Sampling-based nonholonomic motion planning in belief space via dynamic feedback linearization-based FIRM

Ali-akbar Agha-mohammadi, Suman Chakravorty, Nancy M. Amato
{aliagha,schakrav,amato}@tamu.edu

Technical Report TR12-004
Parasol Lab.
Department of Computer Science and Engineering
Texas A&M University
February 01, 2012

Abstract

In roadmap-based methods, such as the Probabilistic Roadmap Method (PRM) in deterministic environments or the Feedback-based Information RoadMap (FIRM) in partially observable probabilistic environments, a stabilizing controller is needed to guarantee node reachability in state or belief space. In belief space, it has been shown that belief-node reachability can be achieved using stationary Linear Quadratic Gaussian (LQG) controllers, for linearly controllable systems. However, for nonholonomic systems such as a unicycle model, belief reachability is a challenge. In this paper, we construct a roadmap in information space, where the local planners in partially-observable space are constructed by utilizing a Kalman filter as an estimator along with a Dynamic Feedback Linearization-based (DFL-based) controller as the belief controller. As a consequence, the task of belief stabilization to pre-defined nodes in belief space is accomplished even for nonholonomic systems. Therefore, a query-independent roadmap is generated in belief space that preserves the "principle of optimality", required in dynamic programming solvers. This method serves as an offline POMDP solver for motion planning in belief space, which can seamlessly take obstacles into account. Experimental results show the efficiency of both individual local planners and the overall planner over the information graph for a nonholonomic model.

I. Introduction

This work aims at providing a sampling-based feedback solution for nonholonomic motion planning in belief space, i.e., under motion and sensing unertainties.

Nonholonomic motion planning (NMP) deals with planning open-loop or feedback (closed-loop) plans for moving an object that is subject to nonholonomic constraints. The unicycle model is an important example of nonholonomic systems, which is commonly used to approximately model a variety of systems ranging from differential drive and synchro drive single-body robots [1] to steerable needles in surgery [2]. One of the challenges in nonholonomic motion planning is stabilizing the system to a point. Thus, if we consider two basic motion tasks: *point-to-point motion*, which deals with driving a moving object from an initial state to a goal state, and *trajectory following*, which deals with following a trajectory in state space, then, in contrast to the holonomic case, point-to-point motion in nonholonomic systems is a much more difficult task than trajectory tracking [3]. As mentioned, the main challenge is the state stabilization to the target node.

Motion planning under uncertainty (MPUU) is an instance of sequential decision making problem under uncertainty. Considering the uncertainty in an object's motion and sensory readings, the state of the system is not fully observable, and thus is not available for decision making. In such a situation, a state estimation module can provide a probability distribution function (pdf) over the possible states of the system, and therefore decision making is performed in the space of these distributions, the so called information space (or belief space). Planning in belief space in its most general form is formulated as a Partially Observable Markov Decision Process (POMDP) problem[4], [5]. However, only a very small class of POMDP problems can be solved exactly. In particular, planning (i.e., solving POMDPs) in continuous state, control, and observation spaces, where our problem resides, is a formidable challenge.

Sampling-based motion planning methods have shown great success in dealing with many motion planning problems in complex environments and are divided into two main classes: Roadmap-based (graph-based) methods such as the Probabilistic Roadmap Method (PRM) and its variants [6], [7] and tree-based, e.g., [8], [9]. In dealing with MPUU, roadmap-based methods

have a desirable feature: since the solution of POMDP is a feedback policy over the entire belief space, it does not depend on the initial belief and from any given belief it produces a best action. This property matches well with roadmap-based methods that are multi query so that a feedback policy can be defined on the graph, which produces the best action at each graph node. On the other hand, tree-based methods are usually query-dependent and rooted in the start point of the query.

Similar to motion planning in state space, in belief space the basic motion tasks can be defined as: *point-to-point motion*, which deals with driving the belief of the moving object from a given belief to a target belief, and *trajectory following*, which deals with following a trajectory in belief space. Both these tasks are even more challenging in belief space than in state space. The point-to-point motion in belief space is required to construct a query-independent roadmap in belief space. A number of pioneering methods have been proposed in [10], [11], [12] for applying sampling-based ideas for motion planning in belief space. However, one challenge they all face is that they do not support point-to-point motion functionality in belief space. Thus, they break the *principle of optimality* (or *optimal substructure property*) needed to solve the DP equation, and needed for Dijkstra's algorithm [13]. Therefore, roadmap construction depends on the start point of the submitted query, and hence a new roadmap has to be constructed for every query. Prentice *et al.* [10] and Huynh *et al.* [11] reuse a significant portion of the computation by using covariance factorization techniques to reduce the computational burden imposed by the query dependence.

In fully-observable environments, Generalized PRM [14] performs point-to-point motion under motion uncertainty. In partially observable environments, under motion and sensing uncertainty, the Feedback-based Information RoadMap (FIRM) [15], [16] utilizes feedback controllers for the purpose of belief stabilization, and hence embeds the point-to-point motion behaviour in belief space. As a consequence, the generated roadmap in belief space is query-independent and only needs to be constructed once. Also, the principle of optimality is preserved on FIRM, and the connection between its solution and the original POMDP can be established rigorously [15], [16]. It is also shown to be probabilistically complete [17]. FIRM is an abstract framework for planning in belief space, and a Linear Quadratic Gaussian-based (LQG-based) instantiation of FIRM is reported in [15], [16]. The main shortcoming of the LQG-based FIRM is that it only works for systems which are linearly controllable about a fixed node point. This condition excludes nonholonomic systems, which are not stabilizable to a fixed point under linear controllers.

The main contribution of this work is to present an instantiation of the abstract FIRM framework [15], [16] for nonholonomic systems using Dynamic Feedback Linearization-based (DFL-based) controllers. The DFL-based control of nonholonomic models is a well-established technique, but its usage in belief space with a stationary Kalman filter to construct a Feedback Information RoadMap (FIRM) is both novel and powerful. This construct leads to a sampling-based feedback motion planner in belief space for nonholonomic systems that shows promising results in conducted experiments. It builds on PRM, where corresponding to each PRM node, we define a unique belief node and design a feedback controller with partial observations in such a way that the robot's belief is steered toward the pre-defined belief node independent of the starting point. Using the KF as the estimator, and the DFL-based controller as the belief controller, we embed point-to-point motion in belief space for nonholonomic systems. Thus, we can construct a roadmap in belief space independent of the query, which can be used efficiently for planning (replanning) purposes. Also, collision probabilities are seamlessly incorporated in planner's construction (see Section III-A).

II. Controllability in nonholonomic systems

An implicit assumption in roadmap based methods such as PRM is that on every edge there exists a controller to drive the robot from the start node to the end node of the edge or to an ϵ-neighborhood of the end node, for some small $\epsilon > 0$. For a linearly controllable robot, a linear controller can locally track a PRM edge and drive the robot to its endpoint node. However, for a nonholonomic robot such as a unicycle, the linearized model at any state point (and zero velocity) is not controllable, and hence, a linear controller cannot stabilize the robot to the PRM nodes. Consider the discrete unicycle model:

$$x_{k+1} = f(x_k, u_k, w_k) = \begin{pmatrix} \mathsf{x}_k + (V_k + n_v)\delta t \cos\theta_k \\ \mathsf{y}_k + (V_k + n_v)\delta t \sin\theta_k \\ \theta_k + (\omega_k + n_\omega)\delta t \end{pmatrix}, \tag{1}$$

where $x_k = (\mathsf{x}_k, \mathsf{y}_k, \theta_k)^T$ describes the robot state, in which $(\mathsf{x}_k, \mathsf{y}_k)^T$ is the 2D position of the robot and θ_k is the heading angle of the robot, at time step k. The vector $u_k = (V_k, \omega_k)^T$ is the control vector consisting of linear velocity V_k and angular velocity ω_k. The motion noise vector is denoted by $w_k = (n_v, n_\omega)^T \sim \mathcal{N}(0, \mathbf{Q}_k)$. Linearizing this system about the point (node) $\mathbf{v} = (\mathsf{x}^p, \mathsf{y}^p, \theta^p)$, one can conclude that the system is linearly controllable if and only if $V^p > 0$. Thus, in stabilizing the robot to a PRM node, where the nominal control is zero, $u^p = (V^p, \omega^p)^T = (0, 0)^T$, the system is not linearly controllable. Therefore, a linear controller cannot stabilize the unicycle to a PRM node. Moreover, based on the necessary condition in Brockett's paper [18], even a smooth time-invariant nonlinear control law cannot drive the unicycle to a PRM node, and the stabilizing controller has to be either discontinuous and/or time-varying.

On roadmaps in belief space, the situation is even more complicated, since the controller has to drive the robot to the ϵ-neighborhood of a belief node in belief space. Again, if the linearized system is controllable, using a linear stochastic

controller such as the stationary LQG controller, one can drive the robot belief to the belief node [15], [16]. However, if the linearized system about the desired point is not controllable, the belief stabilization, if possible, is much more difficult than state stabilization. Consider a unicycle robot, equipped with sensors measuring the range and bearings from a set of landmarks in the environment. Linearizing the motion and sensing models of this system for stabilization purposes, we get a linearly observable system but not a linearly controllable system. In this paper, we handle this situation by utilizing a DFL-based controller along with a Kalman filter to steer the system belief toward a pre-defined node in belief space.

III. Motion planning under uncertainty

Partially Observable Markov Decision Processes (POMDPs) are the most general formulation for motion planning under motion and sensing uncertainties. Note that in this paper, the environment map is assumed to be known. The solution of the POMDP problem is an optimal feedback (policy) $\pi(\cdot)$, which maps the information (belief) space to the control space. Let us denote the state, control and observation at time step k by x_k, u_k, and z_k, respectively, which belong to spaces $\mathbb{X}$, $\mathbb{U}$, and $\mathbb{Z}$, respectively. The belief in stochastic setting is defined as the pdf of the system state conditioned on the available data (measurements and controls up to the current time step), i.e., $b_k = p(x_k|z_{0:k}; u_{0:k-1})$ and $\mathbb{B}$ denotes the belief space, containing all possible beliefs. It is well known that the POMDP problem can be posed as a belief MDP problem [19], [20], whose solution $\pi(\cdot)$ is computed by solving the following Dynamic Programming (DP) equation:

$$J(b) = \min_u \{c(b,u) + \int_{\mathbb{B}} p(b'|b,u) J(b') db'\}, \tag{2a}$$

$$\pi(b) = \arg\min_u \{c(b,u) + \int_{\mathbb{B}} p(b'|b,u) J(b') db'\}, \tag{2b}$$

where $J(\cdot) : \mathbb{B} \to \mathbb{R}$ is the optimal cost-to-go function, $p(b'|b,u)$ is the belief transition pdf under control u, and $c(b,u)$ is the one-step cost of taking control u at belief b.

A. FIRM: Feedback-based Information RoadMap

It is well known that the above DP equation is exceedingly difficult to solve since it is defined over an infinite-dimensional belief space. In this paper, we approach this problem through the FIRM framework [15], [16] and accordingly, transform the POMDP in (2) into a tractable MDP problem. In this section, we briefly present the abstract FIRM framework and in the next section we explain how we can make a concrete instantiation of this framework that works for nonholonomic systems. In FIRM, we aim to sample in belief space and compute the cost-to-go of the sampled beliefs instead of cost-to-go for all beliefs in belief space. Then, we show that although this sampling in belief space leads to an intractable belief SMDP (Semi-Markov Decision Process) problem, it can be arbitrarily well approximated by a tractable FIRM MDP on the FIRM nodes. The main tools in this transformation are the local feedback planners, which are designed to ensure that at decision making stages we reach pre-defined FIRM nodes in belief space. It is worth noting that the power of the local feedback planners resides in the fact that each local planner is a sequence of closed-loop policies (macro-policies) that can steer the belief toward a predefined belief-node, as opposed to a macro-action [21], which is a sequence of actions, i.e., a sequence of open-loop policies, which is not capable of belief stabilization.

Underlying PRM: First, a PRM is constructed in the state space, with nodes denoted by $\mathcal{V} = \{\mathbf{v}^j\}_{j=1}^n$ and edges denoted by $\mathcal{E} = \{\mathbf{e}_{ij}\}$. Note that $\mathcal{V}$ includes the goal node to whose vicinity we want to transfer the robot.

FIRM nodes: FIRM nodes are disjoint small sets in belief space. Corresponding to each PRM node, we design a unique FIRM node. The FIRM node corresponding to $\mathbf{v}^i$ is denoted by $B^i \subset \mathbb{B}$, whose actual construction depends on the choice of local controller and is discussed in Section IV. The set of all FIRM nodes is denoted by $\mathbb{V} = \{B^i\}_{i=1}^n$.

Local planners: We use the phrases "local planner" and "local controller" interchangeably in this paper. The local controller $\mu(\cdot) : \mathbb{B} \to \mathbb{U}$ is a feedback policy. The role of the (i,j)-th local controller, denoted by μ^{ij}, is to take a belief from the FIRM node B^i to the FIRM node B^j. We denote the set of local controllers by $\mathbb{M} = \{\mu^{ij}\}$. In other words, μ^{ij} generates controls based on the belief at each stage, until the belief reaches the region B^j. Thus, the local controller μ^{ij} has to be a proper policy with respect to the region B^j over the initial region B^i as stated in the following property:

Property 1. (Reachability property of FIRM)*: For every PRM edge $\mathbf{e}_{ij}$, there exists a controller μ^{ij}, such that for all $b \in B^i$ we have $\mathbb{P}(B^j|b,\mu^{ij}) = 1$, in the absence of obstacles. In other words, in the obstacle-free environment, the feedback controller $\mu^{ij}(\cdot)$ has to drive the belief state from B^i into a B^j in a finite time with probability one.*

Satisfying reachability property of FIRM: In [15], [16] it is proved that using stationary LQG controllers as local planners for linearly controllable systems about a fixed point, Property 1 (reachability condition) is satisfied. However, as discussed in Section II, LQG controllers cannot satisfy Property 1 for nonholonomic systems. In Section IV we propose a way to construct the DFL-based local controllers and FIRM nodes such that under a mild assumption Property 1 is satisfied for nonholonomic models such as a unicycle. In all the experimental runs we conducted, the DFL-based controller satisfied the above reachability property. However, if needed one can relax Property 1 by setting a deterministic maximum time T_{max},

such that if the reachability under the local controller is not achieved before T_{max}, then the run is considered to be failed, as if the robot collided with an obstacle.

Roadmap of local controllers: Local planners are parametrized by underlying PRM edges. So, one can view the procedure as "sampling the local planners". Similar to roadmap-based methods in which the path is constructed by concatenating edges of the roadmap, in FIRM the policy π is constructed (see Section III-B) by concatenating local feedback policies. At each FIRM node B^i, the set of local controllers that can be invoked is $\mathbb{M}^i = \{\mu^{ij} \in \mathbb{M} | \exists \mathrm{e}_{ij} \in \mathcal{E}\}$. However, it is worth noting that through this construction we still perform planning on continuous spaces, and do not discretize any of the state, control, or observation spaces.

Generalizing transition costs and probabilities: We generalize the concept of one-step cost $c(b,u) : \mathbb{B} \times \mathbb{U} \to \mathbb{R}$ to the concept of $C(b,\mu) : \mathbb{B} \times \mathbb{M} \to \mathbb{R}$. The step cost $C(b,\mu^{ij})$ represents the cost of invoking local controller $\mu^{ij}(\cdot)$ at belief state b, i.e.,

$$C(b,\mu^{ij}) = \sum_{t=0}^{\mathcal{T}^{ij}} c(b_t, \mu^{ij}(b_t)|b_0 = b), \tag{3}$$

$$\mathcal{T}^{ij}(b) = \inf_t \{t | b_t \in B^j, b_0 = b\}, \tag{4}$$

where $\mathcal{T}^{ij}$ is a random stopping time denoting the time at which the belief state enters the region B^j under the controller μ^{ij}. Similarly, we generalize the transition probabilities from $p(b'|b,u) : \mathbb{B}^2 \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ to $\mathbb{P}(\cdot|b,\mu) : \mathbb{V} \times \mathbb{B} \times \mathbb{M} \to [0,1]$, where $\mathbb{P}(B^j|b,\mu^{ij})$ is the transition probability from b to B^j under the local planner μ^{ij}.

Roadmap level transitions: Now, we transform the transition cost and probabilities to $C^g : \mathbb{V} \times \mathbb{M} \to \mathbb{R}$ and $\mathbb{P}^g : \{\mathbb{V}, F\} \times \mathbb{V} \times \mathbb{M} \to [0,1]$ that are defined on the FIRM graph, i.e., over the finite space $\{\mathbb{V}, F\}$. In other words, $\mathbb{P}^g(B^j|B^i,\mu^{ij})$ and $\mathbb{P}^g(F|B^i,\mu^{ij})$ are the transition probability from B^i to B^j and F, respectively, under the local planner μ^{ij}. Similarly, $C^g(B^i,\mu^{ij})$ denotes the cost of invoking local planner μ^{ij} at FIRM node B^i. Accordingly, $J^g : \{\mathbb{V}, F\} \to \mathbb{R}$ is the cost-to-go function over the FIRM nodes, where $J^g(F)$ is a user-defined suitably high cost for hitting obstacles. These roadmap level quantities are defined using the following "piecewise constant approximation", which is an arbitrarily good approximation for smooth transition probability and cost functions, and sufficiently small B^i's:

$$\forall b \in B^i, \forall i,j \begin{cases} J^g(B^i) := J(b_c^i) \approx J(b), \\ C^g(B^i,\mu^{ij}) := C(b_c^i,\mu^{ij}) \approx C(b,\mu^{ij}), \\ \mathbb{P}^g(\cdot|B^i,\mu^{ij}) := \mathbb{P}(\cdot|b_c^i,\mu^{ij}) \approx \mathbb{P}(\cdot|b,\mu^{ij}), \end{cases} \tag{5}$$

where b_c^i is a point in B^i. For example, if B^i is a ball, then b_c^i can be considered as the center of B^i. The approximation essentially states that any belief in the region B^i is represented by b_c^i for the purpose of decision making.

FIRM MDP: By this construction of local planners and FIRM nodes, the original POMDP in (2) is reduced (for details see [16]) to the following finite-state MDP, called the FIRM MDP:

$$\begin{aligned} J^g(B^i) = \min_{\mu^{ij} \in \mathbb{M}^i} \{ & C^g(B^i,\mu^{ij}) + J^g(F)\mathbb{P}^g(F|B^i,\mu^{ij}) \\ & + J^g(B^j)\mathbb{P}^g(B^j|B^i,\mu^{ij})\}, \end{aligned} \tag{6a}$$

$$\begin{aligned} \pi^g(B^i) = \arg \min_{\mu^{ij} \in \mathbb{M}^i} \{ & C^g(B^i,\mu^{ij}) + J^g(F)\mathbb{P}^g(F|B^i,\mu^{ij}) \\ & + J^g(B^j)\mathbb{P}^g(B^j|B^i,\mu^{ij})\}, \end{aligned} \tag{6b}$$

where, the solution of FIRM MDP $\pi^g : \mathbb{V} \to \mathbb{M}$ is a feedback policy that returns the optimal local planner (as opposed to optimal action) for each FIRM node. It is worth noting that the approximation $\pi^g(B^i) := \pi(b_c^i) \approx \pi(b)$ for all $b \in B^i$, results from the approximations in (5).

B. Overall policy π

The overall feedback policy π is generated by combining the policy π^g on the graph and the local controllers μ^{ij}s. Suppose at the k-th time step the active local controller and its corresponding stopping region are given by $\mu_*^k \in \mathbb{M}$ and $B_*^k \in \mathbb{V}$. They remain the same, i.e., $\mu_*^{k+1} = \mu_*^k$, and μ_*^k keeps generating control signals based on the belief at each time step, until the belief reaches the corresponding stopping region, B_*^k, where the higher level decision making is performed on the graph by π^g that chooses the next local controller. For example if the controller $\mu_*^k = \mu^{ij}$ is chosen, the stopping region is $B_*^k = B^j$. Once the state enters the stopping region B_*^k, the higher level decision making is performed on the graph by π^g that chooses the next local controller, i.e., $\mu_*^{k+1} = \pi^g(B_*^k)$. Thus, this hybrid policy $\pi : \mathbb{B} \to \mathbb{U}$ is stated as follows:

$$u_k = \pi(b_k) = \begin{cases} \mu_*^k(b_k), \ \mu_*^k = \pi^g(B_*^{k-1}), & \text{if } b_k \in B_*^{k-1} \\ \mu_*^k(b_k), \ \mu_*^k = \mu_*^{k-1}, & \text{if } b_k \notin B_*^{k-1} \end{cases} \tag{7}$$

Initial controller: Given the initial state is b_0, if b_0 is in one of the FIRM nodes, then we just choose the best local controller using π^g. However, if b_0 does not belong to any of the FIRM nodes, we first compute the mean state $\mathbb{E}[x_0]$ based on b_0 and then connect the $\mathbb{E}[x_0]$ to the PRM nodes using a set of edges denoted by $\mathcal{E}(0)$. Afterwards, for every $e_{ij} \in \mathcal{E}(0)$, we design a local controller μ^{ij}, and denote the set of newly added local controllers by $\mathbb{M}(0)$. Computing the transition cost $C(b_0, \mu^{ij})$, and probabilities $\mathbb{P}(B^j|b_0, \mu^{ij})$, and $\mathbb{P}(F|b_0, \mu^{ij})$, for invoking local controllers $\mu^{ij} \in \mathbb{M}(0)$ at b_0, we choose the best initial controller μ^0_* as:

$$\mu^0_*(\cdot) = \begin{cases} \arg \min\limits_{\mu^{ij} \in \mathbb{M}(0)} \{C(b_0, \mu^{ij}) + \mathbb{P}(B^j|b_0, \mu^{ij}) J^g(B^j) + \\ \qquad \mathbb{P}(F|b_0, \mu^{ij}) J^g(F)\}, & \text{if } \nexists l, b_0 \in B^l \\ \pi^g(B^l), & \text{if } \exists l, b_0 \in B^l \end{cases} \tag{8}$$

Summary: In summary, FIRM reduces the original POMDP into a tractable MDP on a finite subset of its nodes, which can be solved offline. Thus, performing the online phase of planning (or replanning) is computationally feasible. It is also shown that this construct is probabilistically complete [17].

IV. DFL-BASED FIRM

To exploit the generic FIRM framework, one has to determine proper (B, μ) pairs of the FIRM nodes and local controllers. Also, there has to be a way of computing transition costs and probabilities. In this section, we propose one such approach for nonholonomic systems, where we construct a FIRM in which belief stabilization is performed by compositing a Kalman Filter as the estimator and a Dynamic Feedback Linearization-based (DFL-based) controller as the belief controller. Accordingly, we design the reachable FIRM nodes B^j, and local planners μ^{ij}, required in (6). Then we discuss how the transition probabilities $\mathbb{P}^g(\cdot|B^i, \mu^{ij})$, and costs $C^g(B^i, \mu^{ij})$ in (6) are computed. Finally, we solve the corresponding FIRM MDP and provide the algorithms for offline construction of DFL-based FIRM and online planning with it. We start by defining notation needed for dealing with Gaussian beliefs.

Gaussian belief space: Let us denote the estimation vector by x^+, whose distribution is $b_k = p(x^+_k) = p(x_k|z_{0:k}; u_{0:k-1})$. 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, i.e., $b \equiv (\widehat{x}^+, P)$. Abusing notation, we denote this using an "equality relation", i.e., $b = (\widehat{x}^+, P)$.

Control with partial information: In partially-observable environments, decision making at the k-th time step is made based on the observation history $z_{0:k}$ up to time step k and the control history $u_{0:k-1}$ up to time step $k-1$. Thus, in a partially observable environment, the controller is a history-dependent mapping μ^{hist} from information history to the control space $\mu^{hist} : \mathbb{Z}^{k+1} \times \mathbb{U}^k \to \mathbb{U}$, i.e., $u_k = \mu^{hist}(z_{0:k}, u_{0:k-1})$. However, usually the design of μ^{hist} in partially-observable environments is decomposed into two tasks: (i) designing an estimator that generates belief based on information history $b_k = \overline{\tau}(z_{0:k}, u_{0:k-1})$ (or recursively $b_k = \tau(b_{k-1}, z_k, u_{k-1})$) and (ii) designing a belief controller (principally called "separated controller", e.g. [23]) that generates a control signal based on the available belief, here denoted by μ, i.e., $u_k = \mu(b_k)$. Therefore, we have $\mu^{hist} = \mu \circ \overline{\tau}$. In designing the LQG controller, this decomposition leads to designing a Kalman filter as the estimator and an LQR as the controller, whose combination preserves the optimality of the overall local controller. However, in most cases designing an optimal estimator and belief controller is a challenge, which leads to suboptimal solutions.

Unicycle control with partial information: In [15], [16] the optimal LQG controller is utilized for the construction of local controllers in FIRM. However, it is limited to models which are linearly stabilizable to a point in state space, which excludes the class of nonholonomic systems such as a unicycle model. Here, we combine the Kalman filter and a DFL-based controller to construct the μ^{hist} for the unicycle model with partial information. This construct is a suboptimal design for the controller in a partially-observable environment. However, it is very efficient in practice, and shows a promising solution for constructing a Feedback-based Information RoadMap (FIRM) for nonholonomic systems such as the unicycle model.

A. Estimator Design

For the state estimation we adopt the Kalman Filter (KF), which is commonly utilized for localizing unicycle robots [20]. Thus, the belief dynamic $b_{k+1} = \tau(b_k, u_k, z_{k+1})$ is a result of the Kalman filtering equations. In Appendix A, we review the Kalman filter and its stationary behaviours in detail. Here, we only discuss the limiting belief behaviour under the stationary KF (SKF).

State Space Model: Consider a PRM node $\mathbf{v}$. Let us denote the linear (linearized) system about the node $\mathbf{v}$ by the tuple $\Upsilon = (\mathbf{A}, \mathbf{B}, \mathbf{G}, \mathbf{Q}, \mathbf{H}, \mathbf{M}, \mathbf{R})$ that represents the following state space model:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{G}w_k, \quad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{9a}$$

$$z_k = \mathbf{H}x_k + \mathbf{M}v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}). \tag{9b}$$

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

Consider a stationary KF (SKF), which is designed to estimate the state of the system in (9). Let us also define the matrix $\check{\mathbf{Q}}$ such that $\mathbf{G}\mathbf{Q}\mathbf{G}^T = \check{\mathbf{Q}}\check{\mathbf{Q}}^T$. Now, consider the class of systems that satisfy the following property:

Property 2. *The pair $(\mathbf{A}, \check{\mathbf{Q}})$ is a controllable pair[19], and the pair $(\mathbf{A}, \mathbf{H})$ is an observable pair[19].*

Lemma 1. *Given Property 2, the estimation covariance under the SKF, designed for the system in* (9)*, converges to the matrix P_s, independent of its initial covariance:*

$$P_s = P_s^- - P_s^-\mathbf{H}^T(\mathbf{H}P_s^-\mathbf{H}^T + \mathbf{M}\mathbf{R}\mathbf{M}^T)^{-1}\mathbf{H}P_s^-, \tag{10}$$

where, P_s^- is the unique symmetric positive semidefinite solution of the following Discrete Algebraic Riccati Equation (DARE):

$$\begin{aligned} P_{k+1}^- = \mathbf{A}(P_k^- - P_k^-\mathbf{H}^T(\mathbf{H}P_k^-\mathbf{H}^T + \mathbf{M}\mathbf{R}\mathbf{M}^T)^{-1}\mathbf{H}P_k^-)\mathbf{A}^T \\ + \mathbf{G}\mathbf{Q}\mathbf{G}^T. \end{aligned} \tag{11}$$

Proof: See [19]. ∎

The estimation mean, however, evolves randomly, as it is a function of obtained observations. In SKF, the evolution of estimation mean is as follows:

$$\begin{aligned} \widehat{x}_{k+1}^+ = (I - \mathbf{K}\mathbf{H})\mathbf{A}\widehat{x}_k^+ + (I - \mathbf{K}\mathbf{H})\mathbf{B}u_k \\ + \mathbf{K}z_{k+1} + (I - \mathbf{K}\mathbf{H})(I - \mathbf{A})\mathbf{v}, \end{aligned} \tag{12}$$

where, $\mathbf{K} = P_s^-\mathbf{H}^T(\mathbf{H}P_s^-\mathbf{H}^T + \mathbf{M}\mathbf{R}\mathbf{M}^T)^{-1}$.

B. Belief controller (separated controller) design

The belief system is an underactuated system, i.e., the dimension of control space is less than the dimension of belief space. However, we can have full control of the estimation mean, while based on Lemma 1 the estimation covariance under the SKF, tends to P_s. As a result, if we design a feedback controller to control the estimation mean towards node $\mathbf{v}$ in state space, and assuming the system remains in the valid linearization region of the SKF (which is a reasonable assumption), then the belief will approach $b_c = (\mathbf{v}, P_s)$ in belief space. Considering a stopping region in belief space, whose interior contains b_c, the belief process under the feedback control will hit the stopping region in a finite time.

For a nonholonomic system such as the unicycle model, the system is not linearly controllable. Thus, we resort to the original nonlinear model, and utilize a Dynamic Feedback Linearization-based (DFL-based) controller to control the estimation mean for the nonholonomic unicycle model. The nonlinear form of (12) is:

$$\widehat{x}_{k+1}^+ = f(\widehat{x}_k^+, u_k, 0) + \mathbf{K}\widetilde{z}_{k+1}, \tag{13}$$

where, $\widetilde{z}_{k+1}$ is the observation error, defined as

$$\begin{aligned} \widetilde{z}_{k+1} &= h(f(x_k, u_k, w_k), v_{k+1}) - h(f(\widehat{x}_k^+, u_k, 0), 0) \\ &\approx \mathbf{H}\mathbf{A}\widehat{e}_k^+ + \mathbf{H}\mathbf{G}w_k + \mathbf{M}v_{k+1}, \end{aligned} \tag{14}$$

in which the function h is the observation model that maps the states to observations, i.e., $z_k = h(x_k, v_k)$, where v_k models the zero-mean Gaussian sensing noise. Random variable $\widehat{e}_k^+$ is the estimation error defined by $\widehat{e}_k^+ = x_k - \widehat{x}_k^+$. The approximation in (14) results from linearizing functions h and f. The important point is that the equation in the right hand side of (14) does not depend on u_k. Also, note that $\widehat{e}_k^+$ is unknown as it depends on the (unknown) true state x_k.

Kalman filters are vastly used for state estimation in non-holonomic systems and show great success in practice; thus, it is reasonable to assume that the estimation error $\widehat{e}_k^+$ is small. Therefore, the whole term $\mathbf{K}\widetilde{z}_{k+1}$ in (13) can be treated as a small control-independent perturbation affecting the system. Therefore, we adopt a controller to control the unicycle model, which is effectively robust to the noise injected by $\mathbf{K}\widetilde{z}_{k+1}$. The controller of choice is a DFL-based controller proposed in [3], as it offers a robust behaviour with respect to disturbances. Moreover, it provides an exponentially fast stabilization procedure, and has a natural, and smooth, transient performance. Experimental results verify the robustness of this controller to the above-mentioned disturbance $\mathbf{K}\widetilde{z}_{k+1}$.

To construct the DFL-based controller for unicycle model, first we transform the system state such that the target node coincides with the origin, i.e., if we denote $\mathbf{v} = (\mathbf{v}_\mathtt{x}, \mathbf{v}_\mathtt{y}, \mathbf{v}_\theta)$ and $\widehat{x}_k^+ = (\widehat{\mathtt{x}}_k^+, \widehat{\mathtt{y}}_k^+, \widehat{\theta}_k^+)$, we can transfer the system state as:

$$\begin{pmatrix} \breve{\mathtt{x}}_k^+ \\ \breve{\mathtt{y}}_k^+ \end{pmatrix} = \begin{pmatrix} \cos\mathbf{v}_\theta & -\sin\mathbf{v}_\theta \\ \sin\mathbf{v}_\theta & \cos\mathbf{v}_\theta \end{pmatrix}^{-1} \begin{pmatrix} \widehat{\mathtt{x}}_k^+ - \mathbf{v}_\mathtt{x} \\ \widehat{\mathtt{y}}_k^+ - \mathbf{v}_\mathtt{y} \end{pmatrix} \tag{15a}$$

$$\breve{\theta}_k^+ = \widehat{\theta}_k^+ - \mathbf{v}_\theta \tag{15b}$$

Now the controller has to drive the $\breve{x}_k^+ = (\breve{\mathrm{x}}_k^+, \breve{\mathrm{y}}_k^+, \breve{\theta}_k^+)$ to the origin.

Ignoring the disturbance term in estimation mean dynamics, and assuming $\widehat{x}_{k+1}^+ = f(\widehat{x}_k^+, u_k, 0)$, we compute the estimation mean derivative in the last time step, based on the previous control signal $u_{k-1} = (V_{k-1}, \omega_{k-1})$:

$$\begin{pmatrix} \dot{\breve{\mathrm{x}}}_{k-1}^+ \\ \dot{\breve{\mathrm{y}}}_{k-1}^+ \\ \dot{\breve{\theta}}_{k-1}^+ \end{pmatrix} = \begin{pmatrix} V_{k-1}\cos\breve{\theta}_{k-1}^+ \\ V_{k-1}\sin\breve{\theta}_{k-1}^+ \\ \omega_{k-1} \end{pmatrix} \tag{16}$$

Accordingly, we compute the intermediate controls:

$$u_1' = -k_{p1}\breve{\mathrm{x}}_{k-1}^+ - k_{d1}\dot{\breve{\mathrm{x}}}_{k-1}^+ \tag{17}$$

$$u_2' = -k_{p2}\breve{\mathrm{y}}_{k-1}^+ - k_{d2}\dot{\breve{\mathrm{y}}}_{k-1}^+ \tag{18}$$

where, as described in [3], the condition on the gains are $k_{pi}, k_{di} > 0$ for $i = 1, 2$ and also $k_{d1}^2 - 4k_{p1} = k_{d2}^2 - 4k_{p2} > 0$ and $k_{d2} - k_{d1} > 2(k_{d2}^2 - 4k_{p2})^{.5}$.

Finally, we compute the control signal at time step k:

$$V_k = V_{k-1} + (u_1'\cos\breve{\theta}_{k-1}^+ + u_2'\sin\breve{\theta}_{k-1}^+)\delta t \tag{19}$$

$$\omega_k = (u_2'\cos\breve{\theta}_{k-1}^+ - u_1'\sin\breve{\theta}_{k-1}^+)V_{k-1}^{-1} \tag{20}$$

Therefore, the controller, parametrized by the target point $\mathbf{v}$, receives current estimation mean $\widehat{x}_k^+$ and the previous control u_k to generate the next control u_{k+1}. We show this mapping by $u_{k+1} = \mu(\widehat{x}_k^+, u_k)$.

C. Designing FIRM Nodes $\{B^j\}$

FIRM Nodes: As mentioned, to construct a FIRM, we first construct its underlying PRM, characterized by its nodes and edges $\{\{\mathbf{v}^j\}, \{\mathbf{e}^{ij}\}\}$. Linearizing the system about the PRM node $\mathbf{v}^j$ results in a linear system $\Upsilon^j = (\mathbf{A}^j, \mathbf{B}^j, \mathbf{G}^j, \mathbf{Q}^j, \mathbf{H}^j, \mathbf{M}^j, \mathbf{R}^j)$:

$$x_{k+1} = \mathbf{A}^j x_k + \mathbf{B}^j u_k + \mathbf{G}^j w_k, \quad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^j) \tag{21a}$$

$$z_k = \mathbf{H}^j x_k + \mathbf{M}^j v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^j) \tag{21b}$$

Then, we design a stationary Kalman filter τ^j and a DFL-based belief controller μ^j corresponding to the system Υ^j. The controller μ^j is called the j-th *node-controller*. Accordingly, we choose the belief nodes B^j such that B^j is an ϵ-ball in belief space, centered at $b_c^j \equiv (\mathbf{v}^j, P_s^j)$, where P_s^j is the stationary covariance of the SKF designed for the system Υ^j, computed using (10). Therefore, B^j can be written as:

$$B^j = \{b \equiv (x, P) : \|x - \mathbf{v}^j\| < \delta_1, \|P - P_s^j\|_m < \delta_2\}, \tag{22}$$

where $\|\cdot\|$ and $\|\cdot\|_m$ denote suitable vector and matrix norms, respectively. The size of FIRM nodes are determined by δ_1 and δ_2, which are sufficiently small to satisfy the approximation in (5). Based on Lemma 1, the condition $\|P_k - P_s^j\|_m < \delta_2$ is satisfied after a deterministic finite time $k > N$ and based on the adopted DFL design, which has a global attractive behaviour in state space (and exponentially fast stabilization), the condition $\|x - \mathbf{v}^j\| < \delta_1$ is satisfied in a finite random time.

D. DFL-based Local Controllers μ^{ij}

The role of the local controller μ^{ij} is to drive the belief from the node B^i to node B^j. To construct the local controller μ^{ij}, we precede the node-controller μ^j with a so called edge-controller $\overline{\mu}_k^{ij}$.

Edge-controller: The main role of the edge-controller $\overline{\mu}_k^{ij}$ is that it takes the belief at node B^i and drives it to the vicinity of the node B^j, where it hands the system over to the node-controller μ^j, which in turn takes the system into a FIRM node B^i. As opposed to the point-stabilization procedure, if we linearize the unicycle model along the PRM edge $\mathbf{e}^{ij}$, where the nominal linear velocity is greater than zero, the unicycle is linearly controllable. As a result, we use a time-varying LQG controller to track the edge $\mathbf{e}^{ij}$.

Local controller: Thus, overall, the local controller μ^{ij} is the concatenation of the edge-controller $\overline{\mu}_k^{ij}$ and the node-controller μ^j. By this construct, the expected stopping time of the node-controller decreases as the initial belief of the node-controller is closer to the target node B^j, due to the usage of the edge-controllers.

E. Transition probabilities and costs

In general, it can be a computationally expensive task to compute the transition probabilities $\mathbb{P}(\cdot|B^i, \mu^{ij})$ and costs $C(B^i, \mu^{ij})$ associated with invoking local controller μ^{ij} at node B^i. However, owing to the offline construction of FIRM, this is not an issue. We utilize sequential Monte-carlo methods [24] to compute the collision and absorption probabilities. For the transition cost, we first consider estimation accuracy to find the paths, on which the estimator, and consequently, the controller can perform better. A measure of estimation error is the trace of estimation covariance. Thus, we use $\Phi^{ij} = \mathbb{E}[\sum_{k=1}^{\mathcal{T}} \text{tr}(P_k^{ij})]$, where P_k^{ij} is the estimation covariance at the k-th time step of the execution of local controller μ^{ij}. The outer expectation operator is useful in dealing with the Extended Kalman Filter (EKF), whose covariance is stochastic[25], [26]. Moreover, since we are also interested in faster paths, we take into account the corresponding mean stopping time, i.e., $\widehat{\mathcal{T}}^{ij} = \mathbb{E}[\mathcal{T}^{ij}]$, and the total cost of invoking μ^{ij} at B^i is considered as a linear combination of estimation accuracy and expected stopping time, with suitable coefficients ξ_1 and ξ_2.

$$C(B^i, \mu^{ij}) = \xi_1 \Phi^{ij} + \xi_2 \widehat{\mathcal{T}}^{ij}. \tag{23}$$

F. Construction of DFL-based FIRM and Planning With it

Algorithm 1 details the construction of FIRM. A crucial feature of FIRM is that it can be constructed offline and stored, independent of future queries. In the construction, presented in Algorithm 1, it is independent of the starting point of query. However, it can also be independent of the goal location, by postponing the DP equation solver (Line 16 of Algorithm 1) to the online phase (beginning of Algorithm 2). Moreover, owing to the reduction from the original POMDP to an n-state MDP on belief nodes, the FIRM MDP can be solved using standard DP techniques such as value/policy iteration to yield the optimal policy π^g that picks the optimal local planner $\mu^* = \pi^g(B^i)$ at each FIRM node B^i among all controller $\mu \in \mathbb{M}(\alpha, i)$.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. If the given initial belief b_0 does not belong to any B_i, we create a singleton set $B_0 = b_0$. To connect the B_0 to FIRM, we first, compute the expected value of the robot state, i.e. $\mathbb{E}[x_0]$ using its distribution b_0 and add the $\mathbb{E}[x_0]$ to the PRM nodes, and connect it to the PRM graph. The set of newly added edges going from $\mathbb{E}[x_0]$ to the nodes on PRM are called $\mathcal{E}(0)$. We design the local controllers associated with each edge in $\mathcal{E}(0)$ and call the set of them as $\mathbb{M}(0)$. Then choosing a local controller in $\mathbb{M}(0)$, the belief enters one of FIRM nodes, if no collision occurs. Thus, given the current node, we use policy π^g defined in (6) over FIRM nodes to find μ^*, and pick μ^* to move the robot into $B(\mu^*)$.Algorithm 2 illustrates this procedure.

Algorithm 1: Offline Construction of DFL-based FIRM

1 **input** : Free space map, $\mathbb{X}_{free}$
2 **output** : FIRM graph $\mathcal{G}$
3 Construct a PRM with nodes $\mathcal{V} = \{\mathbf{v}^j\}$, and edges $\mathcal{E} = \{\mathbf{e}^{ij}\}$, where $i, j = 1, \cdots, n$;
4 **forall the** *PRM nodes* $\mathbf{v}^j \in \mathcal{V}$ **do**
5 Design the node-controller (DFL-based) μ^j to stabilize the system to $\mathbf{v}^j$;
6 Compute the FIRM node center $b_c^j = (\mathbf{v}^j, P_s^j)$ using (10);
7 Construct FIRM node B^j using (22) centered at b_c^j;
8 Collect all FIRM nodes $\mathbb{V} = \{B^j\}$;
9 **forall the** $(B^i, \mathbf{e}^{ij})$ *pairs* **do**
10 Design the edge-controller $\overline{\mu}_k^{ij}$, as discussed in Section IV-D;
11 Construct the local controller μ_k^{ij} by concatenating edge-controller $\overline{\mu}_k^{ij}$ and node-controller μ_k^j;
12 Set the initial belief b_0 equal to the center of B^i, based on the approximation in (5);
13 Generate sample belief paths $b_{0:\mathcal{T}}$ and state paths $x_{0:\mathcal{T}}$ induced by controller μ^{ij} invoked at B^i;
14 Compute the transition probabilities $\mathbb{P}^g(F|B^i, \mu^{ij})$ and $\mathbb{P}^g(B^j|B^i, \mu^{ij})$ and transition costs $C^g(B^i, \mu^{ij})$;
15 Collect all local controllers $\mathbb{M} = \{\mu^{ij}\}$;
16 Compute cost-to-go J^g and feedback π^g over the FIRM by solving the DP in (6);
17 $\mathcal{G} = (\mathbb{V}, \mathbb{M}, J^g, \pi^g)$;
18 **return** $\mathcal{G}$;

V. Experimental Results

In this section, we illustrate the results of FIRM construction on a simple PRM. As a motion model, we consider the nonholonomic unicycle model whose kinematics are given in (1). As the observation model, in experiments, the robot is

Algorithm 2: Online Phase Algorithm (Planning with DFL-based FIRM)

input : Initial belief b_0, FIRM graph $\mathcal{G}$, Underlying PRM graph
if $\exists B^i \in \mathbb{V}$ *such that* $b_0 \in B^i$ **then**
  Choose the next local controller $\mu^{ij} = \pi^g(B^i)$;
else
  Compute $\mathbf{v}_0 = \mathbb{E}[x_0]$ based on b_0, and connect $\mathbf{v}_0$ to the PRM nodes. Call the set of newly added edges $\mathcal{E}(0) = \{\mathbf{e}^{0j}\}$;
  Design local planners associated with edges in $\mathcal{E}(0)$; Collect them in set $\mathbb{M}(0) = \{\mu^{0j}\}$;
  forall the $\mu \in \mathbb{M}(0)$ **do**
    Generate sample belief and state paths $b_{0:\mathcal{T}}$, $x_{0:\mathcal{T}}$ induced by taking μ at b_0;
    Compute the transition probabilities $\mathbb{P}(\cdot|b_0, \mu)$ and transition costs $C(b_0, \mu)$;
  Set $i = 0$; Choose the best initial local planner μ^{0j} within the set $\mathbb{M}(0)$ using (8);
while $B^i \neq B_{goal}$ **do**
  while $(\nexists B^j, s.t.,\ b_k \in B^j)$ *and "no collision"* **do**
    Apply the control $u_k = \mu_k^{ij}(b_k)$ to the system;
    Get the measurement z_{k+1} from sensors;
    if *Collision happens* **then return** Collision;
    Update belief as $b_{k+1} = \tau(b_k, \mu_k^{ij}(b_k), z_{k+1})$;
  Update the current FIRM node $B^i = B^j$;
  Choose the next local controller $\mu^{ij} = \pi^g(B^i)$;

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

$$
\begin{aligned}
{}^j z &= [\|{}^j\mathbf{d}\|, \text{atan2}({}^j d_2, {}^j d_1) - \theta]^T + {}^j v, \quad {}^j v \sim \mathcal{N}(\mathbf{0}, {}^j\mathbf{R}), \\
{}^j\mathbf{R} &= \text{diag}((\eta_r \|{}^j\mathbf{d}\| + \sigma_b^r)^2, (\eta_\theta \|{}^j\mathbf{d}\| + \sigma_b^\theta)^2),
\end{aligned} \tag{24}
$$

where ${}^j\mathbf{d} = [{}^j d_1, {}^j d_2]^T := [\mathtt{x}, \mathtt{y}]^T - L_j$. The uncertainty (standard deviation) of sensor readings increases as the robot gets farther from the landmarks. The parameters $\eta_r = \eta_\theta = 0.3$ determine this dependency, and $\sigma_b^r = 0.01$ meters and $\sigma_b^\theta = 0.5$ degrees are the bias standard deviations. A similar model for range sensing is used in [10]. The robot observes all N_L landmarks at all times and their observation noises are independent. Thus, the total measurement vector is denoted by $z = [{}^1 z^T, {}^2 z^T, \cdots, {}^{N_L} z^T]^T$ and due to the independence of measurements of different landmarks, the observation model for all landmarks can be written as $z = h(x) + v$, where $v \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{R} = \text{diag}({}^1\mathbf{R}, \cdots, {}^{N_L}\mathbf{R})$.

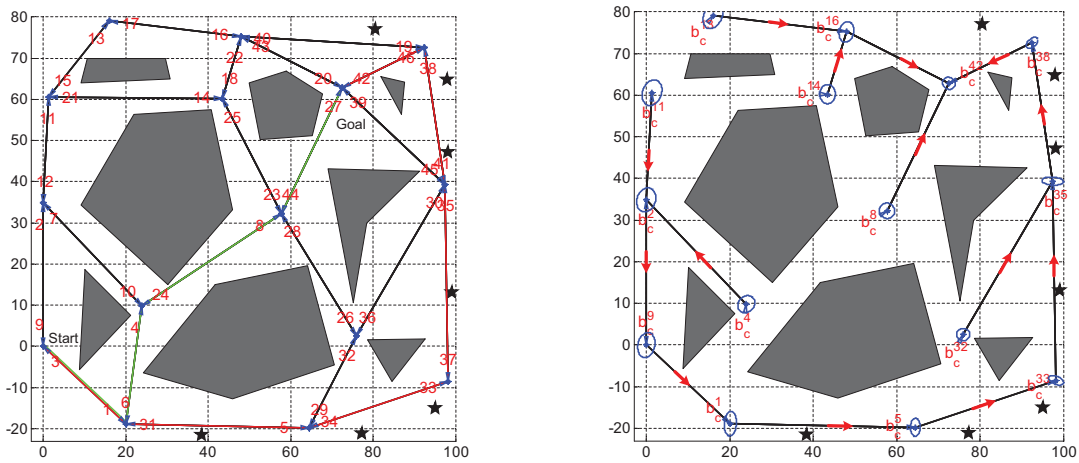
Figure 1(a) shows a simple environment with nine radio beacons (black stars). A PRM is constructed in the 3D space of $(\mathtt{x}, \mathtt{y}, \theta)$ with 46 nodes and 102 edges. PRM nodes are shown by blue triangles with their numbers in red. To construct the FIRM nodes, we first solve the DARE corresponding to each PRM node and design its corresponding DFL-based node-controller. Then, we form the node centers $b_c^j = (\mathbf{v}^j, P_s^j)$, some of which are drawn in Fig.1(a), by the 3σ ellipse (in blue) of the covariance P_s^j. Finally, to handle the error scale difference in position and orientation variables, we construct the FIRM nodes based on the component-wise version of (22), as follows:

$$
B^j = \{b = (x, P) |\ \ |x - \mathbf{v}^j| \dot{<} \epsilon, |P - P_s^j| \dot{<} \Delta\}, \tag{25}
$$

where $|\cdot|$ and $\dot{<}$ stand for the absolute value and component-wise comparison operators, respectively. We set $\epsilon = [0.8, 0.8, 5^\circ]^T$ and $\Delta = \epsilon\epsilon^T$ to quantify the B^j's.

After designing FIRM nodes and local controllers, the transition costs and probabilities are computed in the offline construction phase. Here, we use sequential weighted Monte-carlo based algorithms [24] to compute these quantities. In other words, for every (B^i, μ^{ij}) pair, we perform M runs and accordingly approximate the transition probabilities $\mathbb{P}^g(B^j|B^i, \mu^{ij})$, $\mathbb{P}^g(F|B^i, \mu^{ij})$, and costs $C^g(B^i, \mu^{ij})$. A similar approach is detailed in [16]. Table I shows these quantities along the best path resulting from the FIRM policy (see Fig.1(a)), where $M = 101$ and the coefficients in (23) are $\xi_1 = 0.98$ and $\xi_2 = 0.02$. Along edges where none of the 101 particles have collided with obstacles the collision probability is approximated by a value less than $1/101 = 0.0099$. The expected value and standard deviation of the time it takes for the controller to drive the belief into the target node is also reported in Table I. Table II shows the same quantities for the edges along the shortest path. Comparing these two tables, it is seen that the path returned by FIRM policy is safer, in the sense of collision probability, and more informative, in the sense of localization uncertainty, when compared to the shortest path.

Using the computed transition costs and probabilities in (6), we can solve the DP problem and compute the policy π^g on the graph. This process is performed only once offline, independent of the starting point of the query. Fig. 1(b) shows



(a) (b)

Fig. 1. A sample PRM, with numbered nodes. Seven landmarks are shown by black stars and obstacles are shown by gray polygons. (a) Node 9 is the start node and nodes 20, 27, 39, and 42 are goal nodes. Shortest path and the most-likely path under FIRM policy are shown in green and red, respectively. (b) The center of FIRM node, i.e., b_c is drawn for a selected number of PRM nodes. The feedback π^g is visualized for those FIRM nodes by red arrows.

TABLE I
COMPUTED COSTS FOR SEVERAL PAIRS OF NODE-AND-CONTROLLER USING 101 PARTICLES ALONG THE PATH RETURNED BY π^g.

(B_i,μ^{ij}) pair	$B_9,\mu^{9,1}$	$B_1,\mu^{1,5}$	$B_5,\mu^{5,33}$	$B_{33},\mu^{33,35}$	$B_{35},\mu^{35,38}$	$B_{38},\mu^{38,42}$
$\mathbb{P}^g(F\|B_i,\mu^{ij})$	15.3846%	7.6923%	<0.99%	<0.99%	<0.99%	15.3846%
Φ^{ij}	4.5967	1.9831	0.68936	1.6048	0.58705	0.53226
$\mathbb{E}[\mathcal{T}^{ij}]$	144.4545	217.3077	86.1538	161.2308	73	180.5455
$\sigma[\mathcal{T}^{ij}]$	66.7224	28.2396	9.109	5.7757	2.7433	40.6924

TABLE II
COMPUTED COSTS FOR SEVERAL PAIRS OF NODE-AND-CONTROLLER USING 101 PARTICLES ALONG THE SHORTEST PATH.

(B_i,μ^{ij}) pair	$B_9,\mu^{9,1}$	$B_1,\mu^{1,4}$	$B_4,\mu^{4,8}$	$B_8,\mu^{8,27}$
$\mathbb{P}^g(F\|B_i,\mu^{ij})$	15.3846%	38.4615%	46.1538%	38.4615%
Φ^{ij}	4.5967	2.0181	2.8001	2.1664
$\mathbb{E}[\mathcal{T}^{ij}]$	144.4545	168.375	127.2857	111.25
$\sigma[\mathcal{T}^{ij}]$	66.7224	50.3841	12.9192	38.1042

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 should be invoked, which in turn aims to take the robot belief to the next FIRM node.

Thus, the online part of planning is quite efficient, i.e., it only requires executing the controller and generating the control signal, which is done online. An important consequence of feedback π^g is efficient replanning. In other words, since π^g is independent of query, if due to some unmodeled large disturbance, the robot's belief deviates significantly from the planned path, it suffices to bring the robot back to the closest FIRM node and from there π^g drives the robot to the goal region as shown in Fig. 1(b). In Fig. 1(a), we also show the shortest path (green) and the resulting path under policy π^g (red). It can be seen that the path returned by the best policy detours from the shortest path to a path along which the filtering uncertainty is smaller, and on which it is easier for the controller to avoid collisions.

VI. CONCLUSION

In this paper, we propose a sampling-based roadmap in belief space for nonholonomic motion planning. The crucial feature of the roadmap is that it preserves the "optimal substructure property," and establishes a rigorous connection with the original POMDP problem that models the problem of planning under uncertainty. The method lends itself to the FIRM framework. Exploiting the properties of Kalman filter and DFL-based controllers, the local planners in FIRM are designed such that the belief node reachability is achieved for systems with nonholonomic constraints. As a result, the roadmap provides a sampling-based feedback solution for nonholonomic motion planning in belief space.

REFERENCES

[1] Z. Li and J. Canny, Eds., *Nonholonomic motion planning*. Kluwer Academic Press, 1993.

[2] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *IJRR*, vol. 25, no. 5-6, pp. 509–525, 2006.
[3] G. Oriolo, A. De Luca, and M. Vandittelli, "WMR control via dynamic feedback linearization: design, implementation, and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 835–851, 2002.
[4] K. Astrom, "Optimal control of markov decision processes with incomplete state estimation," *Journal of Mathematical Analysis and Applications*, vol. 10, pp. 174–205, 1965.
[5] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.
[6] L. Kavraki, P. Švestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
[7] N. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo, "OBPRM: An Obstacle-Basaed PRM for 3D workspaces," in *International Workshop on the Algorithmic Foundations of Robotics*, 1998, pp. 155–168.
[8] S. Lavalle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 378-400, 2001.
[9] D. Hsu, "Randomized single-query motion planning in expansive spaces," Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, CA, 2000.
[10] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *International Journal of Robotics Research*, vol. 28, no. 11-12, October 2009.
[11] V. Huynh and N. Roy, "icLQG: combining local and global optimization for control in information space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.
[12] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *IJRR*, vol. 30, no. 7, pp. 895–913, 2011.
[13] M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Control and Cybernetics*, vol. 35, no. 3, pp. 599–620, 2006.
[14] S. Chakravorty and S. Kumar, "Generalized sampling-based motion planners," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 41, no. 3, pp. 855–866, 2011.
[15] A. Agha-mohammadi, S. Chakravorty, and N. Amato, "FIRM: Feedback controller-based Information-state RoadMap -a framework for motion planning under uncertainty-," in *IROS*, 2011.
[16] ——, "Motion planning in belief space using sampling-based feedback planners," *Technical Report: TR11-007, Parasol Lab., CSE Dept., Texas A&M University*, 2011.
[17] ——, "On the probabilistic completeness of the sampling-based feedback motion planners in belief space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
[18] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, R. S. M. R. W. Brockett and H. J. Sussmann, Eds. Boston: Birkhauser, 1983, pp. 181–191.
[19] D. Bertsekas, *Dynamic Programming and Optimal Control: 3rd Ed.* Athena Scientific, 2007.
[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
[21] R. He, E. Brunskill, and N. Roy, "PUMA: Planning under uncertainty with macro-actions," in *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, 2010.
[22] A. Agha-mohammadi, S. Chakravorty, and N. Amato, "Sampling-based nonholonomic motion planning in belief space via dynamic feedback linearization-based FIRM," *Technical Report: TR12-004, Parasol Lab., CSE Dept., Texas A&M University*, 2012.
[23] P. R. Kumar and P. P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
[24] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. New York: Springer, 2001.
[25] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley and Sons, Inc, 2006.
[26] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC, 2004.

APPENDIX A
STATIONARY KALMAN FILTERING

In this section, we first discuss the system linearization around the planned point, and then discuss the Stationary Kalman Filtering (SKF) procedure. Consider the nonlinear partially-observable state-space equations of the system as follows:

$$x_{k+1} = f(x_k, u_k, w_k), \quad w_k \sim \mathcal{N}(0, Q_k) \tag{26a}$$
$$z_k = h(x_k, v_k), \quad v_k \sim \mathcal{N}(0, R_k) \tag{26b}$$

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

$$x^p = f(x^p, 0, 0) \tag{27}$$

The role of a closed-loop stochastic controller, during the state regulation, is compensating robot deviations from the desired point due to the noise effects and driving the robot close to the desired point. When only some imperfect information of the state x_k is available, an estimator can make the estimate x_k^+ of the state based on the available partial observations $z_{0:k}$ up to the current time. Accordingly, a controller can generate the control signal based on the estimated value of the state, i.e. belief. Based on separation principle [19], in linear system with Gaussian noises, the above minimization in is equivalent to performing two separate minimizations that lead to the LQG controller. However, in nonlinear systems, this procedure leads to a suboptimal design.

In the following, we first discuss the linearization of a nonlinear model and then we discuss how the stationary Kalman filtering for this linearized system.

Model linearization: Given a desired point x^p, we linearize the dynamics and observation model in Eq. (26), as follows:

$$x_{k+1} = f(x^p, 0, 0) + A_s(x_k - x^p) + B_s(u_k - 0) + G_s w_k, \quad w_k \sim \mathcal{N}(0, Q_s) \tag{28a}$$
$$z_k = h(x^p, 0) + H_s(x_k - x^p) + M_s v_k, \quad v_k \sim \mathcal{N}(0, R_s) \tag{28b}$$

where

$$A_k = \frac{\partial f}{\partial x}(x^p, 0, 0), \quad B_s = \frac{\partial f}{\partial u}(x^p, 0, 0), \quad G_s = \frac{\partial f}{\partial w}(x^p, 0, 0),$$
$$H_s = \frac{\partial h}{\partial x}(x^p, 0), \quad M_s = \frac{\partial h}{\partial v}(x^p, 0) \tag{29}$$

Now, let us define the following errors:

- Main error: $e_k = x_k - x^p$.
- SKF error (estimation error): $\widetilde{e}_k = x_k - \widehat{x}_k^+$, where $\widehat{x}_k^+ = \mathbb{E}[x_k^+] = \mathbb{E}[x_k|z_{0:k}]$.
- Belief controller 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 above linearized models as follows:

$$e_{k+1} = A_s e_k + B_s \delta u_k + G_s w_k \tag{30a}$$
$$\delta z_k = H_s e_k + M_s v_k \tag{30b}$$

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:

$$b_k = p(x_k^+) = p(x_k|z_{0:k}) = \mathcal{N}(\widehat{x}_k^+, P_k) \Leftrightarrow b_k \equiv (\widehat{x}_k^+, P_k) \tag{31}$$
$$\widehat{x}_k^+ = \mathbb{E}[x_k|z_{0:k}], \qquad P_k = \mathbb{C}[x_k|z_{0:k}] \tag{32}$$

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: prediction step and update step. In the prediction step, the mean and covariance of prior x_k^- is computed. For the system in Eq. (30) prediction step is:

$$\widehat{e}_{k+1}^- = A_s \widehat{e}_k^+ + B_s \delta u_k \tag{33}$$
$$P_{k+1}^- = A_s P_k^+ A_s^T + G_s Q_s G_s^T \tag{34}$$

In the update step, the mean and covariance of posterior x_k^+ is computed. For the error system in Eq. (30), the update step is:

$$K_k = P_k^- H_s^T (H_s P_k^- H_s^T + M_s R_s M_s^T)^{-1} \tag{35}$$
$$\widehat{e}_{k+1}^+ = \widehat{e}_{k+1}^- + K_{k+1}(\delta z_{k+1} - H_s \widehat{e}_{k+1}^-) \tag{36}$$
$$P_{k+1}^+ = (I - K_{k+1} H_s) P_{k+1}^- \tag{37}$$

Note that

$$\widehat{x}_k^+ = x^p + \widehat{e}_k^+, \qquad P_k = P_k^+ \tag{38}$$

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 [19]. The P_s^- can be computed by solving following DARE. Having P_s^-, stationary gain K_s and estimation covariance P_s is computed as follows:

$$P_s^- = G_s Q_s G_s^T + A_s (P_s^- - P_s^- H_s^T (H_s P_s^- H_s^T + M_s R_s M_s^T)^{-1} H_s P_s^-) A_s^T, \tag{39}$$
$$K_s = P_s^- H_s^T (H_s P_s^- H_s^T + M_s R_s M_s^T)^{-1}, \tag{40}$$
$$P_s = (I - K_s H_s) P_s^- \tag{41}$$