

# A Framework for Planning Motions in Stochastic Maps

Alireza Nakhaei

CNRS-LAAS

University of Toulouse

7 avenue du Colonel Roche, Toulouse

anakhaei@laas.fr

Florent Lamiroux

CNRS-LAAS

University of Toulouse

7 avenue du Colonel Roche, Toulouse

florent.lamiroux@laas.fr

**Abstract**—In this paper we propose a framework for motion planning in stochastic map. Most of the recent planners are good enough to solve motion planning problems. However, they need a complete and accurate model of the environment and such an assumption may cause a collision in executing the results in a real world with its uncertainties. Considering uncertainties in the model of environment, we reformulate the path planning problem in a stochastic map and then propose a way to modify classical path planning methods in order to fit into this new framework. Our work shares ideas with previous work in this area [11] but follows a different approach. In this framework, sensors and landmarks need to be taken into account. The core computations lie in the evaluation of the probability of collision of configurations with the map.

## I. INTRODUCTION

To perform a task autonomously, a robot should be able to build a map of the environment and to plan motions in this map. In this respect, motion planning and SLAM (simultaneous localization and map building) have been active research fields for the past twenty years. However, these fields have evolved separately. Many robots are able to navigate in indoor 2D maps built by sensors [1], [2], [3]. The problem is well understood for these robots and some companies are starting to sell products with indoor navigation capabilities.

However, for robots with high dimensional configuration spaces like robotic arms mounted on platforms or humanoid robots, performing autonomous motion is still a very difficult problem. Several reasons explain this state of fact. First, 3D SLAM techniques usually build sparse maps of landmarks containing no obstacles [4], [5], [6], [7] and planning motions for a high-dimensional robots requires a 3D-map of obstacles. Second, motion planning algorithms for high-dimensional systems build geometric structures based on the assumption that the map of the world is exact and static. These structures are mainly hierarchies of bounding boxes around obstacles [8] and roadmaps of local paths in the free configuration space [9], [10]. Therefore, these algorithms do not allow robust operations for high-dimensional robots interacting with real world with many uncertainties. In fact, we need to deal with uncertainties in models of the environment evolving along time.

The objective of the work described in this paper is to propose preliminary ideas that will enable path planning techniques for high dimensional systems to deal with models built by sensors. The motivation of this work is crucial in the way to autonomous behaviors for mobile manipulators or humanoid robots in unknown environments. In fact, the concept of autonomy means generating the model of environment with

sensor and implement the techniques of motion planning in this generated model. As such a model which is generated by sensors is not exact and accurate and we have many uncertainties in these models, we are going to propose a new framework for motion planning to deal with this uncertainties.

The contribution of the paper is to propose a new framework to plan motions in uncertain environments represented by stochastic maps as built by SLAM techniques. We do not address yet the non static aspect of the environment. We assume that we have a stochastic Gaussian map composed of landmarks the position of which is represented by a mean vector and a covariance matrix. To make the reasoning simpler, we assume that the landmarks are also obstacles. However, this assumption could be easily relaxed.

Our work shares ideas with previous work in this area [11] but follows a different approach. The main difference comes from the localization notion that is taken into account in the path planning step. The idea is to minimize the probability of collision of a robot that would follow the path planned by the algorithm. This probability naturally depends on the set of landmarks the robot is going to perform localization on during motion. Our approach significantly increases the complexity of the problem but we strongly believe that landmarks should be taken into account right at the path planning step.

The paper is organized as follows. In Section II, definitions are presented and the main concepts of the framework are introduced. In Section III, the method devised to plan paths in stochastic maps is described. Moreover, the main algorithmic point of the method which is the computation of the probability of collision of a random configuration is presented at the end of this section. Section IV gives a very simple example of a path planned by the method.

We would like to emphasize that the contribution of the paper is mainly conceptual. This work is very preliminary and we do not have impressive experimental results yet. However, we truly believe this preliminary work is worth reporting in a paper.

## II. DEFINITIONS

Suppose that a model of an environment and desired initial and goal configurations are given to a planner. The result of this motion planning problem is a path and the robot is supposed to execute this reference path in a real environment. During navigation in the the environment, the robot usually:

- 1) performs localization on landmarks of the map,

- 2) tries to make the result of localization converge toward the configuration along the reference path.

To make the reasoning simpler, let us assume that the closed loop control law of the robot is powerful enough to maintain equality between the result of localization and the reference configuration:

$$\mathbf{q}^{loc}(\mathbf{q}, M) = \mathbf{q}^{ref} \quad (1)$$

where  $\mathbf{q}^{loc}$  is the result of localization that depends on the actual position of the robot  $\mathbf{q}$  and the actual position of landmarks  $M$  (the map). This equation therefore defines a relation between  $\mathbf{q}$ ,  $\mathbf{q}^{ref}$  and  $M$  that we can inverse to get an equality of the form:

$$\mathbf{q} = g(\mathbf{q}^{ref}, M) \quad (2)$$

This equation can be understood as: ‘‘The executed path is the result of both the reference trajectory expressed in a landmark-based frame and the actual position of the landmarks’’. This statement is very intuitive. As a simple example, a planner plan a path for a robot and at the result, the robot is supposed to follow the trajectory along a wall. If the robot perform localization on the wall, the actual trajectory will depend on the real position of the wall also.

If the map does not correspond exactly to the actual position of landmarks, the actual trajectory will be different from the reference trajectory. If the map is represented by a random variable defining the position of each landmark and the correlations between these positions, the trajectory followed by the robot is also a random variable. Therefore, the fact that this trajectory is in collision will be a random event. This idea is the basis of our work. We now give precise definitions for the different notions we will use in our reasoning.

*a) Stochastic map.*: A stochastic map is a random vector  $M$ , defining the position of several landmarks  $\{\mathbf{l}_1, \dots, \mathbf{l}_p\}$  in the environment. Each landmark is represented by a vector  $\mathbf{l}_i$  of dimension  $m_i$ ,  $i \in \{1, \dots, p\}$

$$M = \begin{pmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_p \end{pmatrix} \in \mathbb{R}^m \quad (3)$$

where  $m = \sum_{i=1}^p m_i$ .

*b) Sensor.*: A sensor maps the position of a landmark(the relative position of landmark) to the image space of the sensor:

$$im_i = im_i(\mathbf{q}, \mathbf{l}_i) \in \mathbb{R}^{p_i}, \quad i \in \{1, \dots, l\} \quad (4)$$

where  $im_i$  is the image of  $\mathbf{l}_i$  in the associated sensor. For instance, considering camera as the sensor, a 3D-point ( $\mathbb{R}^3$ ) is projected to the image space as a point of dimension 2. Also, a vertical plane is projected in the image plane of an horizontal laser range scanner as a straight-line.

If  $k$  landmarks are perceived simultaneously, we can put the above equations in a vector as follow:

$$IM = f(\mathbf{q}, M), IM \in \mathbb{R}^r, \mathbf{q} \in \mathcal{C} \quad (5)$$

where

$$IM = \begin{pmatrix} im_1 \\ \vdots \\ im_k \end{pmatrix} \quad r = \sum_{i=1}^k p_i. \quad (6)$$

*c) Localization.*: consists in computing the maximum likelihood configuration, given a perception and a stochastic map:

$$\mathbf{q}^{loc} = f_1(IM, M) \quad (7)$$

$$= f_2(\mathbf{q}, M) \quad (8)$$

after substituting (5).

*d) Closed-loop trajectory tracking.*: To track a reference trajectory, a mobile robot usually implements a closed loop control feedback. The effect of this control process is to make the localization  $\mathbf{q}^{loc}$  converge toward the reference trajectory.

For simplification purposes, we assume that the closed-loop control law is good enough to maintain equality between the reference configuration and the localization:

$$\mathbf{q}^{loc} = \mathbf{q}^{ref} \quad (9)$$

Using (7), this equality defines a relation between the  $\mathbf{q}$ ,  $\mathbf{q}^{ref}$  and the map:

$$\mathbf{q} = f_3(\mathbf{q}^{ref}, M) \quad (10)$$

As a conclusion, given a reference trajectory  $\mathbf{q}^{ref}(s)$ ,  $s \in [0, 1]$  planned in a map of the environment, the trajectory  $\mathbf{q}(s)$ ,  $s \in [0, 1]$  actually followed by the robot depends on the stochastic map  $M$ . This trajectory is therefore a random variable and its collision is a random event.

In the classical formulation of path planning, a trajectory is said to be admissible (or collision-free) if any configuration along this trajectory is collision-free.

In our framework,

**Definition:** A reference trajectory  $\mathbf{q}^{ref}(s)$ ,  $s \in [0, 1]$  is said to be admissible if the probability of collision of the trajectory  $\mathbf{q}(s)$ ,  $s \in [0, 1]$  actually followed by the robot is less than a tolerance  $\varepsilon > 0$ .

The goal of our work is to propose a method for planning admissible paths according to the above definition.

### III. DESCRIPTION OF THE METHOD

In this section, we develop the computations necessary to our method and we introduce some approximations or simplifications.

#### A. Input Data

The data given as input to our path planning algorithms are

- 1) A Gaussian stochastic map built beforehand by SLAM techniques represented by a mean vector  $\bar{M}$  and a covariance matrix  $\Sigma_M$ ,
- 2) An initial configuration  $\mathbf{q}_{init}$
- 3) A goal configuration  $\mathbf{q}_{goal}$ .

## B. Localization

Assuming that the actual configuration of the robot keeps close to the reference configuration, we perform localization by linearizing the relation (5) between the actual configuration and images seen in the sensors about the reference configuration:

$$IM^{ref} + \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q} - \mathbf{q}^{ref}) + \frac{\partial f}{\partial M}(M - \bar{M}) = IM \quad (11)$$

$IM^{ref}$  is the expected image, that is the image seen from reference configuration when  $M = \bar{M}$  and  $IM$  is the image actually perceived.

To find the best estimator  $\mathbf{q}^{loc}$  of  $\mathbf{q}$ , we use Gauss-Markov theorem as follows.

**Theorem:** *Gauss Markov.*

Let  $X$  and  $Y$  be Gaussian random vectors such that

$$Y = H.X + b \quad (12)$$

where  $H$  is a matrix and  $b$  a centered random vector of covariance matrix Identity. The optimal estimator  $\hat{X}$  of  $X$  given an observation of  $Y$  is given by:

$$\hat{X} = (H^T H)^{-1} H^T Y \quad (13)$$

We consider  $\frac{\partial f}{\partial M}(M - \bar{M})$  as a Gaussian noise in (11). The covariance matrix  $\Sigma_b$  of this noise is:

$$\Sigma_b = \frac{\partial f}{\partial M} \Sigma_M \frac{\partial f^T}{\partial M} \quad (14)$$

Based on Gauss-Markov theorem, the best estimator of  $\mathbf{q}$  is:

$$\mathbf{q}^{loc} = \mathbf{q}^{ref} + \left( \frac{\partial f^T}{\partial \mathbf{q}} \Sigma_b^{-1} \frac{\partial f}{\partial \mathbf{q}} \right)^{-1} \frac{\partial f^T}{\partial \mathbf{q}} \Sigma_b^{-1} (IM - IM^{ref}) \quad (15)$$

## C. Feedback control law

As explained earlier, the robot is subject to a closed loop control law that tends to make  $\mathbf{q}^{loc}$  converge toward  $\mathbf{q}^{ref}$ . As an approximation, we assume that these two configurations are constantly equal, condition equivalent to:

$$\frac{\partial f^T}{\partial \mathbf{q}} \Sigma_b^{-1} (IM - IM^{ref}) = 0 \quad (16)$$

On the other hand, according to (11), the images  $IM$  in the sensor depend on the configuration of the robot and on the map. Therefore, substituting this later expression into (16), the result will be:

$$\frac{\partial f^T}{\partial \mathbf{q}} \Sigma_b^{-1} \left( \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q} - \mathbf{q}^{ref}) + \frac{\partial f}{\partial M}(M - \bar{M}) \right) = 0$$

which can be inverted as:

$$\mathbf{q} = \mathbf{q}^{ref} - \left( \frac{\partial f^T}{\partial \mathbf{q}} \Sigma_b^{-1} \left( \frac{\partial f}{\partial \mathbf{q}} \right)^{-1} \frac{\partial f^T}{\partial \mathbf{q}} \Sigma_b^{-1} \frac{\partial f}{\partial M} \right) (M - \bar{M}) \quad (17)$$

This latter equation can be understood in two ways:

- 1) Given a reference configuration  $\mathbf{q}^{ref}$  and an actual position of landmarks  $M$ , we can predict the position

to which the robot will converge when trying to reach  $\mathbf{q}^{ref}$  in the environment.

- 2) Given a reference configuration  $\mathbf{q}^{ref}$  and the stochastic map of the environment, we can express the position the robot will converge to as a random variable depending on the map.

We will use the second one to devise our path planning algorithm.

## D. Path Planning

In Section II, we have given a definition of admissible paths in a stochastic map. In this section, we are going to explain how we plan admissible paths in a meaning close to this definition. To make computations simpler, we indeed consider that a path  $\mathbf{q}^{ref}(s), s \in [0, 1]$  is admissible if and only if

$$\forall s \in [0, 1], P(\mathbf{q}^{ref}(s) \text{ is in collision}) < \varepsilon$$

We then modify existing random motion planning methods such as RRT or PRM. In the classical path planning framework, roadmap based path planning methods pick random configurations and build paths between these configurations. The configurations and local paths are then tested for collision using a collision detection algorithm. We replace the collision detection algorithm by a computation of the probability of collision of random configurations as defined by (17).

To approximate the probability of collision of a configuration with random obstacles of the map, we approximate the vector of random distances of the configuration to obstacles by a Gaussian vector and compute the probability that one of these distances is less than 0.

1) *Distance to Obstacle:* As the configuration of the robot and the obstacles (or landmark) have been considered as Gaussian variables, the distance between the robot and each obstacle is also approximated by a Gaussian random variable.

Given a reference configuration  $\mathbf{q}^{ref}$ , Equation (17) defines a random configuration (depending on  $M$ ). The distance of this configuration to each landmark  $\mathbf{l}_i$  of the map is thus also a random variable depending on  $M$ . We denote  $d_i(\mathbf{q}, \mathbf{l}_i)$  this random distance. We build a random vector by gathering these distances as follows:

$$d(\mathbf{q}, M) = \begin{pmatrix} d(\mathbf{q}, \mathbf{l}_1) \\ d(\mathbf{q}, \mathbf{l}_2) \\ \vdots \\ d(\mathbf{q}, \mathbf{l}_p) \end{pmatrix} \quad (18)$$

To approximate this vector by a Gaussian vector, we linearize the distance function to get:

$$d(\mathbf{q}, M) - d(\mathbf{q}^{ref}, \bar{M}) = \frac{\partial d}{\partial M}(M - \bar{M}) + \frac{\partial d}{\partial \mathbf{q}}(\mathbf{q} - \mathbf{q}^{ref}) \quad (19)$$

Replacing  $\mathbf{q}$  by expression (17), we get a linear relation between  $d(\mathbf{q}, M) - d(\mathbf{q}^{ref}, \bar{M})$  and  $(M - \bar{M})$  that we do not express in this paper for clarity. All the computations can however be done by considering:

- 1) The expression of the distance between the robot and landmarks in a given configuration

- 2) The expression of the images in sensors with respect to the configuration of the robot.

Point 1 requires the kinematic model of the robot and point 2 requires in addition the model of the sensors.

The last equation will result in a Gaussian approximation of vector  $d(\mathbf{q}, M)$  with mean value  $d(\mathbf{q}^{ref}, \bar{M})$  and variance-covariance matrix  $\Sigma_d$ .

After approximating the distance vector, the probability of collision for an instance configuration should be calculated.

2) *Calculating the Probability of collision:* In an uncertain world, the actual positions of obstacles are unknown. It is unclear whether an instance configuration is collides with some obstacles or is in free zone, thus whether we should accept or reject it. We propose a collision probability parameter for making decision on accepting or rejecting a configuration. Therefore, based on the calculated mean value of distance  $\bar{d}$  to obstacles and its variance-covariance matrix  $\Sigma_d$ , the probability of collision for a proposed configuration will be calculated as follow:

$$P(\text{collision}) = 1 - P(\text{coll}\bar{i}sion) = 1 - P(d(\mathbf{q}, l) > 0) \quad (20)$$

where inequality between two vectors is equivalent to inequality between each component. The probability of non-collision will be as follows:

$$P(\text{coll}\bar{i}sion) = \int_{\mathbb{R}} 1_{d>0} dp(d) \quad (21)$$

$$P(\text{coll}\bar{i}sion) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det(\Sigma_d)}} \int_{(\mathbb{R}^+)^n} \exp(-\frac{1}{2}(x - \bar{d})^T \Sigma_d^{-1} (x - \bar{d})) dx \quad (22)$$

As matrix  $\Sigma_d^{-1}$  is a symmetric positive definite matrix, there is a  $P$  such that  $\Sigma_d^{-1} = P^T \cdot P$ . Also, by changing the variable as  $y = P \cdot x$  and  $dy = \det(P) \cdot dx$ , the probability will be ,

$$P(\text{coll}\bar{i}sion) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det(\Sigma_d) \det(P)}} \int_{P[(\mathbb{R}^+)^n]} \exp(-\frac{1}{2}(y - P \cdot \bar{d})^T \Sigma_d^{-1} (y - P \cdot \bar{d})) dy \quad (23)$$

we propose 2 methods for calculating this integral: Monte–Carlo Method and Direct approximation.

a) *Monte-Carlo method:* The integral that we want to calculate is the probability of a Gaussian variable with a mean value of  $P \cdot \bar{d}$  and matrix variance-covariance identity. So, we can use  $n$  independent Gaussian variables with the following algorithm:

b) *Direct approximation method:* In fact, in the second method that we propose, we find a lower bound for this integral which is much faster than the Monte-Carlo method. Actually, we reduce the space of integration from a cone to a hyper-spheres which centered on  $P \cdot \bar{d}$  having the greatest radius (Figure 1). The radius of this hyper-sphere will be equal or minimum distances of  $P \cdot \bar{d}$  to various hyper-plane which limits the cone. We call this radius and the hyperbole of this radius centered on  $P \cdot \bar{d}$  respectively  $R$  and  $B_n(R)$ . So, if the point  $P \cdot \bar{d}$  be inside the cone, there will be a negative distance to one

---

### Algorithm 1 Monte-Carlo Algorithm

---

- 1: Input: Number of test( $N_{max}$ )
  - 2: Initialization: Probability ( $p = 0$ )
  - 3: For  $i=1$  to  $N_{max}$
  - 4: Generating a Gaussian vector  $v$  with mean value of  $P \cdot \bar{d}$  and matrix variance-covariance of identity.
  - 5: IF  $v$  is inside cone  $P[(\mathbb{R}^+)^n]$
  - 6:  $p++$
  - 7: End
  - 8: End
  - 9:  $p \leftarrow \frac{p}{N_{max}}$
  - 10: Output: Probability ( $p$ )
- 

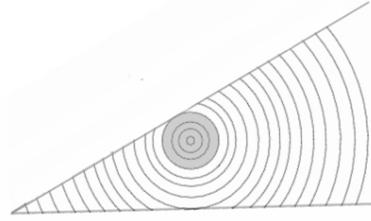


Figure 1. Lower bound of probability of non collision of a configuration. The integral is computed over a ball included in the cone defining non-collision.

of the obstacle and so, the calculation of probability will stop and the result will be:

$$P(\text{collision}) = 1 \quad (24)$$

As the iso-surfaces of our integral are hyper-spheres, we can easily have a hyper-sphere change of variable as below:

$$\int_{P[(\mathbb{R}^+)^n]} \exp(-\frac{1}{2}(y - P \cdot \bar{d})^T (y - P \cdot \bar{d})) dy > \int_{B_n(R)} \exp(-\frac{1}{2}y^T y) dy > \text{const}(n-1) \int_0^R r^{n-1} \exp(-0.5r^2) dr \quad (25)$$

as,

$$\text{If } n \text{ is even, then: } \text{const}(n-1) = \frac{2\pi^{p-1}}{(p-1)!}, n = 2p$$

$$\text{If } n \text{ is even, then: } \text{const}(n-1) = \frac{2^{2p+1} \pi^p p!}{(2p)!}, n = 2p + 1$$

So, the only thing that should be calculated is  $\int_0^R r^{n-1} \exp(-\frac{1}{2}r^2) dr$  which can be calculated by recursive method. So, we can calculate a lower bound for the non collision probability or a upper band for the collision probability as follow:

$$p(\text{collision}) < 1 - \frac{1}{(2P)^{\frac{n}{2}}} \text{const}(n-1) \int_0^R r^{n-1} \exp(-0.5r^2) dr \quad (26)$$

### IV. EXAMPLE

To illustrate the method, we consider 42 obstacles in the environment which are considered as landmarks also. Figure 2 shows the model of environment which the obstacles are located in their mean value of their position. Given the map of environment (the mean vector of obstacle position and its variance-covariance matrix), robots size, and the start and

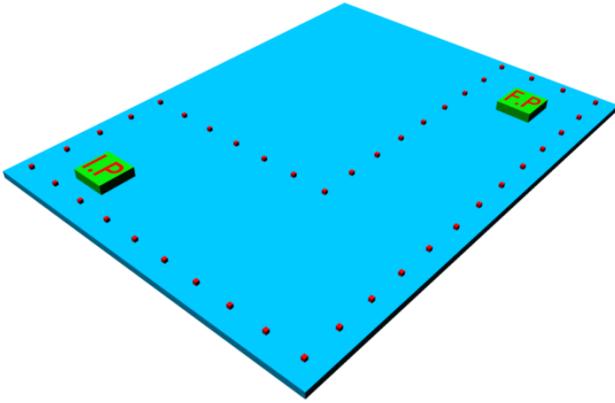


Figure 2. The mean value of the obstacle configurations and the robot initial (I.P) and final (F.P) configuration.

goal configurations, the planner aims to produce a valid path between start and goal.

In fact, the robot is supposed to find a path between the initial point  $I.P$  and final point  $F.P$  in the map and tracks the planned trajectory based on the captured images from the environment. The robot is a two wheeled mobile robot and is equipped with a 2D laser scanner. As it was explained in the last sections, the planner uses the mean value of obstacles (landmark) configuration build the road map based on the proposed probabilistic method and find the appropriate path between  $I.P$  and  $F.P$ . The probability threshold was set to 30% in this example.

Figure 3 shows the planned trajectory in the map of the environment. Red dots represent the mean value of the stochastic map. The red line represents the trajectory planned based on this stochastic map. Blue dot represent a possible real position of the landmarks and the blue line represent the trajectory that would be actually followed by the robot in this environment by performing localisation on the landmarks.

## V. CONCLUSIONS

We have presented in this paper a new framework to take into account uncertainty of the map in path planning. We believe that this framework is more complete than previously published work on this topic.

The main originality of our framework resides in the prevision of the trajectory actually followed by the robot given actual position of landmarks (simulation of localization and motion control process), and in the estimation of collision probability by linking trajectories with landmarks in a stochastic framework.

Let us notice that the input of the path planning algorithm is the output of a SLAM process. We then proposed an algorithm to plan a path in stochastic map. A simple example illustrates this preliminary work on the way to binding path planning with SLAM.

## REFERENCES

[1] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva:

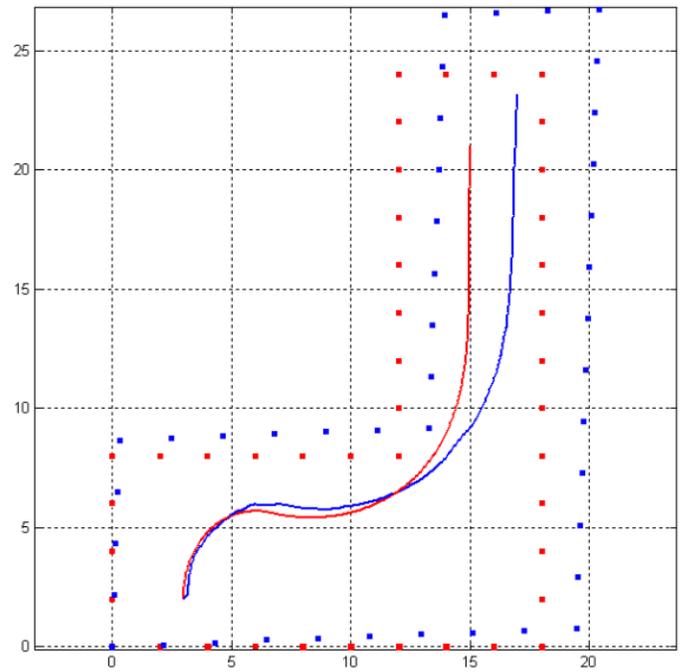


Figure 3. The planned trajectory in the map of the environment. Red dots represent the mean value of the stochastic map. The red line represents the trajectory planned based on this stochastic map. Blue dot represent a possible real position of the landmarks and the blue line represent the trajectory that would be actually followed by the robot in this environment by performing localisation on the landmarks.

a second-generation museum tour-guide robot," in *International Conference on Robotics and Automation*. Detroit (USA): IEEE, May 1999, pp. 1999–2005.

[2] R. Alami, R. Chatila, S. Fleury, M. Herrb, F. Ingrand, M. Khatib, B. Morisset, P. Moutarlier, and T. Siméon, "Around the lab in 40 days..." in *International Conference on Robotics and Automation*. San Francisco, (USA): IEEE, Apr. 2000, pp. 88–94.

[3] B. Jensen, G. Froidevaux, X. Greppin, A. Lorotte, L. Mayor, M. Meisser, G. Ramel, and R. Siegwart, "The interactive autonomous mobile system roblox," in *International Conference on Intelligent Robots and Systems*. Lausanne (Switzerland): IEEE/RSJ, Oct. 2002, pp. 1221–1227.

[4] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transaction on pattern analysis and machine intelligence*, vol. 29, no. 6, June 2007.

[5] T. Lemaire, S. Lacroix, and J. Solá, "A practical 3d bearing-only slam algorithm," in *International Conference on Intelligent Robots and Systems*. Edmonton, Canada: IEEE/RSJ, Aug. 2005.

[6] J. Leonard, R. Carpenter, and H. Feder, "Stochastic mapping using forward look sonar," in *International Conference on Field and Service Robotics*, 1999, pp. 69–74.

[7] C. Estrada, J. Neira, and J. Tardós, "Hierarchical slam: real-time accurate mapping of large environments," *IEEE-TRO*, vol. 21, no. 4, pp. 588–596, 2005.

[8] M. Lin, D. Manocha, J. Cohen, and S. Gottschalk, "Collision detection: Algorithms and applications," in *Algorithmic Foundations of Robotics*, K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, Eds. Wellesley, MA: A K Peters, Ltd., 1995, pp. 129–141.

[9] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for fast path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[10] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *International Conference on Robotics and Automation*. San Francisco, (USA): IEEE, Apr. 2000, pp. 473–479.

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

[12] *International Conference on Robotics and Automation*. San Francisco, (USA): IEEE, Apr. 2000.