such that $\|\mathbf{v} - \check{\mathbf{v}}\| < r$. The local controller associated with node $\check{\mathbf{v}}$ is referred to as $\check{\mu}$, whose successful absorption region is denoted by $\check{\mathcal{B}}$ and stopping region is $\check{\mathcal{S}}$. Similarly the corresponding transient operator and recurrent function are referred to as $\check{\mathbf{T}}_{\check{\mathcal{S}}}$ and $\check{R}$. Finally, the success probability associated with the perturbed node $\check{\mathbf{v}}$ is $\mathbb{P}(\check{\mathcal{B}}|\mathcal{X}, \check{\mu})$. To shorten the statements, we refer to $\mathbb{P}(\mathcal{B}|\mathcal{X}, \mu)$ and $\mathbb{P}(\check{\mathcal{B}}|\mathcal{X}, \check{\mu})$ respectively by $\mathfrak{P}(\mathcal{X})$ and $\check{\mathfrak{P}}(\mathcal{X})$. As a result of node perturbation, the success probability is perturbed as:

$$\begin{aligned}
&\mathbb{P}(\mathcal{B}|\mathcal{X}, \mu) - \mathbb{P}(\check{\mathcal{B}}|\mathcal{X}, \check{\mu}) := \mathfrak{P} - \check{\mathfrak{P}} = R + \mathbf{T}_{\mathcal{S}}[\mathfrak{P}] - \check{R} - \check{\mathbf{T}}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}] \\
&= R - \check{R} + \mathbf{T}_{\mathcal{S}}[\mathfrak{P}] - \mathbf{T}_{\mathcal{S}}[\check{\mathfrak{P}}] + \mathbf{T}_{\mathcal{S}}[\check{\mathfrak{P}}] - \mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}] + \mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}] - \check{\mathbf{T}}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}] \\
&= (R - \check{R}) + \mathbf{T}_{\mathcal{S}}[\mathfrak{P} - \check{\mathfrak{P}}] + (\mathbf{T}_{\mathcal{S}} - \mathbf{T}_{\check{\mathcal{S}}})[\check{\mathfrak{P}}] + (\mathbf{T}_{\check{\mathcal{S}}} - \check{\mathbf{T}}_{\check{\mathcal{S}}})[\check{\mathfrak{P}}],
\end{aligned}$$

where

$$\mathbf{T}_{\check{\mathcal{S}}}\left[f(\cdot)\right](\mathcal{X}) := \int_{\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) f(\mathcal{X}') d\mathcal{X}'. \tag{129}$$

Let us define the operators $\mathbf{T}_{\Delta\mathcal{S}} := (\mathbf{T}_{\mathcal{S}} - \mathbf{T}_{\check{\mathcal{S}}})$ and $\Delta\mathbf{T}_{\check{\mathcal{S}}} := (\mathbf{T}_{\check{\mathcal{S}}} - \check{\mathbf{T}}_{\check{\mathcal{S}}})$. Now, based on Corollary 1, we can write:

$$\mathfrak{P} - \check{\mathfrak{P}} = (I - \mathbf{T}_{\mathcal{S}})^{-1}\left[R - \check{R} + \mathbf{T}_{\Delta\mathcal{S}}[\check{\mathfrak{P}}] + \Delta\mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}]\right], \tag{130}$$

and thus the following inequality holds on the supremum norm of the perturbation of the absorption probability:

$$\begin{aligned}
&\|\mathfrak{P} - \check{\mathfrak{P}}\| \\
&\leq \|(I - \mathbf{T}_{\mathcal{S}})^{-1}\|_{op}\left(\|R - \check{R}\| + \|\mathbf{T}_{\Delta\mathcal{S}}[\check{\mathfrak{P}}]\| + \|\Delta\mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}]\|\right) \\
&\leq c\left(\|R - \check{R}\| + \|\mathbf{T}_{\Delta\mathcal{S}}[\check{\mathfrak{P}}]\| + \|\Delta\mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}]\|\right) \\
&= c\left(\|K_1(\mathcal{X})\| + \|K_2(\mathcal{X})\| + \|K_3(\mathcal{X})\|\right),
\end{aligned} \tag{131}$$

where $K_1(\mathcal{X}) := R(\mathcal{X}) - \check{R}(\mathcal{X})$, $K_2(\mathcal{X}) := \mathbf{T}_{\Delta\mathcal{S}}[\check{\mathfrak{P}}(\cdot)](\mathcal{X})$, and $K_3(\mathcal{X}) := \Delta\mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}(\cdot)](\mathcal{X})$. In the following we bound K_1, K_2, and K_3, and thus bound $\|\mathfrak{P} - \check{\mathfrak{P}}\|$, accordingly.

1) Bound for $K_1(\mathcal{X})$: The supremum norm of $K_1(\mathcal{X})$ is:

$$\begin{aligned}
&\|K_1(\mathcal{X})\| = \|R(\mathcal{X}) - \check{R}(\mathcal{X})\| \\
&= \| \int\limits_{\mathcal{B}} p^{\mu}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' - \int\limits_{\check{\mathcal{B}}} p^{\check{\mu}}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' \| \\
&= \| \int\limits_{\mathcal{B}\cap\check{\mathcal{B}}} [p^{\mu}(\mathcal{X}'|\mathcal{X}) - p^{\check{\mu}}(\mathcal{X}'|\mathcal{X})] d\mathcal{X}' \\
&+ \int\limits_{\mathcal{B}-\check{\mathcal{B}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' - \int\limits_{\check{\mathcal{B}}-\mathcal{B}} p^{\check{\mu}}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' \| \\
&\leq \int\limits_{\mathcal{B}\cap\check{\mathcal{B}}} \|p^{\mu}(\mathcal{X}'|\mathcal{X}) - p^{\check{\mu}}(\mathcal{X}'|\mathcal{X})\| d\mathcal{X}' \\
&+ \| \int\limits_{\mathcal{B}-\check{\mathcal{B}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' + \int\limits_{\check{\mathcal{B}}-\mathcal{B}} p^{\check{\mu}}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' \| \\
&\overset{\text{from (128)}}{\leq} \int\limits_{\mathcal{B}\cap\check{\mathcal{B}}} c_2 \|\mathbf{v} - \check{\mathbf{v}}\| d\mathcal{X}' + \|\mathbb{P}_1(\mathcal{B} \ominus \check{\mathcal{B}}|\mathcal{X}, \mu)\| \\
&\qquad + \|\mathbb{P}_1(\check{\mathcal{B}} \ominus \mathcal{B}|\mathcal{X}, \check{\mu})\| \\
&\overset{\text{from (50)}}{\leq} c_2' \|\mathbf{v} - \check{\mathbf{v}}\| + 2c' \|\mathbf{v} - \check{\mathbf{v}}\| = \gamma_1 \|\mathbf{v} - \check{\mathbf{v}}\|,
\end{aligned} \tag{132}$$

where $c_2' < \infty$ and $\gamma_1 = c_2' + 2c' < \infty$. In the penultimate inequality, we also used the fact that $\mathbb{P}_1(\check{\mathcal{B}} - \mathcal{B}|\mathcal{X}, \check{\mu}) \leq \mathbb{P}_1(\check{\mathcal{B}} \ominus \mathcal{B}|\mathcal{X}, \check{\mu})$ and $\mathbb{P}_1(\mathcal{B} - \check{\mathcal{B}}|\mathcal{X}, \mu) \leq \mathbb{P}_1(\mathcal{B} \ominus \check{\mathcal{B}}|\mathcal{X}, \mu)$ because $\check{\mathcal{B}} - \mathcal{B} \subseteq \check{\mathcal{B}} \ominus \mathcal{B}$ and $\mathcal{B} - \check{\mathcal{B}} \subseteq \mathcal{B} \ominus \check{\mathcal{B}}$.

2) Bound for $K_2(\mathcal{X})$: We have:

$$\begin{aligned}
&\|K_2(\mathcal{X})\| = \|\mathbf{T}_{\Delta\mathcal{S}}[\check{\mathfrak{P}}]\| = \|\mathbf{T}_{\mathcal{S}}[\check{\mathfrak{P}}] - \mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}]\| \\
&= \| \int\limits_{\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' - \int\limits_{\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' \| \\
&= \| \int\limits_{\overline{\mathcal{S}}-\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' - \int\limits_{\overline{\check{\mathcal{S}}}-\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' \| \\
&\leq \| \int\limits_{\overline{\mathcal{S}}-\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' + \int\limits_{\overline{\check{\mathcal{S}}}-\overline{\mathcal{S}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' \| \\
&= \| \int\limits_{\overline{\mathcal{S}}\ominus\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' \| \overset{\text{from (121)}}{\leq} \| \int\limits_{\overline{\mathcal{S}}\ominus\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) d\mathcal{X}' \| \\
&= \|\mathbb{P}_1(\overline{\mathcal{S}} \ominus \overline{\check{\mathcal{S}}}|\mathcal{X}, \mu)\| \leq \|\mathbb{P}_1(\overline{\mathcal{B}} \ominus \overline{\check{\mathcal{B}}}|\mathcal{X}, \mu)\| \\
&= \|\mathbb{P}_1(\mathcal{B} \ominus \check{\mathcal{B}}|\mathcal{X}, \mu)\| \overset{\text{from (50)}}{\leq} \gamma_2 \|\mathbf{v} - \check{\mathbf{v}}\|,
\end{aligned} \tag{133}$$

where $\gamma_2 = c' < \infty$. The penultimate inequality and equality follow from the relations $\overline{\mathcal{S}} \ominus \overline{\mathcal{S}'} \subseteq \overline{\mathcal{B}} \ominus \overline{\mathcal{B}'}$ and $\overline{\mathcal{B}} \ominus \overline{\mathcal{B}'} = \mathcal{B} \ominus \mathcal{B}'$, respectively.

3) Bound for $K_3(\mathcal{X})$: We have:

$$\begin{aligned}
&\|K_3(\mathcal{X})\| = \|\Delta\mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}]\| = \|\mathbf{T}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}] - \check{\mathbf{T}}_{\check{\mathcal{S}}}[\check{\mathfrak{P}}]\| \\
&= \| \int\limits_{\overline{\check{\mathcal{S}}}} p^{\mu}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' - \int\limits_{\overline{\check{\mathcal{S}}}} p^{\check{\mu}}(\mathcal{X}'|\mathcal{X}) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' \| \\
&= \| \int\limits_{\overline{\check{\mathcal{S}}}} \left(p^{\mu}(\mathcal{X}'|\mathcal{X}) - p^{\check{\mu}}(\mathcal{X}'|\mathcal{X})\right) \check{\mathfrak{P}}(\mathcal{X}') d\mathcal{X}' \| \\
&\leq \int\limits_{\overline{\check{\mathcal{S}}}} \|p^{\mu}(\mathcal{X}'|\mathcal{X}) - p^{\check{\mu}}(\mathcal{X}'|\mathcal{X})\| \|\check{\mathfrak{P}}(\mathcal{X}')\| d\mathcal{X}'
\end{aligned}$$

$$\overset{\text{from (128)}}{\leq} \int_{\bar{\mathcal{S}}} c_2 \|\mathbf{v} - \check{\mathbf{v}}\| d\mathcal{X}' = \gamma_3 \|\mathbf{v} - \check{\mathbf{v}}\|, \tag{134}$$

where $\gamma_3 < \infty$.

Therefore, based on Eq. (132), Eq. (133), Eq. (134), and Eq. (131), we can conclude that:

$$\|\mathbb{P}(\mathcal{B}|\mathcal{X}, \mu) - \mathbb{P}(\check{\mathcal{B}}|\mathcal{X}, \check{\mu})\| \leq \gamma \|\mathbf{v} - \check{\mathbf{v}}\|, \tag{135}$$

where $\gamma = c(\gamma_1 + \gamma_2 + \gamma_3) < \infty$, which completes the proof that the absorption probability under the controller μ is continuous in the PRM node $\mathbf{v}$. ■

APPENDIX H
PROOF OF THEOREM 1

Before starting with the proof of Theorem 1, we state the following proposition that concludes the continuity of the success probability of π (overall planner) given the continuity of the success probability of the individual local planners (μ^{ij}s).

Proposition 2. (Continuity of success probability of π): The success probability $\mathbb{P}(\text{success}|b_0, \pi)$ is continuous in $\mathcal{V}$, if the absorption probabilities $\mathbb{P}(B^j|b, \mu^{ij})$ are continuous in $\mathbf{v}^j$ for all i, j, and b.

Proof: Given that $\mathbb{P}(B^j|b, \mu^{ij})$ is continuous in $\mathbf{v}^j$, for all i, j, we want to show that $\mathbb{P}(\text{success}|\pi, b_0)$ is continuous in all $\mathbf{v}^j$. First, let us look at the structure of the success probability.

$$\mathbb{P}(\text{success}|b_0, \pi) = \mathbb{P}(B(\mu_0)|b_0, \mu_0)\mathbb{P}(\text{success}|B(\mu_0), \pi^g), \tag{136}$$

where μ_0 is computed using Eq. (34). The term $\mathbb{P}(B(\mu_0)|b_0, \mu_0)$ in the right hand side of Eq. (136) is continuous because the continuity of $\mathbb{P}(B^j|b, \mu^{ij})$ for all i, j is assumed in this lemma. Thus, we only need to show the continuity of the second term in Eq. (136). Without loss of generality we can consider $B^i = B(\mu_0)$. Then, we need to show that $\mathbb{P}(\text{success}|B^i, \pi^g)$ is continuous in $\mathbf{v}^i$ for all i.

As we saw in Section VI-G, the probability of success from the i-th FIRM node is as follows:

$$\mathbb{P}(\text{success}|B^i, \pi^g) = \Gamma_i^T (I - \mathcal{Q})^{-1} \mathcal{R}_g, \tag{137}$$

Moreover, we can consider $B^{goal} = B^N$ without loss of generality; then, the (i, j)-th element of matrix $\mathcal{Q}$ is $\mathcal{Q}[i, j] = \mathbb{P}(B^i|B^j, \pi^g(B^j))$, and the j-th element of vector $\mathcal{R}_g$ is $\mathcal{R}_g[j] = \mathbb{P}(B^N|B^j, \pi^g(B^j))$. Since we considered the B^j as the stopping region of the local controller μ^{ij}, we have:

$$\mathbb{P}(B^j|B^i, \mu^{il}) = 0, \text{ if } l \neq j. \tag{138}$$

Therefore, all the non-zero elements in the matrices $\mathcal{R}_g$ and $\mathcal{Q}$ are of the form $\mathbb{P}(B^j|B^i, \mu^{ij})$. Thus, given the continuity of $\mathbb{P}(B^j|b, \mu^{ij})$, the transition probability $\mathbb{P}(B^j|B^i, \mu^{ij})$ is continuous and the matrices $\mathcal{R}_g$ and $\mathcal{Q}$ are continuous. Therefore, $\mathbb{P}(\text{success}|B^i, \pi^g)$ and thus $\mathbb{P}(\text{success}|b_0, \pi)$ are continuous in underlying PRM nodes. ■

Now we are ready to prove Theorem 1:

Proof: Based on the definition of probabilistic completeness under uncertainty, if there exists a successful policy $\check{\pi}$, FIRM has to find a successful policy π as the number of FIRM nodes increases unboundedly. Thus, we start by assuming that there exists a successful policy $\check{\pi} \in \Pi$ for a given initial belief b_0. Since each policy in Π is parametrized by a PRM graph, there exists a PRM with nodes $\check{\mathcal{V}} = \{\check{\mathbf{v}}^i\}_{i=1}^N$ that parametrizes the policy $\check{\pi}$. Since $\check{\pi}$ is a successful policy, we know $\mathbb{P}(\text{success}|b_0, \check{\pi}) > p_{min}$. Thus, we can define $\epsilon^* = \mathbb{P}(\text{success}|b_0, \check{\pi}) - p_{min} > 0$.

Given Assumptions 1, 2, and 3, and based on Propositions 1 and 2, we know that $\mathbb{P}(\text{success}|b_0, \pi)$ is continuous with respect to the parameters of the local planners, i.e., for any $\epsilon > 0$, there exists a $\delta > 0$, such that if $\|\mathcal{V} - \check{\mathcal{V}}\| < \delta$, then $|\mathbb{P}(\text{success}|b_0, \pi(\cdot; \mathcal{V})) - \mathbb{P}(\text{success}|b_0, \check{\pi}(\cdot; \check{\mathcal{V}}))| < \epsilon$. The notation $\|\mathcal{V} - \check{\mathcal{V}}\| < \delta$ means that $\|\mathbf{v}^i - \check{\mathbf{v}}^i\| < \delta$, for all i, or equivalently, $\mathbf{v}^i \in \check{\Omega}_i$, for all i, where $\check{\Omega}_i$ is a ball with radius δ, centred at $\check{\mathbf{v}}^i$.

Therefore, for the introduced ϵ^*, there exists a δ^* and corresponding regions $\{\check{\Omega}_i\}_{i=1}^N$, such that if we have a PRM whose nodes (or a subset of nodes – A subset of nodes is sufficient, because the success probability is a non-decreasing function in terms of the number of nodes) satisfy the condition $\mathbf{v}_i^* \in \check{\Omega}_i$, for all $i = 1, \cdots, N$, then the planner π parametrized by this PRM has a success probability greater than p_{min}, i.e., $\mathbb{P}(\text{success}|b_0, \pi(\cdot; \mathcal{V})) > p_{min}$, and hence π is successful.

Since $\delta > 0$, the regions $\check{\Omega}_i$ have nonempty interiors. Consider a PRM with a sampling algorithm, under which there is nonzero probability of sampling in $\check{\Omega}_i$, such as uniform sampling. Thus, starting with any PRM, if we increase the number of nodes, a PRM node will eventually be chosen at every $\check{\Omega}_i$, with probability one. Therefore the policy constructed based on these nodes will have a success probability greater than p_{min}, i.e., we eventually get a successful policy if one exists. Thus, FIRM is probabilistically complete. ■