Online Replanning in Belief Space for Dynamical Systems: Towards Handling Discrete Changes of Goal Location

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

***Abstract*— This paper presents an online planning/replanning strategy for dynamical systems, in the presence of uncertainty. Online replanning capability allows us to handle discrete changes in the goal location. Multi-query graph-based methods, such as PRM in the deterministic setting, and FIRM in the stochastic setting, are suitable frameworks to serve this purpose. However, in these methods, local planners (along the edges) are responsible to drive the state/belief to the final node of the edge. Nevertheless, for dynamical systems, driving the system belief to a sampled belief is a challenge. In this paper, we provide an overview of the FIRM strategy that extends PRM-based planning to belief space, and sketch the application of it to dynamical/non-stoppable systems where the belief is stabilized to an orbit (periodic path) in belief space instead of a point. The method takes obstacles into account, while it provides a query-independent graph, as a means for online replanning.**

I. Introduction

Consider an application in which different tasks are assigned to a robotic system over time, where each task can be translated to a goal location for the motion planning module. Therefore, this problem calls for a query-independent motion planning method which can perform online replanning based on the assigned task. Considering uncertainty in system's motion and sensing, such an online replanning needs to be performed in the belief space (space of probability distributions over all possible states). Recent advances in belief space planning offer multi-query methods such as Information RoadMaps (IRM) [1], [2], which allows us to perform real-time replanning in belief space. However, the main instantiation of the abstract IRM framework is the Stationary Linear Quadratic Gaussian Feedback-based IRM (SLQG-FIRM) [1], [2], which is limited to systems that are stabilizable to equilibrium points in the state space.

This paper is concerned with systems that are not necessarily stabilizable to a point in the state space, such as *(i)* dynamical systems whose state is composed of robot's configuration and velocity, and/or *(ii)* systems with kinodynamical constraints, in particular, systems, whose velocity cannot fall below a certain threshold (referred to as non-stoppable systems in this paper). Consider a control problem where the system state is composed of the position and velocity $(x, \dot{x})$ of an object. Stabilizing this system to a state $(x = q, \dot{x} = V)$ where $V \neq 0$ is not possible since to stabilize the x part to q, the $\dot{x}$ must go to zero. As an example task for non-stoppable systems, consider surveillance with a fixed-wing aircraft, which has constraints on its velocity $\dot{x} > V_{min} > 0$ to maintain its lift requirement. Stabilizing such systems to a fixed state is impossible. In this paper, we propose a framework that circumvents the need for point stabilization in graph-based (roadmap-based) methods by means of stabilization to suitably designed periodic maneuvers. Then, we leverage the solution to the belief space, using graph-based planning methods.

Graph-based (roadmap-based) motion planning: Methods such as the Probabilistic Roadmap Method (PRM) and its variants [3]–[5] have shown a great success in dealing with many motion planning problems. To generalize the query-independent PRM to belief space, a point-to-point motion in belief space is required, which was a missing behavior in initial attempts to generalize the PRM to the belief space [6]–[8]. Feedback-based Information RoadMap (FIRM) [1], [2] provides such a graph in belief space by embedding the point-to-point motion behaviour in belief space using belief stabilizers. Therefore, the roadmap construction does *not* depend on the submitted query and hence can be constructed offline independent of the future queries. However, as mentioned, the initial instantiation of this framework (SLQG-FIRM) does not handle non-stoppable systems.

This paper provides an overview presentation of FIRM and sketches the construction of a concrete realization of IRM for systems that are not stabilizable to equilibrium points such as fixed-wing aircraft. To do so, first, periodic-node PRM in state space is introduced. Then, we leverage it to the belief space. Then, we discuss the replanning process and illustrate how this framework can handle online changes in the goal location. Finally, we discuss how the method can be extended to the case where a set of goal locations are available beforehand and we need to plan the order in which the goals need to be attained.

II. Periodic-Node PRM

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 can form a roadmap in belief space such that the belief nodes are reachable without a point-stabilization process. Let us denote

A. Agha-mohammadi and N. Amato are with the Dept. of Computer Science and Engineering and S. Chakravorty is with the Dept. of Aerospace Engineering, Texas A&M University, TX 77843, USA. Emails: {aliagha,schakrav,amato}@tamu.edu

The work of Agha-mohammadi and Chakravorty is supported in part by NSF award RI-1217991 and AFOSR Grant FA9550-08-1-0038 and the work of Agha-mohammadi and Amato is supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-0830753, IIS-0917266, IIS-0916053, EFRI-1240483, RI-1217991, by NSF/DNDO award 2008-DN-077-ARI018-02, by NIH NCI R25 CA090301-11, by DOE awards DE-FC52-08NA28616, DE-AC02-06CH11357, B575363, B575366, by THECB NHARP award 000512-0097-2009, by Samsung, Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

the motion model with $x_{k+1} = f(x_k, u_k, w_k)$, where state, control, and process noise at the k-th time step are denoted by x_k, u_k, and w_k, respectively.

Similar to traditional PRM, PNPRM 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, which 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 $O^j := (x_k^j, u_k^j)_{k\geq 0}$, where $x_{k+1}^j = f(x_k^j, u_k^j, 0)$, $x_{k+T}^j = x_k^j$, and $u_{k+T}^j = u_k^j$. The set of PNPRM nodes chosen on O^j is denoted by $\mathbb{V}^j = \{\mathbf{v}_1^j, \mathbf{v}_2^j, \cdots, \mathbf{v}_m^j\}$ where $\mathbf{v}_\alpha^j = x_{k_\alpha}^j$ for some $k_\alpha \in \{1, \cdots, T\}$. Edges in PNPRM do not connect nodes to nodes, but they connect orbits to orbits respecting all 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 edges are the lines that are tangent to the orbits. Figure 1 shows a simple PNPRM with three orbits O^i, O^r, and O^j. On each orbit four nodes are selected which are drawn (dots) with different colors. Edges $\mathbf{e}^{ij}$ and $\mathbf{e}^{rj}$ connect the corresponding orbits. Shown covariance ellipses in Fig. 1 associated with each PNPRM node is discussed further below.

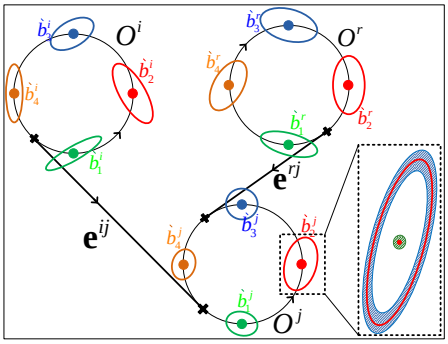


Fig. 1. A simple PNPRM with three orbits, twelve nodes, and two edges. $\grave{b}_\alpha^l = (\mathbf{v}_\alpha^l, \check{P}_{k_\alpha}^l)$ is the center of corresponding belief nodes, where $\check{P}_{k_\alpha}^l$'s are shown by their 3σ-ellipse. As an example of FIRM node, the magnified version of 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. Planning under uncertainty

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 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 [9], [10], whose solution π is computed by solving the following Dynamic Programming (DP) equation:

$$J(b) = \min_u \{c(b,u) + \int_{\mathbb{B}} p(b'|b,u) J(b') db'\}, \tag{1a}$$

$$\pi(b) = \arg\min_u \{c(b,u) + \int_{\mathbb{B}} p(b'|b,u) J(b') db'\}, \tag{1b}$$

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. However, if we can construct a graph in the belief space, we can reduce this intractable DP to a DP over the representative graph, so called the Feedback-based Information RoadMap (FIRM).

FIRM nodes and edges: Corresponding to each PNPRM node $\mathbf{v}_\alpha^i$, we have a FIRM node $B_\alpha^i \subset \mathbb{B}$, whose construction is discussed in Sect. V. Each FIRM edge is a local 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, i.e., to $\cup_\gamma B_\gamma^j$. The construction of these local controllers is discussed in Sect. V. The set of controllers available at each FIRM node B_α^i is denoted by $\mathbb{M}(i,\alpha) := \{\mu^{\alpha,ij} \in \mathbb{M} | \exists \mathbf{e}_{ij}\}$.

FIRM MDP: In the presence of obstacles F, we add F as a node (failure node) to the graph. Therefore, we can form the DP on the FIRM graph as follows [1], [2]:

$$J^g(B_\alpha^i) = \min_{\mu^{\alpha,ij} \in \mathbb{M}(i,\alpha)} C^g(B_\alpha^i, \mu^{\alpha,ij}) + J^g(F)\mathbb{P}^g(F|B_\alpha^i, \mu^{\alpha,ij}) + \sum_{\gamma=1}^m J^g(B_\gamma^j)\mathbb{P}^g(B_\gamma^j|B_\alpha^i, \mu^{\alpha,ij}), \ \forall \alpha, i, j \tag{2a}$$

$$\pi^g(B_\alpha^i) = \arg\min_{\mu^{\alpha,ij} \in \mathbb{M}(i,\alpha)} C^g(B_\alpha^i, \mu^{\alpha,ij}) + J^g(F)\mathbb{P}^g(F|B_\alpha^i, \mu^{\alpha,ij}) + \sum_{\gamma=1}^m J^g(B_\gamma^j)\mathbb{P}^g(B_\gamma^j|B_\alpha^i, \mu^{\alpha,ij}), \ \forall \alpha, i, j \tag{2b}$$

where, $J^g(B)$ is the optimal cost-to-go for the graph node B. The cost-to-go for the goal node is set to be zero. $J^g(F)$ is a user-defined suitably high cost for hitting obstacles. $\mathbb{P}(B_\gamma^j|B_\alpha^i, \mu^{\alpha,ij})$ and $\mathbb{P}(F|B_\alpha^i, \mu^{\alpha,ij})$ are transition probabilities from B_α^i to B_γ^j and F, respectively, under the

local controller $\mu^{\alpha,ij}$. Finally, $C^g(B^i_\alpha, \mu^{\alpha,ij}) > 0$ is the cost of taking the local controller $\mu^{\alpha,ij}$ at node B^i_α. The success probability under this framework and its probabilistic completeness are discussed in [11].

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 (2). 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 (2) are computed.

PLQG 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. Similarly, Periodic LQG (PLQG) is composed of a periodic Kalman filter and a periodic LQR. PLQG is designed to track a given periodic trajectory [12]. In [13] we review the periodic LQG controller in detail.

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

As discussed in [13], designing PLQG controllers along the orbits, the belief can be derived into FIRM nodes designed along that orbit. Let us denote the PLQG designed for the j-th orbit by μ^j_k. We refer to μ^j_k as the j-th node controller. Also, let us denote mean and covariance of estimation by $\widehat{x}^+$ and P. Thus in Gaussian space, $b = \mathcal{N}(\widehat{x}^+, P)$, or $b \equiv (\widehat{x}^+, P)$.

In [13], we have shown that for each $\mathbf{v}^j_\alpha$, we can find a unique belief $\grave{b}^j_\alpha$ and make a FIRM node

$$B^j_\alpha = \{b : \|b - \grave{b}^j_\alpha\| < \epsilon\}, \tag{3}$$

such that $\cup_\alpha B^j_\alpha$ is a reachable region under the node-controller μ^j_k. $\|\cdot\|$ in above equation is a suitable norm in Gaussian belief space (see [13]), and ϵ defines the size of FIRM node.

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, *ii)* the edge itself $\mathbf{e}^{ij}$, and *iii)* a part of O^j that connects the ending point of $\mathbf{e}^{ij}$ to x^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 the 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 the 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 $\mu^{\alpha,ij}$.

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 [14] 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 $\Phi^{\alpha,ij} = \mathbb{E}[\sum_{k=1}^{\mathcal{T}} \mathrm{tr}(P^{\alpha,ij}_k)]$, 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. Moreover, as we are also interested in faster paths, we take into account the corresponding mean stopping time, i.e., $\widehat{\mathcal{T}}^{\alpha,ij} = \mathbb{E}[\mathcal{T}^{\alpha,ij}]$, 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.

$$C(B^i_\alpha, \mu^{\alpha,ij}) = \xi_1 \Phi^{\alpha,ij} + \xi_2 \widehat{\mathcal{T}}^{\alpha,ij}. \tag{4}$$

D. Construction of PLQG-FIRM and Planning With it

The crucial feature of FIRM is that it can be constructed offline and stored, independent of future queries. Algorithm 1 details the construction of FIRM. Algorithm 2 illustrates this procedure.

Algorithm 1: Offline Construction of PLQG-FIRM

```
input: Free space map, X_free
Construct a PNPRM with orbits O = {O^j}, nodes
  V = {v^j_α}, and edges E = {e^ij};
forall the PNPRM orbits O^j ∈ O do
    Design the node-controller (periodic LQG) μ^j_k;
    Construct m FIRM nodes V^j = {B^j_1, ··· , B^j_m}
      using (3);
forall the (B^i_α, e^ij) pairs do
    Design the edge-controller μ̄^{α,ij}_k, (see Section V-B);
    Construct the local controller μ^{α,ij}_k by concatenating
      edge-controller μ̄^{α,ij}_k and node-controller μ^j_k;
    Generate sample belief paths b_{0:T} and state paths
      x_{0:T} induced by controller μ^{α,ij} invoked at B^i_α;
    Compute the transition probabilities
      P^g(F|B^i_α, μ^{α,ij}) and P^g(B^j_γ|B^i_α, μ^{α,ij}) for all γ
      and transition cost C^g(B^i_α, μ^{α,ij});
return G = (V = ∪_{j=1}^n V^j, M = {μ^{α,ij}});
```

Given that the FIRM graph is computed offline, the online phase of planning/replanning is explained in Algorithm 2.

Algorithm 2: Online Planning with PLQG-FIRM

input: Initial and goal beliefs b_0, b^{goal}, FIRM graph $\mathcal{G}$
Connect initial and goal beliefs to the FIRM graph as described in Sect. V;
Compute cost-to-go J^g and feedback π^g by solving the DP in (2);
while *the goal node is not reached* **do**
  Denote the current FIRM node as B^i_α;
  Choose the next local controller $\mu^{\alpha,ij} = \pi^g(B^i_\alpha)$;
  while $(\nexists B^j_\gamma, s.t.,\ b_k \in B^j_\gamma)$ *and "no collision"* **do**
    Apply the control $u_k = \mu^{\alpha,ij}_k(b_k)$ to the system;
    Get the measurement z_{k+1} from sensors;
    if *Collision happens* **then return** Collision;
    Update belief as $b_{k+1} = \tau(b_k, \mu^{\alpha,ij}_k(b_k), z_{k+1})$;

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 as follows:

$$x_{k+1} = f(x_k, u_k, w_k) = \begin{pmatrix} \mathsf{x}_k + (V_k + n_v)\delta t \cos\theta_k \\ \mathsf{y}_k + (V_k + n_v)\delta t \sin\theta_k \\ \theta_k + (\omega_k + n_\omega)\delta t \end{pmatrix}, \quad (5)$$

where $x_k = (\mathsf{x}_k, \mathsf{y}_k, \theta_k)^T$ describes the robot state (2D position and heading angle). The vector $u_k = (V_k, \omega_k)^T$ is the control vector consisting of linear velocity V_k and angular velocity ω_k. The motion noise vector is denoted by $w_k = (n_v, n_\omega)^T \sim \mathcal{N}(0, \mathbf{Q}_k)$. 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. Denoting the vector from robot to the j-th landmark by ${}^j\mathbf{d} = [{}^jd_1, {}^jd_2]^T := [{}^1x, {}^2x]^T - L_j$, measuring L_j can be modeled as follows:

$${}^jz = {}^jh(x, {}^jv) = [\|{}^j\mathbf{d}\|, \text{atan2}({}^jd_2, {}^jd_1) - \theta]^T + {}^jv, \quad (6)$$

where, ${}^jv \sim \mathcal{N}(\mathbf{0}, {}^j\mathbf{R})$ and ${}^j\mathbf{R} = \text{diag}((\eta_r\|{}^j\mathbf{d}\| + \sigma^r_b)^2, (\eta_\theta\|{}^j\mathbf{d}\| + \sigma^\theta_b)^2)$. 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 $\sigma^r_b = 0.01$ meter and $\sigma^\theta_b = 0.5$ degrees are the bias standard deviations. A similar model for range sensing is used in [6]. 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 $v \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{R} = \text{diag}({}^1\mathbf{R}, \cdots, {}^{N_L}\mathbf{R})$.

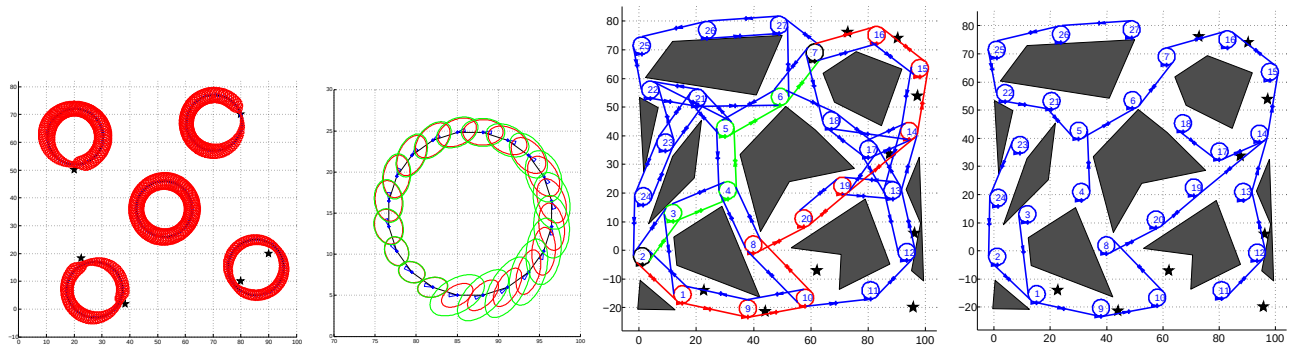
We first show a typical solution of Discrete Periodic Riccati Equation (DPRE) on orbits, which is indeed the periodic covariance trajectory, to which the estimation covariance is converging. Fig. 2(a) shows a simple environment with six radio beacons (black stars). For illustration purposes, we choose five large circular orbits and every orbit is discretized to 100 steps. Thus the solution of DPRE 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. In Fig. 2(b), we illustrate the covariance convergence. As can be seen in Fig. 2(b), the initial covariance is three times larger than the limiting covariance, and in less than one period it converges to the 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, compared to the orbit in Fig. 2(b).

Figure 2(c) 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.2(c) 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 DPREs on each orbit and design its corresponding node-controller (PLQG). Then, we construct the FIRM nodes based on (3) (see [13] for details).

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 [14] 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 [1]. Table I shows these quantities for several $(B^i_\alpha, \mu^{\alpha,ij})$ pairs corresponding to Fig.2(c), where $M = 101$.

Plugging the computed transition costs and probabilities into (2), 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. 2(d) 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 each orbit we may have more than one outgoing arrow in Fig. 2(d).

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 the efficient replanning procedure. In other words, since π^g is independent of query, if due to some unmodeled large disturbances, the system deviates significantly from the planned path, it suffices to bring the system back to the closest FIRM node and from thereon the optimal plan is already known, i.e., π^g drives the robot to the goal region as shown in Fig. 2(d).



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

Fig. 2. (a) Five orbits ($T = 100$) and corresponding periodic estimation covariances (solution of DPRE). (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., $P_0 = 3\tilde{P}_0$. (c) A sample PNPRM with circular orbits. Number of each orbit is written in its center. Nine landmarks (black stars) and obstacles (gray polygons) are also shown. The moving directions on orbits and on edges are shown by little triangles with a cross in their heading direction. Nodes 2 and 7 (distinguished in black) are start and goal nodes, respectively. Shortest path (green) and the most-likely path (red) under FIRM policy are also shown. (d) Assuming on each orbit, there exists a single node, the feedback π^g is visualized for all FIRM nodes.

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
$\mathbb{E}[\mathcal{T}^{\alpha,ij}]$	63.6703	82.6747	62.5882	58.2000	51.7033	50.2755	35.4653

We show the most likely path under the π^g, in red in Fig. 2(c). The shortest path is also illustrated in Fig. 2(c) 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. DISCUSSION ON MULTI-GOAL SCENARIOS

In this paper, we focused on the case where the goal locations are provided by the user over time and we discussed how the framework can replan accordingly. However, from a different perspective one can consider a scenario where a set of goal locations are already given and the method should choose the order in which the goals need to be attained. An example application of such a scenario is a surveillance task where the robot needs to visit a set of places (goal locations). A similar problem in the absence of sensing uncertainty is studied in [15].

In this case, associated with each goal there is a feedback π^g and a success probability. Thus, basically, the problem of planning the order of goals becomes a version of Travelling Salesman Problem (TSP). In other words, starting from each goal (city in TSP), we know the cost and success probability of reaching the next goal through underlying FIRM feedback. Thus, we can run a TSP on the space of goals and find the best order of goals that leads to the least overall cost.

Figure 3 illustrates different layers in this framework. In the lowest layer local controllers are designed between neighboring nodes, and the transition costs and probabilities are transferred to the higher level. In the second layer the FIRM graph gets constructed, where each node is indeed a probability distribution. Solving dynamic programming on this graph, one can compute the optimal graph feedback corresponding to each goal node. Transferring the graph feedbacks, cost-to-go's, and success probabilities to the task layer, one can make a graph whose nodes consist of the start node and all goal nodes. Based on costs and success probabilities (from the start node to goal nodes and from one goal to another) in the graph layer, the solution of the travelling salesman problem on this graph leads to an optimal plan (order) in reaching the set of goals.

VIII. CONCLUSION

This paper proposes a solution to the problem of stochastic planning for non-stoppable systems and systems with dynamics, such as aircraft. The Periodic-Node PRM (PNPRM) is introduced whose nodes lie on periodic trajectories, called orbits. Exploiting the properties of periodic LQG controllers on orbits, the local planners in the PNPRM-based FIRM framework are realized by periodic LQG controllers, such that the distribution over the belief converges to a periodic distribution, and it is shown that by suitably choosing belief node regions along the orbits, the belief node reachability is achieved. Therefore, while taking obstacle collision probabilities into account, the proposed method can generate a graph in the belief space, with reachable nodes, which can be used for stochastic control of the non-stoppable systems and systems with dynamics.

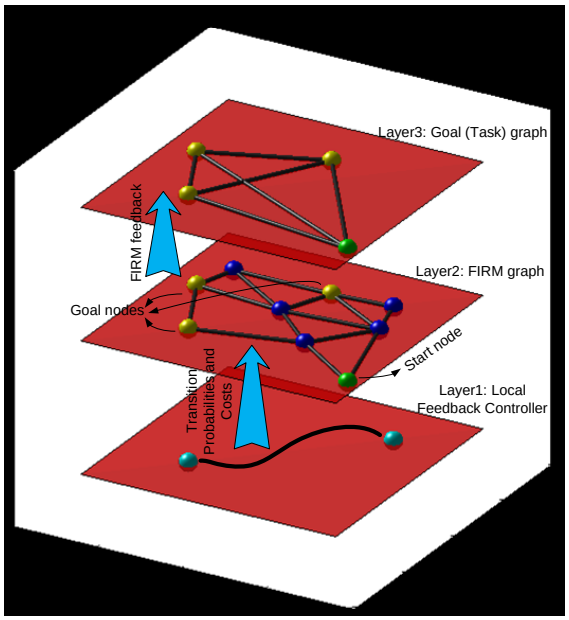


Fig. 3. This figure illustrates different layers of the problem. The first layer contains local feedback controllers along PRM edges; the second layer contains the FIRM graph constructed based on the local controllers; The third contains the task graph (or the goal graph), on which we can solve a TSP to find the best sequence of goals to achieve.

REFERENCES

[1] A. Agha-mohammadi, S. Chakravorty, and N. Amato, "Motion planning in belief space using sampling-based feedback planners," *Technical Report: TR11-007, Parasol Lab., CSE Dept., Texas A&M University*, 2011.

[2] ——, "FIRM: Feedback controller-based Information-state RoadMap -a framework for motion planning under uncertainty-," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[3] 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.

[4] N. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo, "OBPRM: An Obstacle-Basaed PRM for 3D workspaces," in *International Workshop on the Algorithmic Foundations of Robotics*, 1998, pp. 155–168.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

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

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

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

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

[15] S. Kumar and S. Chakravorty, "Multi-agent generalized probabilistic roadmaps: Magprm," in *IROS*, 2012, pp. 3747–3753.