FIRM: Feedback Controller-Based Information-State Roadmap, A Framework for Motion Planning Under Uncertainty

Ali-akbar Agha-mohammadi, Suman Chakravorty, Nancy Amato
Technical Report TR11-001
Parasol Lab.
Texas A&M University January 10, 2011

Abstract

Direct transformation of sampling-based motion planning methods to the Information-state (belief) space is a challenge. The main bottleneck for roadmap-based techniques in belief space is that the incurred cost on different edges of the graph are not independent of each other. In this paper, we generalize the Probabilistic RoadMap (PRM) framework to Feedback controller-based Information-state RoadMap (FIRM) that takes into account motion and sensing uncertainty in planning. The FIRM nodes and edges lie in belief space and the crucial feature of FIRM is that the costs associated with different edges of FIRM are independent of each other. Therefore, this construct essentially breaks the "curse of history" in the original Partially Observable Markov Decision Process (POMDP), which models the planning problem. Further, we show how obstacles can be rigorously incorporated into planning on FIRM. All these properties stem from utilizing feedback controllers in the construction of FIRM.

I. Introduction

Sampling-based path planning algorithms such as Probabilistic Roadmap (PRM) [1] method, Rapidly exploring Randomized Trees (RRT) [2], or their variants have shown a great success in solving robot motion planning problems in the absence of uncertainty. However, direct transformation of these methods to planning under uncertainty is a challenge. The first issue is ensuring that the roadmap nodes are reachable. The second challenge is that the incurred cost on different edges of roadmap depends on each other, which violates the basic assumption in roadmap based methods.

In this paper, we generalize the PRM framework to Feedback controller-based Information-state RoadMap (FIRM) that takes into account both motion and sensing uncertainties. The probability density function (pdf) over the state is called *belief* or *information-state*. The FIRM is constructed as a roadmap in belief space, where its nodes are small subsets of belief space and the edges of FIRM are Markov chains in belief space. It is the first method that generalizes the PRM to belief space in such a way that the incurred cost on different edges of roadmap are independent of each other, while providing a straightforward approach to sample reachable nodes in belief space. These properties are the direct consequence of utilizing feedback controllers in the construction of FIRM. Planning under motion and sensing uncertainty is essentially a Partially Observable Markov Decision Process (POMDP), and indeed FIRM breaks the curse of history in such POMDPs and provides the optimal policy over the roadmap instead of only a nominal path.

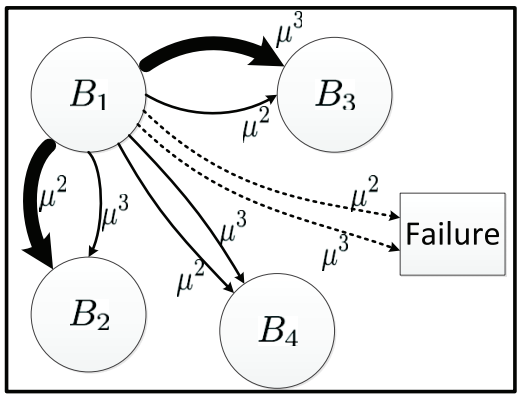

Fig. 1. Decision process in FIRM is depicted, given current belief $\mathbf{b}$ belongs to B_1. Assuming k-nearest neighbors of B_1 are $A(1) = \{2, 3\}$ only controllers μ^2 and μ^3 can be invoked. Choosing any of these controllers induces a stochastic edge (Markov chain) that can stop in any node B_i or failure state. The wide solid lines reflect the fact that if controller μ^j is invoked, the probability of landing in B_j is significantly higher than probability of landing in $B_m,\ m \neq j$. Probability of landing in failure state (dashed arrows) depends on obstacles' configuration.

A. Agha-mohammadi and N. Amato are with the Computer Science and Engineering Department, Texas A&M University, TX 77843, USA.Email: aliagha@tamu.edu and Email: amato@cse.tamu.edu

S. Chakravorty is with the Aerospace Department, Texas A&M University, TX 77843, USA. Email: chakrav@neo.tamu.edu

The incorporation of the obstacles in planning on the roadmaps is also a challenge, without edge independence, because it either entails costly repeated computations of collision probabilities, or some collision measure has to be designed that in general cannot capture the true collision probabilities and may lead to overly conservative plans. However, in FIRM, owing to the independence of edges, we can compute the collision probabilities offline and incorporate obstacles in planning over FIRM that leads to more reliable and less conservative plans.

In next section, we review the most relevant work. In section III we derive FIRM as a computationally tractable approximation of POMDP. In section IV we detail how the assumptions inherent in FIRM can be satisfied and construct a FIRM. Experimental results are presented in section V.

II. Related Work

In recent years, there has been a concerted effort to incorporate uncertainty into the sampling-based motion planning methods. A class of these methods deal with map uncertainty, such as [3]–[5], while the methods in [6]–[8] deal with motion uncertainty. Another class of methods that are most related to FIRM consider both motion and sensing uncertainties in planning, such as [9]–[15]. In following we briefly discuss the planning scheme in these references and place the FIRM into context.

In [9], the planning is done using graph search and constraint propagation on a grid-based representation of the space. In [10], the best path is found among the finite number of RRT paths by simulating the performance of LQG on all of them. [11] plans in continuous space by finding the best nominal path through nonlinear optimization methods. [12] and [13] are PRM-based approaches, where the best path is found through breadth-first search on the Belief roadmap.

In all these methods, a best nominal path is computed offline. This nominal path is fixed regardless of the process and sensing noises in the execution phase. [11] performs replanning when large deviations happen, which is a computationally expensive procedure since all the costs along path has to be reproduced for the new initial belief. In FIRM, however, the best feedback policy, i.e. a mapping from belief space to actions, is computed offline, which is the true goal of planning under motion and sensing uncertainty (POMDPs). In [14], the nominal path is updated dynamically in a receding horizon control approach, which entails repeatedly solving open loop optimal control problems at every time step. In [15], the value function at sampled milestones are computed and thus the optimal policy is computed rather than the optimal nominal controls.

In the methods that account for sensing uncertainty, the state has to be estimated based on measurements. To handle unknown future measurements in planning stage, methods [9]–[15] but [10], [15] consider only the maximum likelihood (ML) observation sequence to predict the estimation performance. In contrast, the FIRM takes all possible future observations into account in planning.

In presence of obstacles, due to the dependency of collision events in different time steps, it is a burdensome task to include the collision probabilities in planning. Thus, the methods such as [9], [10], [14] design some safety measure to account for obstacles in planning. However, in FIRM, collision probabilities can be computed and seamlessly incorporated in planning stage.

A. Contributions and method highlights

FIRM graph is a generalization of the PRM graph, whose nodes are small subsets of belief space and edges are Markov chains induced by feedback controllers. As a result planning on the FIRM is a Markov Decision Process (MDP) defined on FIRM nodes, which can be solved using standard Dynamic Programming techniques.

Inducing reachable belief nodes: FIRM samples nodes in the robot state space and then, utilizes feedback controllers to automatically induce unique beliefs, associated with each of these state space nodes. The controller can drive the belief into the neighborhood of these belief states in finite time and thus ensures reachability. This way the FIRM addresses the hard task of sampling in reachable belief space that is usually required in belief space planning [15]–[17].

Breaking the curse of history: A fundamental contribution of the FIRM is that the optimal action, at some node, does not depend on the traversed nodes and taken actions in past, i.e., it is independent of the history of the information process. This is a direct consequence of inducing reachable belief nodes using the feedback controllers, which essentially breaks the curse of history in POMDPs. In addition, the sampling based nature of the method borrowed from PRM allows us to ameliorate the curse of dimensionality.

Efficient planning: The construction of FIRM is offline and thus, the online planning (and replanning) is feasible. Moreover, in the FIRM, the optimal feedback policy, instead of the nominal path, is computed offline. This is done by solving the dynamic programming problem associated with the higher level MDP on belief nodes induced by the feedback controllers.

Incorporating obstacles in planning: In the FIRM framework, the collision probabilities can be computed, which leads to more accurate plans, as opposed to a simplified collision measure, that may lead to conservative plans. The obstacles induce a *failure node* in the higher level MDP, into which the robot can be absorbed. Further, due to the offline construction of FIRM, the heavy computational burden of estimating collision probabilities can be done offline.

III. The POMDP to FIRM Transformation

In this section, we detail how to transform a POMDP problem into a FIRM. In the first subsection, POMDP problem is briefly outlined. In subsection B, we develop the transformation for the obstacle free case, and in subsection C, we show how to incorporate obstacles into the planning.

A. Preliminaries

Consider a controlled hidden Markov model with hidden state $\mathbf{X} \in \mathbb{X}$, control $\mathbf{u} \in \mathbb{U}$, and observation $\mathbf{z} \in \mathbb{Z}$, with transition probability model $p(\mathbf{X}'|\mathbf{X}, \mathbf{u})$ and observation model $p(\mathbf{z}|\mathbf{X})$. Let $\mathbf{z}_{0:k}$ denote the set of observations $\mathbf{z}_k$ till time k. Then, the information-state (belief) of the system at time k is defined as the probability distribution of the underlying system state $\mathbf{X}$ given $\mathbf{z}_{0:k}$, i.e, $p(\mathbf{X}|\mathbf{z}_{0:k})$, and denoted by $b(\mathbf{X})$ or interchangeably by the parameter vector $\mathbf{b}$ representing $b(\mathbf{X})$. Let the space of all such beliefs be denoted by the belief space $\mathbb{B}$. Then, the infinite horizon POMDP problem can be cast as the following stationary Dynamic Programming problem on the belief space $\mathbb{B}$ [18]:

$$J(\mathbf{b}) = \min_{\mathbf{u}} \{c(\mathbf{b}, \mathbf{u}) + \int_{\mathbb{B}} p(\mathbf{b}'|\mathbf{b}, \mathbf{u}) J(\mathbf{b}') d\mathbf{b}'\}, \forall \mathbf{b} \in \mathbb{B}, \tag{1}$$

where $c(\mathbf{b}, \mathbf{u})$ is the incremental cost of taking action $\mathbf{u}$ at belief state $\mathbf{b}$, $J(\mathbf{b})$ is the optimal cost-to-go from belief state $\mathbf{b}$, and $p(\mathbf{b}'|\mathbf{b}, \mathbf{u})$ represents the transition probability density over belief states, given that control $\mathbf{u}$ is taken at belief state $\mathbf{b}$. This transition probability can be derived using Bayes rule and the law of total probability. However, as is well known, the above DP equation is exceedingly difficult to solve since it is defined over whole belief space, and suffers from curse of history. In the following, we show how the POMDP can be reduced to a computationally tractable FIRM.

B. Obstacle-free FIRM

Let us consider the DP in (1), through which the cost-to-go function can be computed for the belief MDP problem. We further restrict the problem's scope by following assumption.

Assumption 1: It is assumed that the planning goal is to transfer the robot into some pre-specified neighbourhood B_g of a uniquely defined belief state $\mathbf{b}^g_\infty$ associated with the goal state $\mathbf{X}_g$, with probability 1, following which the system can remain there without incurring any further cost.

This is known as a stochastic shortest path problem [18]. Now, consider a set of sampled nodes in the state space of the robot $\{\mathbf{n}_i\}_{i=1}^N$ that includes the goal node, i.e. the node into whose vicinity we want to transfer the robot.

Assumption 2: We assume that corresponding to every state node $\mathbf{n}_i$, there exists a unique belief state $\mathbf{b}^i_\infty$, an associated feedback controller $\mathbf{u} = \mu^i(\mathbf{b})$ and a neighbourhood $N(\mathbf{b}^i_\infty)$ of the belief $\mathbf{b}^i_\infty$, such that given any $\mathbf{b} \in N(\mathbf{b}^i_\infty)$, the controller can drive the belief state into $\cup_m B_m$ in finite time with probability one, where B_m is a small neighbourhood of $\mathbf{b}^m_\infty$ and termed m-th belief neighbourhood or m-th belief node. Further, we assume that the goal neighbourhood B_g is one of the belief nodes B_i, i.e. the problem is solved if the system is transferred into B_g.

Note that node $B_i \subset N(\mathbf{b}^i_\infty)$. The feedback controller μ^i is called i-th node-controller. Under the node controller $\mu^i(\mathbf{b})$, the belief state evolves according to a Markov chain whose transition density function is denoted by $p^{\mu^i}(\mathbf{b}'|\mathbf{b}) := p(\mathbf{b}'|\mathbf{b}, \mu^i(\mathbf{b}))$. Thus, the node-controller essentially induces the Markov chain $p^{\mu^i}(\mathbf{b}'|\mathbf{b})$ over belief space. Irreducibility of this chain is the sufficient condition for the assumption 2 to be satisfied. Irreducibility essentially implies that the Markov chain can go from any point in the belief space to any non-zero measure set in the belief space in finite time with probability one. Due to the irreducibility, under node-controller μ^i which tries to draw the belief into neighbourhood B_i, there is still a finite probability that the system ends up in a neighbourhood B_m instead of the target neighbourhood B_i, and hence, the absorption into the set $\cup_m B_m$ instead of only the target neighbourhood B_i (see Fig. I). In next section we discuss how such a node-controller may be constructed.

Assumption 3: It is assumed that the belief process can stop if and only if it enters the region $\cup_m B_m$. Further, once the system is in any of the nodes B_i, it is allowed to invoke one of the controllers $\mu^j(\cdot)$ among $j \in A(i)$, the k-nearest neighbour set of i, that in turn will draw the system toward the region $\cup_m B_m$ with highest probability of landing in B_j.

Based on these assumptions, the original MDP in belief space, formulated using DP in (1), is now turned into a Semi-Markov Decision Process (SMDP) [19] on the belief space or equivalently an MDP on the continuous regions B_i. We call this restricted form of original POMDP, the Feedback controller-based Information-state RoadMap (FIRM), whose DP formulation is:

$$J(\mathbf{b}) = \min_{j \in A(i)} C^{\mu^j}(\mathbf{b}) + \int_{\cup_m B_m} p^{\mu^j}(\mathbf{b}'|\mathbf{b}) J(\mathbf{b}') d\mathbf{b}', \forall \mathbf{b} \in B_i, \forall i. \tag{2}$$

In the equation above, $C^{\mu^j}(\mathbf{b})$ represents the expected cost of invoking node-controller $\mu^j(\cdot)$ starting at belief state $\mathbf{b}$ till the node-controller stops executing once the belief enters the region $\cup_m B_m$. Mathematically:

$$C^{\mu^j}(\mathbf{b}) = \sum_{t=0}^{\mathcal{T}} c(\mathbf{b}_t, \mu^j(\mathbf{b}_t)|\mathbf{b}_0 = \mathbf{b}), \tag{3}$$

where $\mathcal{T}$ is a random stopping time denoting the time at which the belief state enters one of the nodes B_m. The pdf $p^{\mu^j}(\mathbf{b}'|\mathbf{b})$ represents the belief transition pdf given that μ^j is invoked at $\mathbf{b}$.

The FIRM, though computationally more tractable than the original POMDP, is defined on the continuous neighbourhoods B_i and thus, still formidable to solve. Instead, let us consider the following piecewise constant approximation:

$$J(\mathbf{b}) \approx J(\mathbf{b}^i_\infty), \quad C^{\mu^j}(\mathbf{b}) \approx C^{\mu^j}(\mathbf{b}^i_\infty), \quad \forall \mathbf{b} \in B_i, \forall i. \tag{4}$$

Given the above approximation, FIRM in (2) can be approximated as follows:

$$J(\mathbf{b}^i_\infty) = \min_{j \in A(i)} C^{\mu^j}(\mathbf{b}^i_\infty) + \sum_{m=1}^{N} P^{\mu^j}(B_m|\mathbf{b}^i_\infty) J(\mathbf{b}_m), \forall i, \tag{5}$$

where, $P^{\mu^j}(B_m|\mathbf{b}^i_\infty)$ represents the probability that the controller μ^j invoked at $\mathbf{b}^i_\infty$ takes $\mathbf{b}^i_\infty$ into the B_m before it gets absorbed into any B_l, where $l \neq m$. Note that $\sum_{m=1}^{N} P^{\mu^j}(B_m|\mathbf{b}^i_\infty) = 1$ if $\mathbf{b}^i_\infty$ can be driven into $\cup_m B_m$ by μ^j, which is guaranteed if assumption 2 holds.

Equation (5) is an arbitrarily accurate approximation to the original FIRM in (2) given that the functions $C^{\mu^j}(\cdot)$ and $P^{\mu^j}(\cdot|\cdot)$ 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 is represented by $\mathbf{b}^i_\infty$ for the purpose of decision making. Abusing the notation and defining $J(B_i) := J(\mathbf{b}^i_\infty)$, $C^{\mu^j}(B_i) := C^{\mu^j}(\mathbf{b}^i_\infty)$, and $P^{\mu^j}(\cdot|B_i) := P^{\mu^j}(\cdot|\mathbf{b}^i_\infty)$ leads to the equation:

$$J(B_i) = \min_{j \in A(i)} C^{\mu^j}(B_i) + \sum_{m=1}^{N} P^{\mu^j}(B_m|B_i) J(B_m), \forall i. \tag{6}$$

Thus, (6) turns the original POMDP into a finite N-state MDP defined on the abstract "belief nodes" $\{B_i\}_{i=1}^N$. Given $C^{\mu^j}(\cdot)$ and $P^{\mu^j}(\cdot|\cdot)$, this problem can easily be solved using standard DP techniques such as value/policy iteration to yield a feedback policy π on the higher level embedded MDP defined on the belief nodes B_i. Given that the system stops in node B_i, this policy determines which node-controller μ^{j^*} has to be invoked, where $j^* = \pi(B_i)$. In order to solve (6), the generalized costs $C^{\mu^j}(\cdot)$ and transition probabilities $P^{\mu^j}(\cdot|\cdot)$ need to be evaluated. We discuss how to compute these, in Section IV on FIRM construction.

C. Incorporating Obstacles into FIRM

In the presence of obstacles, we can never assure that any $\mathbf{b} \in B_i$ can be driven into $\cup_m B_m$ with probability one, under node-controller $\mu^j(\cdot)$. 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., $F = \mathbb{X} - \mathbb{X}_{free}$). The extended state (e-state) $\mathfrak{X}_k = (\mathbf{X}_k, \mathbf{b}_k)$, consisting of the state-belief pair, under the action of node-controller μ^j, evolves according to a Markov chain, called extended chain (e-chain), whose transition pdf is denoted by $p^{\mu^j}(\mathfrak{X}_{k+1}|\mathfrak{X}_k)$. The system is deemed to have failed if $\mathbf{X}_k \in F$ for some time step k, or is deemed to have been successful if $\mathbf{b}_k \in \cup_m B_m$ before hitting F for some time step k.

Assumption 4: In the presence of obstacles, it is assumed that the belief process can stop in two cases: (i) successful stop: which happens if and only if belief enters into a node $\mathbf{b}_k \in B_m$ and $p(\mathbf{X}|\mathbf{z}_{0:k}) = b^m_\infty(\mathbf{X})$, for any m; (ii) failed stop: which happens if and only if $\mathbf{X}_k \in F$.

Equality $p(\mathbf{X}|\mathbf{z}_{0:k}) = b^m_\infty(\mathbf{X})$ always holds in obstacle-free space for sufficiently small B_m. However, in presence of obstacles, if the estimator does not consider the obstacles in estimation procedure, this condition may be violated. In section IV we discuss how this assumption can be satisfied in presence of obstacles.

Let $P^{\mu^j}(F|\mathbf{b}^i_\infty)$ denote the probability that under node-controller μ^j the system enters the failure set F before successful stop happens, given that the chain starts at $\mathbf{b}^i_\infty$. Under the assumption 4, these failure probabilities can be defined based on the Markov chain of extended state $\mathfrak{X}_k$ as follows:

$$P^{\mu^j}(F|\mathbf{b}^i_\infty) = \int P^{\mu^j}(F|\mathfrak{X} = (\mathbf{X}, \mathbf{b}^i_\infty)) b^i_\infty(\mathbf{X}) d\mathbf{X}, \tag{7}$$

where $P^{\mu^j}(F|\mathfrak{X} = (\mathbf{X}, \mathbf{b}^i_\infty))$ represents the probability that the extended chain fails given it starts at the extended state $\mathfrak{X} = (\mathbf{X}, \mathbf{b}^i_\infty)$. Again, it can be shown if the transition pdf of the extended Markov chain, $p^{\mu^j}(\mathfrak{X}_{k+1}|\mathfrak{X}_k)$ is smooth in its arguments, and given that the sets B_j are suitably small, and abusing the notation to $P^{\mu^j}(\cdot|B_i) := P^{\mu^j}(\cdot|\mathbf{b}^i_\infty)$, we can

modify (6) to incorporate obstacles in the state space:

$$J(B_i) = \min_j C^{\mu^j}(B_i) + J(F)P^{\mu^j}(F|B_i) + \sum_m J(B_m)P^{\mu^j}(B_m, \overline{F}|B_i), \tag{8a}$$

$$j^* = \pi(B_i) = arg\min_j C^{\mu^j}(B_i) + J(F)P^{\mu^j}(F|B_i) + \sum_m J(B_m)P^{\mu^j}(B_m, \overline{F}|B_i), \tag{8b}$$

where $P^{\mu^j}(B_m, \overline{F}|B_i)$ denotes the probability of the successful stop in B_m, under controller μ^j invoked at B_i and $J(F)$ is a user-defined suitably high cost-to-go value for failure. 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 $C^{\mu^j}(B_i)$ and transition probability functions $P^{\mu^j}(B_m, \overline{F}|B_i)$ and $P^{\mu^j}(F|B_i)$. Thus, the main difference from the obstacle free case is the addition of a "failure" state to the higher level MDP along with the associated probabilities of failure from the various nodes B_i.

We would also like to quantify the quality of the solution that is obtained by the FIRM. To this end, we require the probability of success of a policy π at the higher level Markov chain on B_i given by (8b). The higher level MDP now has $N+1$ states $\{S_1, S_2, \cdots, S_{N+1}\}$ that can be decomposed into three disjoint classes: the goal class $S_1 = B_g$, the failure class $S_2 = F$, and the transient class $\{S_3, S_4, \cdots, S_{N+1}\} = \{B_1, B_2, \cdots, B_N\} \setminus B_g$. The goal and failure classes are recurrent classes of this Markov chain. As a result, the transition probability matrix of this higher level $N+1$ state Markov can be decomposed as follows:

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_g & 0 & 0 \\ 0 & \mathcal{P}_f & 0 \\ \mathcal{R}_g & \mathcal{R}_f & \mathcal{Q} \end{bmatrix}. \tag{9}$$

The (i,j)-th components of $\mathcal{P}$ represents the transition probability from S_j to S_i. Moreover $\mathcal{P}_g = 1$ and $\mathcal{P}_f = 1$, since goal and failure classes are recurrent classes, i.e. the system stops once it reaches the goal or it fails. $\mathcal{Q}$ is a matrix that represents the transition probabilities between belief nodes B_i in transient class. $\mathcal{R}_g$ and $\mathcal{R}_f$ are $(N-1)\times 1$ vectors that represent the probability of the transient nodes $\{B_i\} \setminus B_g$ getting absorbed into the goal node and the failure set, respectively. Then, it can be shown that the success probability from any desired node B_i is given by the i-th component of the vector $\mathcal{P}^s$, denoted by $\mathcal{P}^s_i$:

$$\Pr(\text{success}|B_i) = \mathcal{P}^s_i, \quad \mathcal{P}^s = (I - \mathcal{Q})^{-1}\mathcal{R}_g. \tag{10}$$

Given that we can suitably construct the node-controllers $\mu^i(\cdot)$, the sets B_i, evaluate the transition costs $C^{\mu^i}(\cdot)$ and the transition probabilities $P^{\mu^i}(\cdot|\cdot)$, we can transform the POMDP into a FIRM. Generic planning algorithm on FIRM is presented in Algorithm 2.

Algorithm 1: Generic planning algorithm on FIRM

1 Given an initial belief, invoke some controller $\mu^i(\cdot)$ in the FIRM such that the robot is eventually absorbed into one of FIRM nodes B_m;
2 Given the system is in set B_m, invoke the higher level feedback policy π to choose the lower level feedback controller $\mu^{j^*}(\cdot)$ where $j^* = \pi(B_m)$ is given by (8b);
3 Let the node controller $\mu^{j^*}(\cdot)$ execute till absorption into the next neighbourhood $B_{m'}$ or failure;
4 Repeat steps 2-3 till absorption into the goal node B_g or failure;

A concrete instantiation of this generic algorithm is given in section IV.

IV. FIRM CONSTRUCTION

In this section, we address how the four elements in FIRM, i.e. nodes B_i, node-controllers μ^i, transition probabilities $P^{\mu^j}(\cdot|B_i)$, and costs $C^{\mu^j}(B_i)$ can be constructed such that the assumptions 1-4 in section III are satisfied.

A. FIRM Nodes B_i and Control Policies μ^j

PRM samples its nodes $\{\mathbf{n}_j\}_{j=1}^N$ from $\mathbb{X}_{free}$ based on some appropriate probabilistic measure [1]. Similarly, in planning in belief space it is desired to sample the belief space, where the main problem is if the sampled belief is reachable or not. In general, characterizing the whole reachable region of $\mathbb{B}$ is computationally infeasible. A main contribution of FIRM is

that instead of sampling in belief space and characterizing if the sampled belief is reachable or not, FIRM exploits node controllers to induce reachable regions in belief space $\mathbb{B}$ as is explained in the following.

The main sampling in FIRM is done in the state space using PRM techniques. After sampling PRM nodes $\{\mathbf{n}_j\}_{j=1}^{N}$ in $\mathbb{X}_{free}$, we associate a FIRM node $B_j \subset \mathbb{B}$ for each PRM node $\mathbf{n}_j$. Reaching the node B_i in FIRM means reaching a point $\mathbf{b} \in B_i$. Obviously, entire B_i or a part of B_i has to be in the reachable region of $\mathbb{B}$. In FIRM, the node controller μ^j associated with PRM node $\mathbf{n}_j$, leads to an irreducible Markov chain in belief space, and thus guarantees the reachability. We call μ^j the j-th node-controller. Suppose the system (linearized at $\mathbf{n}_j$) has the state-space form

$$\mathbf{X}_{k+1} = \mathbf{A}^j \mathbf{X}_k + \mathbf{B}^j \mathbf{u}_k + \mathbf{G}^j \mathbf{W}_k, \quad \mathbf{W}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^j) \tag{11}$$

$$\mathbf{z}_k = \mathbf{H}^j \mathbf{X}_k + \mathbf{V}_k, \qquad \mathbf{V}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^j). \tag{12}$$

Node controller: We choose the node-controller μ^j as the Linear Quadratic Gaussian (LQG) controller, which is an optimal controller for linear systems with Gaussian process and observation noises. In Gaussian case belief is characterized by a pair consisting of estimation mean and covariance $\mathbf{b}_k = (\widehat{\mathbf{X}}_k^+, P_k)$. We denote the governing dynamics of belief under LQG by $\mathbf{f_b}$, which is indeed encapsulates Kalman filtering equations:

$$\mathbf{b}_{k+1} = \mathbf{f_b}(\mathbf{b}_k, \mathbf{u}_k, \mathbf{z}_k), \quad \mathbf{u}_k = \mu^j(\mathbf{b}_k). \tag{13}$$

From control theory it can be shown that (as we detailed in the appendices) if the pair $(\mathbf{A}^j, \mathbf{B}^j)$ is controllable and the pair $(\mathbf{A}^j, \mathbf{H}^j)$ is observable, belief chain in (13) under μ^j is ergodic, i.e. $\lim_{k\to\infty} \mathbf{b}_k = \mathbf{b}_s = (\widehat{\mathbf{X}}_\infty^{+^j}, P_\infty^j)$. Actually, the dynamics of estimation covariance P_k is deterministic and it converges to the deterministic covariance P_∞^j [18]. Covariance P_∞^j is computed as $P_\infty^j = (I - \mathbf{L}^j \mathbf{H}^j) P_\infty^{j^-}$, where $P_\infty^{j^-}$ is the solution following Discrete Algebraic Riccati Equation (DARE) within the class of positive semidefinite symmetric matrices.

$$\begin{aligned} P_\infty^{j^-} &= \mathbf{G}^j \mathbf{Q}^j \mathbf{G}^{j^T} \\ &+ \mathbf{A}^j (P_\infty^{j^-} - P_\infty^{j^-} \mathbf{H}^{j^T} (\mathbf{H}^j P_\infty^{j^-} \mathbf{H}^{j^T} + \mathbf{R}^j)^{-1} \mathbf{H}^j P_\infty^{j^-}) \mathbf{A}^{j^T}, \end{aligned} \tag{14}$$

$$\mathbf{L}^j = P_\infty^{j^-} \mathbf{H}^{j^T} (\mathbf{H}^j P_\infty^{j^-} \mathbf{H}^{j^T} + \mathbf{R}^j)^{-1}. \tag{15}$$

The dynamics of estimation mean $\widehat{\mathbf{X}}_k^+$ is random and it can be shown that the appendices it converges to a stationary random vector $\widehat{\mathbf{X}}_\infty^{+^j}$, whose mean is $\mathbf{n}_j = \mathbb{E}[\widehat{\mathbf{X}}_\infty^{+^j}]$.

FIRM Node: According to the irreducible belief chain induced by μ^j, we define $\mathbf{b}_\infty^j$ and its corresponding FIRM node $B_j \subset \mathbb{B}$ as an ϵ-region centered at $\mathbf{b}_\infty^j$:[1]

$$\mathbf{b}_\infty^j = [\mathbf{n}_j^T, P_\infty^j(:)^T]^T, \tag{16}$$

$$B_j = \{\mathbf{b} | -\epsilon \dot{<} \mathbf{b} - \mathbf{b}_\infty^j \dot{<} \epsilon\}, \tag{17}$$

where symbol $\dot{<}$ denotes the componentwise inequality. Also, $\epsilon = [\delta^T, \Delta(:)^T]^T$, where $\Delta = \delta\delta^T$ and given that the robot's state is defined in (x, y, θ) space, $\delta = [\epsilon_x, \epsilon_y, \epsilon_\theta]^T$, where ϵ_x, ϵ_y, and ϵ_θ are user-defined tolerances. Indeed, δ determines the neighbourhood of estimation mean and Δ determines the neighbourhood of estimation covariance.

Stopping Condition: Based on assumption 3, in obstacle free case the controller stops at time step k if $\mathbf{b}_k \in \cup_m B_m$. The time this condition is satisfied is called stopping time of j-th node-controller, denoted by $\mathcal{T}^j(\mathbf{b})$. The stopping time $\mathcal{T}^j(\mathbf{b})$ is a random time, but finite, due to the ergodicity of belief chain.

To define the stopping condition in presence of obstacles, first let us denote the governing dynamics of e-state $\mathfrak{X}_k = (\mathbf{X}_k, \mathbf{b}_k)$ under LQG by $\mathbf{f}_\mathfrak{X}$:

$$\mathfrak{X}_{k+1} = \mathbf{f}_\mathfrak{X}(\mathfrak{X}_k, \mathbf{u}_k, \mathbf{W}_k, \mathbf{V}_{k+1}). \tag{18}$$

It is known from control theory that under the LQG controller and in the absence of obstacles, the e-chain is ergodic (as is detailed also in the appendices). Therefore, even in the presence of obstacles, if the nodes are selected far enough from the obstacles the estimator will have enough time to correct the belief and capture the distribution of $\mathbf{X}$ before $\mathbf{b}_k \in \cup_m B_m$ happens. Therefore, by choosing nodes far enough from the obstacles, for example by choosing $\mathbf{n}_j$ such that there is no obstacles within the 3σ uncertainty ellipse of P_∞^j, we can experimentally show that the condition $p(\mathbf{X}|\mathbf{z}_{0:k}) = b_k^m(\mathbf{X})$ is automatically satisfied with high accuracy when $\mathbf{b}_k \in B_m$, and thus for sufficiently small B_m, we have $p(\mathbf{X}|\mathbf{z}_{0:k}) = b_\infty^m(\mathbf{X})$.

Once the system is in any of the nodes B_i, it is allowed to invoke one of the controllers $\mu^j(\cdot)$ among $j \in A(i)$, the k-nearest neighbour set of i, that in turn will draw the system toward the region $\cup_m B_m$ with highest probability of landing in B_j compared to landing in any B_r, $r \neq j$ (see Fig. I). Optimal j for a belief $\mathbf{b}$ is computed through the Dynamic Programming equation defined on the belief nodes in (8).

[1]The operator $(:)$ after a matrix converts that matrix to a column vector by stacking the columns of the matrix into a single column.

If the system is nonlinear in following form, which is the case in our experiments,

$$\begin{aligned}\mathbf{X}_{k+1}&= \mathbf{f}(\mathbf{X}_k)+ \mathbf{g}(\mathbf{X}_k)\mathbf{u}_k+ \mathbf{g}'(\mathbf{X}_k)\mathbf{W}_k,\ \mathbf{W}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})\\ \mathbf{z}_k&= \mathbf{h}(\mathbf{X}_k)+ \mathbf{V}_k, \qquad\qquad\qquad\quad \mathbf{V}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}),\end{aligned}$$

we linearize it about $\mathbf{n}_j$ to get $\mathbf{A}^j = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(\mathbf{n}_j)$, $\mathbf{B}^j = \mathbf{g}(\mathbf{n}_j)$, $\mathbf{G}^j = \mathbf{g}'(\mathbf{n}_j)$, and $\mathbf{H}^j = \frac{\partial \mathbf{g}'}{\partial \mathbf{X}}(\mathbf{n}_j)$. In nonlinear systems, the main constraint imposed on the FIRM construction is that if we expect the controller μ^j to take a belief $\mathbf{b} \in B_i$ to B_j, the region $B_i \subset N(B_j)$, which can be satisfied by increasing the number of samples. However, we usually even do not need to satisfy $B_i \subset N(B_j)$ if we precede the node-controller with a time-varying edge controller that takes the $\mathbf{b} \in B_i$ to $N(B_j)$, and then run the node controller to take the belief $\mathbf{b} \in N(B_j)$ to B_j.

B. Transition probabilities and Edge costs

In this section, we compute the transition probabilities $P^{\mu^j}(\cdot|B_i)$, and costs $C^{\mu^j}(B_i)$ associated with invoking node controller μ^j at node B_i. We approximate $p^{\mathfrak{X}_k}(\mathfrak{x}_k)$ using particle-based representations, as $p^{\mathfrak{X}_k}(\mathfrak{x}_k) = \{\mathfrak{X}_k^{(q)}, w^{(q)}\}_{q=1}^M$, in which $\mathfrak{X}_k^{(q)}$ is the q-th particle and $w^{(q)}$ is its corresponding weight, where we have the constraint $\sum_{i=1}^M w^{(q)} = 1$. Owing to the designed stopping condition in previous section, we draw samples $\{\mathbf{X}_0^{(q)}\}_{q=1}^M$ from $p^{\mathbf{X}_0}(\mathbf{x}) = b_\infty^i(\mathbf{x}) = \mathcal{N}(\mathbf{n}_i, P_\infty^i)$. Then we construct $\{\mathfrak{X}_0^{(q)}\}_{q=1}^M = \{\mathbf{X}_0^{(q)}, \mathbf{b}_\infty^j\}_{q=1}^M$, and generate noise samples to propagate each of particles along the time using (18) to get $\{\mathfrak{X}_{0:\mathcal{T}^q}^{(q)}\}_{q=1}^M$, where $\mathcal{T}^q$ is the stopping time of q-th particle.

The particle-based approximation can reach any desired accuracy by increasing the number of particles M. However, its main problem is the incurred high computational cost, which might preclude its use in online scenarios. Nevertheless, owing to the offline construction of FIRM, it can tolerate this high computational burden. Remembering that $\{\mathfrak{X}_k^{(q)}\}_{q=1}^M = \{\mathbf{X}_k^{(q)}, \widehat{\mathbf{X}}_k^{+(q)}, P_k^{(q)}\}_{q=1}^M$, the particle representation $\{\mathfrak{X}_{0:\mathcal{T}^q}^{(q)}, w^{(q)}\}_{q=1}^M$, generated by controller μ^j invoked at B_i, includes all information needed to compute the transition probabilities. Let us define $\mathbb{I}_F$ as the collision indicator, which is one if the particle hits an obstacle and is zero otherwise. Similarly, $\mathbb{I}_{B_m}$ is one if the particle stops at B_m, and zero otherwise. Thus, we have:

$$\begin{aligned}P^{\mu^j}(F|B_i) &= \mathbb{E}[\mathbb{I}_F|B_i, \mu^j] \approx \sum_{q=1}^M w^{(q)} \mathbb{I}_F(\mathfrak{X}_{0:\mathcal{T}^q}^{(q)}), \qquad (19)\\ P^{\mu^j}(B_m, \overline{F}|B_i) &= \mathbb{E}[\mathbb{I}_{B_m}|B_i, \mu^j] \approx \sum_{q=1}^M w^{(q)} \mathbb{I}_{B_m}(\mathfrak{X}_{0:\mathcal{T}^q}^{(q)}).\end{aligned}$$

One can define a variety of relevant cost functions for taking node controller μ^j at B_i. Here, we first consider estimation accuracy to find the paths, on which the estimator and accordingly controller can perform better. A measure of estimation error is the trace of estimation covariance. Thus, we use $\mathbb{E}[\sum_{k=1}^{\mathcal{T}} \mathrm{tr}(P_k)]$, which is approximated as:

$$\Phi^{ij} = \mathbb{E}[\sum_{k=1}^{\mathcal{T}} \mathrm{tr}(P_k)|B_i, \mu^j] \approx \sum_{q=1}^M \sum_{k=1}^{\mathcal{T}^q} w^{(q)} \mathrm{tr}(P_k^{(q)}), \qquad (20)$$

where, $\mathcal{T}^q$ is the stopping time of q-th particle. This measure can take into account the stochasticity of the covariance matrix in Extended Kalman Filter (EKF) framework, where covariance dynamics is not deterministic.

Also, we consider the mean stopping time as a cost, which can be computed as $\widehat{\mathcal{T}} = \mathbb{E}[\mathcal{T}] \approx \sum_{q=1}^M w^{(q)} \mathcal{T}^q$. Total cost of invoking μ^j at B_i is considered as a linear combination of estimation accuracy and expected stopping time.

$$C^{\mu^j}(B_i) = \alpha_1 \Phi + \alpha_2 \widehat{\mathcal{T}}. \qquad (21)$$

C. Offline Construction of FIRM

The crucial feature of FIRM is that it can be constructed offline and stored, independent of future queries. Moreover, owing to the huge reduction from original POMDP to N-state MDP on belief nodes, the FIRM can be solved using standard DP techniques such as value/policy iteration to yield the optimal policy $j^* = \pi(B_i)$ on the higher level MDP defined on the belief nodes. Indeed at each belief $\mathbf{b} \in B_i$, the policy $j^* = \pi(B_i)$ decides which node-controller μ^{j^*} has to be invoked among $j \in A(i)$. Algorithm 2 details the construction of FIRM.

D. Planning on FIRM

Given that the FIRM 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 $\mathbf{b}_0$ does not belong to any B_i, we create a singleton set $B_0 = \mathbf{b}_0$ and connect it to FIRM through its k-nearest neighbors $A(0)$. Afterwards, due to the designed stopping condition, the belief is guaranteed to be in one of nodes B_i at the decision stages, if no collision occurs. Thus, we use policy π over nodes defined in (8b) to find j^*, given the current node.

Algorithm 2: Offline Construction of FIRM

input : Free space map, $\mathbb{X}_{free}$
output : FIRM graph $\mathcal{G}$
Sample PRM nodes $\{\mathbf{n}_j\}_{j=1}^{N}$;
forall $\mathbf{n}_i \in \mathcal{V}$ **do**
    Design μ^i and compute associated $\mathbf{b}_\infty^i$ using (16);
    Construct FIRM node B_i using (17);
forall i **do**
    forall $j \in A(i)$ **do**
        Set $\mathbf{b}_0 = \mathbf{b}_\infty^i$;
        Characterize belief chain $\mathbf{b}_{0:\mathcal{T}}$ and e-chain $\mathfrak{X}_{0:\mathcal{T}}$ induced by controller μ^j using (13) and (18);
        Compute the transition probabilities and costs associated with $\mathbf{b}_{0:\mathcal{T}}$ and $\mathfrak{X}_{0:\mathcal{T}}$ using (19-21);
Compute $J(B_i)$ by solving DP in (8a);
$\mathcal{G}.\overline{B} = \{B_i\}$; $\mathcal{G}.\overline{J} = \{J(B_i)\}$; $\mathcal{G}.\overline{C} = \{C^{\mu^j}(B_i)\}$;
$\mathcal{G}.\overline{P} = \{P^{\mu^j}(B_m, \overline{F}|B_i),\ P^{\mu^j}(F|B_i)\}$;
return $\mathcal{G}$;

V. Experimental Results

In this section we construct FIRM on a sample environment. A 3-wheel omnidirectional mobile robot is adopted in experiments with the nonlinear kinematic model given in [20]. The state vector composed of 2D location and heading angle $\mathbf{x} = [x, y, \theta]^T$. In experiments, robot is equipped with exteroceptive sensors that provide range and bearing measurements from existing landmarks (radio beacons) in the environment. 2D location of the j-th landmark is denoted by L_j. Measuring L_j can be modeled as follows:

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

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

$$^j\mathbf{R} = \text{diag}((\eta_r\|^j\mathbf{d}\| + \sigma_b^r)^2, (\eta_\theta\|^j\mathbf{d}\| + \sigma_b^\theta)^2). \tag{22}$$

In other words, the uncertainty (standard deviation) of sensor reading increases as the robot gets farther from the landmarks. $\eta_r = \eta_\theta = 0.3$ determines this dependence, and $\sigma_b^r = 0.01$ meter and $\sigma_b^\theta = 0.5$ degrees are the bias standard deviations. Similar model for range sensing is used in [12]. We assume robot observes all N_L landmarks all the time and their observation noises are independent.

Algorithm 3: Online Phase Algorithm

input : Initial belief $\mathbf{b}_0$, FIRM graph $\mathcal{G}$
if $\exists B_m$ *such that* $\mathbf{b}_0 \in B_m$ **then**
    Set $i = m$ and compute $j^* = \pi(B_m)$ using (8b);
else
    Define the singleton set $B_0 = \mathbf{b}_0$;
    forall $j \in A(0)$ **do**
        Characterize belief chain $\mathbf{b}_{0:\mathcal{T}}$ and e-chain $\mathfrak{X}_{0:\mathcal{T}}$ induced by controller μ^j using (13) and (18) ;
        Compute the transition probabilities and costs associated with $\mathbf{b}_{0:\mathcal{T}}$ and $\mathfrak{X}_{0:\mathcal{T}}$ using (19-21);
    Set $i = 0$ and compute $j^* = \pi(B_0)$ using (8b);
while $B_i \neq B_{goal}$ **do**
    while $\mathbf{b}_k \notin \cup_m B_m$ *and "no collision"* **do**
        Apply the control $\mathbf{u}_k = \mu^{j^*}(\mathbf{b}_k)$ to the system;
        Update belief as $\mathbf{b}_{k+1} = \mathbf{f}_\mathbf{b}(\mathbf{b}_k, \mu^{j^*}(\mathbf{b}_k), \mathbf{z}_{k+1})$;
    if *Collision happens* **then return** Collision;
    Set $i = l$, where $\mathbf{b}_{k+1} \in B_l$;
    Compute $j^* = \pi(B_i)$ using (8b);

Figure 2(a) shows a sample environment, including obstacles, landmarks, and enumerated nodes in (x, y, θ) space. Nodes are shown by blue triangles, that encodes position (x, y) and heading angle θ of the robot. Landmarks are shown by black stars. The corresponding FIRM nodes are computed and shown in Fig. 2(b). All elements in Fig. 2(b) are defined in (x, y, θ) space but only (x, y) portion of them are shown here. Each $\mathbf{b}_\infty^j = [\mathbf{n}_j^T, P_\infty^j(:)^T]^T$ is illustrated by a red dot representing $\mathbf{n}_j$ and a green ellipse, representing 3σ ellipse of covariance P_∞^j. Each FIRM node B_j is a neighborhood around $\mathbf{b}_\infty^j$. In experiments, we set $\epsilon_x = \epsilon_y = 0.07$ meter and $\epsilon_\theta = 1$ degree to quantify B_j's. Part of this neighborhood that is defined for estimation mean $\widehat{\mathbf{X}}^+$ is shown by a cyan rectangle centered at $\mathbf{n}_j$. The other part of this neighborhood is illustrated by two dashed green ellipses that are representing 3σ covariances of $P_\infty^j - \Delta_d$ and $P_\infty^j + \Delta_d$, where Δ_d is the Δ matrix, whose off-diagonal elements are set to zero. For illustration purposes, both of these neighbourhoods are five times magnified in this figure.

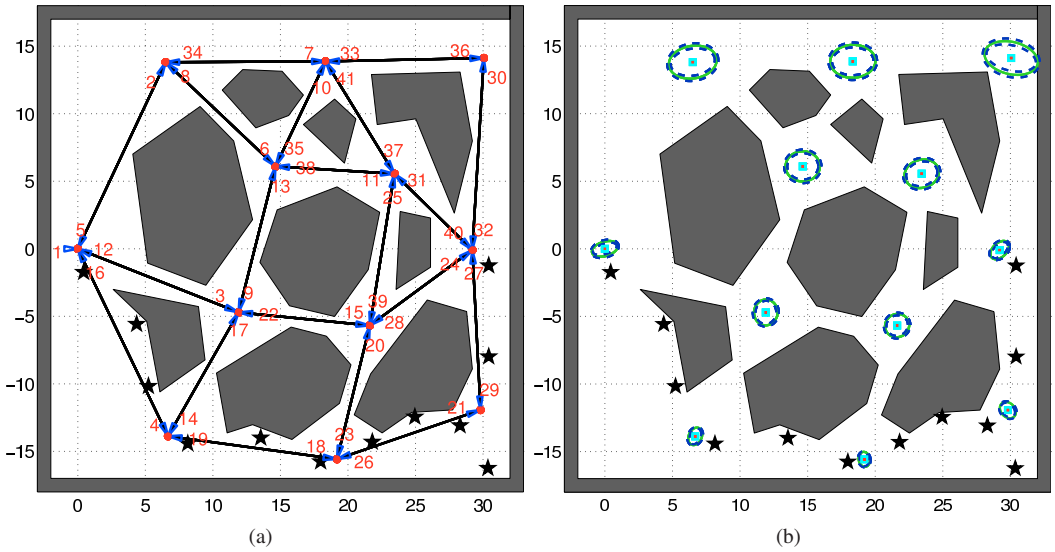

Fig. 2. (a) Figure depicts the underlying PRM graph. Gray polygons are the obstacles and black stars represent the landmarks' locations. (b) FIRM nodes $\{B_1, B_2, B_3, B_4, B_6, B_7, B_{15}, B_{18}, B_{21}, B_{27}, B_{31}, B_{36}\}$.

TABLE I

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

B_i, μ^j pair	B_1, μ^4	B_4, μ^{18}	B_{18}, μ^{21}	B_{21}, μ^{27}	B_{27}, μ^{31}	B_1, μ^3	B_3, μ^{13}	B_{13}, μ^{11}
$1 - P^{\mu^j}(F\|B_i)$	%97	%95	%99	%77	%79	%87	%55	%79
Φ	18.5967	11.2393	6.8229	15.1148	26.2942	23.6183	48.8189	43.6207
$\mathbb{E}[\mathcal{T}], \sigma[\mathcal{T}]$	238.2, 21.8	193.0, 28.7	150.0, 15.1	209.6, 24.5	170.8, 22.6	200.3, 22.7	242.4, 30.1	219.2, 26.7

Remember that an e-chain in particle based representation is $\{\mathfrak{X}_{0:\mathcal{T}^q}^{(q)}, w^{(q)}\}_{q=1}^M = \{\mathbf{X}_{0:\mathcal{T}^q}^{(q)}, \widehat{\mathbf{X}}_{0:\mathcal{T}^q}^{+(q)}, P_{0:\mathcal{T}^q}^{(q)}, w^{(q)}\}_{q=1}^M$. Defining $\overline{\mathcal{T}} = \max_q\{\mathcal{T}^q\}$, Fig. 3(a) depicts the $\mathbf{X}_{0:\overline{\mathcal{T}}}^{(q)}$ in green, $\widehat{\mathbf{X}}_{0:\overline{\mathcal{T}}}^{+(q)}$ in dark red for $M = 100$ particles. As it is seen in Fig. 3(a) the behavior of ground truth that have access to accurate observations is remarkably close to the planned behavior. However, on the edges that have access to less informative observations, controller cannot effectively compensate the deviations of the ground truth path, which can lead to collision with obstacles or landing in a node different than the planned node. Figure 3(b) depicts $P_{0:\mathcal{T}^q}^{(q)}$ only for $q = 1$ to avoid clutter in the figure. We set the $q = 1$, to be the zero-noise particle by setting corresponding process and observation noises to zero. Thus, Fig. 3(b) can be seen as the maximum-likelihood estimation uncertainty tube over roadmap.

To complete the construction of FIRM, we compute the properties associated with invoking j-th controller at i-th node, such as collision probability, filtering performance, and stopping time. Table I shows these quantities for some node-controller pairs in FIRM. Finally, we perform planning on FIRM to find the optimal policy based on the defined costs in (21). It is worth noting that the best policy can lead to any path in online phase based on occurred noises and we cannot define a best path in planning stage. However, for illustration purposes, we show the most likely path under the best policy of FIRM, i.e. (8b), in red in Fig. 3(b). The shortest path is also illustrated in Fig. 3(b) in yellow. It can be seen that the "most likely path under the best policy" detours from the shortest path to a path along which the filtering uncertainty is smaller and is easier for controller to avoid collisions.

VI. Conclusion

In this paper, we have proposed the Feedback controller-based Information-state road map (FIRM) for solving the motion planning problem under motion and sensing uncertainties. This problem originally is a POMDP, whose solution is intractable. Exploiting feedback controllers, we reduce it to a tractable FIRM that can be solved by standard DP techniques. FIRM utilizes feedback controllers to create the reachable node regions in belief space, and construct a graph, on which a higher level policy is defined to provide the optimal plans. This procedure indeed overcomes the curse of history and curse of dimensionality in the original POMDP problem. Finally, obstacles are also taken into account planning through FIRM, through computing the collision probabilities. We believe that FIRM can be a considerable step toward solving POMDPs and utilizing them as a practical tool for robot motion planning under uncertainty.

References

[1] L. Kavraki, P. Svestka, 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.

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

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

[4] L. Guibas, D. Hsu, H. Kurniawati, and E. Rehman, "Bounded uncertainty roadmaps for path planning," in *International Workshop on Algorithmic Foundations of Robotics*, 2008.

[5] P. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

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

[7] 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.

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

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

[10] J. van den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," in *Proceedings of Robotics: Science and Systems (RSS)*, June 2010.

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

[12] 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.

[13] 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.

[14] N. D. Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," in *ICRA*, May 2010.

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

[16] J. Pineau, G. Gordon, and S. Thrun, "Anytime point based approximations for large pomdps," *Journal of Artificial Intelligence Research*, vol. 27, pp. 335–380, 2006.

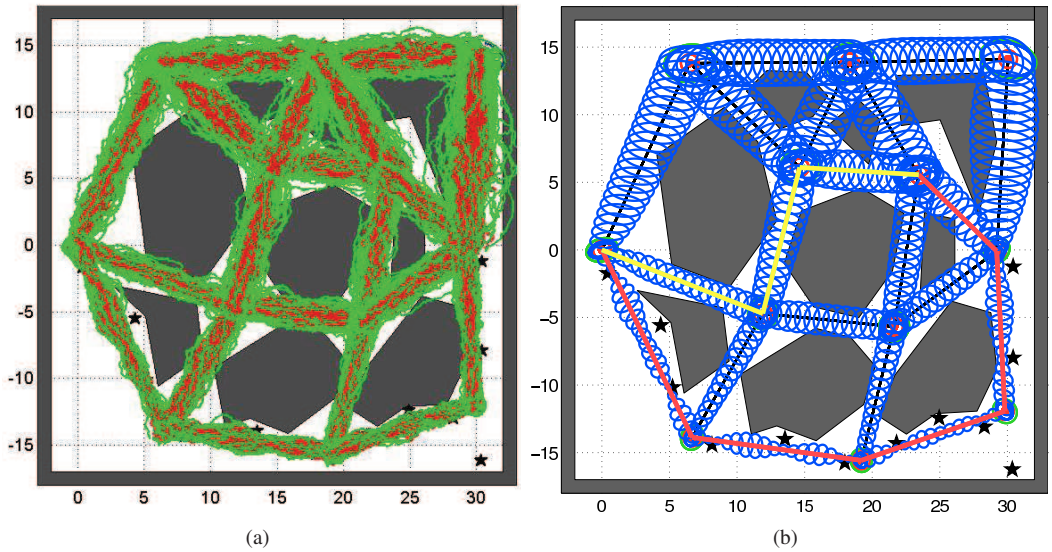

Fig. 3. Induced chains by the controllers invoked in different nodes. (a) For $M = 100$ particles, the ground truth chain $\mathbf{X}^{(q)}_{0:\mathcal{T}}$ and the estimation mean chain $\widehat{\mathbf{X}}^{+(q)}_{0:\mathcal{T}}$ are shown in green and red respectively. (b) The most likely path under the optimal policy and shortest path are shown in red and yellow respectively. The 3σ ML estimation uncertainty tube is drawn in blue.

[17] M. Spaan and N. Vlassis, "Perseus: Randomized point-based vallue iteration for pomdps," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.
[18] D. Bertsekas, *Dynamic Programming and Optimal Control: 3rd Edition*. Athena Scientific, 2007.
[19] 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.
[20] T. Kalmár-Nagy, R. D'Andrea, and P. Ganguly, "Near-optimal dynamics trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 46, no. 1, pp. 47–64, January 2004.
[21] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley and Sons, Inc, 2006.

APPENDIX I
CONVERGENCE LEMMA OF E-STATE

Let us call the distribution over e-state $\mathfrak{X}_k$ as hyper-belief (h-belief) and denote it by β_k. If β_k is Gaussian, it is characterized by the pair $\beta_k = (\mathbb{E}[\mathfrak{X}_k], \mathbb{C}[\mathfrak{X}_k])$. Under linear/Gaussian system assumption, we adopt stationary Linear Quadratic Gaussian (LQG) controller as the node-controller. The j-th node-controller corresponding to the j-th node, provides a stationary feedback policy μ_s^j that maps the belief at each time step to an optimal control signal, i.e. $U_k = \mu_s^j(b_k)$. Accordingly, j-th node-controller induces the following governing dynamics over the e-state and h-belief:

$$\mathfrak{X}_{k+1} = \mathbf{f}_{\mathfrak{X}}(\mathfrak{X}_k, \mu_s^j(b_k), \mathbf{W}_k, \mathbf{V}_{k+1}). \tag{23}$$

$$\beta_{k+1} = f_\beta^j(\beta_k, \mu_s^j(b_k)) \tag{24}$$

which is analytically characterized in appendix II for the following linear Gaussian system

$$X_{k+1} = \mathbf{A}_s^j X_k + \mathbf{B}_s^j U_k + \mathbf{G}_s^j W_k \ , \quad W_k \sim \mathcal{N}(0, \mathbf{Q}_s^j) \tag{25}$$

$$Z_k = \mathbf{H}_s^j X_k + V_k \ , \quad V_k \sim \mathcal{N}(0, \mathbf{R}_s^j) \tag{26}$$

where,

$$\mathbf{A}_s^j = \frac{\partial \mathbf{f}}{\partial x}(\mathbf{n}_j, \mathbf{0}, \mathbf{0}) \ , \quad \mathbf{B}_s^j = \frac{\partial \mathbf{f}}{\partial u}(\mathbf{n}_j, \mathbf{0}, \mathbf{0}) \ , \quad \mathbf{H}_s^j = \frac{\partial \mathbf{h}}{\partial x}(\mathbf{n}_j, \mathbf{0}) \tag{27}$$

and $\mathbf{Q}_s^j$ and $\mathbf{R}_s^j$ respectively are the covariances of motion and observation noise, computed at $\mathbf{n}_j$.

Moreover, from control theory, we know the following lemma holds. In the following lemma all vectors and matrices have to have superscript j as they correspond to the linearized model in $\mathbf{n}_j$, but we do not write them to unclutter the formulae.

Lemma 1: Based on the dynamics in (24) induced by the j-th node-controller, if the pair $(\mathbf{A}_s, \mathbf{B}_s)$ is a controllable pair, and the pair $(\mathbf{A}_s, \mathbf{H}_s)$ is an observable pair, the h-belief sequence generated by (24) is a converging sequence and we denote its limit by β_s^j:

$$\beta_s^j = \lim_{k \to \infty} \beta_k \tag{28}$$

where,

$$\beta_s^j = \left(\begin{pmatrix} \mathbf{n}_j \\ \mathbf{n}_j \\ P_s^+ \end{pmatrix}, \begin{pmatrix} \mathcal{P}_s = \begin{pmatrix} \mathcal{P}_{s,11} & \mathcal{P}_{s,12} \\ \mathcal{P}_{s,21} & \mathcal{P}_{s,22} \end{pmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \right) \tag{29}$$

where, P_s^+ is the stationary estimation covariance, associated with the j-th node-controller, and is computed by first finding the P_k^- as the solution of the following Discrete Algebraic Riccati Equation (DARE) in the class of positive semi-definite matrices, and then plugging P_k^- into the (31):

$$P_s^- = \mathbf{A}_s(P_s^- - P_s^- \mathbf{H}_s^T (\mathbf{H}_s P_s^- \mathbf{H}_s^T + \mathbf{R}_s)^{-1} \mathbf{H}_s P_s^-) \mathbf{A}_s^T + \mathbf{Q}_s \tag{30}$$

$$P_s^+ = P_s^- - \mathbf{L}_s \mathbf{H}_s P_s^- \ , \quad \mathbf{L}_s = P_s^- \mathbf{H}_s^T (\mathbf{H}_s P_s^- \mathbf{H}_s^T + \mathbf{R}_s)^{-1} \tag{31}$$

In addition, the matrix $\mathcal{P}_s$ in (29) is the solution of following Lyapunov equation:

$$\mathcal{P}_s = \overline{\mathbf{F}}_s \mathcal{P}_s \overline{\mathbf{F}}_s^T - \overline{\mathbf{G}}_s \overline{\mathbf{Q}}_s \overline{\mathbf{G}}_s^T \tag{32}$$

where,

$$\overline{\mathbf{F}}_s = \begin{pmatrix} \mathbf{A}_s & -\mathbf{B}_s \mathbf{K}_s \\ \mathbf{L}_s \mathbf{H}_s \mathbf{A}_s & \mathbf{A}_s - \mathbf{B}_s \mathbf{K}_s - \mathbf{L}_s \mathbf{H}_s \mathbf{A}_s \end{pmatrix}, \ \overline{\mathbf{G}}_s = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L}_s \mathbf{H}_s & \mathbf{L}_s \end{pmatrix}, \ \overline{\mathbf{Q}}_s = \begin{pmatrix} \mathbf{Q}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_s \end{pmatrix} \tag{33}$$

and feedback gain $\mathbf{K}_s$ computed by solving following DARE in (38) and plugging the results into (37) as follows:

$$\mathbf{S}_s = \mathbf{A}_s^T \left(\mathbf{S}_s - \mathbf{S}_s \mathbf{B}_s (\mathbf{B}_s^T \mathbf{S}_s \mathbf{B}_s + W_u)^{-1} \mathbf{B}_s^T \mathbf{S}_s \right) \mathbf{A}_s + W_x, \tag{34}$$

$$\mathbf{K}_s = (\mathbf{B}_s^T \mathbf{S}_s \mathbf{B}_s + W_u)^{-1} \mathbf{B}_s^T \mathbf{S}_s \mathbf{A}_s. \tag{35}$$

Proof: See appendix II. ■

APPENDIX II
PROOF OF β CONVERGENCE LEMMA

Under linear/Gaussian system assumption, we adopt stationary Linear Quadratic Gaussian (LQG) controller as the node-controller. Stationary LQG is an optimal controller for linear Gaussian systems, whose minimization objective for a given node $\mathbf{n}$ is:

$$\mathbb{E}\left[\sum_{k=0}^{\infty}\left(X_k-\mathbf{n}\right)^T W_x\left(X_k-\mathbf{n}\right)+\delta U_k^T W_u \delta U_k\right] \tag{36}$$

where, δU_k denotes the feedback control signal and nominal control is zero, $u_k^p=0$, and thus $U_k=\delta U_k$. The weight matrix of state deviation from $\mathbf{n}$ is W_x and the weight matrix of control signal is W_u. Weight matrices are chosen as positive definite matrices.

Stationary LQG is composed of a stationary Linear Quadratic Regulator (LQR) and a stationary Kalman filter (KF). Feedback gain $\mathbf{K}_s$ of a stationary LQR designed for node $\mathbf{n}$ is computed by solving the Discrete Algebraic Riccati Equation (DARE) in (38) and plugging the results into (37) as follows: [18]

$$\mathbf{K}_s=(\mathbf{B}_s^T\mathbf{S}_s\mathbf{B}_s+W_u)^{-1}\mathbf{B}_s^T\mathbf{S}_s\mathbf{A}_s, \tag{37}$$

$$\mathbf{S}_s=\mathbf{A}_s^T\left(\mathbf{S}_s-\mathbf{S}_s\mathbf{B}_s(\mathbf{B}_s^T\mathbf{S}_s\mathbf{B}_s+W_u)^{-1}\mathbf{B}_s^T\mathbf{S}_s\right)\mathbf{A}_s+W_x. \tag{38}$$

where,

$$\mathbf{A}_s=\frac{\partial\mathbf{f}}{\partial x}(\mathbf{n},\mathbf{0},\mathbf{0})\ ,\quad \mathbf{B}_s=\frac{\partial\mathbf{f}}{\partial u}(\mathbf{n},\mathbf{0},\mathbf{0}). \tag{39}$$

and therefore feedback control, δU_k, at time step k is computed as follows:

$$\delta U_k=-\mathbf{K}_s(\widehat{X}_k^+-\mathbf{n}):=\mu_s^j(b_k) \tag{40}$$

where, $\widehat{X}_k^+$ is produced by the stationary KF. Filter gain, $\mathbf{L}_s$, of stationary KF designed for node $\mathbf{n}$ is computed by solving the DARE in (42) and plugging the results into (41) as follows: [21]

$$\mathbf{L}_s\ =P_s^-\mathbf{H}_s^T(\mathbf{H}_sP_s^-\mathbf{H}_s^T+\mathbf{R}_s)^{-1} \tag{41}$$

$$P_s^-=\mathbf{A}_s(P_s^--P_s^-\mathbf{H}_s^T(\mathbf{H}_sP_s^-\mathbf{H}_s^T+\mathbf{R}_s)^{-1}\mathbf{H}_sP_s^-)\mathbf{A}_s^T+\mathbf{Q}_s \tag{42}$$

where, $\mathbf{Q}_s$ and $\mathbf{R}_s$ respectively are the covariances of motion and observation noise, computed at $\mathbf{n}$. Also $\mathbf{H}_s=\frac{\partial\mathbf{h}}{\partial x}(\mathbf{n},\mathbf{0})$.

At each step of stationary KF, first a prediction of state at the next step is constructed. Then, based on the observations, prediction is updated to produce estimation. The predicted state at $(k+1)$-th step is denoted by X_{k+1}^-, which is computed as:

$$X_{k+1}^-=\mathbf{f}(X_k^+,u_k^p+\delta U_k,W_k) \tag{43}$$

whose, mean and covariance are approximated as:

$$\widehat{X}_{k+1}^-=\mathbf{f}(\widehat{X}_k^+,u_k^p+\delta U_k,\mathbf{0}) \tag{44}$$

$$P_{k+1}^-=\mathbf{A}_sP_k^+\mathbf{A}_s^T+\mathbf{Q}_s, \tag{45}$$

In execution phase, when the observation z_k is obtained from sensors at k-th step, stationary KF produces the state estimation at k-th step as:

$$\widehat{X}_{k+1}^+=\widehat{X}_{k+1}^-+\mathbf{L}_s(z_{k+1}-\mathbf{h}(\widehat{X}_{k+1}^-,\mathbf{0})) \tag{46}$$

$$P_{k+1}^+=P_{k+1}^--\mathbf{L}_s\mathbf{H}_sP_{k+1}^- \tag{47}$$

However in planning phase the observation realizations, i.e. z_k, has not been obtained yet. Thus, we take into account all possible observation probabilistically by considering random observation Z_k at k-th time step:

$$Z_{k+1}=\mathbf{h}(X_{k+1},V_{k+1}) \tag{48}$$

Thus, overall, the explained procedure maps a given $\widehat{X}_k^+$, X_{k+1}, and V_k into a $\widehat{X}_{k+1}^+$. We denote this mean mapping as stationary Kalman Filter's mean mapping $\mathbf{f}_s^\mu$:

$$\widehat{X}_{k+1}^+=\mathbf{f}_s^\mu(\widehat{X}_k^+,X_{k+1},V_k). \tag{49}$$

Besides, substituting P_k^- from (45) into (47), we can define the mapping $\mathbf{f}_s^\Sigma$ that governs estimation covariance:

$$P_{k+1}^+=(I-\mathbf{L}_s\mathbf{H}_s)(\mathbf{A}_sP_k^+\mathbf{A}_s^T+\mathbf{Q}_s)=:\mathbf{f}_s^\Sigma(P_k^+) \tag{50}$$

Moreover, the X_{k+1} is governed by mapping $\mathbf{f}_s$:

$$\begin{aligned} X_{k+1} &= \mathbf{f}(X_k, u_k^p + \delta U_k, W_k) \\ &= \mathbf{f}(X_k, u_k^p - \mathbf{K}_s(\widehat{X}_k^+ - \mathbf{n}), W_k) =: \mathbf{f}_s(X_k, \widehat{X}_k^+, W_k) \end{aligned} \tag{51}$$

These three mappings, i.e. $\mathbf{f}_s$, $\mathbf{f}_s^\mu$, and $\mathbf{f}_s^\Sigma$ define the evolution of h-state $\mathfrak{X}_k$ of the node-controller designed for $\mathbf{n}$:

$$\begin{aligned} \mathfrak{X}_{k+1} &= \begin{pmatrix} X_{k+1} \\ \widehat{X}_{k+1}^+ \\ P_{k+1}^+(:) \end{pmatrix} = \begin{pmatrix} \mathbf{f}(X_k, \widehat{X}_k^+, W_k) \\ \mathbf{f}^+(\widehat{X}_k^+, X_{k+1}, V_{k+1}) \\ \mathbf{f}_s^\Sigma(P_k^+)(:) \end{pmatrix} \\ &=: \mathbf{f}^{\text{node}}(\mathfrak{X}_k, W_k, V_{k+1}) \end{aligned} \tag{52}$$

$\mathbf{f}^{\text{node}}$ in Eq. (52) defines a Markov chain that governs the h-state $\mathfrak{X}_k$. Under appropriate conditions this Markov chain has an ergodic distribution and converges to it. Ergodicity of this Markov chain means that β_k converges to some stationary $\beta_s \in \mathbb{B}_h$. In following we characterize this stationary h-belief β_s for linear Gaussian systems regulated under node-controller.

In linear Gaussian systems, as it is seen in (52) the last part of h-state $\mathfrak{X}_k$, i.e. estimation covariance P_k^+, does not depend on process and observation noises. From control theory we know that if the pair $(\mathbf{A}_s, \mathbf{B}_s)$ is a controllable pair and the pair $(\mathbf{A}_s, \mathbf{H}_s)$ is an observable pair, there exist a stationary covariance matrix P_s^+ such that for every possible initial matrix P_0^+ the P_k^+ converges to P_s^+ as $k \to \infty$ [18]. P_s^+ is the solution of $P_s^+ = \mathbf{f}_s^\Sigma(P_s^+)$ within the class of positive semidefinite symmetric matrices. In other words, to compute P_s^+ we solve the DARE in (42) and we have $P_s^+ = (I - \mathbf{L}_s\mathbf{H}_s)P_s^-$.

Therefore, even if P_0^+ is a random matrix P_k^+ will converge to a deterministic matrix P_s^+. In other words the distribution of P_k^+ converges to the Dirac delta at P_s^+. As a result in $k \to \infty$, h-belief can be factored as:

$$p^{\mathfrak{X}_s}(\mathfrak{x}_s) = p^{\mathcal{X}_s}(\psi_s)\delta_{P_s^+}(\rho_s) \tag{53}$$

where, $\mathcal{X}_k = (X_k, \widehat{X}_k^+)$ and $\mathfrak{B}_k$ is its distribution. $p^{\mathfrak{X}_s}(\mathfrak{x}_s; \beta_s)$ and $p^{\mathcal{X}_s}(\psi_s; \mathfrak{B}_s)$ are the stationary distribution of $\mathfrak{X}_k$ and $\mathcal{X}_k$, respectively, parameterized by β_s and $\mathfrak{B}_s$. $\delta_{P_s^+}$ is the Dirac delta function that is only nonzero at P_s^+. As a result, we can define $\beta_s = [\mathfrak{B}_s^T, P_s^+(:)^T]^T$.

We already know how to compute P_s^+ corresponding to a node $\mathbf{n}$ of underlying PRM. We need to compute the $\mathfrak{B}_s$ for the node $\mathbf{n}$ to characterize the whole β_s for node $\mathbf{n}$. To do so, let us define the e-state error and write (52) for linear systems (or linearized about $\overline{\mathbf{n}}$) as follows:

$$\zeta_k := \mathcal{X}_k - \overline{\mathbf{n}}, \quad \overline{\mathbf{n}} := [\mathbf{n}^T, \mathbf{n}^T]^T \tag{54}$$

$$\zeta_{k+1} = \overline{\mathbf{F}}_s\zeta_k - \overline{\mathbf{G}}_s\mathbf{q}_k, \quad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \overline{\mathbf{Q}}_s) \tag{55}$$

where,

$$\overline{\mathbf{F}}_s = \begin{pmatrix} \mathbf{A}_s & -\mathbf{B}_s\mathbf{K}_s \\ \mathbf{L}_s\mathbf{H}_s\mathbf{A}_s & \mathbf{A}_s - \mathbf{B}_s\mathbf{K}_s - \mathbf{L}_s\mathbf{H}_s\mathbf{A}_s \end{pmatrix}, \tag{56}$$

$$\overline{\mathbf{G}}_s = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L}_s\mathbf{H}_s & \mathbf{L}_s \end{pmatrix}, \quad \mathbf{q}_k = \begin{pmatrix} W_k \\ V_k \end{pmatrix} \tag{57}$$

It can be shown that if $\overline{\mathbf{F}}_s$ is a stable matrix, i.e. $\lim_{\kappa\to\infty}(\overline{\mathbf{F}}_s)^\kappa = 0$, the pdf over ζ_k converges, where we get $\lim_{k\to\infty}\mathbb{E}[\zeta_k] = \mathbf{0}$ and $\lim_{k\to\infty}\mathbb{C}[\zeta_k] = \mathcal{P}_s$. Stationary covariance $\mathcal{P}_s$ is the solution of Lyapunov equation in (59). It can be proven that in stationary LQG, stability of matrix $\overline{\mathbf{F}}_s$ is a direct consequence of preceding controllability and observability conditions [18]. Thus, we get:

$$\widehat{\mathcal{X}}_s := \lim_{k\to\infty}\mathbb{E}[\mathcal{X}_k] = \lim_{k\to\infty}\mathbb{E}[\zeta_k] + \overline{\mathbf{n}} = \overline{\mathbf{n}} \tag{58}$$

$$\mathcal{P}_s := \lim_{k\to\infty}\mathbb{C}[\mathcal{X}_k] = \lim_{k\to\infty}\mathbb{C}[\zeta_k], \mathcal{P}_s = \overline{\mathbf{F}}_s\mathcal{P}_s\overline{\mathbf{F}}_s^T - \overline{\mathbf{G}}_s\overline{\mathbf{Q}}_s\overline{\mathbf{G}}_s^T \tag{59}$$

$$\mathfrak{B}_s = [\widehat{\mathcal{X}}_s^T, \mathcal{P}_s(:)^T]^T \tag{60}$$

Computing $\mathfrak{B}_s$, the HBRM node β_s corresponding to underlying PRM node $\mathbf{n}$ is defined as:

$$\beta_s = [\mathfrak{B}_s^T, P_s^+(:)^T]^T \in \mathbb{B}_h \tag{61}$$

or equivalently, for the j-th node:

$$\beta_s^j = \left(\begin{pmatrix} \mathbf{n}_j \\ \mathbf{n}_j \\ P_s^+ \end{pmatrix}, \begin{pmatrix} \mathcal{P}_s = \begin{pmatrix} \mathcal{P}_{s,11} & \mathcal{P}_{s,12} \\ \mathcal{P}_{s,21} & \mathcal{P}_{s,22} \end{pmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \right) \tag{62}$$