Proceedings of the 5th RSI
International Conference on Robotics and Mechatronics
October 26-28, 2017, Tehran, Iran

Implementation of Multi-Goal Motion Planning Under Uncertainty on a Mobile Robot

Ali Noormohammadi-Asl, Hamid D. Taghirad
Advanced **R**obotics and **A**utomated **S**ystems (ARAS),
Faculty of Electrical Engineering,
K. N. Toosi University of Technology,
(Email: a.noormohammadi@ieee.org, taghirad@kntu.ac.ir)

Amirhossein Tamjidi
Department of Aerospace Engineering
Texas A&M University,
(Email: ahtamjidi@tamu.edu)

***Abstract*—Multi-goal motion planning under motion and sensor uncertainty is the problem of finding a reliable policy for visiting a set of goal points. In this paper, the problem is formulated as a formidable traveling salesman problem in the belief space. To solve this intractable problem, we propose an algorithm to construct a TSP-FIRM graph which is based on the feedback-based information roadmap (FIRM) algorithm. Also, two algorithms are proposed for the online planning of the obtained policy in the offline mode and overcoming changes in the map of the environment. Finally, we apply the algorithms on a physical nonholonomic mobile robot in the presence of challenging situations like the discrepancy between the real and computation model, map updating and kidnapping.**

I. Introduction

The multi-goal path planning objective is to guide the robot to visit a sequence of goal points while the traversed path is optimized. These goal points have the main role in the planning and are representative of the environment. For example, they can be points of a structure which should be inspected [1], parts of an environment containing important information of it [2] or they can be positions where a service robot should do its tasks [3]. Many problems, such as environment exploration, search, inspection and coverage can be modeled as a multi-goal path planning problem. In this paper, we assume background information such as a prior map, semantic map or a layout of the environment is available to help the robot.

Many studies have accomplished on methods of solving and modeling robotics problems as a multi-goal motion planning. Reference [4] uses an improved self-organizing map-based algorithm for the TSP in the polygonal domain. In [5], a generalized traveling salesman problem with neighborhoods is adopted. In [6] the coverage problem is modeled as a multi-goal path planning problem in an environment with incomplete prior map information. Reference [2] uses the multi-goal motion planning for the environment exploration. Despite the much work on the topic, they mostly focus on finding a short and obstacle free path, and neglect uncertainties in the motion planning. This is while many real robotics problems are involved with uncertainties and ignoring them may cause poor results or even failure in many applications.

In this paper, we take into account the motion and observation uncertainties in the planning. Motion and sensor uncertainties transform the motion planning in the state space to a challenging sequential decision-making problem in the belief space where the state of the robot is available as a probability distribution function over all possible states. The presence of uncertainty leads us to planning in the belief space rather than state space. In [7], the path planning and resource allocation under uncertainty is considered, but it is limited to the discrete space. Reference [8] adds auxiliary nodes to the vicinity of goal nodes to reduce the uncertainty in the multi-goal path planning, but the uncertainty on the edges and the replanning ability are ignored. Hence, we model the multi-goal path planning problem as an asymmetric TSP in the belief space where some background information about the environment is available. However, solving this problem is notoriously difficult and intractable, because the path between two edges is not the direct line connecting them, edges cost are not deterministic, and finding the path between each two goal nodes needs motion planning in the belief space.

In order to solve the problem, we utilize the Feedback-based Information RoadMap (FIRM) algorithm proposed in [9], [10]. FIRM is a motion planning algorithm which formulates the problem as a Partially Observable Markov Decision Process (POMDP) framework and helps to obtain a policy for driving the robot from an initial belief to a target belief. In [11] FIRM is implemented on a physical robot, and they have shown that the FIRM outperforms other motion planning algorithms in the belief space. By using FIRM a graph is obtained, called TSP-FIRM graph, which helps for decision making and obtaining the best policy for searching the goal points using background information of the environment. Then, we propose an algorithm for executing the obtained policy online. Another important challenge that the robot faces in the online mode is the necessity of online replanning when the initial information changes owing to finding new obstacles or the robot's belief changes due to factors such as deviation from the planned path, missing observation causing high uncertainty and kidnapping. Therefore, we propose some methods to overcome these situations and also an algorithm is provided for the case when the map is updated. Finally, we adapt the proposed algorithms to a nonholonomic unicycle robot and evaluate them by implementing on a physical robot. In what follows, we first provide some preliminaries about POMDP and FIRM. Then, we formulate the problem and propose our algorithm, and we implement the algorithm on a nonholonomic mobile robot.

II. Preliminaries

A. POMDP Problem

Decision making under uncertainty is formalized as POMDP or equivalently the stochastic control with imperfect state information problem. By defining the one-step-cost, $c\left(b_k, u_k\right)$, as the cost of taking action u_k at belief b_k where the action $u_k = \pi_k\left(b_k\right)$ is a function of current belief, called policy or planner, POMDP problem can be defined as the following minimization problem:

$$
\begin{aligned}
&J\left(\cdot\right) = \min_{\Pi} \sum_{k=0}^{\infty} \mathbb{E}\left[c\left(b_k, \pi\left(b_k\right)\right)\right] \\
&\pi^* = \underset{\Pi}{argmin} \sum_{k=0}^{\infty} \mathbb{E}\left[c\left(b_k, \pi\left(b_k\right)\right)\right] \\
&s.t. \quad b_{k+1} = \tau\left(b_k, \pi\left(b_k\right), z_k\right), \quad z_k \sim p\left(z_k \mid x_k\right)
\end{aligned} \tag{1}
$$

where $J\left(\cdot\right)$ is the optimal cost-to-go function and τ is a function for estimating the next belief based on the current observation and the last belief and action. The dynamic programming equations of (1) can be written as follows:

$$
\begin{aligned}
&J\left(b\right) = \min_{u} \left\{ c\left(b, u\right) + \int_{\mathbb{B}} p\left(b' \mid b, u\right) J\left(b'\right) db' \right\}, \quad \forall b \in \mathbb{B} \\
&\pi^*\left(b\right) = \underset{u}{argmin} \left\{ c\left(b, u\right) + \int_{\mathbb{B}} p\left(b' \mid b, u\right) J\left(b'\right) db' \right\}, \forall b \in \mathbb{B}
\end{aligned} . \tag{2}
$$

B. A Short Review On FIRM

FIRM is a graph based algorithm for the path planning in the belief space and is independent of the initial belief, in other words, it is a multi-query algorithm. FIRM helps to reduce the MDP over the entire belief space to a tractable MDP over a graph. The FIRM graph node is a small region, $B = \{b : \|b - b'\| \leq \epsilon\}$, around the sampled belief b'. Also, FIRM graph edges are local controllers which each one is a concatenation of the node controller and the edge controller. The feedback structure of the local controller helps to drive the belief to the target node of the edge. Briefly, we have a graph with the set of nodes, $\mathbb{V} = \left\{B^i\right\}_{i=1}^{N}$, and the set of edges $\mathbb{M} = \left\{\mu^{ij}\right\}$. For each FIRM node, B^i, it is assumed that the cost-to-go of all beliefs in the node are approximately equal. Therefore, the transition cost, $C\left(b_c^i, \mu^{ij}\right)$, and the transition probability, $\mathbb{P}^g\left(B^j \mid B^i, \mu^{ij}\right)$, which is the transition probability from B^i to B^j using the local controller μ^{ij} are defined as:

$$
\forall b \in B^i, \forall i, j \begin{cases} B^i = C\left(b_c^i, \mu^{ij}\right) \approx C\left(b, \mu^{ij}\right) \\ \mathbb{P}^g\left(\cdot \mid B^i, \mu^{ij}\right) := \mathbb{P}\left(\cdot \mid b_c^i, \mu^{ij}\right) \\ \qquad \approx \mathbb{P}\left(\cdot \mid b, \mu^{ij}\right) \end{cases} \tag{3}
$$

where b_c^i is a point in B^i. By using this approximation and considering the failure set, F, e.g. obstacles, the intractable DP equation 2 is simplified as:

$$
\begin{aligned}
J^g\left(B^i\right) = \min_{\mu \in \mathbb{M}(i)} C^g & \left(B^i, \mu\right) + J^g\left(F\right) \mathbb{P}^g\left(F \mid B^i, \mu\right) \\
& + \sum_j \mathbb{P}^g\left(B^j \mid B^i, \mu\right) J^g\left(B^j\right), \ \forall i \\
\pi^g\left(B^i\right) = \underset{\mu \in \mathbb{M}(i)}{argmin}\, C^g & \left(B^i, \mu\right) + J^g\left(F\right) \mathbb{P}^g\left(F \mid B^i, \mu\right) \\
& + \sum_j \mathbb{P}^g\left(B^j \mid B^i, \mu\right) J^g\left(B^j\right), \ \forall i
\end{aligned} \tag{4}
$$

where $\mathbb{P}^g\left(F \mid B^i, \mu\right)$ is the probability of hitting the failure set. The FIRM graph offline construction algorithm and the online planning algorithm are presented in [10, Algorithm 3] and [10, Algorithm 4]. Also, the replanning algorithms of FIRM are presented in [11, Algorithm 1] and [11, Algorithm 2].

III. Multi-Goal Motion Planning Under Uncertainty

A. Problem Formulation

In order to formulate the multi-goal path planning problem in the belief space, first we assume the environment is obstacle free, $F = \emptyset$, and there are N_g goal points. We consider the $b^i, i \in V$ as the belief of i-th goal point, and subsequently B^i_{goal} as the i-th goal region where the system stops as the robot's belief enters into it. We define the one-step-cost, the cost of taking the action u at the belief x, as:

$$
\begin{cases}
c_i\left(b, u\right) = 0 & if \quad b \in B^i_{goal} \\
c_i\left(b, u\right) = \mathbb{E}\left[c\left(x, u\right) | \mathcal{H}\right] & \\
\qquad = \int_{\mathbb{X}} c\left(x, u\right) p\left(x | \mathcal{H}\right) dx \geq \varepsilon > 0 & if \quad b \notin B^i_{goal}
\end{cases} \tag{5}
$$

where $\mathcal{H}_k = \{z_{0:k}, u_{0:k-1}\}$ is the data history. We take a positive value for the one-step-cost before reaching the goal region to avoid the robot to stop before reaching it and to trap in an infinite cycle. Consequently, the problem is formulated as follows:

$$
\begin{aligned}
\underset{\{p,\Pi\}}{min} \quad & \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} \sum_{k=0}^{\infty} \mathbb{E}\left[c_j\left(b_k^i, \pi\left(b_k^i\right)\right)\right] \\
s.t. \quad & \sum_{j=1}^{N_g} p_{ij} = 1 \quad (i \neq j, i \in V) \\
& \sum_{i=1}^{N_g} p_{ij} = 1 \quad (i \neq j, j \in V) \\
& \sum p_{ij} \leq |s| - 1 \qquad (s \subset V \quad , \quad 2 \leq |s| \leq N_g - 2) \\
& p_{ij} \in \{0, 1\} \qquad (i, j) \in A \\
& b_{k+1} = \tau\left(b_k, \pi\left(b_k\right), z_k\right)
\end{aligned} \tag{6}
$$

where V and A are the set of graph's vertex and arc, respectively.

B. Multi-Goal Belief Space Planning Using FIRM

As mentioned before, solving the optimization problem in (6) is intractable. Therefore, we exploit FIRM to solve ATSP in the belief space.

1) ATSP With Multi-Path On The FIRM Graph: To construct the TSP-FIRM graph, we form a Probabilistic Road Map (PRM) including the goal points and the sampled nodes $\mathcal{V} = \left\{ \left\{ v^j \right\}_{j=1}^{N_s}, \left\{ v^i_{goal} \right\}_{i=1}^{N_g} \right\}$. Therefore, we design a stabilizer (node controller) for each node in $\mathcal{V}$. Consequently, a TSP-FIRM graph with the nodes $\left\{ \mathcal{B}^i \right\}_{i=1}^{N_t}$ and the edges, i.e. local controller, $\mathbb{M} = \left\{ \mu^{ij} \right\}$ is constructed. N_s is the number of sampled nodes and $N_t = N_s + N_g$ is the number of all TSP-FIRM nodes. In the obtained graph the path between each two goal nodes can be more than one. Therefore, the optimization over the entire belief space is simplified as stated below:

$$\begin{aligned} \underset{\{p,y,\Pi^g\}}{min} \quad & \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} \sum_{q \in Q_{ij}} \sum_{k=0}^{\infty} \mathbb{E} \left[C^g_j \left(B^i_k, \pi^g \left(B^i_k \right) \right) \right] y^q_{ij} \\ s.t. \quad & \sum_{j=1}^{N_g} p_{ij} = 1 \quad (i \neq j, i \in V) \\ & \sum_{i=1}^{N_g} p_{ij} = 1 \quad (i \neq j, j \in V) \\ & \sum p_{ij} \leq |s| - 1 \qquad (s \subset V \quad , \quad 2 \leq |s| \leq N_g - 2) \\ & p_{ij} \in \{0,1\} \qquad (i,j) \in A \\ & y^q_{ij} \in \{0,1\} \qquad q \in Q_{ij}, \ (i,j) \in A \\ & \sum_{q \in Q_{ij}} y^q_{ij} = 1 \qquad \text{for each } (i,j) \\ & \mathbb{P} \left(B^i_{k+1} \mid B^i_k; \pi^g \left(B^i_k \right) \right) \end{aligned} \tag{7}$$

where Q_{ij} is the set of all paths between the goal points i and j, and y^k_{ij} is set to one if the path is selected.

2) Asymmetric TSP on FIRM: Although the optimization problem in (7) is more tractable than (6), but it is not straightforward to solve still, because the selected path between each two goal nodes, y, and the optimal policy to move from one goal node to another goal node, Π^g, are defined as decision variables in the optimization. In order to cope with this difficulty, we use FIRM to find the optimal policy and the best path between each two goal nodes by solving the DP equation (4). Now, we can write the (7) as follows which is the common asymmetric traveling salesman problem:

$$\begin{aligned} \underset{\{p\}}{min} \quad & \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} J^g_j \left(B^i_0 \right) \\ s.t. \quad & \sum_{j=1}^{N_g} p_{ij} = 1 \quad (i \neq j, i \in V) \\ & \sum_{i=1}^{N_g} p_{ij} = 1 \quad (i \neq j, j \in V) \\ & \sum p_{ij} \leq |s| - 1 \qquad (s \subset V \quad , \quad 2 \leq |s| \leq N_g - 2) \\ & p_{ij} \in \{0,1\} \qquad (i,j) \in A \end{aligned} \quad . \tag{8}$$

3) Considering Obstacles in Planning: There are three types of obstacles in the environment. In the first type, obstacles are known in advance and their positions are given in the initial map. In the second type, some areas are potentially obstacle and it is risky to move on them. The last type of the obstacles are completely unknown and the robot detects them in the online mode. After detecting the unknown or potentially obstacles areas, the robot estimates their positions and updates its map. In order to incorporate potentially obstacle areas in the planning, the one-step cost in (5) is rewritten as:

$$\begin{cases} c_i\left(b,u\right) = 0 & \text{if } \left(\left(b \in B^i_{goal} \right) \text{ or } \left(F \text{ happens} \right) \right) \\ & \quad \text{and } \left(b \notin F_{sus} \right) \\ c_i\left(b,u\right) \geq \beta > \varepsilon > 0 & \text{if } \quad \left(b \in F_{sus} \right) \\ \beta > c_i\left(b,u\right) \geq \varepsilon > 0 & \text{if } \quad \text{otherwise} \end{cases} \quad . \tag{9}$$

4) Multi-Goal Motion Planning Algorithm: Algorithm 1 is the general algorithm of TSP-FIRM graph construction. Also, algorithm 2 is presented for online planning and executing the policy obtained in algorithm 1. For the case of finding new obstacles in the online phase, the algorithm 3 is presented. Algorithm 3 decides whether replanning or resolving ATSP is necessary or not and helps the robot to update its map, graph and policy.

IV. Implementation on a Mobile Robot

In this section, we consider generalizing and implementing the proposed algorithms on nonholonomic unicycle mobile robots. The main challenge in the roadmap-based algorithms is designing a proper controller to steer the robot from a start point toward a goal on the planned path. Reference [12] shows that for the point-to-point stabilization problem (node controller) a time-varying or a discontinuous controller is needed. Therefore, we suggest a switching based controller for the posture stabilization of the robot. However, what we need first is designing an estimator for the robot state estimation and constructing the TSP-FIRM nodes. In this implementation, we consider Gaussian noise for the system, and we adopt a Kalman filter for the state estimation.

A. TSP-FIRM Elements

In the following, we describe the elements which are required in the construction of TSP-FIRM.

Algorithm 1: TSP-FIRM graph offline construction

Get the set of search points, $\left\{v^i_{goal}\right\}$;
Sample a set of stabilizer parameters $\mathcal{V} = \left\{v^i\right\}$;
Concatenate search points and stabilizer parameters, $\mathcal{V} = \left\{v^i, v^i_{goal}\right\}$ and construct their corresponding stabilizers, $\mathbb{M} = \left\{\mu^i\right\}$;
Form FIRM nodes $\mathbb{V} = \left\{B^i\right\}$;
Use local controlles for connecting the belief nodes;
Compute the transition cost $C^g\left(B^i, \mu\right)$, transition probabilities $\mathbb{P}^g\left(B^j \mid B^i, \mu\right)$ and $\mathbb{P}^g\left(F \mid B^i, \mu\right)$ for each B^i and $\mu \in \mathbb{M}(i)$ by applying μ at B^i;
forall $\left\{v^i_{goal}\right\}$ **do**
  Solve the graph DP in Eq.(4) to compute pair $\left(\pi_i^{g^*}, J_i^{g^*}\right)$ to take the robot to the B^j_{goal}, $j \neq i$
end
Construct the TSP cost matrix and solve it, $Tour^*$;

Algorithm 2: TSP-FIRM graph online planning

Given an initial belief b_0, operate the best policy π_0, to take the robot into one of the the search nodes, $B_{start} = B^i_{goal}$;
$Cur\_Goal \leftarrow$ The neighbor of B_{start} in the $Tour^*$ as the next goal;
forall *search nodes* **do**
  while $B \neq Cur\_Goal$ **do**
    Considering the robot is in the FIRM node B, choose the local feedback policy $\mu(\cdot) = \pi^g(B)$ where π^g is the global feedback policy;
    Continue applying the local controller $\mu(\cdot)$ until the system falls into a FIRM node B' or it hits the failure set;
    if *Collision happens* **then return** Collision ;
    Update current node $B \leftarrow B'$
  end
  $Cur\_Goal \leftarrow$ The neighbor of Cur_Goal in the $Tour^*$ as the next goal;
end

Algorithm 3: TSP-FIRM graph updating in finding new obstacles

Estimate obstacle position;
Update map;
$F \leftarrow$ Retrieve surrounding edges of the obstacles;
forall *edges,* $\mu \in F$ **do**
  $\boldsymbol{f} = \left[C^g\left(B^i, \mu\right), \mathbb{P}^g\left(B^j \mid B^i, \mu\right), \mathbb{P}^g\left(F \mid B^i, \mu\right)\right]$;
  Recompute the transition cost , transition probability and collision probability $\boldsymbol{f_{new}} = \left[C^g_{new}\left(B^i, \mu\right), \mathbb{P}^g_{new}\left(B^j \mid B^i, \mu\right), \mathbb{P}^g_{new}\left(F \mid B^i, \mu\right)\right]$;
end
if *any TSP-FIRM edge intersect obstacles* **then**
  if *any search node is in the obstacle area* **then**
    Delete all TSP-FIRM nodes in the obstacle area ;
    $newTSP \leftarrow true$;
  end
  Delete all edges and nodes intersect obstacles; Update F and corresponding $\mathbf{f}$;
  $newPlanning \leftarrow true$;
end
if *exists* $\mu \in F$ *such that* $|\boldsymbol{f}_{new} - \boldsymbol{f}| \nless \boldsymbol{\alpha}_{min}$ **then**
  if *exists* $\mu \in F$ *such that* $|\boldsymbol{f}_{new} - \boldsymbol{f}| \nless \boldsymbol{\alpha}_{max}$ **then**
    $newTSP \leftarrow true$;
  end
  $newPlanning \leftarrow true$;
end
if $newPlanning$ **then**
  Replace previous transition costs, transition probabilities and collision probabilities with new computed values;
  if $newTSP$ **then**
    Recompute π^{g^*} and its corresponding J^{g^*} between each search node;
    Construct TSP cost matrix, solve it and assign new goals sequence;
  end
  Replan($b_{current}$);
else
  Graph does not change
end

1) TSP-FIRM Nodes: To construct TSP-FIRM nodes, first the PRM nodes are sampled, and the goal points are chosen. Then, by linearizing the system around each node, we design a stationary Kalman filter (SKF) and a switching based controller in the belief space. Subsequently, the j-th TSP-FIRM node B^j with the center $b^j_c \equiv \left(\mathbf{v}^j, P^j_s\right)$ is obtained where the P^j_s is the covariance matrix obtained in the SKF. The B^j can be shown as:

$$B^j = \left\{b \equiv (x, P) : \left\|x - \mathbf{v}^j\right\| < \delta_1, \left\|P - P^j_s\right\|_m < \delta_2\right\} \tag{10}$$

where $\|\cdot\|$ and $\|\cdot\|_m$ are proper vector and matrix norms.

2) Local Controller: The local controller consists of the edge and node controller. In the starting point of an edge, first the edge controller is activated and steers the robot to the vicinity of the target node of the edge. Then, the node controller is activated to stabilize the system in the target node. We use a switching controller as the node controller. In order to design the edge controller, first we design a series of nominal states and control inputs to drive the robot from the start point to the target point at the end of the edge. Since the unicycle robot is linearly controllable along the PRM edge, a linear controller can be utilized. Therefore, an LQG controller is adopted to help the robot to track the path in the online mode.

3) Motion Model: The motion model of the unicycle mobile robot is as:

$$X_{k+1} = f\left(X_k, u_k, w_k\right) = \begin{pmatrix} x_k + \left(V_k + n_v\right)\delta t \cos\theta_k \\ y_k + \left(V_k + n_v\right)\delta t \sin\theta_k \\ \theta_k + \left(w_k + n_w\right)\delta t \end{pmatrix} \tag{11}$$

where the vector $w_k = \left(n_v, n_w\right)^T \sim \mathcal{N}\left(0, \mathbf{Q}_k\right)$ is the motion noise. The motion noise of the system is considered as a combination of a fixed uncertainty and a part proportional to the control input values as:

$$\mathbf{Q}_k = \mathbf{diag}\left(\left(\eta_v V_k + \sigma_b^V\right)^2, \left(\eta_w w_k + \sigma_b^w\right)^2\right) \tag{12}$$

where in the implementation we take its parameters as $\eta_v = 0.1$, $\eta_w = 0.01$, $\sigma_b^V = 6\ cm/s$ and $\sigma_b^w = 0.08\ rad/s$.

4) Sensor Model: we use a camera mounted on the robot for sensing purpose. The camera detects some black and white patterns called ArUco markers using ArUco library provided in OpenCv [13] and computes the relative range and bearing to the markers. The sensor model is:

$${}^j z_k = \left[\left\|{}^j\mathbf{d}_k\right\|, atan2\left({}^j d_{2_k}, {}^j d_{1_k}\right) - \theta\right]^T + {}^j v, {}^j v \sim \mathcal{N}\left(\mathbf{0}, {}^j\mathbf{R}\right) \tag{13}$$

where ${}^j\mathbf{d}_k = \left[{}^j d_{1_k}, {}^j d_{2_k}\right]^T := \left[x_k, y_k\right]^T - L_j$ and ${}^j v$ is the measurement noise of jth landmark. The measurement noise, in addition to the fixed uncertainty is proportional to the relative distance to the landmark and the angle between the line connecting the camera to the landmark and the wall

$$\begin{aligned} {}^j\mathbf{R}_k = \mathbf{diag}\Big(&\left(\eta_{r_d}\left\|{}^j\mathbf{d}_k\right\| + \eta_{r_\phi}\left|\phi_k\right| + \sigma_b^r\right)^2, \\ &\left(\eta_{\theta_d}\left\|{}^j\mathbf{d}_k\right\| + \eta_{\theta_\phi}\left|\phi_k\right| + \sigma_b^\theta\right)^2\Big). \end{aligned} \tag{14}$$

5) Transition Cost and Probability: The transition cost and probability are computed using the sequential Monte Carlo method. Although this method is time-consuming, it is endurable owing to the offline construction of TSP-FIRM graph. In order to define the transition cost, we consider the estimation accuracy, Φ^{ij}, the mean stopping time of local controller, $\hat{\mathcal{T}}^{ij} = \mathbb{E}\left[\mathcal{T}^{ij}\right]$, and the mean time that the robot moves in the high-risk area, $\hat{\mathcal{T}}^{ij}_{obs} = \mathbb{E}\left[\mathcal{T}^{ij}\right]$. The estimation accuracy is evaluated by the weighted trace of the estimation covariance $\Phi^{ij} = \mathbb{E}\left[\sum_{k=1}^{\mathcal{T}} tr\left(wP_k^{ij}\right)\right]$ where $w = diag\left(\left[w_x, w_y, w_\theta\right]\right)$ is weighting matrix and P_k^{ij} is the system covariance at the k-th time step. Consequently, the transition cost is obtained as $C\left(B^i, \mu^{ij}\right) = \xi_1\Phi^{ij} + \xi_2\hat{\mathcal{T}}^{ij} + \xi_3\hat{\mathcal{T}}^{ij}_{obs}$ where ξ_1, ξ_2 and ξ_3 are proper weighting coefficients.

6) Environment: The experiment has done in the second floor of the Electrical Engineering department at K. N. Toosi University of Technology. Fig. 1 and Fig. 2 show the map of the environment and a real view of the environment, respectively.

B. Experiment Results

1) Offline Phase of TSP-FIRM: Fig. 3a shows the constructed TSP-FIRM graph where red nodes are selected goal points. The black and gray blocks represent the known

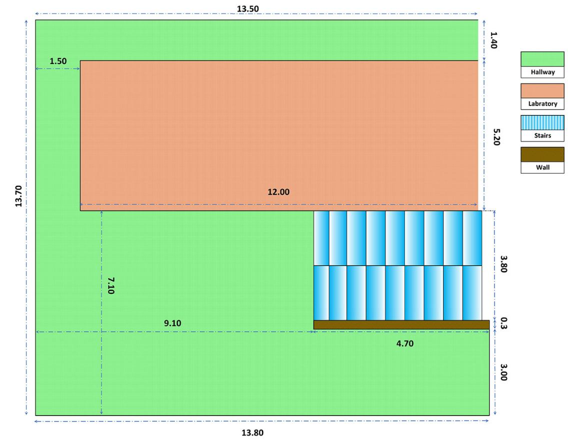
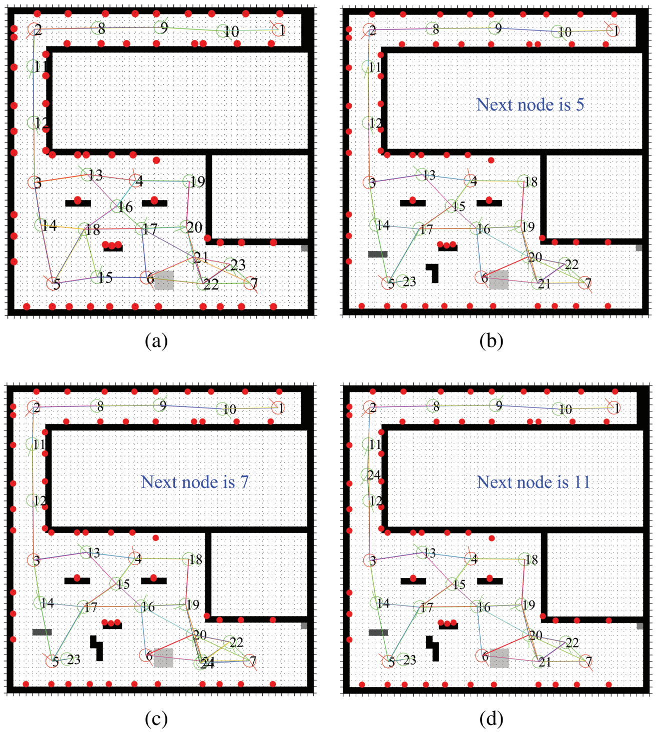


Fig. 1. Environment's layout in real experiment.

Fig. 2. An overview of the real experiment's environment

and potentially (false) obstacles, respectively. The red points on the blocks depict the position of the landmarks. By computing the transition cost and probability as well as the failure probability, the best path and its cost-to-go value between each two goal nodes are computed. Then, the TSP matrix is formed and solved. The nominal sequence of main goal points is obtained as $[1, 2, 3, 4, 5, 6, 7, 1]$, and the nominal path for searching the goal points is $[1, 10, 9, 8, 2, 11, 12, 3, 14, 5, 15, 6, 22, 7, 21, 17, 16, 4, 13, 3, 12, 11, 2, 8, 9, 10, 1]$.

2) Online Phase of TSP-FIRM: We place the robot near to node 1 (laboratory entrance) and the robot starts to search hallway according to the obtained policy in the offline mode. Therefore, it visits goal points 1,2,3 and 5. When the robot starts to move from goal node 5 to goal node 6, it detects unknown obstacles near node 15. Therefore, it updates the map, deletes node 15 and adds the current belief of the robot as the permanent node 23 to the graph (Fig. 3b). It should be noted that, in the replanning, we can add the current belief of the robot to the graph if its covariance is close to the covariance of the robot at this point. Then, the transition cost, transition probability and failure probability of the newly added edge and the neighboring edges of new obstacles is computed. According to the large difference between the previous cost of going from node 5 to 6 and the new one, the ATSP is resolved, and the new search plan is obtained as $[6, 7, 4, 1]$. In resolving



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

Fig. 3. Updates in the Map and graph

the ATSP, we should add a constraint that the robot should return to the start point again. Therefore, we add a dummy node to ATSP where the cost of going from this node to the current node and the cost of going from the start node to the dummy node are set to zero. The cost of other outgoing and incoming edges of the dummy node are set to zero. The robot moves toward node 6 on the path $[23, 5, 17, 16, 6]$, and then it moves toward node 7 on the path $[6, 22, 7]$. However, the robot observes few markers, thus gets highly uncertain about its position and therefore starts gathering information and detecting markers by small movements until the covariance matrix decreases. Then, temporary node 24 is added to the graph (Fig. 3c), and the robot makes a replanning to reach node 7. After that, the robot moves toward node 4 and finally the start point, node 1. However, the robot is kidnapped at a point between node 8 and 9 and is placed on a point between node 11 and 12. The robot after detecting the kidnapping starts gathering information and makes an acceptable estimation of its position. Then, node 24 is added temporarily (Fig. 3d), and the robot makes a replanning to reach node 1. Therefore, it moves toward node 11 and continues its path toward node 1. The video of this experiment is available online at [14].

V. Conclusions

In this paper, we propose TSP-FIRM algorithm for the problem of multi-goal motion planning in the belief space. The problem is formulated as an asymmetric traveling salesman problem in the belief space which is an intractable problem to solve due to the computational burden. To cope with this problem, we exploit FIRM algorithm and propose an algorithm to generate the TSP-FIRM graph which makes the problem more tractable. After solving the problem and obtaining the policy for visiting goal points, we propose an algorithm to execute the policy in the online mode. Furthermore, we propose an algorithm to update the map, graph and policy when a new obstacle is founded in the environment. Then, we implement the proposed algorithm on a nonholonomic mobile robot in a real environment where brings challenges such as finding new obstacles, getting highly uncertain and kidnapping. We use a Kalman filter for the robot localization. Also, the switching based and the LQG controller is used for the posture stabilization and path tracking, respectively. The experiments results show the applicability and efficiency of the proposed algorithms in finding the policy for searching the goal points.

References

[1] Andreas Bircher, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6423–6430. IEEE, 2015.

[2] Stefan Oßwald, Maren Bennewitz, Wolfram Burgard, and Cyrill Stachniss. Speeding-up robot exploration by exploiting background information. *IEEE Robotics and Automation Letters*, 1(2):716–723, 2016.

[3] Richard Bormann, Joshua Hampp, and Martin Hägele. New brooms sweep clean-an autonomous robotic cleaning assistant for professional office cleaning. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4470–4477. IEEE, 2015.

[4] Jan Faigl. On the performance of self-organizing maps for the non-euclidean traveling salesman problem in the polygonal domain. *Information Sciences*, 181(19):4214–4229, 2011.

[5] Kevin Vicencio, Brian Davis, and Iacopo Gentilini. Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2985–2990. IEEE, 2014.

[6] Ling Xu and Tony Stentz. A fast traversal heuristic and optimal algorithm for effective environmental coverage. 2010.

[7] Charles Freundlich, Philippos Mordohai, and Michael M Zavlanos. Optimal path planning and resource allocation for active target localization. In *American Control Conference (ACC), 2015*, pages 3088–3093. IEEE, 2015.

[8] Jan Faigl, Tomáš Krajník, Vojtěch Vonásek, and Libor Přeučil. On localization uncertainty in an autonomous inspection. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1119–1124. IEEE, 2012.

[9] ALI-AKBAR AGHA-MOHAMMADI. *Feedback-based Information Roadmap (FIRM) : Graph-based Estimation and Control of Robotic Systems Under Uncertainty*. PhD thesis, Texas A&M University, 2014.

[10] Ali-akbar Agha-mohammadi, Suman Chakravorty, and Nancy Amato. Firm : Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *International Journal of Robotics Research (IJRR)*, 33(2):268–304, 2014.

[11] Ali-akbar Agha-mohammadi, Saurav Agarwal, Aditya Mahadevan, Suman Chakravorty, Daniel Tomkins, Jory Denny, and Nancy M Amato. Robust online belief space planning in changing environments : Application to physical mobile robots. 2014.

[12] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. Wmr control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on control systems technology*, 10(6):835–852, 2002.

[13] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[14] Multi-goal motion planning under uncertainty using tsp-firm with unknown obstacles. https://www.youtube.com/watch?v=Uuo9Exle9Po&feature=youtu.be.