Periodic-Feedback Motion Planning in Belief Space for Nonholonomic and/or Nonstoppable Robots

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

Technical Report TR12-003
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 the Linear Quadratic Gaussian-based (LQG-based) instantiation of FIRM, it has been shown that for controllable linear systems, belief node reachability can be achieved using a stationary LQG controller. However, for nonholonomic and/or non-stoppable systems (whose velocity cannot be zero), belief reachability is a challenge. In this paper, we propose a novel method based on periodic trajectories, in which instead of stabilizing the belief to a predefined point the belief is stabilized to a periodic path in the belief space through which it is driven into predefined belief nodes. Therefore, the belief-node reachability is achieved and an instantionation of the FIRM framework that can handle the nonholonomic and/or non-stoppable systems is introduced. While taking obstacles into account, this method serves as an offline POMDP solver for motion planning in belief space. It is a query-independent solution, and preserves the optimal substructure property, required in dynamic programming solvers. Experiments illustrate the planning procedure on a unicycle model.

I. INTRODUCTION

The work presented here lies at the intersection of the research areas of nonholonomic motion planning, motion planning under uncertainty, and sampling-based motion planning.

Nonholonomic motion planning (NMP) deals with planning open-loop or feedback (closed-loop) plans for moving an object that is subject to nonholonomic constraints. A common example of a nonholonomic model is a unicycle model that is kinematically equivalent to a vast class of systems ranging from differential drive and synchro drive single-body robots [1], to steerable needles in surgery [2]. Let us consider two basic motion tasks: *point-to-point motion*, which deals with driving a moving object from a given state/configuration to another given state/configuration, and *trajectory following*, that deals with following a trajectory in state/configuration space. In contrast to the holonomic case, in nonholonomic systems, point-to-point motion is a much more difficult task than trajectory tracking [3]. A major challenge in performing a point-to-point motion task 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 object's motion and uncertainty in sensory readings, the state of 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 over the possible states of the system, and therefore decision making is performed in the space of these distributions, the so called belief space. Planning in the 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], [8] and tree-based methods such as methods in [9], [10], [8]. In dealing with MPUU, roadmap-based methods have a desirable feature: since the solution of POMDP is a feedback over whole 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 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 motion planning, 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 another given 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. To construct a query-independent roadmap in belief space, point-to-point motion in belief space is required. A number of pioneering methods have been proposed in [11], [12], [13] for applying sampling-based ideas for motion planning in belief space. However, one challenge they all face is they do not support point-to-point motion functionality in belief space. Thus, they do not preserve the "optimal substructure property"[14] needed in the solving DP equation or in Dijkstra's algorithm[15]. Therefore, the 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.*[11] and Huynh *et al.*[12] reuse a significant portion of the computation by using covariance factorization techniques to reduce the computation burden imposed by the query dependence.

In fully observable environments, Generalized PRM (GPRM)[16], [17] perform point-to-point motion under motion uncertainty. In partially observable environments, under motion and sensing uncertainty, the Feedback-based Information RoadMap (FIRM) [18], [19] 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 optimal substructure property is preserved on FIRM, and the connection between its solution and the original POMDP can be established rigorously [18], [19]. It is also shown to be probabilistically complete [20]. FIRM is an abstract framework for planning in belief space, and a Linear Quadratic Gaussian-based (LQG-based) instantiation of FIRM is reported in [18], [19]. 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.

As the main contribution of this work, we propose an instantiation of the FIRM method[18], [19] that can handle nonholonomic motion models. Additionally, the proposed method provides a sampling-based feedback motion planner for another type of systems, called non-stoppable systems here, i.e., systems whose velocity must be greater than a given threshold, for instance, aircraft models that cannot hover. A variant of PRM in state space is introduced, whose nodes lie on periodic trajectories, called orbit. Corresponding to each state node, we define a unique belief node and for each orbit, we design a feedback controller in such way that the belief node reachability is guaranteed independent of starting point, in the absence of obstacles. By inducing this reachability, the roadmap is constructed offline, independent of query, and can be used efficiently for replanning purposes. Also, the collision probabilities are rigorously incorporated in the construction of planner.

II. Controllability and Periodic Node PRM

An implicit assumption in road-map based methods such as PRM is that on every edge there exists a controller to drive the robot from the start node of the edge to the end node of the edge or to an ε-neighborhood of the end node, for some small $\varepsilon > 0$. 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 point is not controllable, and hence, a linear controller cannot stabilize the robot to the PRM nodes. Consider the discrete unicycle model:

$$x_{k+1} = f(x_k, u_k, w_k) = \begin{pmatrix} \mathsf{x}_k + (V_k + n_v)\delta t \cos\theta_k \\ \mathsf{y}_k + (V_k + n_v)\delta t \sin\theta_k \\ \theta_k + (\omega_k + n_\omega)\delta t \end{pmatrix}, \quad w_k \sim \mathcal{N}(0, \mathbf{Q}_k) \tag{1}$$

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 $w_k = (n_v, n_\omega)^T$. Linearizing this system about the point (node) $\mathbf{v} = (\mathsf{x}^p, \mathsf{y}^p, \theta^p)$, nominal control $u^p = (V^p, \omega^p)$, and zero noise, we get:

$$x_{k+1} = f(x_k, u_k, w_k) \approx f(\mathbf{v}, u^p, 0) + \mathbf{A}(x_k - \mathbf{v}) + \mathbf{B}(u_k - u^p) + \mathbf{G}w_k \tag{2}$$

where $\mathbf{A} = \frac{\partial f}{\partial x}(\mathbf{v}, u^p, 0)$, $\mathbf{B} = \frac{\partial f}{\partial u}(\mathbf{v}, u^p, 0)$,, $\mathbf{G} = \frac{\partial f}{\partial w}(\mathbf{v}, u^p, 0)$. Checking the rank of the controllability matrix of the linearized system (detailed in Appendix C), we get: $rank([\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}]) = 2 + \mathbb{I}(V^p > 0)$, where $\mathbb{I}$ is the indicator function, which is one if $V^p > 0$ and is zero, otherwise. Therefore, if the nominal control is zero, $u^p = (V^p, \omega^p)^T = (0, 0)^T$, which is the case when we stabilize the robot to a PRM node, the resulting linear system is not controllable, since $rank([\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}]) = 2 < 3$. Thus, a linear controller cannot stabilize the unicycle to a PRM node. Moreover, based on the necessary condition in Brockett's paper [21], 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 in (2) is controllable, using a linear stochastic controller such as the stationary LQG controller, one can drive the robot belief to the belief node [18], [19]. However, if the linearized system about the desired point is not controllable, the belief stabilization, if possible, is much more difficult than state stabilization. In addition to nonholonomic robots, controlling non-stoppable robots on a PRM roadmap is a challenging task since they have constraints on their controls and cannot reduce their velocity below a specific threshold u_{min}, and hence, stabilization is not feasible for them.

A. Periodic-node PRM

In this paper, we circumvent the problem of stabilization to roadmap nodes by designing a variant of PRM, called Periodic-Node PRM (PNPRM). Although there are different ways to address this problem in state space, the critical property of PNPRM is that it can be extended to the belief space and forms a roadmap in belief space such that the belief nodes are reachable without a point-stabilization process. In other words, in belief space, instead of reaching a belief node through a point-stabilizing controller, the belief nodes are designed along a periodic path, called belief-orbit, and are reached using orbit-stabilizing controllers. Circumventing the stabilization process also resolves the challenges in using state/belief space roadmaps for non-stoppable robots, such as aircraft.

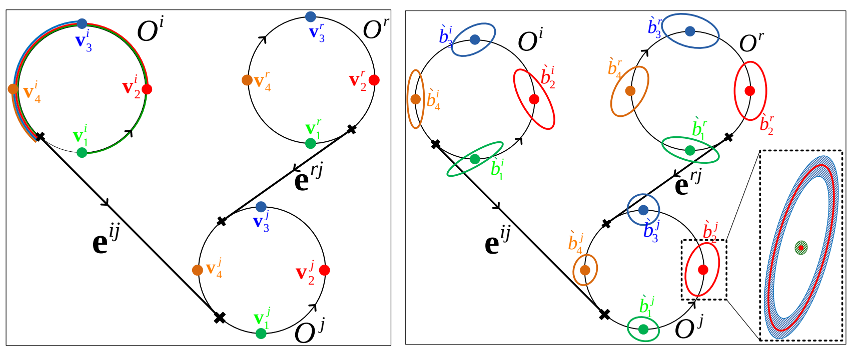
PNPRM is the underlying roadmap of Periodic LQG-based FIRM (PLQG-based FIRM) in belief space. Similar to traditional PRM, PNPRM also consists of nodes and edges. However, in PNPRM, the nodes lie on small T-periodic trajectories (trajectories with period T) in the state space, called orbits, that satisfy the control constraints and nonholonomic constraints of the moving robot. To construct a PNPRM, we first sample a set of orbits in the state space, and then on each orbit, a number of state nodes are selected. Let us denote the j-th orbit trajectory by O^j:

$$O^j := (x_k^{p^j}, u_k^{p^j})_{k\geq 0}, \text{ where } x_{k+1}^{p^j} = f(x_k^{p^j}, u_k^{p^j}, 0), \quad x_{k+T}^{p^j} = x_k^{p^j}, \quad u_{k+T}^{p^j} = u_k^{p^j} \tag{3}$$

The set of PNPRM nodes that are chosen on O^j is denoted by $\mathbb{V}^j = \{\mathbf{v}_1^j, \mathbf{v}_2^j, \cdots, \mathbf{v}_m^j\}$ where $\mathbf{v}_\alpha^j = x_{k_\alpha}^{p^j}$ for some $k_\alpha \in \{1, \cdots, T\}$. Edges in PNPRM do not connect nodes to nodes, but they connect orbits to orbits in a way that respect all the control constraints and nonholonomic constraints. Thus, the (i,j)-th edge denoted by $\mathbf{e}^{ij}$ connects O^i to O^j.

As a result, a node $\mathbf{v}_\alpha^i$ is connected to the node $\mathbf{v}_\gamma^j$ through concatenation of three path segments: *i)* the first segment is a part of O^i that connects $\mathbf{v}_\alpha^i$ to the starting point of $\mathbf{e}^{ij}$. This part is called *pre-edge* and is denoted by $\mathbf{e}^{i\alpha j}$, *ii)* the second segment is the edge $\mathbf{e}^{ij}$ itself that connects O^i to O^j, and *iii)* the third segment is a part of O^j that connects the ending point of $\mathbf{e}^{ij}$ to the $\mathbf{v}_\gamma^j$. This part is called *post-edge* and is denoted by $\mathbf{e}^{ij\gamma}$.

One form of constructing orbits is based on circular periodic trajectories, where the edges are the lines that are tangent to the orbits. This construct is a time-optimal construct in the deterministic case [22]. Figure 1(a) shows a simple PNPRM with three orbits O^i, O^r, and O^j. On each orbit four nodes are selected which are drawn with different colors. Edges $\mathbf{e}^{ij}$ and $\mathbf{e}^{rj}$ connect the corresponding orbits. Pre-edges $\mathbf{e}^{i\alpha j}$ are also shown on the i-th orbit for $\alpha = 1, \cdots, 4$ with four different colors.



(a) (b)

Fig. 1. (a) A simple PNPRM with three orbits, twelve nodes, and two edges. (b) $\grave{b}_\alpha^l = (\mathbf{v}_\alpha^l, \check{P}_{k_\alpha}^l)$ is the center of corresponding belief nodes, where $\check{P}_{k_\alpha}^l$'s are shown by their 3σ-ellipse. As an example of FIRM node, the magnified version of B_2^j, which is a small neighborhood centered at $\grave{b}_2^j$, is shown in the dotted box, where the blue shaded region depicts the covariance neighborhood and green shaded region depicts the mean neighborhood.

III. General planning in partially-observable environments

Partially Observable Markov Decision Processes (POMDPs) are the most general formulation for motion planning problem under motion and sensing uncertainties (In this paper, the environment map is assumed to be known). The solution of the POMDP problem is an optimal feedback (mapping) π, 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 the of the system state conditioned on the obtained information (measurements and controls) up to the k-th 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 [23], [24], whose solution π is computed by solving the following Dynamic Programming (DP) equation:

$$J(b) = \min_u \{c(b,u) + \int_{\mathbb{B}} p(b'|b,u) J(b') db'\}, \tag{4a}$$

$$\pi(b) = \arg\min_u \{c(b,u) + \int_{\mathbb{B}} p(b'|b,u) J(b') db'\}, \tag{4b}$$

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.

IV. FIRM FRAMEWORK BASED ON PNPRM

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 [18], [19], and consider the union of pre-defined FIRM nodes on each orbit as the stopping region of the local planner, and accordingly, transform the POMDP in (4a),(4b) into a tractable MDP problem.

In this section, we first describe the elements we need to construct a FIRM. Then, we show that for the given admissible class of admissible policies, we can transform the POMDP into a belief SMDP, which is still computationally intractable. Then the computationally tractable FIRM MPD serves as an arbitrarily good approximation of this intractable belief SMDP, over FIRM nodes (a subset of belief space). We discuss this approximation first in obstacle-free case and then we add the obstacles to the planning framework. We discuss the quality of solution obtained by FIRM, via its success probability, and provide a generic algorithm for planning with FIRM.

A. Feedback Controllers and Reachability

Feedback controllers in partially observable spaces: In partially observable environments, at each stage, the decision making process is performed based on the belief at that stage. Thus, a feedback controller in partially observable spaces is a mapping from belief space to the control space, i.e., $\mu(\cdot):\mathbb{B}\to\mathbb{U}$. Generating controls based on a given controller μ, the belief evolves according to a Markov chain whose one-step transition density function is denoted by $p^{\mu}(b'|b) := p(b'|b,\mu(b))$. Thus, the feedback controller μ essentially induces the Markov chain $p^{\mu}(b'|b)$ over the belief space $\mathbb{B}$. In a similar way, $\mathbb{P}_1(\cdot|b,\mu):\mathscr{B}_{\mathbb{B}}\to[0,1]$ is defined as the probability measure over the belief space, induced by the local controller μ after one step, starting from the belief b. The set $\mathscr{B}_{\mathbb{B}}$ is the sigma-algebra on the belief space $\mathbb{B}$.

Stopping region: We call region $B\subset\mathbb{B}$ a stopping region of controller μ if the controller stops executing as the belief reaches the region B, i.e., $\mathbb{P}_1(b|b,\mu)=1$ for all $b\in B$. In a similar way, $\mathbb{P}_n(\cdot|b,\mu):\mathscr{B}_{\mathbb{B}}\to[0,1]$ for the controller μ with stopping region B is defined as the probability measure over the belief space, induced by the local controller μ after n steps, starting from the belief b and conditioned on the event that belief process has not stopped, i.e., $\mathbb{P}_n(B|b,\mu) := \Pr(b_n\in B|b_k\notin B,\ \forall k=0,\cdots,n-1,\ b_0=b,\ \mu)$.

Probability of landing in the stopping region: Consider the controller μ that starts executing from belief b and stops executing when the belief enters region B. Thus, we can define $p(b'|b,\mu)$ as the pdf over belief space, when the controller μ, invoked at b, stops executing. Similarly, $\mathbb{P}(\cdot|b,\mu)$ represents the probability measure induced by μ invoked at b, when controller stops executing. Thus, the probability of landing in stopping region B in a finite time is $\mathbb{P}(B|b,\mu)$, which is computed as:

$$\mathbb{P}(B|b,\mu)=\sum_{n=0}^{\infty}\mathbb{P}_n(B|b,\mu) \tag{5}$$

Reachability and proper feedback policy: The stopping region B is reachable under controller μ from b in a finite time, if $\mathbb{P}(B|b,\mu)=1$. The controller μ is called proper with respect to a stopping region B, over the region B_0, if for all $b\in B_0$, we have $\mathbb{P}(B|b,\mu)=1$. This can be considered as an extension of the "proper policy" definition in [23]. In this case, the controller and stopping region pair, i.e., (μ,B), is called a proper pair over the region B_0. The largest B_0, i.e., the set of all beliefs from which the stopping region B is reachable under μ is called the effective region of pair (μ,B) and is shown by $\mathbb{B}_{(\mu,B)}$. In other words,

$$\mathbb{B}_{(\mu,B)}=\{b\in\mathbb{B}\ :\ \mathbb{P}(B|b,\mu)=1\},\quad \text{if}\quad \mathbb{X}_{free}=\mathbb{X} \tag{6}$$

It is worth noting that $B\subset\mathbb{B}_{(\mu,B)}$, and in practical cases, B is much smaller than $\mathbb{B}_{(\mu,B)}$.

B. The obstacle-free FIRM MDP with Periodic Nodes

In this section we assume there are no obstacles in the state space, i.e. $\mathbb{X}_{free}=\mathbb{X}$. Consider the DP in (4), through which the cost-to-go function can be computed for the belief MDP problem. It is well known that this DP equation is not tractable over continuous spaces. Thus, inspired by sampling-based motion planning methods, in FIRM, we aim to sample the belief space and compute the cost-to-go of the sampled beliefs instead of cost-to-go for all beliefs in the belief space. Then, we show that this sampling in belief space leads to an intractable belief SMDP problem, which can be arbitrarily well approximated by a tractable FIRM MDP on the FIRM nodes.

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

FIRM nodes: FIRM nodes are disjoint small sets in the belief space. Corresponding to each PNPRM node, we have a FIRM node. The FIRM node corresponding to $\mathbf{v}_{\alpha}^{i}$ is denoted by $B_{\alpha}^{i} \subset \mathbb{B}$, whose construction is discussed in Section V. FIRM nodes are measurable sets, i.e. $B_{\alpha}^{i} \in \mathscr{B}_{\mathbb{B}}$. The set of FIRM nodes that correspond to the i-th orbit is denoted by $\mathbb{V}^{i} = \{B_{\alpha}^{i}\}_{\alpha=1}^{m}$ and the set of all FIRM nodes are the union of these nodes over all orbits, which is denoted by $\mathbb{V} = \cup_{i=1}^{n} \mathbb{V}^{i} = \cup_{i=1}^{n} \{B_{\alpha}^{i}\}_{\alpha=1}^{m}$.

Local planners: First, it is worth mentioning that the words "local planner" and "local controller" are used interchangeably in this manuscript. The local controller $\mu(\cdot) : \mathbb{B} \to \mathbb{U}$ is a feedback controller. The role of (i_{α}, j)-th local controller, denoted by $\mu^{\alpha,ij}$, is to take a belief from the FIRM node B_{α}^{i} to a FIRM node on orbit O^{j}. We denote the set of local controllers by $\mathbb{M} = \{\mu^{\alpha,ij}\}$. For $m > 1$, the target node of the local controller $\mu^{\alpha,ij}$ is random, but is restricted to the nodes in $\mathbb{V}^{j}$. In other words, $\mu^{\alpha,ij}$ generates controls based on the belief at each stage, until the belief reaches the region $\cup_{\gamma} B_{\gamma}^{j}$. Thus, the local controller $\mu^{\alpha,ij}$ has to be a proper policy with respect to the region $\cup_{\gamma} B_{\gamma}^{j}$ over initial region B_{α}^{i} as stated in the following property:

Property 1. (Reachability property of FIRM): *For every PNPRM edge $\mathbf{e}_{ij}$ and an $\alpha \in \{1, 2, \cdots, m\}$, there exists a controller $\mu^{\alpha,ij}$, such that the pair $(\mu^{\alpha,ij}, \cup_{\gamma} B_{\gamma}^{j})$ is a proper pair over region B_{α}^{i}, i.e., for all $b \in B_{\alpha}^{i}$ we have $\mathbb{P}(\cup_{\gamma} B_{\gamma}^{j} | b, \mu^{\alpha,ij}) = 1$, in the absence of obstacles. In other words, in the obstacle-free environment, the feedback controller $\mu^{\alpha,ij}(\cdot)$ can drive the belief state from B_{α}^{i} into a $B \in \mathbb{V}^{j}$ in a finite time with probability one.*

Satisfying reachability property of FIRM: In [18], [19] 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, if the FIRM nodes are designed appropriately, we prove in Section V that the periodic LQG controller can accomplish the above reachability condition (Property 1), for nonholonomic models such as a unicycle that are not linearly controllable about a fixed point.

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

Local controllers versus Macro-actions: 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 important to note that local controllers are not a sequence of open-loop actions (macro-actions)[25], [26], but rather a sequence of closed-loop policies (macro-policies), each is a mapping from belief space to the continuous control space. Using macro-actions results in an open-loop policy, that cannot compensate the belief state deviation from the planned path, however, under local-controllers (macro-policies), the effect of noise can be compensated due to the feedback nature of the controllers and thus, the belief can be steered towards a stopping region.

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 $C(b, \mu^{\alpha,ij})$ represents the cost of invoking local controller $\mu^{\alpha,ij}(\cdot)$ at belief state b, i.e.,

$$C(b, \mu^{\alpha,ij}) = \sum_{t=0}^{\mathscr{T}^{\alpha,ij}} c(b_t, \mu^{\alpha,ij}(b_t) | b_0 = b), \tag{7}$$

$$\mathscr{T}^{\alpha,ij}(b) = \inf_{t} \{t | b_t \in \cup_{\gamma} B_{\gamma}^{j}, b_0 = b\}. \tag{8}$$

where $\mathscr{T}^{\alpha,ij}$ is a random stopping time denoting the time at which the belief state enters the region $\cup_{\gamma} B_{\gamma}^{j}$ under the controller $\mu^{\alpha,ij}$. 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 $\mathbb{P}(B_{\gamma}^{j} | b, \mu^{\alpha,ij})$ is the transition probability from b to B_{γ}^{j} under the local planner $\mu^{\alpha,ij}$.

POMDP to belief SMDP transformation: According to this construction, the original POMDP, formulated using DP in (4), is now turned into a Semi-Markov Decision Process (SMDP) [27] in the belief space, which is also called *belief SMDP*, on the continuous regions B_{α}^{i}. To see this, note that if the pair $(\cup_{\gamma} B_{\gamma}^{j}, \mu^{\alpha,ij})$ is proper, DP formulation corresponding to this belief SMDP is:

$$J(b) = \min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C(b,\mu^{\alpha,ij}) + \int_{\cup_\gamma B^j_\gamma} p(b'|b,\mu^{\alpha,ij})J(b')db',\ \forall b\in B^i_\alpha,\ \forall i,\alpha \tag{9}$$

$$= \min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C(b,\mu^{\alpha,ij}) + \sum_{\gamma=1}^{m}\int_{B^j_\gamma} p(b'|b,\mu^{\alpha,ij})J(b')db',\ \forall b\in B^i_\alpha,\ \forall i,\alpha. \tag{10}$$

The integration over whole belief space in (4) is reduced to the integration over regions $\cup_\gamma B^j_\gamma$ in (9) because according to Property 1, when $\mu^{\alpha,ij}$ stops executing, the belief is in $\cup_\gamma B^j_\gamma$ with probability one.

Roadmap level transitions: The DP in (9), though computationally more tractable than the original POMDP, is defined on the continuous neighborhoods B_i and thus, still formidable to solve. However, for sufficiently small B_i's, the cost-to-go of all the beliefs in B_i, are approximately equal. Similar statement holds for the incremental cost. Thus, we can define the transition cost and probabilities $C^g:\mathbb{V}\times\mathbb{M}\rightarrow\mathbb{R}$ and $\mathbb{P}^g:\mathbb{V}\times\mathbb{V}\times\mathbb{M}\rightarrow[0,1]$ on the FIRM graph, i.e., over the finite space $\mathbb{V}$, such that $\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij})$ is the transition probability from B^i_α to B^j_γ under the local planner $\mu^{\alpha,ij}$. Similarly, $C^g(B^i_\alpha,\mu^{\alpha,ij})$ denotes the cost of invoking local planner $\mu^{\alpha,ij}$ at FIRM node B^i_α. Accordingly, $J^g:\mathbb{V}\rightarrow\mathbb{R}$ 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_α's:

$$\forall b\in B^i_\alpha,\forall i,\alpha,j\quad \begin{cases} J^g(B^i_\alpha):=J(\grave{b}^i_\alpha)\approx J(b),\\ C^g(B^i_\alpha,\mu^{\alpha,ij}):=C(\grave{b}^i_\alpha,\mu^{\alpha,ij})\approx C(b,\mu^{\alpha,ij}),\\ \mathbb{P}^g(\cdot|B^i_\alpha,\mu^{\alpha,ij}):=\mathbb{P}(\cdot|\grave{b}^i_\alpha,\mu^{\alpha,ij})\approx\mathbb{P}(\cdot|b,\mu^{\alpha,ij}),\end{cases} \tag{11}$$

where $\grave{b}^i_\alpha$ is a point in B^i_α, for example, its center, if B^i_α is a ball. The approximation essentially states that any belief in the region B^i_α is represented by $\grave{b}^i_\alpha$ for the purpose of decision making.

Obstacle-free FIRM MDP: Given this approximation, the DP equation in (9) becomes:

$$\begin{aligned} J^g(B^i_\alpha) &= \min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + \sum_{\gamma=1}^{m}\int_{B^j_\gamma} p(b'|b,\mu^{\alpha,ij})J^g(B^j_\gamma)db',\ \forall b\in B^i_\alpha,\ \forall\alpha,i,j\\ &= \min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + \sum_{\gamma=1}^{m} J^g(B^j_\gamma)\mathbb{P}(B^j_\gamma|b,\mu^{\alpha,ij}),\ \forall b\in B^i_\alpha,\ \forall\alpha,i,j\\ &= \min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + \sum_{\gamma=1}^{m} J^g(B^j_\gamma)\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij}),\ \forall\alpha,i,j \end{aligned} \tag{12}$$

Note that in the absence of obstacels, Property 1 implies that $\sum_{\gamma=1}^{m}\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij})=1$.

Equation (12) is an arbitrarily good approximation to the original DP in (9) given that the functions $C(b,\mu)$ and $\mathbb{P}(B|b,\mu)$ are smooth with respect to their arguments (i.e., at least continuous), and given that the belief nodes B_i are sufficiently small. The approximation essentially states that any belief in the region B^i_α is represented by $\grave{b}^i_\alpha$ for the purpose of decision making. In other words, the approximation $\pi^g(B^i_\alpha):=\pi(\grave{b}^i_\alpha)\approx\pi(b),\forall b\in B^i_\alpha,\forall i,\alpha$ results from the approximations in (11), which leads to:

$$J^g(B^i_\alpha) = \min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + \sum_{\gamma=1}^{m} J^g(B^j_\gamma)\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij}),\ \forall\alpha,i,j \tag{13a}$$

$$\pi^g(B^i_\alpha) = \arg\min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + \sum_{\gamma=1}^{m} J^g(B^j_\gamma)\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij}),\ \forall\alpha,i,j \tag{13b}$$

Thus, the original POMDP over the entire belief space, becomes an MDP in (13) defined on the finite set of FIRM nodes $\mathbb{V}=\{B^i_\alpha\}$, where $i\in\{1,2,\cdots,n\}$, $\alpha\in\{1,2,\cdots,m\}$. We call the MDP in (13), the FIRM MDP in the absence of obstacles. It is worth noting that $J^g(\cdot):\mathbb{V}\rightarrow\mathbb{R}$ is the cost-to-go function over the FIRM nodes, which assigns a cost-to-go for every FIRM node B^i_α and the mapping $\pi^g(\cdot):\mathbb{V}\rightarrow\mathbb{M}$ is a mapping over the FIRM graph, from FIRM nodes into the set of local controllers that computes the optimal best local controller that has to be taken at any FIRM node. Given $C^g(B,\mu)$ for all (B,μ) pairs and $\mathbb{P}^g(B'|B,\mu)$ for all (B',B,μ) triples, the DP equation in (13) can be solved *offline* using standard DP techniques such as value/policy iteration to yield a feedback policy π^g over the FIRM nodes B^i_α.

C. Incorporating Obstacles into FIRM MDP

In the presence of obstacles, we cannot assure that the local controller $\mu^{\alpha,ij}(\cdot)$ can drive any $b \in B^i_\alpha$ into $\cup_\gamma B^j_\gamma$ with probability one. Instead, we have to specify the failure probabilities that the robot collides with an obstacle. Let us denote the failure set on $\mathbb{X}$ by F (i.e., $F = \mathbb{X} - \mathbb{X}_{free}$).

Let us generalize the $\mathbb{P} : \mathscr{B}_\mathbb{B} \to [0,1]$ to $\mathbb{P} : \{\mathscr{B}_\mathbb{B} \cup F\} \to [0,1]$ that also measure the failure probability, such that $\mathbb{P}(F|b,\mu^{\alpha,ij})$ denote the probability that under local controller $\mu^{\alpha,ij}$ the system enters the failure set F before it enters the region $\cup_\gamma B^j_\gamma$, given that the initial belief is b. Similarly, we generalize $\mathbb{P}^g$ such that $\mathbb{P}^g(F|B^i_\alpha,\mu) := \mathbb{P}(F|\grave{b}^i,\mu)$. Again, given function $\mathbb{P}(\cdot|b,\mu)$ is smooth and given that the sets B^i_α are suitably small, we can make the approximation $\mathbb{P}^g(\cdot|B^i_\alpha,\mu) := \mathbb{P}(\cdot|\grave{b}^i,\mu) \approx \mathbb{P}(\cdot|b,\mu)$ for all $b \in B^i_\alpha$ and for all i. Finally, we generalize the cost-to-go function by adding F to its input set, i.e. $J^g : \{\mathbb{V}, F\} \to \mathbb{R}$, such that $J^g(F)$ is a user-defined suitably high cost for hitting obstacles. Therefore, we can modify (13) to incorporate obstacles in the state space, by repeating the procedure in the previous subsection to get the FIRM MDP in presence of obstacles:

$$
\begin{aligned}
J^g(B^i_\alpha) = &\min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + J^g(F)\mathbb{P}^g(F|B^i_\alpha,\mu^{\alpha,ij}) \\
&+ \sum_{\gamma=1}^{m} J^g(B^j_\gamma)\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij}),\ \forall\alpha,i,j \qquad (14a)\\
\pi^g(B^i_\alpha) = \arg&\min_{\mu^{\alpha,ij}\in\mathbb{M}(i,\alpha)} C^g(B^i_\alpha,\mu^{\alpha,ij}) + J^g(F)\mathbb{P}^g(F|B^i_\alpha,\mu^{\alpha,ij}) \\
&+ \sum_{\gamma=1}^{m} J^g(B^j_\gamma)\mathbb{P}^g(B^j_\gamma|B^i_\alpha,\mu^{\alpha,ij}),\ \forall\alpha,i,j \qquad (14b)
\end{aligned}
$$

It is assumed that the system can enter the goal region or the failure set and remain there subsequently without incurring any additional cost. Thus, all that is required to solve the above DP equation are the values of the costs $C^g(B^i_\alpha,\mu^{\alpha,ij})$ and transition probability functions $\mathbb{P}^g(\cdot|B^i_\alpha,\mu^{\alpha,ij})$. Thus, the main difference from the obstacle free case is the addition of a "failure" state to the FIRM MDP along with the associated probabilities of failure from the various nodes B^i_α.

D. Overall policy π

The overall feedback 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 shown by $\mu^k_* \in \mathbb{M}$ and $B^k_* \in \mathbb{V}$. They remain the same, i.e., $\mu^{k+1}_* = \mu^k_*$, and 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 over the graph by π^g that chooses the next local controller. For example if the controller $\mu^k_* = \mu^{\alpha,ij}$ is chosen, the stopping region is $B^k_* = \cup_\gamma B^j_\gamma$. Once the state enters the stopping region B^k_*, the higher level decision making is performed over graph by π^g that chooses the next local controller, i.e., $\mu^{k+1}_* = \pi^g(B^k_*)$. Thus, this hybrid policy is stated as follows:

$$
\pi : \mathbb{B} \to \mathbb{U}, \quad u_k = \pi(b_k) = \begin{cases} \mu^k_*(b_k),\ \mu^k_* = \pi^g(B^{k-1}_*), & \text{if } b_k \in B^{k-1}_* \\ \mu^k_*(b_k),\ \mu^k_* = \mu^{k-1}_*, & \text{if } b_k \notin B^{k-1}_* \end{cases} \qquad (15)
$$

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 PNPRM orbits using a set of edges denoted by $\mathscr{E}(0)$. Afterwards, for every $e_{ij} \in \mathscr{E}(0)$, 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 $\mathbb{P}(B^j_\gamma|b_0,\mu^{ij})$, 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:

$$
\mu^0_*(\cdot) = \begin{cases} \arg\min_{\mu^{ij}\in\mathbb{M}(0)} \{C(b_0,\mu^{ij}) + \sum_{\gamma=1}^{m} \mathbb{P}(B^j_\gamma|b_0,\mu^{ij})J^g(B^j_\gamma) + \\ \qquad \mathbb{P}(F|b_0,\mu^{ij})J^g(F)\},\ \forall i,j, & \text{if } \nexists l,\alpha, \text{s.t. } b_0 \in B^l_\alpha \\ \pi^g(B^l_\alpha), & \text{if } \exists l,\alpha, \text{s.t. } b_0 \in B^l_\alpha \end{cases} \qquad (16)
$$

It is worth noting that computing the μ_*^0 is the only part of computation that depends on the initial belief and has to be reproduced for every query with a new initial belief. However, it is computationally feasible in real-time.

Generic FIRM Algorithms: The generic algorithms for offline construction of FIRM and online planning on FIRM are presented in Algorithms 1 and 2, respectively. The concrete instantiations of these algorithms for PLQG-based FIRM are given in the next section.

Algorithm 1: Generic Construction of FIRM with periodic nodes (Offline)

1 Construct a PNPRM with orbits $\mathscr{O} = \{O^j\}_{j=1}^n$, nodes $\mathscr{V} = \{\mathbf{v}_\alpha^i\}_{\alpha=1,i=1}^{m,n}$, and edges $\mathscr{E} = \{\mathbf{e}_{ij}\}_{i,j=1}^n$;
2 For each PNPRM node-edge pair $\mathbf{v}_\alpha^i, \mathbf{e}_{ij}$, design a local controller $\mu^{\alpha,ij}$;
3 Construct m FIRM nodes $\{B_1^j, \cdots, B_m^j\}$ corresponding to $\{\mathbf{v}_1^j, \cdots, \mathbf{v}_m^j\}$ such that $(\cup_\gamma B_\gamma^j, \mu^{\alpha,ij})$ is a proper pair.
4 For each B_α^i and $\mu^{\alpha,ij} \in \mathbb{M}(i,\alpha)$, compute the transition costs and probabilities associated with taking $\mu^{\alpha,ij}$ at B_α^i, i.e., $C^g(B_\alpha^i, \mu^{\alpha,ij})$, $\mathbb{P}^g(F|B_\alpha^i, \mu^{\alpha,ij})$ and $\mathbb{P}^g(B_\gamma^j|B_\alpha^i, \mu^{\alpha,ij})$ for all γ;
5 Solve the FIRM MDP in (14) to compute feedback π^g over FIRM nodes, and compute the π accordingly.

Algorithm 2: Generic planning (or replanning) on FIRM (Online)

1 Given an initial belief b_0, invoke the controller $\mu_0(\cdot)$ in (16), to absorb the robot into some FIRM node B_α^i;
2 Given the system is in set B_α^i, invoke the higher level feedback policy π^g to choose the lower level local feedback controller $\mu^{\alpha,ij}(\cdot) = \pi^g(B_\alpha^i)$;
3 Let the local controller $\mu^{\alpha,ij}(\cdot)$ execute until absorption into the B_γ^j for some $\gamma \in \{1, \cdots, m\}$ or failure;
4 Repeat steps 2-3 until absorption into the goal node B_{goal} or failure.

E. 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 (14b). The DP in (14b) has $mn+1$ states $\{B_1^1, \cdots, B_m^1, B_1^2, \cdots, B_m^2, \cdots, B_1^n, \cdots, B_m^n, F\}$. Let us denote these $mn+1$ states by $\{S_1, S_2, \cdots, S_{mn+1}\}$ that can be decomposed into three disjoint classes: the goal class $S_1 = B_{goal}$, the failure class $S_2 = F$, and the transient class $\{S_3, S_4, \cdots, S_{n+1}\} = \mathbb{V} \setminus B_{goal}$. 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[28]:

$$\mathscr{P} = \begin{bmatrix} \mathscr{P}_{goal} & 0 & 0 \\ 0 & \mathscr{P}_f & 0 \\ \mathscr{R}_{goal} & \mathscr{R}_f & \mathscr{Q} \end{bmatrix}. \tag{17}$$

The (i,j)-th component of $\mathscr{P}$ represents the transition probability from S_j to S_i, i.e., $\mathscr{P}[i,j] = \mathbb{P}^g(S_i|S_j, \pi^g(S_j))$. Moreover $\mathscr{P}_{goal} = \mathbb{P}^g(S_1|S_1, \pi^g(S_1)) = 1$ and $\mathscr{P}_f = \mathbb{P}^g(F|F,\cdot) = 1$, since goal and failure classes are the absorbing recurrent classes, i.e., the system stops once it reaches the goal or it fails. $\mathscr{Q}$ is a matrix that represents the transition probabilities between belief nodes B_i in transient class, whose (i,j)-th element is $\mathscr{Q}[i,j] = \mathbb{P}^g(S_{i+2}|S_{j+2}, \pi^g(S_{j+2}))$. Vectors $\mathscr{R}_{goal}$ and $\mathscr{R}_f$ are $(n-1) \times 1$ vectors that represent the probability of the transient nodes $\mathbb{V} \setminus B_{goal}$ getting absorbed into the goal node and the failure set, respectively, i.e., $\mathscr{R}_{goal}[j] = \mathbb{P}^g(S_1|S_{j+2}, \pi^g(S_{j+2}))$ and $\mathscr{R}_f[j] = \mathbb{P}^g(S_2|S_{j+2}, \pi^g(S_{j+2}))$. Then, it can be shown that the success probability from any desired node $S_i \in \mathbb{V} \setminus B_{goal}$ is given as following [28]:

$$\mathbb{P}(\text{success}|S_i, \pi^g) := \mathbb{P}(B_{goal}|S_i, \pi^g) = \Gamma_{i-2}^T (I - \mathscr{Q})^{-1} \mathscr{R}_{goal}, \quad \forall i \geq 3, \tag{18}$$

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

According to the computed $\mathbb{P}(\text{success}|S_i, \pi^g)$, we define the success probability from any given initial belief b_0, denoted by $\mathbb{P}(\text{success}|b_0, \pi)$:

$$\mathbb{P}(\text{success}|b_0, \pi) = \mathbb{P}(B_0^*|b_0, \mu_*^0) \mathbb{P}(\text{success}|B_0^*, \pi^g), \tag{19}$$

where μ_*^0 is given by (16) and B_0^* is its corresponding stopping region.

Then, this success probability is compared with the minimum acceptable success probability, denoted by p_{min}, and if the condition $\mathbb{P}(\text{success}|b_0,\pi) > p_{min}$ is not satisfied, the number of nodes in FIRM 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 has been shown in [20].

F. Discussion

In summary, in FIRM, we aim to transform the original POMDP into a belief SMDP and solve it on a subset of belief space. Condition on the smoothness of cost function and transition probabilities, FIRM solution is arbitrarily close to the solution of belief SMDP over FIRM nodes. The important characteristic of FIRM is that it is solved offline and thus performing online phase of planning (or replanning) is computationally feasible. To exploit the generic FIRM framework, one has to find a proper (B,μ) pairs as the FIRM nodes and local controllers. Also, there has to be a way of computing transition costs and probabilities. There can be different ways of finding and computing these elements. In the next section, we propose one such approach, called Periodic LQG-based FIRM (or PLQG-based FIRM), in which the design of local controllers $\mu^{\alpha,ij}$ and FIRM nodes B^i_α are based on the properties of periodic LQG controllers. Moreover, we discuss how the transition costs $C^g(B^i_\alpha,\mu^{\alpha,ij})$ and the transition probabilities $\mathbb{P}^g(\cdot|B^i_\alpha,\mu^{\alpha,ij})$ can be evaluated in PLQG-based FIRM. Finally, we solve the corresponding FIRM MDP and provide the algorithms of offline construction of PLQG-based FIRM and online planning with it.

V. PLQG-BASED FIRM CONSTRUCTION

In this section, we construct a FIRM, in which local controllers are Periodic LQG (PLQG) controllers. Utilizing PLQG controllers, we design reachable FIRM nodes B^j_γ, and local planners $\mu^{\alpha,ij}$, required in (14). Then we discuss how the transition probabilities $\mathbb{P}^g(\cdot|B^i_\alpha,\mu^{\alpha,ij})$, and costs $C^g(B^i_\alpha,\mu^{\alpha,ij})$ in (14) are computed. 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 $b_k = p(x^+_k) = p(x_k|z_{0:k})$. 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 also show this using "equality relation", i.e., $b = (\widehat{x}^+, P)$.

A. Designing PLQG-based FIRM Nodes $\{B^j_\alpha\}$

LQG controllers: An LQG controller is composed of a Kalman filter as the state estimator and an LQR controller. Thus, the belief dynamics $b_{k+1} = \tau(b_k,u_k,z_{k+1})$ 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. LQG is an optimal controller for linear systems with Gaussian noise [23]. However, it is most often used for stabilizing nonlinear systems to a given trajectory or to a given point.

Periodic LQG: Periodic LQG (PLQG) is a time-varying LQG that is designed to track a given periodic trajectory[29], [30]. In Appendix A we review the periodic LQG controller in detail. Here, we only state the reachability result under the PLQG.

State Space Model: Consider a T-periodic PNPRM orbit $O = (x^p_k, u^p_k)_{k\geq 1}$ and the set of nodes $\{\mathbf{v}_\alpha\}$ on it. Let us denote the time-varying linear (linearized) system along the orbit O by the tuple $\Upsilon_k = (\mathbf{A}_k,\mathbf{B}_k,\mathbf{G}_k,\mathbf{Q}_k,\mathbf{H}_k,\mathbf{M}_k,\mathbf{R}_k)$ that represents the following state space model, where $\Upsilon_k = \Upsilon_{k+T}$:

$$x_{k+1} = \mathbf{A}_k x_k + \mathbf{B}_k u_k + \mathbf{G}_k w_k, \quad w_k \sim \mathcal{N}(\mathbf{0},\mathbf{Q}_k) \tag{20a}$$

$$z_k = \mathbf{H}_k x_k + \mathbf{M}_k v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0},\mathbf{R}_k), \tag{20b}$$

Consider the Periodic LQG (PLQG) controller that is designed for the system in (20) to track the orbit $(x^p_k, u^p_k)_{k\geq 1}$, through minimizing the following quadratic cost:

$$J = \mathbb{E}\left[\sum_{k\geq 0}(x_k - x^p_k)^T \mathbf{W}_x (x_k - x^p_k) + (u_k - u^p_k)^T \mathbf{W}_u (u_k - u^p_k)\right] \tag{21}$$

where $\mathbf{W}_x$ and $\mathbf{W}_u$ are the positive definite weight matrices for state and control cost, respectively. Let us also define the matrices $\check{\mathbf{Q}}_k$ and $\check{\mathbf{W}}_x$ such that $\mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T = \check{\mathbf{Q}}_k\check{\mathbf{Q}}_k^T$, $\mathbf{W}_x = \check{\mathbf{W}}_x^T\check{\mathbf{W}}_x$, for all k. Now, consider the class of systems, and associated LQG controllers, that satisfy the following property:

Property 2. *The pairs $(\mathbf{A}_k,\mathbf{B}_k)$ and $(\mathbf{A}_k,\check{\mathbf{Q}}_k)$ are controllable pairs[23], and the pairs $(\mathbf{A}_k,\mathbf{H}_k)$ and $(\mathbf{A}_k,\check{\mathbf{W}}_x)$ are observable pairs[23], for all $k = 1,\cdots,T$.*

Although the controllability of the pair $(\mathbf{A}_k,\mathbf{B}_k)$ is not met in stabilizing the nonholonomic unicycle model to a fixed point, it is met along orbits as discussed in Section II. The observability condition on the pair $(\mathbf{A}_k,\mathbf{H}_k)$ is related to the sensor model and the sensors has to be designed in such a way to satisfy this condition. Thus, it is important noting that the unicylce model and the large class of nonholonomic systems, equipped with a suitable sensors, satisfy the property 2 along the periodic trajectories with non-zero velocities, including orbits in PNPRM.

In the following, we present three lemmas, through which we can construct pairs of periodic LQG controllers, and reachable nodes in belief space, for nonholonomic and/or non-stoppable robots.

Lemma 1. *Consider the PLQG controller designed for the system in* (20) *to track the orbit $(x_k^p,u_k^p)_{k\geq 1}$. Given Property 2 is satisfied, in the absence of stopping region, the belief process under PLQG converges to a Gaussian cyclostationary process [31], i.e., the distribution over belief converges to a T-periodic Gaussian distribution, where we denote the mean and covariance of this process by b_k^c and $\mathtt{C}_k$, respectively:*

$$b_k \equiv (\widehat{x}_k^+,P_k) \sim \mathcal{N}(b_k^c,\mathtt{C}_k) = \mathcal{N}(b_{k+T}^c,\mathtt{C}_{k+T}), \tag{22}$$

where $b_k^c \equiv (x_k^p,\check{P}_k)$. The covariance matrices $\check{P}_k$ is characterized in Lemma 2 and covariance $\mathtt{C}_k$ is characterized in Appendix A.

Proof: In Appendix A, we review the periodic LQG and prove Lemma 1. ■

Lemma 2. *Given Property 2, the following Discrete Periodic Riccati Equation (DPRE) has a unique Symmetric T-Periodic Positive Semidefinite (SPPS) solution [29], denoted by $\check{P}_k^-$:*

$$\check{P}_{k+1}^- = \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T + \mathbf{A}_k(\check{P}_k^- - \check{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\check{P}_k^-\mathbf{H}_k^T + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^T)^{-1}\mathbf{H}_k\check{P}_k^-)\mathbf{A}_k^T \tag{23}$$

Moreover, the covariance matrix $\check{P}_k$ introduced in Lemma 1 is computed as

$$\check{P}_k = \check{P}_k^- - \check{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\check{P}_k^-\mathbf{H}_k^T + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^T)^{-1}\mathbf{H}_k\check{P}_k^- \tag{24}$$

Proof: See [29] or Appendix A. ■

Now, we state the main result, through which we can construct the proper pairs of periodic LQG controller and nodes in belief space for nonholonomic/non-stoppable robots.

Lemma 3. *Consider the PLQG controller μ designed for the system in* (20) *to track the orbit $(x_k^p,u_k^p)_{k\geq 1}$. Suppose the matrix $\mathbf{H}_k$ is full rank, and Property 2 is satisfied. Also, consider the sets $B_1,B_2,\cdots,B_m$ in belief space, such that interior of B_α contains $b_{k_\alpha}^c$ for some $k_\alpha \in \{1,\cdots,T\}$. Then, under μ, the region $\cup_\alpha B_\alpha$ is reachable in a finite time with probability one.*

Proof: See Appendix B. ■

FIRM Nodes: As mentioned, to construct a FIRM, we first construct its underlying PNPRM, characterized by the triple $\{\{O^j\},\{\mathbf{v}_\alpha^j\},\{\mathbf{e}_{ij}\}\}$. Linearizing the system along the j-th orbit $O^j = (x_k^{p^j},u_k^{p^j})_{k\geq 0}$ results in a time-varying T-periodic system $\Upsilon_k^j = (\mathbf{A}_k^j,\mathbf{B}_k^j,\mathbf{G}_k^j,\mathbf{Q}_k^j,\mathbf{H}_k^j,\mathbf{M}_k^j,\mathbf{R}_k^j)$:

$$x_{k+1} = \mathbf{A}_k^j x_k + \mathbf{B}_k^j u_k + \mathbf{G}_k^j w_k, \quad w_k \sim \mathcal{N}(\mathbf{0},\mathbf{Q}_k^j) \tag{25a}$$

$$z_k = \mathbf{H}_k^j x_k + \mathbf{M}_k^j v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0},\mathbf{R}_k^j). \tag{25b}$$

where w_k and v_k are motion and measurement noises, respectively, drawn from zero-mean Gaussian distributions with covariances $\mathbf{Q}_k^j$ and $\mathbf{R}_k^j$. The important property of the system in (25) is that is a T-periodic system, i.e.,$\Upsilon_k^j = \Upsilon_{k+T}^j$. Then, we design a PLQG controller μ_k^j corresponding to the system Υ_k^j. The controller μ_k^j is called the j-th *node-controller*. Since the orbits are designed such that Property 2 is satisfied on them, based on Lemma 1 the belief converges to a Gaussian cyclostationary process, with mean $b_k^{c^j}$, which can be computed using Lemma 2, where its existence and uniqueness are also guaranteed. Knowing that $\mathbf{v}_\alpha^j$, for $\alpha = 1,\cdots,m$, lies on orbit O^j, such that

$\mathbf{v}^j_\alpha = x^{p^j}_{k_\alpha}$, we choose the belief nodes B^j_α, for $\alpha = 1,\cdots,m$ such that B^j_α is an ε-ball in belief space, centered at $\check{b}^j_\alpha := b^{c^j}_{k_\alpha} \equiv (x^{p^j}_{k_\alpha}, \check{P}^j_{k_\alpha}) = (\mathbf{v}^j_\alpha, \check{P}^j_{k_\alpha})$: (See Fig.1(b).)

$$B^j_\alpha = \{b \equiv (x,P) \ : \ \|x - \mathbf{v}^j_\alpha\| < \delta_1 \ , \ \|P - \check{P}^j_{k_\alpha}\|_m < \delta_2\}, \tag{26}$$

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. Based on Lemma 3, $\cup_\alpha B^j_\alpha$ is a reachable region under the node-controller μ^j_k. Note that δ_1 and δ_2 are sufficiently small thresholds that determine the size of FIRM node B_j that satisfy the approximation in (11).

B. PLQG-based Local Controllers $\mu^{\alpha,ij}$

The role of the local controller $\mu^{\alpha,ij}$ is to drive the belief from the node B^i_α to $\cup_\gamma B^j_\gamma$, i.e., to a node B^j_γ, for some $\gamma = 1,\cdots,m$. To construct the local controller $\mu^{\alpha,ij}$, we precede the node-controller μ^j_k, with a time-varying LQG controller $\overline{\mu}^{\alpha,ij}_k$, which is called the *edge-controller* here.

Edge-controller: Consider a finite trajectory that consists of three segments: *i)* the pre-edge $\mathbf{e}^{i\alpha j}$ as defined in Section II (See Fig. 1(a)), *ii)* the edge itself $\mathbf{e}^{ij}$ (See Fig. 1(a)), and *iii)* a part of O^j that connects the ending point of $\mathbf{e}^{ij}$ to $x^{p^j}_0$. Edge-controller $\overline{\mu}^{i\alpha j}_k$ is a time-varying LQG controller that is designed to track this finite trajectory. The main role of the edge-controller is that it takes the belief at node B_i and drives it to the vicinity of a starting point of orbit O^j, where it hands over the system to the node-controller, and node-controller in turn takes the system to a FIRM node.

Local controllers: Thus, overall, the local controller $\mu^{\alpha,ij}$ is a concatenation of the edge-controller $\overline{\mu}^{\alpha,ij}_k$ and the node-controller μ^j_k. Note that since reachability is guaranteed by the node-controller (periodic LQG controller), by this construction, the stopping region $\cup_\gamma B^j_\gamma$ is also reachable under the local controller $\mu^{\alpha,ij}$. Hence the Property 1 is satisfied by this construction.

C. Transition probabilities and costs

In general, it can be a computationally expensive task to compute the transition probabilities $\mathbb{P}(\cdot|B^i_\alpha,\mu^{\alpha,ij})$ and costs $C(B^i_\alpha,\mu^{\alpha,ij})$ associated with invoking local controller $\mu^{\alpha,ij}$ at node B^i_α. However, owing to the offline construction of FIRM, it is not an issue in FIRM. We utilize sequential Monte-carlo methods [32] 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 $\Phi^{\alpha,ij} = \mathbb{E}[\sum_{k=1}^{\mathscr{T}} \mathrm{tr}(P^{\alpha,ij}_k)]$, where $P^{\alpha,ij}_k$ is the estimation covariance at the k-th time step of the execution of local controller $\mu^{\alpha,ij}$. The outer expectation operator is useful in dealing with the Extended Kalman Filter (EKF), whose covariance is stochastic[33], [34]. Moreover, as we are also interested in faster paths, we take into account the corresponding mean stopping time, i.e., $\widehat{\mathscr{T}}^{\alpha,ij} = \mathbb{E}[\mathscr{T}^{\alpha,ij}]$, and the total cost of invoking $\mu^{\alpha,ij}$ at B^i_α is considered as a linear combination of estimation accuracy and expected stopping time, with suitable coefficients ξ_1 and ξ_2.

$$C(B^i_\alpha,\mu^{\alpha,ij}) = \xi_1 \Phi^{\alpha,ij} + \xi_2 \widehat{\mathscr{T}}^{\alpha,ij}. \tag{27}$$

D. Construction of PLQG-based FIRM and Planning With it

The crucial feature of FIRM is that it can be constructed offline and stored, independent of future queries. 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 $\mu^* = \pi^g(B^i_\alpha)$ at each FIRM node B^i_α among all controller $\mu \in \mathbb{M}(\alpha,i)$.Algorithm 3 details the construction of FIRM. 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 $\mathscr{E}(0)$. We design the local controllers associated with each edge in $\mathscr{E}(0)$ 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 (14b) over FIRM nodes to find μ^*, and pick μ^* to move the robot into $B(\mu^*)$.Algorithm 4 illustrates this procedure.

Algorithm 3: Offline Construction of PLQG-based FIRM

input : Free space map, $\mathbb{X}_{free}$

output : FIRM graph $\mathscr{G}$

Construct a PNPRM with T-periodic orbits $\mathscr{O} = \{O^j = (x_k^{p^j}, u_k^{p^j})_{k\geq 0}\}$, nodes $\mathscr{V} = \{\mathbf{v}_\alpha^j\}$, and edges $\mathscr{E} = \{\mathbf{e}^{ij}\}$, where $i, j = 1, \cdots, n$ and $\alpha = 1, \cdots, m$;

forall the *PNPRM orbits* $O^j \in \mathscr{O}$ **do**

    Design the node-controller (periodic LQG) μ_k^j along the periodic trajectory;

    Compute the periodic mean belief trajectory $b_k^{c^j} = (x_k^{p^j}, \check{P}_k^j)$ using (24);

    Construct m FIRM nodes $\mathbb{V}^j = \{B_1^j, \cdots, B_m^j\}$ using (26), where B_α^j is centered at $b_{k_\alpha}^{c^j}$;

Collect all FIRM nodes $\mathbb{V} = \cup_{j=1}^n \mathbb{V}^j$;

forall the $(B_\alpha^i, \mathbf{e}^{ij})$ *pairs* **do**

    Design the edge-controller $\overline{\mu}_k^{\alpha,ij}$, as discussed in Section V-B;

    Construct the local controller $\mu_k^{\alpha,ij}$ by concatenating edge-controller $\overline{\mu}_k^{\alpha,ij}$ and node-controller μ_k^j;

    Set the initial belief b_0 equal to the center of B_α^i, based on the approximation in (11);

    Generate sample belief paths $b_{0:\mathscr{T}}$ and state paths $x_{0:\mathscr{T}}$ induced by controller $\mu^{\alpha,ij}$ invoked at B_α^i;

    Compute the transition probabilities $\mathbb{P}^g(F|B_\alpha^i, \mu^{\alpha,ij})$ and $\mathbb{P}^g(B_\gamma^j|B_\alpha^i, \mu^{\alpha,ij})$ for all γ and transition cost $C^g(B_\alpha^i, \mu^{\alpha,ij})$;

Collect all local controllers $\mathbb{M} = \{\mu^{\alpha,ij}\}$;

Compute cost-to-go J^g and feedback π^g over the FIRM by solving the DP in (14);

$\mathscr{G} = (\mathbb{V}, \mathbb{M}, J^g, \pi^g)$;

return $\mathscr{G}$;

VI. Experimental Results

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

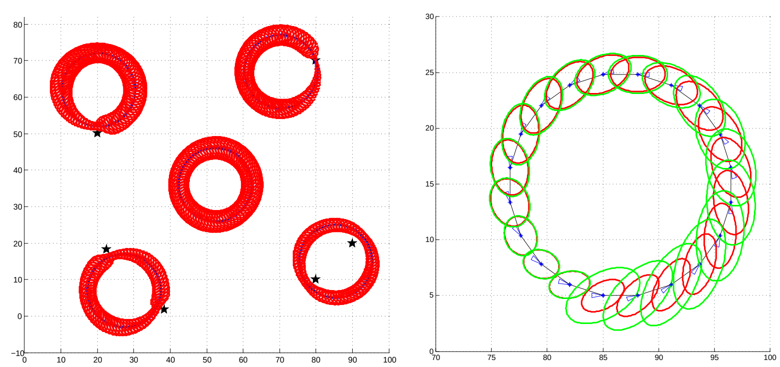
$$\begin{aligned} {}^jz &= {}^jh(x, {}^jv) = [\|{}^j\mathbf{d}\|, \text{atan2}({}^jd_2, {}^jd_1) - \theta]^T + {}^jv, \quad {}^jv \sim \mathscr{N}(\mathbf{0}, {}^j\mathbf{R}), \\ {}^j\mathbf{R} &= \text{diag}((\eta_r\|{}^j\mathbf{d}\| + \sigma_b^r)^2, (\eta_\theta\|{}^j\mathbf{d}\| + \sigma_b^\theta)^2), \end{aligned} \tag{28}$$

where ${}^j\mathbf{d} = [{}^jd_1, {}^jd_2]^T := [{}^1x, {}^2x]^T - L_j$. The uncertainty (standard deviation) of sensor reading increases as the robot gets farther from the landmarks. The parameters $\eta_r = \eta_\theta = 0.3$ determine this dependency, and $\sigma_b^r = 0.01$ meter and $\sigma_b^\theta = 0.5$ degrees are the bias standard deviations. A similar model for range sensing is used in [11]. 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 $z = h(x) + v$, where $v \sim \mathscr{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{R} = \text{diag}({}^1\mathbf{R}, \cdots, {}^{N_L}\mathbf{R})$.

We first show a typical SPPS solution of DPRE on the orbits. Fig. 2(a) shows a simple environment with six radio beacons, which are shown by the black stars. For illustration purposes, we choose five large circular orbits and every orbit is discretized to 100 steps. Thus the SPPS solution of the DPRE in (23) on each orbit leads to hundred covariance matrices that are superimposed on the graph in red. As is seen from Fig. 2(a), the localization uncertainty along the orbit is not homogeneous and varies periodically.Another important observation from the Fig. 2(a) is obtained by noticing the left top orbit in the Fig. 2(a). As it can be seen, the localization uncertainty in the right hand side of the landmark, which lies close to orbit, is greater than its left hand side. In other words, although the phrase "The closer to landmark, the lesser the sensing uncertainty" is true, the phrase "The closer to landmark, the lesser the localization uncertainty" is not true, which emphasizes the role of dynamic model in filtering and its interaction with the observation model. In Fig. 2(b), we illustrate the covariance convergence in the periodic belief process. As can be seen in Fig. 2(b), the initial covariance is three times larger than the limiting

Algorithm 4: Online Phase Algorithm (Planning with PLQG-based FIRM)

input : Initial belief b_0, FIRM graph $\mathscr{G}$, Underlying PNPRM graph
if $\exists B^i_\alpha \in \mathbb{V}$ *such that* $b_0 \in B^i_\alpha$ **then**
  Choose the next local controller $\mu^{\alpha,ij} = \pi^g(B^i_\alpha)$;
else
  Compute $\mathbf{v}_0 = \mathbb{E}[x_0]$ based on b_0, and connect $\mathbf{v}_0$ to the PNPRM orbits. Call the set of newly added edges $\mathscr{E}(0) = \{\mathbf{e}^{0j}\}$;
  Design local planners associated with edges in $\mathscr{E}(0)$; Collect them in set $\mathbb{M}(0) = \{\mu^{0,0j}\}$;
  forall the $\mu \in \mathbb{M}(0)$ **do**
    Generate sample belief and state paths $b_{0:\mathscr{T}}$, $x_{0:\mathscr{T}}$ 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 $\alpha, i = 0$; Choose the best initial local planner $\mu^{0,0j}$ within the set $\mathbb{M}(0)$ using (16);
while $B^i_\alpha \neq B_{goal}$ **do**
  while $(\nexists B^j_\gamma, s.t., \ b_k \in B^j_\gamma)$ *and "no collision"* **do**
    Apply the control $u_k = \mu_k^{\alpha,ij}(b_k)$ to the system;
    Get the measurement z_{k+1} from sensors;
    if *Collision happens* **then return** Collision;
    Update belief as $b_{k+1} = \tau(b_k, \mu_k^{\alpha,ij}(b_k), z_{k+1})$;
  Update the current FIRM node $B^i_\alpha = B^j_\gamma$;
  Choose the next local controller $\mu^{\alpha,ij} = \pi^g(B^i_\alpha)$;



(a) (b)

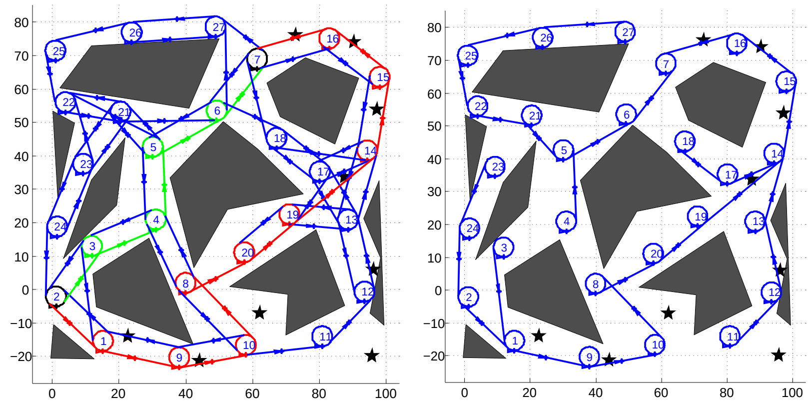
Fig. 2. (a) Five orbits ($T = 100$) and corresponding periodic estimation covariances as the SPPS solution of DPRE in (23). (b) Sample covariance convergence on an orbit ($T = 20$) under PLQG. Red ellipses are the solution of DPRE and green ellipses are the evolution of estimation covariance. The initial covariance is three times bigger than the SPPS solution of DPRE, i.e., $P_0 = 3\check{P}_0$.

covariance, and in less than one period it converges to the SPPS solution of DPRE. The convergence time is a random quantity, whose mean and variance can be estimated through simulations. However, in practical cases it usually converges in less than one full period, because the initial covariance is closer to the actual solution (due to the use of edge-controllers) and also the orbit size is much smaller, when compared to Fig. 2(b).

Figure 3(a) shows a sample PNPRM with 23 orbits and 67 edges. To simplify the explanation of the results, we assume $m = 1$, i.e., we choose one node on each orbit. All elements in Fig.3(a) are defined in $(\mathtt{x}, \mathtt{y}, \theta)$ space but only the $(\mathtt{x}, \mathtt{y})$ portion is shown here. To construct the FIRM nodes, we first solve the corresponding DPRE's on each orbit and design its corresponding node-controller (PLQG). Then, we pick the node centers $\grave{b}^j_\alpha = (\mathbf{v}^j_\alpha, \check{P}^j_{k_\alpha})$ and construct the FIRM nodes based on the component-wise version of (26), to handle the error scale difference in position and orientation variables:

$$B^j_\alpha = \{b = (x,P)| \quad |x - \mathbf{v}^j_\alpha| \dot{<} \varepsilon, |P - \check{P}^j_{k_\alpha}| \dot{<} \Delta\}, \tag{29}$$

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



(a) (b)

Fig. 3. A sample PNPRM with circular orbits. Number of each orbit is written in its center. Nine landmarks are shown by black stars and obstacles are shown by gray polygons. The moving directions on orbits and on edges are shown by little triangles whose head is specified by a little cross. (a) Nodes 2 and 7 are start and goal nodes, respectively, which are drawn in black. Shortest path and the most-likely path under FIRM policy are shown in green and red, respectively. (b) Assuming on each orbit, there exists a single node, the feedback π^g is visualized for all FIRM nodes.

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 [32] to compute these quantities. In other words, for every $(B^i_\alpha, \mu^{\alpha,ij})$ pair, we perform M runs and accordingly approximate the transition probabilities $\mathbb{P}^g(B^j_\gamma|B^i_\alpha, \mu^{\alpha,ij})$, $\mathbb{P}^g(F|B^i_\alpha, \mu^{\alpha,ij})$, and costs $C^g(B^i_\alpha, \mu^{\alpha,ij})$. A similar approach is detailed in [19]. Table I shows these quantities for several $(B^i_\alpha, \mu^{\alpha,ij})$ pairs corresponding to Fig.3(a), where $M = 101$ and the coefficients in (27) are $\xi_1 = 0.98$ and $\xi_2 = 0.02$.

TABLE I
COMPUTED COSTS FOR SEVERAL PAIRS OF NODE-AND-CONTROLLER USING 101 PARTICLES.

$(B^i_\alpha,\mu^{\alpha,ij})$ pair	$B^2_1,\mu^{1,(2,3)}$	$B^4_1,\mu^{1,(4,5)}$	$B^6_1,\mu^{1,(6,7)}$	$B^{11}_1,\mu^{1,(11,12)}$	$B^2_1,\mu^{1,(2,1)}$	$B^8_1,\mu^{1,(8,20)}$	$B^{16}_1,\mu^{1,(16,7)}$
$\mathbb{P}^g(F\|B^i_\alpha,\mu^{\alpha,ij})$	9.9010%	17.8218%	15.8416%	29.7030%	7.9208%	1.9802%	0.9901%
$\Phi^{\alpha,ij}$	2.1386	2.2834	1.9181	0.9152	2.1695	1.1857	0.4385
$\mathbb{E}[\mathscr{T}^{\alpha,ij}]$	63.6703	82.6747	62.5882	58.2000	51.7033	50.2755	35.4653

Plugging the computed transition costs and probabilities into (14), 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. 3(b) shows the policy π^g on the constructed FIRM in this example. Indeed, at every FIRM node B^i_α, 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. It is worth noting that if we had more than one node on each orbit, the feedback π^g may return different controllers for each of them and for every orbit we may have more than one outgoing arrow in Fig. 3(b).

Thus, the online part of planning is quite efficient, i.e., it only requires executing the controller and generating the control signal, which is an instantaneous computation. An important consequence of the 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. 3(b).

We show the most likely path under the π^g, in red in Fig. 3(a). The shortest path is also illustrated in Fig. 3(a) in green. 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 collisions.

VII. 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 and considers a collection of disjoint sets along a periodic orbit as the stopping regions of the FIRM local planners. Exploiting the properties of periodic LQG controllers, the local planners in FIRM are designed such that the belief reachability is achieved along periodic maneuver, satisfying nonholonomic/control constraints. As a result, while taking collision probabilities into account, the roadmap provides a sampling-based feedback solution for nonholonomic systems and/or nonstoppable systems 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 *wafr*, 1998, pp. 155–168.

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

[9] S. Lavalle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 378-400, 2001.

[10] D. Hsu, "Randomized single-query motion planning in expansive spaces," Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, CA, 2000.

[11] 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.

[12] 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.

[13] 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.

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

[15] M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Control and Cybernetics*, vol. 35, no. 3, pp. 599–620, 2006.

[16] S. Chakravorty and S. Kumar, "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, October 2009.

[17] ——, "Generalized sampling-based motion planners," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 41, no. 3, pp. 855–866, 2011.

[18] 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.

[19] ——, "Motion planning in belief space using sampling-based feedback planners," *Technical Report: TR11-007, Parasol Lab., CSE Dept., Texas A&M University*, 2011.

[20] ——, "On the probabilistic completeness of the sampling-based feedback motion planners in belief space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[21] 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.

[22] A. Agrachev and Y. Sachkov, *Control theory from the geometric viewpoint*, ser. Encyclopaedia of Mathematical Sciences, 87, Control Theory and Optimization, II. Springer-Verlag, 2004.

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

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[25] 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.

[26] ——, "Efficient planning under uncertainty with macro-actions," *Journal of Artificial Intelligence Research*, vol. 40, pp. 523–570, February 2011.

[27] R. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, pp. 181–211, 1999.

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

[29] S. Bittanti and P. Colaneri, *Periodic systems filtering and control*. Springer-Verlag, 2009.

[30] P. Colaneri, R. Scattolini, and N. Schiavoni, "LQG optimal control for multirate samled-data systems," *IEEE Transactions on Automatic Control*, vol. 37, no. 5, pp. 675–682, 1992.

[31] S. Bittanti, P. Bolzern, G. De Nicolao, and L. Piroddi, "Representation, prediction and identification of cyclostationary processes," *Cyclostationarity in Communications and Signal Processing*, Edited by W.A. Gardner, IEEE Press, 1994.

[32] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. New York: Springer, 2001.
[33] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley and Sons, Inc, 2006.
[34] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC, 2004.
[35] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability: 2nd Edition*. Cambridge University Press, 2009.

APPENDIX A
PERIODIC LQG CONTROLLER

Periodic Linear Quadratic Gaussian (PLQG) controller is a time-varying LQG controller that is designed to track a pre-planned periodic trajectory (also called nominal, desired, or open-loop trajectory) in the presence of process and observation noise.

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:

$$x_{k+1} = f(x_k, u_k, w_k), \quad w_k \sim \mathcal{N}(0, Q_k) \tag{30a}$$

$$z_k = h(x_k, v_k), \quad v_k \sim \mathcal{N}(0, R_k) \tag{30b}$$

A T-periodic 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:

$$x_{k+1}^p = f(x_k^p, u_k^p, 0), \quad x_{k+T}^p = x_k^p, \quad u_{k+T}^p = u_k^p \tag{31}$$

The role of a closed-loop stochastic controller, during the trajectory tracking execution, is compensating robot deviations from the planned trajectory due to the noise effects and keeping the robot close to the planned trajectory in the sense of minimizing following quadratic cost:

$$J = \mathbb{E}\left[\sum_{k\geq 0} (x_k - x_k^p)^T W_x (x_k - x_k^p) + (u_k - u_k^p)^T W_u (u_k - u_k^p)\right] \tag{32}$$

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

Since the state space is not fully observable and it is only partially observable, at every step of LQG execution, a Kalman filter estimates the system state and an LQR controller generates the optimal control based on this estimation. We first linearize the system along the nominal path and then describe the KF and LQR designed along this path.

Model linearization: Given a periodic nominal trajectory $(x_k^p, u_k^p)_{k\geq 0}$, we linearize the dynamics and observation model in (30), as follows:

$$x_{k+1} = f(x_k^p, u_k^p, 0) + A_k(x_k - x_k^p) + B_k(u_k - u_k^p) + G_k w_k, \quad w_k \sim \mathcal{N}(0, Q_k) \tag{33a}$$

$$z_k = h(x_k^p, 0) + H_k(x_k - x_k^p) + M_k v_k, \quad v_k \sim \mathcal{N}(0, R_k) \tag{33b}$$

where

$$\begin{cases} A_k &= \frac{\partial f}{\partial x}(x_k^p, u_k^p, 0), \quad B_k = \frac{\partial f}{\partial u}(x_k^p, u_k^p, 0), \quad G_k = \frac{\partial f}{\partial w}(x_k^p, u_k^p, 0), \\ H_k &= \frac{\partial h}{\partial x}(x_k^p, 0), \quad M_k = \frac{\partial h}{\partial v}(x_k^p, 0) \end{cases} \tag{34}$$

It is worth noting that the linearized system is a periodic one, i.e.,

$$A_{k+T} = A_k, \quad B_{k+T} = B_k, \quad G_{k+T} = G_k, \quad H_{k+T} = H_k, \quad M_{k+T} = M_k, \quad Q_{k+T} = Q_k, \quad R_{k+T} = R_k. \tag{35}$$

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 (mean of 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 above linearized models as follows:

$$e_{k+1} = A_k e_k + B_k \delta u_k + G_k w_k, \quad w_k \sim \mathcal{N}(0, Q_k) \tag{36a}$$

$$\delta z_k = H_k e_k + M_k v_k, \quad v_k \sim \mathcal{N}(0, R_k) \tag{36b}$$

which is a periodic linear system due to the (35).

Periodic Kalman Filter: Periodic Kalman Filter (PKF) is a time-varying Kalman filtering, whose underlying linear system is periodic, like (36). 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:

$$b_k = p(x_k^+) = p(x_k|z_{0:k}) \tag{37}$$
$$\widehat{x}_k^+ = \mathbb{E}[x_k|z_{0:k}] \tag{38}$$
$$P_k = \mathbb{C}[x_k|z_{0:k}] \tag{39}$$

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 only be characterized by its mean and covariance, i.e., $b_k \equiv (\widehat{x}_k^+, P_k)$. Similar to the conventional Kalman filtering, PKF 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 (36) prediction step is:

$$\widehat{e}_{k+1}^- = A_k \widehat{e}_k^+ + B_k \delta u_k \tag{40}$$
$$P_{k+1}^- = A_k P_k^+ A_k^T + G_k Q_k G_k^T \tag{41}$$

In the update step, the mean and covariance of posterior x_k^+ is computed. For the system in (36), the update step is:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \tag{42}$$
$$\widehat{e}_{k+1}^+ = \widehat{e}_{k+1}^- + K_{k+1}(\delta z_{k+1} - H_{k+1}\widehat{e}_{k+1}^-) \tag{43}$$
$$P_{k+1}^+ = (I - K_{k+1}H_{k+1})P_{k+1}^- \tag{44}$$

Note that

$$\widehat{x}_k^+ = \mathbb{E}[x_k|z_{0:k}] = x_k^p + \mathbb{E}[e_k|z_{0:k}] = x_k^p + \widehat{e}_k^+ \tag{45}$$
$$P_k = \mathbb{C}[x_k|z_{0:k}] = \mathbb{C}[e_k|z_{0:k}] = P_k^+ \tag{46}$$

Lemma 4. *In Periodic Kalman filtering (PKF), if for all k, the pair (A_k, H_k) is detectable and the pair $(A_k, \check{Q}_k)$ is stabilizable, where $G_k Q_k G_k^T = \check{Q}_k \check{Q}_k^T$, then the prior and posterior covariances P_k^- and P_k and the filter gain K_k, all converge to their T-periodic stationary values, denoted by $\check{P}_t^-$, $\check{P}_t$, and $\check{K}_t$, respectively [29]. Matrix $\check{P}_t^-$ is the unique Symmetric T-Periodic Positive Semi-definite (SPPS) solution [29] of the following Discrete Periodic Riccati Equation (DPRE):*

$$\check{P}_{k+1}^- = G_k Q_k G_k^T + A_k(\check{P}_k^- - \check{P}_k^- H_k^T (H_k \check{P}_k^- H_k^T + M_k R_k M_k^T)^{-1} H_k \check{P}_k^-) A_k^T \tag{47}$$

Having $\check{P}_k^-$, the periodic gain $\check{K}_k$ and estimation covariance $\check{P}_k$ is computed as follows:

$$\check{K}_k = \check{P}_k^- H_k^T (H_k \check{P}_k^- H_k^T + M_k R_k M_k^T)^{-1}, \tag{48}$$
$$\check{P}_k = (I - \check{K}_k H_k)\check{P}_k^- \tag{49}$$

where

$$\check{P}_{k+T}^- = \check{P}_k^-, \quad \check{K}_{k+T} = \check{K}_k, \quad \check{P}_{k+T} = \check{P}_k \tag{50}$$

Proof: See [29]. ■

If the pair (A_k, H_k) is detectable and the pair $(A_k, \check{Q}_k)$ is stabilizable, then the pair (A_k, H_k) is observable and the pair $(A_k, \check{Q}_k)$ is controllable, and hence the Lemma 2 follows.

Periodic LQR controller: LQR controller is an state controller, which is utilized within the structure of the LQG controller, which is an output controller. Once Kalman filter produces the estimation (belief), the LQR controller generates 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. In LQG, the mapping μ_k is the control law of the LQR controller, which is optimal in the sense of minimizing following cost:

$$\begin{aligned} J_{PLQR} &= \mathbb{E}\left[\sum_{k\geq 0} (\widehat{x}_k^+ - x_k^p)^T W_x (\widehat{x}_k^+ - x_k^p) + (u_k - u_k^p)^T W_u (u_k - u_k^p)\right] \\ &= \mathbb{E}\left[\sum_{k\geq 0} (\widehat{e}_k^+)^T W_x (\widehat{e}_k^+) + (\delta u_k)^T W_u (\delta u_k)\right] \end{aligned} \tag{51}$$

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

$$\delta u_k = -L_k \widehat{e}_k^+, \quad L_{k+T} = L_k \tag{52}$$

Lemma 5. *In Periodic LQR (PLQR), if for all k, the pair (A_k, B_k) is stabilizable and the pair $(A_k, \check{W}_x)$ is detectable, where $W_x = \check{W}_x^T \check{W}_x$, then the time-varying feedback gains L_k is a T-periodic gain, i.e., $L_{k+T} = L_k$ and is computed as follows:*

$$L_k = (B_k^T S_{k+1} B_k + W_u)^{-1} B_k^T S_{k+1} A_k \tag{53}$$

where S_k is the SPPS solution of the following DPRE:

$$S_k = W_x + A_k^T S_{k+1} A_k - A_k^T S_{k+1} B_k (B_k^T S_{k+1} B_k + W_u)^{-1} B_k^T S_{k+1} A_k \tag{54}$$

Note that the whole control is $u_k = u_k^p + \delta u_k$.

Periodic LQG controller: Plugging the obtained control law of PLQR into the PKF equations, we can get the following error dynamics, for the defined errors:

$$\begin{aligned}\begin{pmatrix} e_{k+1} \\ \widetilde{e}_{k+1} \end{pmatrix} &= \begin{pmatrix} A_k - B_k L_k & B_k L_k \\ 0 & A_k - \check{K}_{k+1} H_{k+1} A_k \end{pmatrix} \begin{pmatrix} e_k \\ \widetilde{e}_k \end{pmatrix} \\ &+ \begin{pmatrix} G_k & 0 \\ G_k - \check{K}_{k+1} H_{k+1} G_k & -\check{K}_{k+1} M_{k+1} \end{pmatrix} \begin{pmatrix} w_k \\ v_{k+1} \end{pmatrix}\end{aligned} \tag{55}$$

or equivalently,

$$\begin{aligned}\begin{pmatrix} e_{k+1} \\ \widehat{e}_{k+1}^+ \end{pmatrix} &= \begin{pmatrix} A_k & -B_k L_k \\ \check{K}_{k+1} H_{k+1} A_k & A_k - B_k L_k - \check{K}_{k+1} H_{k+1} A_k \end{pmatrix} \begin{pmatrix} e_k \\ \widehat{e}_k^+ \end{pmatrix} \\ &+ \begin{pmatrix} G_k & 0 \\ \check{K}_{k+1} H_{k+1} G_k & \check{K}_{k+1} M_{k+1} \end{pmatrix} \begin{pmatrix} w_k \\ v_{k+1} \end{pmatrix}\end{aligned} \tag{56}$$

Defining $\zeta_k := (e_k, \widehat{e}_k^+)^T$ and $q_k := (w_k, v_{k+1})^T$, we can rewrite (56) in a more compact form as

$$\zeta_{k+1} = \overline{F}_k \zeta_k - \overline{G}_k q_k, \quad q_k \sim \mathcal{N}(0, \overline{Q}_k), \quad \overline{Q}_k = \begin{pmatrix} Q_k & 0 \\ 0 & R_{k+1} \end{pmatrix} \tag{57}$$

with appropriate definitions for $\overline{F}_k$ and $\overline{G}_k$. Thus, ζ_k is a random variable with a Gaussian distribution, i.e.,

$$\zeta_k = \mathcal{N}(0, \mathscr{P}_k) \tag{58}$$

or

$$\begin{pmatrix} x_k \\ \widehat{x}_k^+ \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} x_k^p \\ x_k^p \end{pmatrix}, \mathscr{P}_k\right) \tag{59}$$

where $\mathscr{P}_k$ is the solution of following Discrete Periodic Lyapunov Equation (DPLE):

$$\mathscr{P}_{k+1} = \overline{F}_k \mathscr{P}_k \overline{F}_k^T - \overline{G}_k \overline{Q}_k \overline{G}_k^T \tag{60}$$

which can be decomposed into four blocks

$$\mathscr{P}_k = \begin{pmatrix} \mathscr{P}_{k,11} & \mathscr{P}_{k,12} \\ \mathscr{P}_{k,21} & \mathscr{P}_{k,22} \end{pmatrix} \tag{61}$$

Lemma 6. *Under the preceding assumptions in Lemma 4 and Lemma 5, the solution of DPLE in* (60) *converges to a unique SPPS solution $\check{\mathscr{P}}_k$ independent of the initial covariance $\mathscr{P}_0$, i.e., $\check{\mathscr{P}}_{k+T} = \check{\mathscr{P}}_k$.*

Proof: See [29]. ■

Therefore, the process in (57) converges to a cyclostationary process [31], i.e., the distribution over ζ_k is periodic. Hence, since $\widehat{x}_k^+ \sim \mathcal{N}(x_k^p, \mathscr{P}_{k,22})$, the distribution over estimation mean is also converges to a periodic distribution, i.e., $\widehat{x}_k^+ \sim \mathcal{N}(x_k^p, \check{\mathscr{P}}_{k,22}) = \mathcal{N}(x_{k+T}^p, \check{\mathscr{P}}_{k+T,22})$. Hence, this analysis leads to the following lemma:

Lemma 7. *Under Periodic LQG, belief falls into a Gaussian cyclostationary process, i.e., the distribution over belief $b_k = (\widehat{x}_k^+, P_k)$ converges to the following periodic Gaussian distribution:*

$$b_k \equiv (\widehat{x}_k^+, P_k) \sim \mathscr{N}\left(\left(\begin{array}{c} x_k^p \\ \check{P}_k \end{array}\right), \left(\begin{array}{cc} \check{\mathscr{P}}_{k,22} & 0 \\ 0 & 0 \end{array}\right)\right) \tag{62}$$

The degeneracy of the Gaussian distribution over belief in (62) is due to the fact that $\check{P}_k$ is a deterministic process. It is worth noting that the belief mean converges to the T-periodic belief $\mathbb{E}[b_{k+T}] = \mathbb{E}[b_k] = (x_k^p, \check{P}_k)$. Hence, the Lemma 1 follows, as it is the same as Lemma 7, where we have:

$$b_k^c = \left(\begin{array}{c} x_k^p \\ \check{P}_k \end{array}\right), \quad \mathtt{C}_k = \left(\begin{array}{cc} \check{\mathscr{P}}_{k,22} & 0 \\ 0 & 0 \end{array}\right) \tag{63}$$

APPENDIX B
PROOF OF LEMMA 3

Proof: Let us consider the state space model of a T-periodic linear system of interest as follows:

$$x_{k+1} = \mathbf{A}_k x_k + \mathbf{B}_k u_k + \mathbf{G}_k w_k, \quad w_k \sim \mathscr{N}(\mathbf{0}, \mathbf{Q}_k) \tag{64a}$$

$$z_k = \mathbf{H}_k x_k + v_k, \qquad v_k \sim \mathscr{N}(\mathbf{0}, \mathbf{R}_k). \tag{64b}$$

Based on Lemma 1 and Lemma 2, if $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}, \check{\mathbf{Q}})$ are controllable pairs, where $\mathbf{GQG}^T = \check{\mathbf{Q}}\check{\mathbf{Q}}^T$, 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 T-periodic stationary covariance $\check{P}_k$. Therefore, for any $\varepsilon > 0$, after a deterministic finite time, P_k enters the ε-neighbourhood of the periodic stationary covariance, i.e., $\|P_k - \check{P}_k\|_m \leq \varepsilon$ for all k large enough, where $\|\cdot\|$ stands for a matrix norm.

The estimation mean dynamics, however, is stochastic and is as follows for the system in Eq. (64):

$$\begin{aligned} \widehat{x}_{k+1}^+ &= x_{k+1}^p + (\mathbf{A}_k - \mathbf{B}_k \mathbf{L}_k - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{A}_k)(\widehat{x}_k^+ - x_k^p) \\ &+ \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{A}_k(x_k - x_k^p) + \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{G}_k w_k + \mathbf{K}_{k+1} v_{k+1} \\ &= x_{k+1}^p - (\mathbf{A}_k - \mathbf{B}_k \mathbf{L}_k) x_k^p + (\mathbf{A}_k - \mathbf{B}_k \mathbf{L}_k - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{A}_k)\widehat{x}_k^+ \\ &+ \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{A}_k x_k + \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{G}_k w_k + \mathbf{K}_{k+1} v_{k+1} \end{aligned} \tag{65}$$

(66)

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

$$\mathbf{K}_k = P_k^- \mathbf{H}_k^T (\mathbf{H}_k P_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{67}$$

Since $\mathbf{K}_k$ is full rank (due to the condition on the rank of $\mathbf{H}_k$) for all k and since the v and w are Gaussian noises, (65) induces an irreducible Markov process over the state space[35]. Thus, if we have a stopping region for the estimation mean with size $\varepsilon > 0$, the estimation mean process will hit this stopping region in a finite time [35], with probability one, i.e., for a finite $\mathbf{v} \in \mathbb{X}$, the condition $\|\widehat{x}_k^+ - \mathbf{v}\|$ is satisfied in a finite time. However, $\mathbf{v}$ can be chosen in a way that maximizes the absorption probability and minimizes the hitting time.

Based on the estimation mean dynamics in (65) and the state dynamics in Appendix A, if the estimation mean process and state process start from $\widehat{x}_0^+$ and x_0, respectively, such that $\mathbb{E}[\widehat{x}_0^+] = x_k^p$ and $\mathbb{E}[x_0] = x_k^p$ (which indeed is the case in FIRM due to the usage of edge-controllers), "the mean of estimation mean" remains on the x_k^p, i.e., $\mathbb{E}[\widehat{x}_k^+] = x_k^p$, for all k. As a result, the x_k^p is the optimal choice for the center of stopping region and thus, the condition $\|\widehat{x}_k^+ - x_k^p\|$ is satisfied in a minimum time in the sense of "expected value".

Combining the results for estimation covariance and estimation mean, if we define the region $\check{B}_k$ as a set in the Gaussian belief space with a non-empty interior centered at $(x_{k_\alpha}^p, \check{P}_{k_\alpha})$, then the belief $b_k = (\widehat{x}_k^+, P_k)$ enters region $\cup_k \check{B}_k$ with a minimum finite expected time with probability one. To decrease the number of nodes, one can only look at the subsequence $\grave{b}_\alpha := b_{k_\alpha} = (\widehat{x}_{k_\alpha}^+, P_{k_\alpha})$ and $B_\alpha := \check{B}_{k_\alpha}$ for $\{k_1, k_2, \cdots, k_m\} \subset \{1, 2, \cdots, T\}$, then similarly the belief $\grave{b}_\alpha$ enters region $\cup_\alpha B_\alpha$ in a finite time with probability one. ■

Appendix C
Linear Controllability Analysis of Unicycle Model

An implicit assumption in road-map based methods such as PRM is that on every edge there exists a controller to drive the robot from the start node of the edge to the end node of the edge or to an ε-neighborhood of the end node, for some $\varepsilon > 0$.

For a controllable robot, whose linearized model is also controllable around the PRM nodes, a linear controller can locally drive the robot to the PRM nodes. However, for a nonholonomic robot such as unicycle the linearized model at any point is not controllable and hence a linear controller cannot drive the robot to the PRM nodes. Consider the discrete unicycle model:

$$x_{k+1} = f(x_k, u_k, w_k) \tag{68}$$

where $x_k = (\mathsf{x}_k, \mathsf{y}_k, \theta_k)^T$, 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 $w_k = (n_v, n_\omega)^T$.

$$x_{k+1} = f(x_k, u_k, w_k) = \begin{pmatrix} \mathsf{x}_{k+1} \\ \mathsf{y}_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{pmatrix} \mathsf{x}_k + (V_k + n_v)\delta t \cos\theta_k \\ \mathsf{y}_k + (V_k + n_v)\delta t \sin\theta_k \\ \theta_k + (\omega_k + n_\omega)\delta t \end{pmatrix} \tag{69}$$

Linearizing this system about the point (node) $\mathbf{v} = (\mathsf{x}^p, \mathsf{y}^p, \theta^p)$, nominal control $u^p = (V^p, \omega^p)$, and zero noise, we get:

$$x_{k+1} = f(x_k, u_k, w_k) \approx f(\mathbf{v}, u^p, 0) + \mathbf{A}(x_k - \mathbf{v}) + \mathbf{B}(u_k - u^p) + \mathbf{G}w_k \tag{70}$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & -V^p \delta t \sin\theta^p \\ 0 & 1 & V^p \delta t \cos\theta^p \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \mathbf{G} = \begin{pmatrix} \cos\theta^p & 0 \\ \sin\theta^p & 0 \\ 0 & 1 \end{pmatrix} \delta t \tag{71}$$

Now let us look at:

$$\mathbf{A}^n \mathbf{B} = \begin{pmatrix} \cos\theta^p & -n(V^p \sin\theta^p) \\ \sin\theta^p & n(V^p \cos\theta^p) \\ 0 & 1 \end{pmatrix} \delta t \tag{72}$$

Thus, the controllability grammian for the discrete-time model is:

$$[\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}] = \begin{bmatrix} \cos\theta^p & 0 & \cos\theta^p & -V^p \sin\theta^p & \cos\theta^p & -2V^p \sin\theta^p \\ \sin\theta^p & 0 & \sin\theta^p & V^p \cos\theta^p & \sin\theta^p & 2V^p \cos\theta^p \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \delta t \tag{73}$$

whose rank is:

$$rank([\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}]) = 1 + rank(\begin{bmatrix} \cos\theta^p & -V^p \sin\theta^p \\ \sin\theta^p & V^p \cos\theta^p \end{bmatrix} \delta t) \tag{74}$$

For the $\delta t > 0$, we have:

$$rank([\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}]) = 2 + \mathbb{I}(V^p > 0) \tag{75}$$

If the nominal control is zero, $u^p = (V^p, \omega^p)^T = (0,0)^T$, we get: $rank([\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}]) = 2 < 3$. Thus, a linear controller cannot drive the unicycle to the PRM node.

Moreover, even a continuous time-invariant nonlinear control law cannot drive the unicycle to the PRM node, based on the necessary condition in Brockett's paper [21]: For smooth stabilization of the driftless regular systems to the point $\mathbf{v}$ the number of inputs has to be at least equal to the number of states. A regular system at $\mathbf{v}$ is a system in which the input vector fields are well-defined and independent in $\mathbf{v}$. Thus, since a unicycle model is a regular model and has less number of control inputs than number of states, it does not satisfy the Brockett's necessary condition and the stabilizing controller has to be either discontinuous and/or time-varying controller.

In the roadmaps in belief space, the situation is more complicated. In the belief space roadmaps, such as FIRM [18], [19], the controller has to drive the robot to a belief node in the belief space. Again, if the linearized system is controllable, using a linear stochastic controller such as stationary LQG controller, one can drive the robot belief to the belief node. However, if the linearized system about the desired point is not controllable, the stabilization to a belief node, if possible, is much more difficult than stabilization to a state node.

Here, in this paper, we circumvent the stabilization to the belief nodes and avoid singularities in controllability grammian, by exploiting the periodic controllers. Instead of reaching a belief node using a stabilizing controller, we reach belief nodes by choosing them along a periodic path and utilize a periodic controller to track the desired periodic path.

In addition to nonholonomic robots, controlling the non-stoppable robots on a roadmap such as PRM is a challenge. Non-stoppable robots have constraints on their controls. For example, many aircrafts, which cannot hover, or some surface vehicles, cannot reduce their velocity to less than a specific threshold u_{min}.