Multi-Robot Decentralized Belief Space Planning in Unknown Environments

Tal Regev

Multi-Robot Decentralized Belief Space Planning in Unknown Environments

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Tal Regev

Submitted to the Senate
of the Technion — Israel Institute of Technology
Elul 5776 Haifa September 2016

This research was carried out under the supervision of Assistant Prof. Vadim Indelman from the Faculty of Aerospace Engineering, in the Faculty of Computer Science at the Technion - Israel Institute of Technology.

Acknowledgements

I would like to heartily thank my advisor who not only patiently taught me intricacies of the subject matter, but also suggested new directions of research. This continued support as well as that from my family and parents, was instrumental in making of this thesis. I would also like to thank my lab mates that constantly provided me interesting ideas and suggestions, some of which changed the way of my thinking, especially in the style and content of my presentations. Lastly, I am also grateful for various other friends that I made during these years of studies.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

List of Algorithms

Abstract **1**

Abbreviations and Notations **3**

1 Introduction **5**

2 Literature Review **9**

3 Background **11**

3.1 Factor Graph 11

3.2 Probabilistic Roadmap Planners 13

4 Probabilistic Formulation and Notations **15**

5 Decentralized Sampling-Based Planning **17**

6 Approach and Algorithm Development **21**

6.1 Change in Local Information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ 22

6.2 Impacted Paths and Change in Multi-Robot Factors 22

6.3 Objective Function Re-Evaluation for Candidate Paths 26

6.4 More than 2 Robots 29

6.5 Prior Correlation 30

6.6 Simple Example 32

6.7 A More General Case 33

7 Results **35**

7.1 Basic Scenarios 35

7.2 Large Scale Scenarios 41

8 Conclusion **51**

A Additional Large Scale Scenarios 53

A.1 Scenario 2 53

A.2 Scenario 3 58

Hebrew Abstract i

List of Figures

1.1 Illustration of the proposed concept. The figure shows belief evolution over a few candidate paths of robot r given an announced path $\mathcal{P}^{r'}$ from robot r' and the corresponding multi-robot constraints that can represent, e.g., future mutual observations of environments unknown at planning time [18]. Upon an update in an announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$, a new set of such constraints will be generated (shown in purple), requiring to re-calculate belief evolution for candidate paths. Covariance ellipses are shown for illustration. 7

3.1 An example of a factor graph that the variables X_i can represent the location of the robots and the edges and dot represent the constraints between them. Image taken from [15] 12

3.2 An example of a factor graph representing the joint belief $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ for some candidate path $\mathcal{P}^r$. Different factor graphs are obtained for each path $\mathcal{P}^r$ considering either $\mathcal{P}^{r'}$ or $\mathcal{P}^{r'}_{new}$. 12

3.3 Candidate paths shown on PRM. Robot starting positions are denoted by $\star$. 13

6.1 Graph $G = (V, E)$ along which different candidate paths $\mathcal{P}^r$ of robot r can be defined. Announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ from robot r' facilitate multi-robot factors f_1, f_2, f_3 and f_4. 25

6.2 (a) Illustration of the proposed concept without correlation as show before in Figure 1.1 compare with (b) Illustration of the proposed concept with correlation between robot states at planning time k (referred also as prior correlation in the text) due to mutual past landmark observations. . . . 30

6.3 Prior correlation: (a) with and (b) without multi-robot factors. 31

7.1 Candidate paths shown on PRM. Robot starting positions are denoted by $\star$. 36

7.2 Candidate paths shown on PRM. Robot starting positions are denoted by $\star$. 37

7.3 (Statistics for running time as a function of number of candidate paths for each robot, considering groups of 2 robots. 38

7.4 Statistics for running time as a function of number of candidate paths for each robot, considering groups of 4 robots. 38

7.5 Multi-robot factors (cyan color) and belief evolution (covariance ellipses) for one of the candidate paths from Figure 7.1, considering an announced path $\mathcal{P}^{r'}$. 39

7.6 Illustration of the proposed approach considering a group of two robots (see Figure 6.1). Vertices in V_{inv} for robot r given a (a) previous and (b) new announced path of robot r' are shown as circles. Unchanged multi-robot factors are shown in cyan. Changed multi-robot factors associated with $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ are shown in yellow and magenta, respectively. Only impacted candidate paths of robot r are shown. 40

7.7 `Scenario1`. First planning session. States of different robots are not correlated. (a) Candidate paths to the first goal of each robot; (b) Chosen paths by the planning approach. 43

7.8 `Scenario1`. (a) Multi-robot SLAM given paths determined in the first planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution. 44

7.9 `Scenario1`. Second planning session. (a) Candidate paths to the second goal of each robot; (b) Chosen paths by the planning approach. 45

7.10 `Scenario1`. (a) Multi-robot SLAM given paths determined in the second planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution. 46

7.11 `Scenario1`. Third planning session. (a) Candidate paths to the second goal for the red robot, and to the third goal of each other robot; (b) Chosen paths by the planning approach. 47

7.12 `Scenario1`. (a) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution. 48

7.13 `Scenario1`. Fourth planning session. (a) Candidate paths to the third goal of each robot; (b) Chosen paths by the planning approach. 49

7.14 `Scenario1`. (a) Multi-robot SLAM given paths determined in the fourth planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution. 50

7.15 `Scenario1`. Running time comparison between the proposed and the Standard approach. (a) running time for each planning session. (b) Time ratio. 50

7.16 Scenario3. (a) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution. 50

A.1 Scenario2. First planning session. States of different robots are not correlated. (a) Candidate paths to the first goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the first planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 54

A.2 Scenario2. Second planning session. (a) Candidate paths to the second goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the second planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 55

A.3 Scenario2. Third planning session. (a) Candidate paths to the third goal for the purple robot, and to the second goal of each other robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 56

A.4 Scenario2. Fourth planning session. (a) Candidate paths to the third goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the fourth planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 57

A.5 Scenario2. Running time comparison between the proposed and the Standard approach. (a) running time for each planning session. (b) Time ratio. 57

A.6 Scenario3. First planning session. States of different robots are not correlated. (a) Candidate paths to the first goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the first planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 58

A.7 `Scenario3`. Second planning session. (a) Candidate paths to the second goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the second planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 59

A.8 `Scenario3`. Third planning session. (a) Candidate paths to the third goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution. 60

A.9 `Scenario3`. Running time for each planning session. 60

List of Algorithms

6.1 identifyInvolvedPaths. Identify vertices $V_{inv}^{r} \subseteq V$ involving multi-robot factors considering announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$, and the corresponding multi-robot factors. Each vertex $v \in V_{inv}$ is associated with appropriate multi-robot factors to be later used in Alg. 6.2. 24

6.2 evalObjFunc. Re-evaluate objective function for candidate paths $\mathcal{P}^{r}$ upon update in an announced path from another robot r'. Notations: MR=Multi-Robot; rmv = remove. 27

6.3 updInfo. Update information matrix by adding or subtracting information from factors. 28

Abstract

In this thesis we develop a new approach for decentralized multi-robot belief space planning in high-dimensional state spaces while operating in unknown environments. State of the art approaches often address related problems within a sampling based motion planning paradigm, where robots generate candidate paths and are to choose the best paths according to a given objective function. As exhaustive evaluation of all candidate path combinations from different robots is computationally intractable, a commonly used (sub-optimal) framework is for each robot, at each time epoch, to evaluate its own candidate paths while only considering the best paths announced by other robots. Yet, even this approach can become computationally expensive, especially for high-dimensional state spaces and for numerous candidate paths that need to be evaluated. In particular, upon an update in the announced path from one of the robots, state of the art approaches re-evaluate belief evolution for *all* candidate paths and do so from scratch. In this work we develop a framework to identify and efficiently update *only* those paths that are actually impacted as a result of an update in the announced path. Our approach is based on appropriately propagating belief evolution along impacted paths while employing insights from factor graph and incremental smoothing for efficient inference that is required for evaluating the utility of each impacted path. We demonstrate our approach in a synthetic simulation.

Abbreviations and Notations

BSP	:	Belief space planning
iSAM	:	Incremental smoothing and mapping
SLAM	:	Simultaneous localization and mapping
SFM	:	Structure from motion
PRM	:	Probabilistic roadmap
RRT	:	Rapidly-exploring random tree
RRG	:	Rapidly-exploring random graph
BRM	:	Belief roadmap
RRBT	:	Rapidly-exploring random belief tree
MR	:	Multi robots
PDF	:	Probability density function
MDP	:	Markov decision process
POMDP	:	Partially observable Markov decision process
SEIF	:	Sparse extended information filter
EKF	:	Extended Kalman filter

Chapter 1

Introduction

Collaboration between multiple robots pursuing common or individual tasks is important in numerous problem domains, including cooperative navigation, collaborative mapping and 3D reconstruction in indoor, underwater and urban environments, as well as in various space applications and in the context of autonomous cars. A key required capability is to autonomously determine robot actions while taking into account different sources of uncertainty and to operate autonomously in unknown, uncertain or dynamically changing environments.

The passive instance of the problem (i.e. inference), considering a single-robot setting for now, involves localizing the robot while at the same time constructing or refining a model of the environment. The corresponding problem is known as simultaneous localization and mapping (SLAM), which requires reliable perception, i.e. sensing and correctly interpreting the environment through on-board sensors (e.g. camera, range sensor), and computationally efficient inference. The two processes are commonly known in SLAM community as front-end and back-end, respectively.

The latter stage, involves inference over a high-dimensional state that comprises the robot (past and) current pose and the observed environment thus far represented, for example, by landmarks. In the last decade, much progress has been made in efficiently solving this problem. In particular, state of the art approaches represent the problem via a factor graph graphical model, which naturally encodes the inherent sparsity of the problem (see e.g. [15, 23]). The factor graph will be also used in this work, however, in the context of (belief space) planning.

However, autonomous operation requires not only inference, but also involves a second key-ingredient - determining the best future action(s) while accounting for different sources of uncertainty and given some user-defined high-level objective. The corresponding problem can be formulated within a partially observable Markov decision process (POMDP) framework, which is known to be computationally intractable [33]. Thus, the research community has been extensively investigating approximate approaches to provide better scalability to support real world problems. These methods are commonly referred to as belief space planning (BSP) approaches, as they reason about belief evolution due to different candidate actions. Here, the belief corresponds to the probability density function (pdf) over the state. As in the passive case (SLAM), the latter can be high-dimensional if the environment is uncertain or unknown.

Collaboration between multiple robots can significantly improve performance of both inference and planning phases. In particular, by sharing relevant information between robots, estimation quality can substantially improve, while by appropriately coordinating actions the robots can often finish a task in a shorter time. However, these advantages come with a price of more complicated inference and planning approaches.

In particular, multi-robot belief space planning involves reasoning about all permutations of different candidate actions for different robots, which scales exponentially and thus quickly becomes intractable. This is especially true while operating over high-dimensional state spaces, as we consider here, as each such permutation involves inference, calculating belief evolution given candidate actions, which by itself is computational expensive.

Contribution

In this work we contribute a multi-robot belief space planning approach which further reduces computational complexity, considering the problem of multi-robot autonomous navigation in unknown environments. Instead of re-evaluating from scratch each candidate path, the key observation is that often, belief evolution changes only for part of the candidate paths as a result of an update in the announced path from another robot(s). We show how to identify and efficiently recalculate *only* those candidate paths that are impacted as a result of an update in the announced paths from another robot. See illustration in Figure 1.1. Our approach is based on appropriately propagating belief evolution along impacted paths while employing insights from factor graph for efficient inference that is required for evaluating the utility of each impacted path.

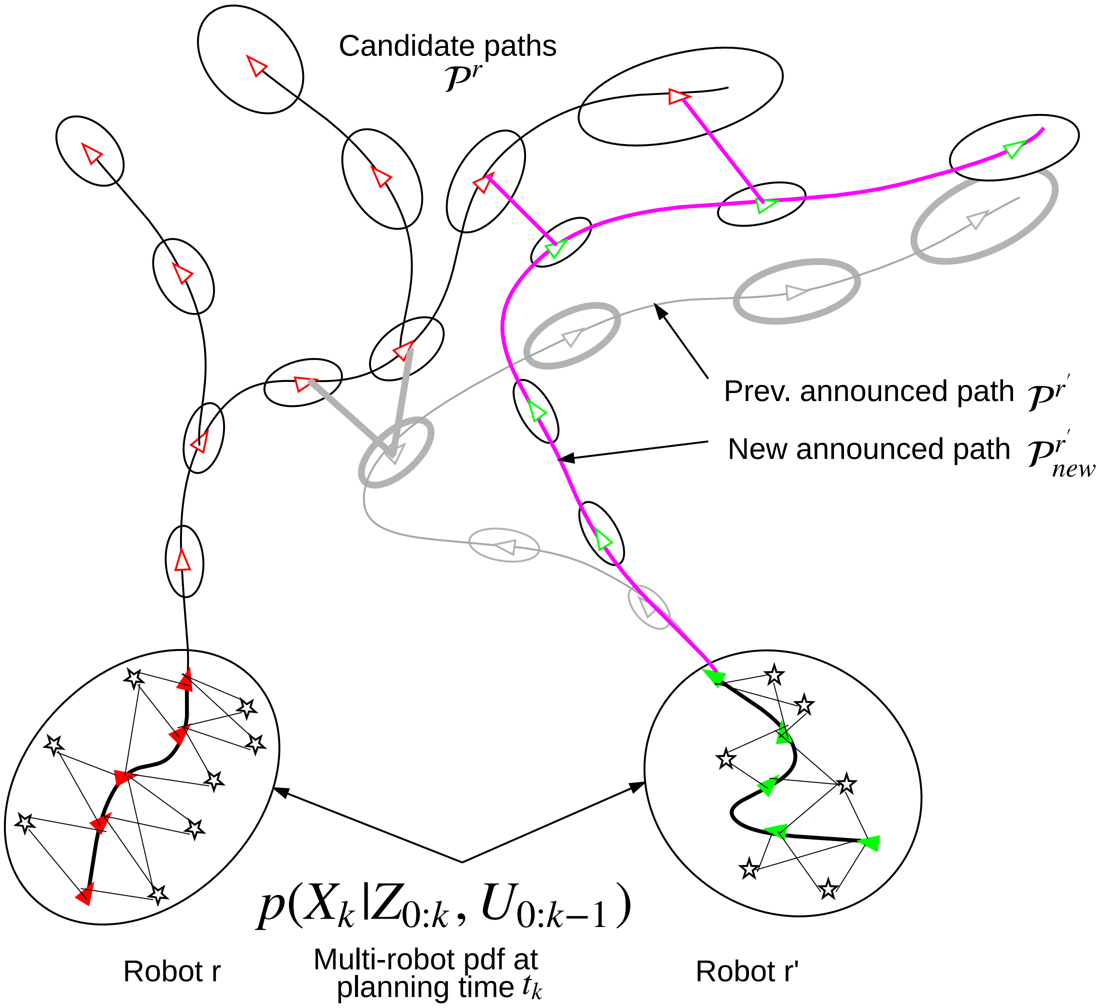

Candidate paths
$\mathcal{P}^r$
Prev. announced path $\mathcal{P}^{r'}$
New announced path $\mathcal{P}^{r'}_{new}$
$p(X_k|Z_{0:k}, U_{0:k-1})$
Robot r
Multi-robot pdf at planning time t_k
Robot r'

Figure 1.1: Illustration of the proposed concept. The figure shows belief evolution over a few candidate paths of robot r given an announced path $\mathcal{P}^{r'}$ from robot r' and the corresponding multi-robot constraints that can represent, e.g., future mutual observations of environments unknown at planning time [18]. Upon an update in an announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$, a new set of such constraints will be generated (shown in purple), requiring to re-calculate belief evolution for candidate paths. Covariance ellipses are shown for illustration.

Chapter 2

Literature Review

As already mentioned, the partially observable Markov decision process (POMDP) framework is computationally intractable [33] for all but the smallest problems. Thus, the research community has been extensively investigating approaches that trade-off computational complexity with sub-optimal performance. These approaches can be roughly classified into four categories, some of which are further discussed below: point-based value iteration methods (e.g. [29]), simulation based approaches (e.g. [37]) in the context of active SLAM, sampling based approaches (e.g. [25, 26, 30]) and direct trajectory optimization approaches (e.g. [20, 35, 38]).

In particular, sampling based approaches (e.g. [5, 10, 17, 36]) discretize the state space using randomized exploration strategies to explore the belief space in search of an optimal plan. While many of these approaches, including probabilistic roadmap (PRM) [26], rapidly exploring random trees (RRT) [30], and RRT* and Rapidly-exploring Random Graph (RRG) [25], assume perfect knowledge of the state, deterministic control and a known environment, efforts have been devoted in recent years to alleviate these restricting assumptions. The corresponding approaches include, for example, the belief roadmap (BRM) [36] and the rapidly-exploring random belief trees (RRBT) [10], where planning is performed in the belief space, thereby incorporating the predicted uncertainties of future position estimates. Similar strategies are used to address also informative planning problems (see, e.g., [17]).

While typically the environment is assumed to be known, recent research focused on facilitating autonomous operation also in the presence of uncertainty in the environment and when the environment is a priori unknown and instead is mapped on the fly, see e.g. [11, 20, 37].

The passive instance of this problem is called simultaneous localization and mapping (SLAM), and has been extensively investigated in the last two decades. As the name suggests, the objective is to solve two problems at once - estimating robot poses and infer the observed environment thus far on the fly. Numerous approaches have been developed over the years, ranging from Davison's EKF-SLAM [13] that considered for the first time real-time performance aspects in a monocular setting, to Eustice's sparse extended information filter (SEIF) [16] and Dellaert's smoothing and mapping (SAM) [15] paradigms. The latter led to the development of incremental SAM approaches [23, 24], most notably iSAM2, which are considered by many as the state of the art in (back-end) SLAM. In this work we utilize these approaches for solving multi-robot SLAM.

The active instance of the problem, i.e. belief space planning in high-dimensional state spaces due to unknown or uncertain environments, is also known as active SLAM. Recent approaches that addressed this problem considering a single robot setting include [20, 28, 37].

A multi-robot belief space framework has been also investigated in different contexts in recent years, including multi-robot tracking, active SLAM and autonomous navigation in unknown environments, planning for coverage tasks, and informative planning (see, e.g. [8,18,19,31]). In particular, recent work [18,19] considered the problem of multi-robot active collaborative estimation while operating in unknown environments and introduced within the belief reasoning regarding future mutual observations of environments that are unknown at planning time. Here, we build upon that work considering a decentralized framework.

Unfortunately, solving exactly the corresponding decentralized POMDP problem is computationally intractable and has been shown to be nondeterministic exponential (NEXP) complete [9], and thus has been typically addressed using approximate approaches. Also, despite the intractable worse case complexity of decentralized POMDP, there has been impressive progress in recent years in solving interesting instances of the problem (e.g. [6]).

A common approach to reduce computational complexity is for each robot, at each time epoch, to solve the belief space planning problem considering its own candidate paths (generated, e.g., by some sampling method) and the best solutions found and announced by other robots (e.g. [8, 31]). The robot then announces its best path, according to a user-defined objective function, to other robots which then proceed with the same procedure. Such an approach avoids solving the problem jointly over all robots and reduces the exponential complexity in the number of robots to linear complexity, with performance guarantees analyzed in [8].

Yet, existing methods calculate the belief evolution over *all* candidate paths from *scratch* each time a new announced plan from another robot is received, which by itself can be computationally extensive operation. In contrast, in this work we develop an approach to identify and efficiently recalculate belief evolution, while re-using calculations, only of impacted paths. As will be seen, our approach yields identical results to the above-mentioned announced path approach, while significantly reducing running time.

Chapter 3

Background

3.1 Factor Graph

Factor graphs are graphical models that are well suited to modeling complex estimation problems, such as Simultaneous Localization and Mapping (SLAM) or Structure from Motion (SFM). A factor graph is a bipartite graph consisting of factors connected to variables. [15]. Variables nodes represent the unknown random variables, while edges represent constrains between appropriate variables, constraints that correspond to motion and measurement models and to prior knowledge. Each such constraint is called a factor.

A toy example for a factor graph is shown in Figure 3.1. As seen, the factor graph has three variable nodes, x_1, x_2, and x_3, that could for example represent robot poses at different time instances. The shown factors include unary and pairwise factors. For example, the variable node x_1 has two unary factors attached to it (factors that only involve a single variable node), that could correspond to prior knowledge and GPS measurements. Pairwise factors, such as the factor that connects x_1 and x_2, could represent a motion model or odometry, for example.

Mathematically, a joint pdf $P(x_1, x_2, x_3|Z, U)$ over the variables x_1, x_2, and x_3 can be factorized as

$$P(x_1, x_2, x_3|Z, U) = \mathbb{P}(x_1)\mathbb{P}(Z_1^{GPS}|x_1)\prod_{i=2}^{3}\mathbb{P}(x_i|x_{i-1}, u_{i-1})\mathbb{P}(Z_i^{GPS}|x_i). \tag{3.1}$$

where the motion and observation models (in this case, $\mathbb{P}(x_i|x_{i-1}, u_{i-1})$ and $\mathbb{P}(Z_i^{GPS}|x_i)$) will be formally defined in the sequel.

The factor graph from Figure 3.1 is merely a graphical representation of this factorization, which abstracts away and encapsulates information. Mathematically, it can be written as

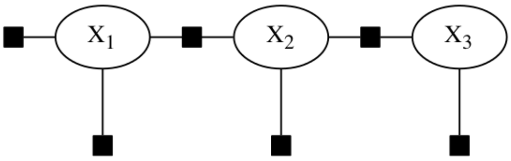


Figure 3.1: An example of a factor graph that the variables X_i can represent the location of the robots and the edges and dot represent the constraints between them. Image taken from [15]

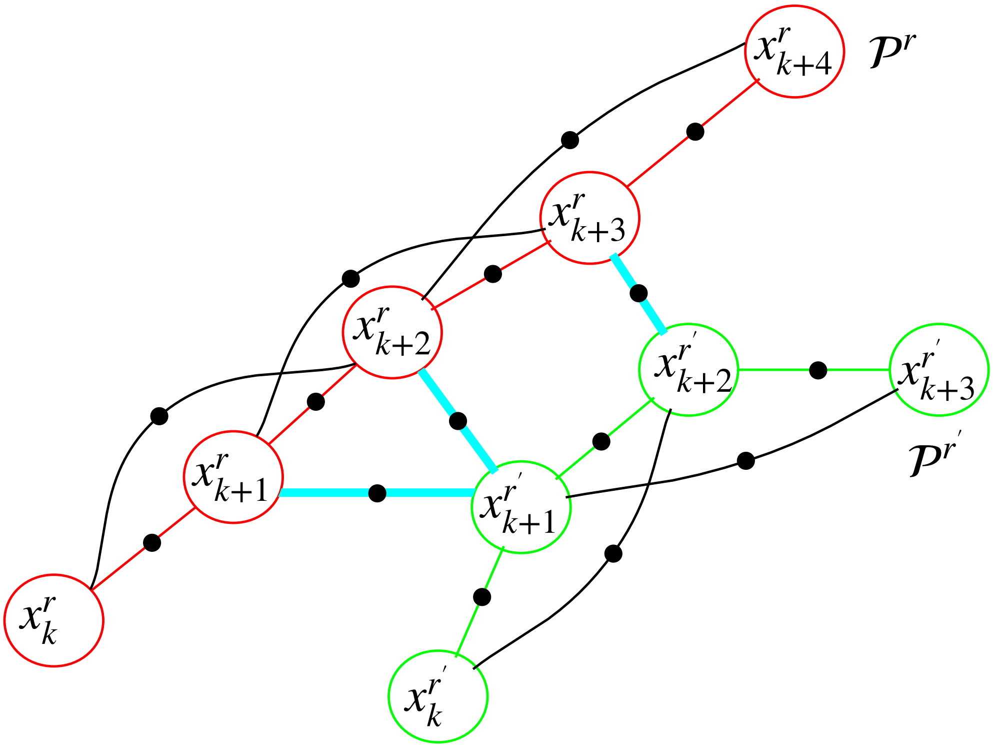


Figure 3.2: An example of a factor graph representing the joint belief $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ for some candidate path $\mathcal{P}^r$. Different factor graphs are obtained for each path $\mathcal{P}^r$ considering either $\mathcal{P}^{r'}$ or $\mathcal{P}^{r'}_{new}$.

$$f(x_1, x_2, x_3) = \prod_i f_i(X_i) \tag{3.2}$$

where $X_i \subseteq X \doteq \{x_1, x_2, x_3\}$ is an appropriate subset of variables for each factor f_i.

In this work we will also use factor graphs to represent the joint pdf - in our context, this joint pdf will represent a future belief over states of multiple robots given specific candidate actions. See illustration in Figure 3.2.

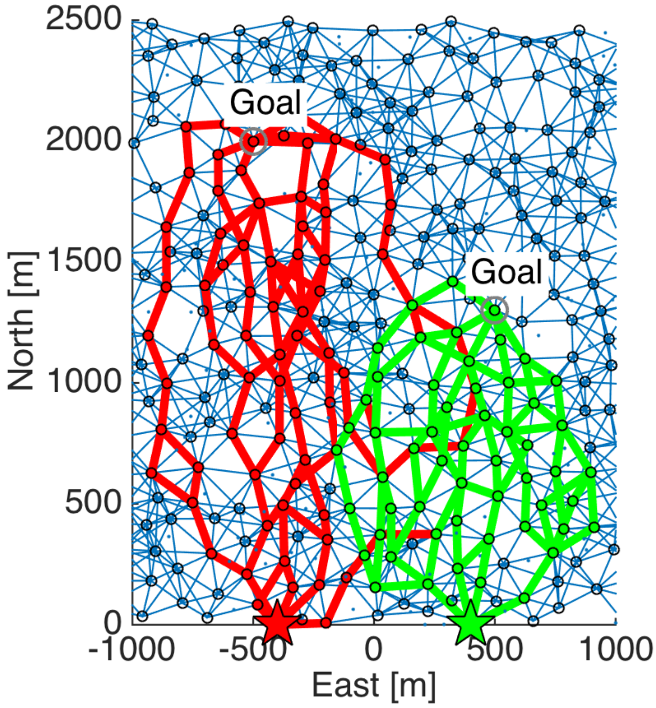


Figure 3.3: Candidate paths shown on PRM. Robot starting positions are denoted by ⋆.

3.2 Probabilistic Roadmap Planners

The Probabilistic RoadMap (PRM) planner [7, 27] is a sampling based method that for given environment (known or unknown) takes a random sample from all over the open space and creates verticies that represent possible future robot locations. Vertices are connected by edges if there is a feasible control action, or robot motion, that brings the robot from one vertex to another.

From a given starting point of the robot, goal and PRM, we can generate multiple candidate paths and reason which path is the best for a given objective function. See illustration for two robots in Figure 3.3.

Until recently, all such approaches considered a Markov Decision Process (MDP) framework, i.e. the state to be fully observable and thus accurately known regardless of the action (which can be stochastic in nature). The seminal work of Prentice and Roy [36] extended the PRM to a partially observable setting, yielding an approach called Belief Roadmap (BRM). The overall idea was to track the evolution of the belief for paths over a PRM, and choose the one that, e.g., minimizes uncertainty at the goal.

Yet, the BRM assumes the environment map is given and relies on known beacons to update the state. In contrast, in this work we consider unknown environments and a multi-robot setting.

Chapter 4

Probabilistic Formulation and Notations

We consider a group of R robots operating in unknown or uncertain environments, aiming to autonomously decide their future actions based on information accumulated thus far and a given objective function J, which is a function of robots' beliefs at different future time instances.

Let $\mathbb{P}(X_k^r | \mathcal{Z}_{0:k}^r, \mathcal{U}_{0:k-1}^r)$ represent the posterior probability distribution function (pdf) at planning time t_k over states of interest X_k^r of robot r (e.g. current and past poses). Here, $\mathcal{Z}_{0:k}^r$ and $\mathcal{U}_{0:k-1}^r$ denote, respectively, all observations and controls by time t_k. Consider conventional state transition and observation models

$$x_{i+1} = f(x_i, u_i, w_i) \ , \ z_{i,j} = h(x_i, x_j, v_{i,j}) \tag{4.1}$$

with zero-mean Gaussian process and measurement noise $w_i \sim N(0, \Omega_w)$ and $v_{i,j} \sim N(0, \Omega_{vij})$, and with known information matrices Ω_w and Ω_{vij}. Denoting the corresponding probabilistic terms to Eq. (4.1) by $\mathbb{P}(x_i | x_{i-1}, u_{i-1})$ and $\mathbb{P}(z_{i,j} | x_i, x_j)$, the pdf $\mathbb{P}(X_k^r | \mathcal{Z}_{0:k}^r, \mathcal{U}_{0:k-1}^r)$ can be written as

$$\mathbb{P}(X_k^r | \mathcal{H}_k^r) \propto \mathbb{P}(x_0^r) \prod_{i=1}^{k} \mathbb{P}(x_i^r | x_{i-1}^r, u_{i-1}^r) p(Z_i^r | X_i^r) \tag{4.2}$$

where the history $\mathcal{H}_k^r$ is defined as $\mathcal{H}_k^r \doteq \{\mathcal{Z}_{0:k}^r, \mathcal{U}_{0:k-1}^r\}$.

The measurement likelihood term $\mathbb{P}(Z_i^r | X_i^r)$ can be expanded in terms of individual observations, $\mathbb{P}(Z_i^r | X_i^r) = \prod_{j=1}^{n_i} \mathbb{P}(z_{i,j}^r | X_{i,j}^r)$. Here, $Z_i^r \doteq \{z_{i,j}^r\}_{j=1}^{n_i}$ and n_i denotes the number of observations acquired at time t_i and $X_{i,j}^r \subseteq X_i^r$ represents involved variables in the jth observation model. Note this formulation assumes known data association and does not consider outliers. Robust perception approaches do exist, however, both in inference (e.g. [32]) and, recently, in belief space planning [34].

We now consider all the R robots in the group, and let $\mathbb{P}(X_k | \mathcal{H}_k)$ represent the pdf over the joint state X_k at time t_k, where $X_k \doteq \{X_k^r\}_{r=1}^R$ and $\mathcal{H}_k \doteq \{\mathcal{Z}_{0:k}, \mathcal{U}_{0:k-1}\}$, with $\mathcal{Z}_{0:k} \doteq \{\mathcal{Z}_{0:k}^r\}_{r=1}^R$ and $\mathcal{U}_{0:k-1} \doteq \{\mathcal{U}_{0:k}^r\}_{r=1}^R$.

Let J denote a user-defined objective function $J(\mathcal{U}) = \mathbb{E}\left[\sum_{l=1}^L c_l(b[X_{k+l}], u_{k+l})\right]$, where $u_{k+l} \doteq \{u_{k+l}^r\}$ and the expectation is taken with respect to future observations of all robots, and where c_l represents an immediate cost function at the lth look ahead step, which can be a function of the joint belief $b[X_{k+l}]$ (to be defined) and of the controls. For simplicity, we use the same planning horizon L for all robots.

In this thesis we consider a special case of the objective function J and assume the latter is of the following form:

$$J(\mathcal{U}) = \mathbb{E}\left[\sum_{l=1}^{L}\sum_{r=1}^{R} c_l^r(b[X_{k+l}^r], u_{k+l}^r)\right], \tag{4.3}$$

where $b[X_{k+l}^r] = \int_{\neg X_{k+l}^r} b[X_{k+l}]$ and thus depends on the multi-robot belief $b[X_{k+l}]$. Such a form naturally supports *collaborative active* state estimation, where each robot aims to improve its estimation accuracy while considering additional terms in c_l, if exist (see e.g. [19]). For example in our simulation, c_l includes two terms: robot pose uncertainty upon reaching the goal and the corresponding path length.

We assume all robots are capable of communicating with each other. Partial communication, or failure in communication, is likely to negatively impact overall performance of the algorithm. However, further analysis of such a scenario is left to future research endeavors.

In this thesis, our objective is to find the optimal controls $\mathcal{U}^\star = \arg\min_{\mathcal{U}} J(\mathcal{U})$ for all robots in the group, considering a multi-robot decentralized framework discussed below.

Chapter 5

Decentralized Sampling-Based Planning

We consider a decentralized framework, where each robot calculates candidate paths using one of the existing sampling-based motion planning approaches (e.g. RRT, RRG, PRM). Adopting typical notations in literature, let $G^r = (V^r, E^r)$ be a graph maintained by robot r, with vertices V^r representing sampled robot states and edges E^r denoting feasible paths between corresponding vertices. Each vertex $v \in V^r$ is associated with a set of belief nodes, with each belief node representing a path $\mathcal{P}^r \doteq \{v_0, \ldots, v\}$ from the initial vertex v_0 that could be followed to reach the vertex v.

In this thesis we interchangeably use $\mathcal{P}^r$ to represent a path and, when clear from context, also the corresponding robot states along that path. Denoting the state at each vertex v by x_v, the corresponding joint belief over the entire path $\mathcal{P}^r$, considering for now only a single robot r, is

$$b[\mathcal{P}^r] \doteq \mathbb{P}(X_k^r, x_{v_0}^r, \ldots, x_v^r | \mathcal{H}_k^r, U(\mathcal{P}^r), Z(\mathcal{P}^r)), \tag{5.1}$$

where $U^r(\mathcal{P}^r)$ and $Z^r(\mathcal{P}^r)$ represent, respectively, the corresponding controls and (unknown) observations to be acquired by following the path $\mathcal{P}^r$. This pdf can be explicitely written in terms of the belief at planning time and the corresponding state transition and observation models as (see Eq. (4.2))

$$b[\mathcal{P}^r] = \mathbb{P}(X_k^r | \mathcal{H}_k^r) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)), \tag{5.2}$$

where, for convenience, the local information (factors) along path $\mathcal{P}^r$ is defined as

$$FG_{local}(\mathcal{P}^r) \doteq \prod_{l=1}^{L(\mathcal{P}^r)} \mathbb{P}(x_{v_l}^r | x_{v_{l-1}}^r, u_{v_{l-1}}^r) \mathbb{P}(Z_{v_l}^r | X_{k+l}^r). \tag{5.3}$$

Throughout the thesis we will often use the factor graph graphical model to represent a pdf. The factor graph for the pdf from Eq. (5.3) is denoted by $FG_{local}(\mathcal{P}^r)$.

The measurement likelihood term $\mathbb{P}(Z^r_{v_l}|X^r_{k+l})$ can be further expanded, similarly to Eq. (4.2). Here, X^r_{k+l} is the joint state up to the lth vertex along the path $\mathcal{P}^r$, i.e.:

$$X^r_{k+l} = X^r_{k+l}(\mathcal{P}^r) \equiv \mathcal{P}^r_{k+l} \doteq \{X^r_k, x^r_{v_0}, \ldots, x^r_{v_l}\}. \tag{5.4}$$

We now proceed to the multi-robot case and consider different paths $\mathcal{P}^r$ for each robot $r \in \{1, \ldots, R\}$. Letting $\mathcal{P} \doteq \{\mathcal{P}^r\}^R_{r=1}$, the multi-robot belief is given by

$$\begin{aligned} b[\mathcal{P}] \quad &= \quad \mathbb{P}(X_k|\mathcal{H}_k) \prod_{r=1}^{R} \left[\prod_{l=1}^{L(\mathcal{P}^r)} \mathbb{P}(x^r_{v_l}|x^r_{v_{l-1}}, u^r_{v_{l-1}}) \right. \\ & \qquad \left. \cdot \mathbb{P}(Z^r_{v_l}|X^r_{k+l}) \prod_{\{i,j\}} \mathbb{P}(z^{r,r'}_{i,j}|x^r_{v_i}, x^{r'}_{v_j}) \right], \end{aligned} \tag{5.5}$$

where the last product corresponds to multi-robot constraints that can involve different time instances, representing mutual observations of a scene. With a slight abuse of notation, we use $x^r_{v_i}$ and $x^{r'}_{v_j}$ in the measurement likelihood term $\mathbb{P}(z^{r,r'}_{i,j}|x^r_{v_i}, x^{r'}_{v_j})$ to represent both a robot state before planning time, i.e. $x^r_{v_i} \subset X^r_k \subseteq X^r(\mathcal{P}^r)$ (likewise for $x^{r'}_{v_j}$), and a future state along the path $\mathcal{P}^r$. The latter case corresponds to a mutual observation of an area that is unknown at planning time, as introduced in our previous work [18].

The index set $\{i, j\}$ in Eq. (5.5) represents the time indices that facilitate multi-robot constraints. We assume a given criteria function $\mathbf{cr_{MR}}(v_i, v_j)$ that determines if there should be a multi-robot constraint between the two vertices v_i and v_j. This function is conceptually similar to the indicator function used in [31], while in our previous work [18] we used a simpler criteria (relative distance between poses). The joint belief (5.5) can be represented by a factor graph graphical model, as illustrated in Figure 6.1. Different candidate paths $\mathcal{P}$ typically yield different factor graphs.

In a decentralized multi-robot framework, each robot maintains the joint belief (5.5) on its own while communicating to each other relevant pieces of information. We assume, for simplicity, each robot is capable of calculating the joint pdf at planning time $\mathbb{P}(X_k|\mathcal{H}_k)$ using one of the recently developed approaches (e.g. [12, 21]). We note that given transition and observation models (4.1), it is sufficient for each robot r' to only transmit (in addition to what is required by multi-robot inference) the corresponding controls to path $\mathcal{P}^{r'}$. Any robot r that receives this information can then formulate the multi-robot belief (5.5) [31].

Evaluating the objective function (4.3) for the considered paths $\mathcal{P}$ involves performing inference over the multi-robot belief (5.5). As shown in prior work (e.g. [11, 20]), this inference can be performed in the information space:

$$\Lambda(\mathcal{P}) = \Lambda_k + \sum_{r=1}^{R} \left[\sum_{l=1}^{L(\mathcal{P}^r)} \Lambda^{r,local}_l + \sum_{\{i,j\}} \Lambda^{r,r'}_{i,j} \right] \tag{5.6}$$

where $\Lambda_l^{r,local} = (F_l^r)^T \Omega_w^r F_l^r + \sum_m (H_{l,m}^r)^T \Omega_{vlm}^r H_{l,m}^r$ and $\Lambda_{i,j}^{r,r'}$ represents the information from the multi-robot constraint term $\mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'})$ in Eq. (5.5). Here, the matrices F and H represent appropriate Jacobians of the state transition and observation models (4.1), linearized about the considered candidate path and the MAP estimate of the joint state at planning (current) time. Observe that the matrices in Eq. (5.6) are assumed to be appropriately augmented (e.g. zero-padded) as the dimensionality of the state increases with l; see similar treatment e.g. in [11, 20].

Recalling that each robot r has numerous candidate paths over the graph G^r, determining the optimal controls involves considering *all* path combinations between different robots, which is computationally intractable. Optimality here refers to choosing the best path from the set of candidate paths.

Instead, a common (sub-optimal) approach for decentralized belief space planning is for each robot r to consider only its own candidate paths and the *announced* paths of other robots, see e.g. [8, 31]. The robot can then select the best path, according to the objective function (4.3), and announce this path to other robots, which then repeat the same procedure on their end. Such an approach reduces the exponential complexity in number of robots to a linear complexity, and can be viewed as a decentralized coordindated descent [8,31], i.e. where robots either repeat this process until convergence [8] or at some frequency [31]. Performance guarantees of such an approach are analyzed in [8].

In particular, when an announced path of some robot r' is updated (e.g. from $\mathcal{P}^{r'}$ to $\mathcal{P}_{new}^{r'}$), robot r has to recalculate the best path by re-evaluating its candidate paths given $\mathcal{P}_{new}^{r'}$. Existing approaches perform this re-evaluation for *all* candidate paths from *scratch*. In contrast, in the following section we develop an approach to identify and efficiently re-evaluate, while re-using calculations, *only* impacted candidate paths due to an update in the announced path.

Chapter 6

Approach and Algorithm Development

Although our approach applies for any number of robots, for simplicity we consider the case of two robots r and r' and re-write the objective function J from Eq. (4.3) as $J(\mathcal{P}^r, \mathcal{P}^{r'}) = \mathbb{E}\left[\sum_{l=1}^{L}[c_l^r(b[X_{k+l}^r], u_{k+l}^r(\mathcal{P}^r)) + c_l^{r'}(b[X_{k+l}^{r'}], u_{k+l}^{r'}(\mathcal{P}^{r'})]\right]$. In Section 6.4 we then generalize back to a general number of robots.

Consider robot r has already calculated belief evolution over *all* candidate paths while accounting for the announced path $\mathcal{P}^{r'}$, and the latter is now updated to $\mathcal{P}^{r'}_{new}$. The corresponding multi-robot beliefs for some candidate path $\mathcal{P}^r$ of robot r are:

$$\begin{aligned} b[\mathcal{P}^r, \mathcal{P}^{r'}] = {} & \mathbb{P}(X_k|\mathcal{H}_k)\mathbb{P}(\mathcal{P}^r|U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \\ & \mathbb{P}(\mathcal{P}^{r'}|U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \prod_{\{i,j\}} \mathbb{P}(z_{i,j}^{r,r'}|x_{v_i}^r, x_{v_j}^{r'}) \end{aligned} \tag{6.1}$$

$$\begin{aligned} b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}] = {} & \mathbb{P}(X_k|\mathcal{H}_k)\mathbb{P}(\mathcal{P}^r|U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \\ & \underline{\mathbb{P}(\mathcal{P}^{r'}_{new}|U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})) \prod_{\{i,j\}} \mathbb{P}(z_{i,j}^{r,r'}|x_{v_i}^r, x_{v_j}^{r'}),} \end{aligned} \tag{6.2}$$

where the changed terms are underlined and denoted in red.

One can consider the joint beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$ to be represented by appropriate two different factor graphs (see Figures 1.1 and 6.1). Re-evaluating the objective function for a candidate path $\mathcal{P}^r$ involves performing MAP inference over the updated factor graph $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. In the general case, the factor graphs will be different for *each* candidate path $\mathcal{P}^r$.

The general concept of our approach is to track the multi-robot factors and local information *change* between the two pdfs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. This information is then used to efficiently perform inference over the updated belief, which is required for re-evaluating the objective function.

Our approach first identifies which candidate paths $\mathcal{P}^r$ of robot r are impacted as a result of the update in the announced plan, and consequently operates *only* over these paths instead of always re-calculating belief evolution over all candidate paths. Second, our approach efficiently calculates the belief evolution over these impacted paths, while re-using calculations where possible.

The main steps of the proposed approach are summarized below and described in detail in the following sections:

1. Section 6.1 calculates the change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$.

2. Section 6.2 identifies the impacted candidate paths $\mathcal{P}^r$ and collects appropriate multi-robot factors to be later used for efficient belief inference.

3. Section 6.3 re-evaluates the objective function for (only) the impacted candidate paths, based on the output of Sections 6.1 and 6.2.

6.1 Change in Local Information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$

We first calculate the change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$. This calculation is used later in Alg. 6.2 for *consistent* inference over appropriate beliefs while avoiding double counting information that is shared by $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$. Specifically, recalling the definition (5.3) of a factor graph $FG_{local}(\mathcal{P}^{r'})$ that represents only the local information along path $\mathcal{P}^{r'}$ we identify which factors only appear in $FG_{local}(\mathcal{P}^{r'})$ or in $FG_{local}(\mathcal{P}^{r'}_{new})$. These factors will then be either added or removed upon re-evaluating belief evolution along impacted candidate paths $\mathcal{P}^r$. We therefore collect these factors into two separate factor graphs:

$$FG^{rmv}_{local} \doteq \{f \mid f \in FG_{local}(\mathcal{P}^{r'}) \wedge f \notin FG_{local}(\mathcal{P}^{r'}_{new})\}$$
$$FG^{add}_{local} \doteq \{f \mid f \notin FG_{local}(\mathcal{P}^{r'}) \wedge f \in FG_{local}(\mathcal{P}^{r'}_{new})\}$$

Additionally, we calculate belief evolution $b[\mathcal{P}^{r'}_{new}]$ along path $\mathcal{P}^{r'}_{new}$ taking into account *only* local information of robot r', and use it to calculate the *change* in the immediate cost functions $c^{r'}_l$ between $b[\mathcal{P}^{r'}_{new}]$ and $b[\mathcal{P}^{r'}_{new}]$. Denoting this change by $\Delta c^{r'}_l$ we let $\Delta J^{r'} \doteq \mathbb{E}\left[\sum_{l=1}^{L} \Delta c^{r'}_l\right]$. This quantity will be used to very efficiently re-evaluate the objective function for candidate paths $\mathcal{P}^r$ that are *not* impacted, as discussed in Section 6.3.

6.2 Impacted Paths and Change in Multi-Robot Factors

Next, we identify, among all the candidate paths of robot r, those paths $\mathcal{P}^r$ that are impacted as a result of the update in the announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$. In other words, recalling Eqs. (6.1)-(6.2), we are interested in finding paths $\mathcal{P}^r$ such that $b[\mathcal{P}^r] \neq b'[\mathcal{P}^r]$, with

$$b[\mathcal{P}^r] = \int b[\mathcal{P}^r, \mathcal{P}^{r'}] d\mathcal{P}^{r'} \ , \ b'[\mathcal{P}^r] = \int b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}] d\mathcal{P}^{r'}_{new} \tag{6.3}$$

Such paths $\mathcal{P}^r$ are marked, indicating that the objective function should be re-evaluated, a process that involves re-calculating belief evolution. On the other hand, belief evolution re-calculation is not required for candidate paths that are *not* impacted. In the latter case, the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'})$ is only updated due to the change in immediate cost functions $c_l^{r'}$ of robot r', as discussed in Section 6.3.

We now describe our approach to identify the impacted paths, as well as collecting the required information that will be used in Section 6.3 (Alg. 6.2) for efficient inference.

The *key observation* is that the belief over path $\mathcal{P}^r$ is impacted due to an announced path $\mathcal{P}^{r'}$ only if there exist multi-robot factors $\mathbb{P}(z_{i,j}^{r,r'}|x_{v_i}^r, x_{v_j}^{r'})$ or, in certain cases, if the states of robots r and r' are already correlated at planning time, i.e.

$$\mathbb{P}(X_k|\mathcal{H}_k) \neq \mathbb{P}(X_k^r|\mathcal{H}_k)\mathbb{P}(X_k^{r'}|\mathcal{H}_k) \tag{6.4}$$

This is the case if, by planning time t_k, the robots have already performed some multi-robot update, e.g. by mutually observing a common scene.

Clearly, in absence of multi-robot factors and prior correlation, the belief over a candidate path $\mathcal{P}^r$ is not impacted by neither $\mathcal{P}^{r'}$ nor $\mathcal{P}_{new}^{r'}$. However, it is also interesting to note that also when there is prior correlation, but *no changes* in multi-robot factors between $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$, the belief over path $\mathcal{P}^r$ typically remains the same. In what follows we treat prior correlation just as a multi-robot factor, see Figure 6.1. However, this observation can be used to switch to a more efficient version of the algorithm that approximately recovers the pdf $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. Investigation of this direction is left to future research.

As mentioned in Chapter 6, our approach tracks the changed multi-robot factors and the local factors of robot r' between the beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. This information is then used in Section 6.3 to efficiently re-evaluate the belief over path $\mathcal{P}^r$. However, such a procedure is required for *each* candidate path $\mathcal{P}^r$ that has some multi-robot factors, even if several paths are identical up to some point. This would lead to the same work (i.e. computational effort) done multiple times.

To address this issue, rather than reasoning about robot r's candidate paths, we reason in terms of the corresponding *vertices* in the graph G^r, that define the paths. Our approach, summarized in Alg. 6.1, considers the corresponding graph vertices and identifies the vertices $V_{inv} \subseteq V^r$ that are involved in at least one multi-robot factor due to either $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$. See illustration in Figure 6.1. We then associate to each such vertex $v_i \in V_{inv}$ the *changed* multi-robot factors that involve v_i, i.e. any such factor f should either appear in $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ or in $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. In the former case, f should be removed from the corresponding factor graph, and as such is added to $v_i.FG_{MR}^{rmv}$ (line 20); in the latter case, f should be added and is thus added to $v_i.FG_{MR}^{add}$ (line 22).

Finally, the algorithm marks all paths $\mathcal{P}^r$ that include at least one vertex in V_{inv}^r as impacted paths (line 25), to indicate belief re-evaluation is required.

```
Inputs:
    $G^r = (V^r, E^r)$: graph of robot $r$
    $P^{r'}, P^{r'}_{new}$: prev. and updated announced path of robot $r'$
    $\mathbf{cr_{MR}}(v_i, v_j)$: multi-robot factor criteria function
Outputs:
    $V^r_{inv}$: involved vertices in multi-robot factors
    $\forall v \in V_{inv} : v.FG^{add}_{MR}, v.FG^{add}_{MR}$
$V^r_{inv} = \phi$ /* Initialization */
foreach $v^{r'} \in \mathcal{P}^{r'} \cup \mathcal{P}^{r'}_{new}$ do
    Find all nearby vertices $\{v\} \subseteq V^r$ to $v^{r'}$ such that
    - at least one candidate path $\mathcal{P}^r$ goes through $v$
    - multi-robot criteria $\mathbf{cr_{MR}}(v, v^{r'})$ is satisfied
    $V^r_{inv} = V^r_{inv} \cup \{v\}$
    foreach $v_i \in \{v\}$ do
        Generate multi-robot factor $f(x^r_{v_i}, x^{r'}_v)$
        if $v_i \in \mathcal{P}^{r'}$ and $v_i \in \mathcal{P}^{r'}_{new}$ then
            continue
        end
        if $v_i \in \mathcal{P}^{r'}$ then
            Add $f(x^r_{v_i}, x^{r'}_v)$ to $v_i.FG^{rmv}_{MR}$
        else
            Add $f(x^r_{v_i}, x^{r'}_v)$ to $v_i.FG^{add}_{MR}$
        end
    end
    Mark all candidate paths $\mathcal{P}^r$ that go through vertex $v_i$
end
return $V^r_{inv}$
```

Algorithm 6.1: `identifyInvolvedPaths`**.** Identify vertices $V^r_{inv} \subseteq V$ involving multi-robot factors considering announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$, and the corresponding multi-robot factors. Each vertex $v \in V_{inv}$ is associated with appropriate multi-robot factors to be later used in Alg. 6.2.

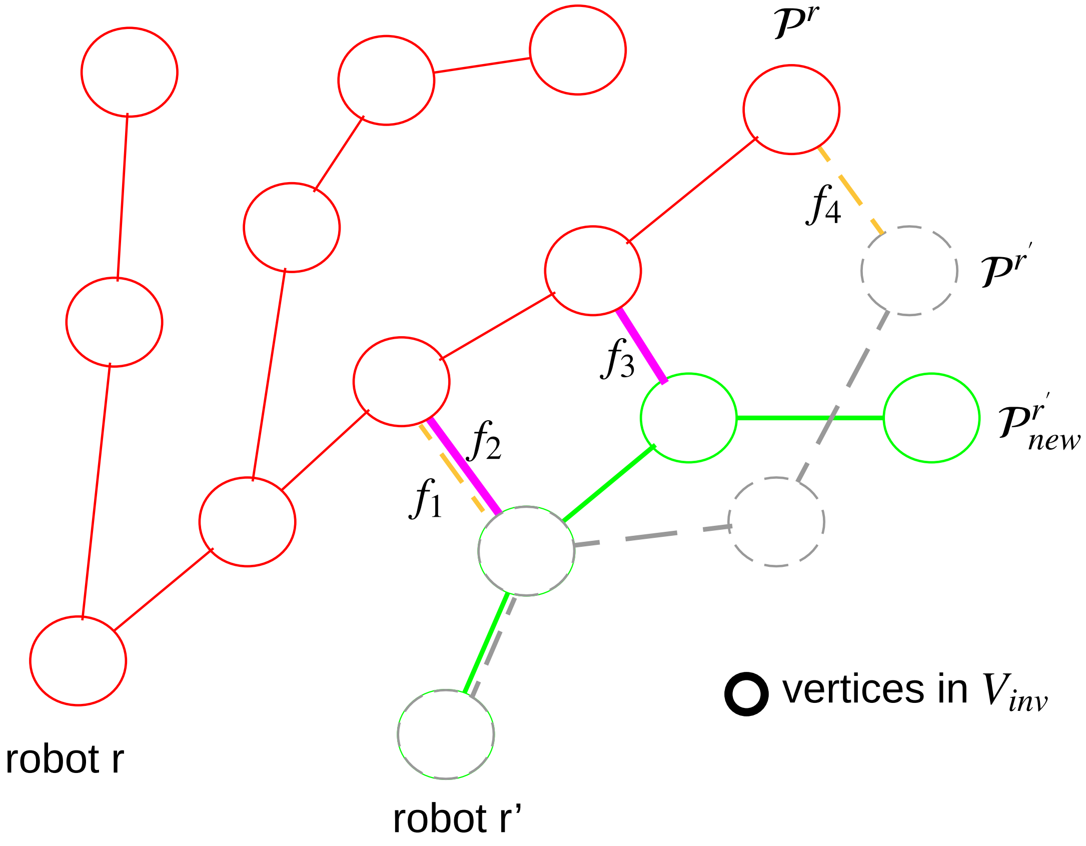


Figure 6.1: Graph $G = (V, E)$ along which different candidate paths $\mathcal{P}^r$ of robot r can be defined. Announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ from robot r' facilitate multi-robot factors f_1, f_2, f_3 and f_4.

6.3 Objective Function Re-Evaluation for Candidate Paths

As mentioned in Chapter 5, each robot r evaluates the objective function by considering its candidate paths and the announced paths of different robots. Such a process requires performing inference over the belief $b[X_{k+l}]$, for each look ahead step l, to recover its first two moments

$$b[\mathcal{P}^r_{k+l}, \mathcal{P}^{r'}_{k+l}] = \mathcal{N}(\mu_{k+l}, \Lambda^{-1}_{k+l}), \tag{6.5}$$

where the general form for the information matrix Λ_{k+l} is given by Eq. (5.6). Observe that if the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'})$ only includes immediate cost functions for some of the look ahead steps l, then the above inference is only required for these time instances. For example, one may be interested only in the uncertainty at the final step (e.g. upon reaching a goal), in which case inference should be performed only for $l = L$. On the other hand, in chance-constrained motion-planning (see e.g. [10]), belief evolution is typically needed for many (or all) look ahead steps l.

Since the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'})$ has been already calculated for different candidate paths $\mathcal{P}^r$ and the announced path $\mathcal{P}^{r'}$, a process that also involves inference over the corresponding beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$, our objective now is to efficiently evaluate the objective function considering the *updated* announced path $\mathcal{P}^{r'}_{new}$.

Our approach for re-evaluating the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ for each candidate path $\mathcal{P}^r$, while exploiting results from the previous inference $b[\mathcal{P}^r, \mathcal{P}^{r'}]$, is summarized in Alg. 6.2 and further discussed below.

The algorithm calculates the maximum a posteriori (MAP) information matrix that corresponds to the belief $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$ for each of the future time instances, which is then used for evaluating the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$. Let $\Lambda \doteq \Lambda(\mathcal{P}^r, \mathcal{P}^{r'})$ and $\Lambda' \doteq \Lambda(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ represent the corresponding MAP information matrices to the beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$, respectively. Denote also by Λ_{k+l} the information matrix that corresponds to the belief over the first l steps, $b[\mathcal{P}^r_{k+l}, \mathcal{P}^{r'}_{k+l}]$, and likewise for Λ'_{k+l}. Since inference over $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ has been already performed, the matrices Λ_{k+l} for all steps l are known. We now focus on calculating Λ'_{k+l}, for each candidate path $\mathcal{P}^r$.

If a candidate path $\mathcal{P}^r$ has been determined in the previous section *not* to be impacted as a result of the update in the announced path (from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$), there is no need to recalculate the immediate functions c^r_l of robot r. We note this holds true due to the considered form of J, where c^r_l only involves $b[\mathcal{P}^r_{k+l}]$ and not also $b[\mathcal{P}^{r'}_{k+l}]$. The latter can still change due to new local information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$, but that change does not affect c^r_l (since $b[\mathcal{P}^r] = b'[\mathcal{P}^r]$). Therefore, to get $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ from $J(\mathcal{P}^r, \mathcal{P}^{r'})$ we only have to update the terms $c^{r'}_l$ (lines 8-11 in Alg. 6.2). This update is the same for all non-impacted paths $\mathcal{P}^r$ and is given by $\Delta J^{r'}$ from Section 6.1. We note, however, that often, $\Delta J^{r'}$ is negligible.

For each marked (impacted) path $\mathcal{P}^r$ and for each $l \in L(\mathcal{P}^r)$, we start with the

```
Inputs:
    $V^r_{inv}$: involved vertices in multi-robot factors
    For each candidate path $\mathcal{P}^r$: $J(\mathcal{P}^r, \mathcal{P}^{r'})$; $\forall l \in L(\mathcal{P}^r) : \Lambda_{k+l}$ from Eq. (6.5)
    $\Delta J^{r'}$ from Sec. 6.1
Outputs:
    For each impacted candidate path $\mathcal{P}^r$: $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$; $\forall l \in L(\mathcal{P}^r) : \Lambda'_{k+l}$
foreach candidate path $\mathcal{P}^r$ do
    if $\neg \mathcal{P}^r.isMarked$ then
        $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new}) = J(\mathcal{P}^r, \mathcal{P}^{r'}) + \Delta J^{r'}$
        continue
    end
    /* re-evaluate belief over $\mathcal{P}^r$                                    */
    $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new}) = 0$
    for $l = 1 : L(\mathcal{P}^r)$ do
        if $\Lambda^{'}_{k+l}$ is not required in Eq. (4.3) then
            continue
        end
        /* Get previous belief $b[\mathcal{P}^r_{k+l}, \mathcal{P}^{r'}_{k+l}]$                  */
        $\Lambda_{k+l} \doteq \Lambda_{k+l}(\mathcal{P}^r, \mathcal{P}^{r'})$ from Eq. (6.5)
        /* Initialize $\Lambda'_{k+l} \doteq \Lambda_{k+l}(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$          */
        $\Lambda'_{k+l} = \Lambda_{k+l}$
        foreach $v \in \mathcal{P}^r$ and $v \in V^r_{inv}$ do
            /* MR factors involving $v \in V^r_{inv}$                              */
            $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, v.FG^{rmv}_{MR}, l$, rmv)
            $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, v.FG^{add}_{MR}, l$, add)
        end
        /* Changed local info. of robot $r'$                                   */
        $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, FG^{add}_{local}, l$, add)
        $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, FG^{rmv}_{local}, l$, rmv)
        Evaluate $c^r_l$ and $c^{r'}_l$ from Eq. (4.3)
    end
end
```

Algorithm 6.2: `evalObjFunc`. Re-evaluate objective function for candidate paths $\mathcal{P}^r$ upon update in an announced path from another robot r'. Notations: MR=Multi-Robot; rmv = remove.

```
Inputs:
    $FG, l$: factor graph and time index
    Linearlization point = graph vertices $V$ and $\hat{X}_k$
    toAddflag: indicates if to add or subtract information
    $\Lambda$: input information matrix to be updated
Outputs:
    $\Lambda$: updated information matrix
$\{f\}$ = getFactorsCausal($FG, l$)
foreach $f \in \{f\}$ do
    Linearize $f$ about linearization point and calculate $\Lambda(f)$
    Adjust size of $\Lambda$, if needed
    if toAddflag then
        $\Lambda = \Lambda + \Lambda(f)$
    else
        $\Lambda = \Lambda - \Lambda(f)$
    end
end
```

Algorithm 6.3: `updInfo`. Update information matrix by adding or subtracting information from factors.

previously calculated information matrix Λ_{k+l} and update it by adding and subtracting the multi-robot and local factors that were collected as explained in Sections 6.1 and 6.2. See lines 16-24 in Alg. 6.2.

Specifically, referring to Eq. (5.6), and resorting to factor graph notation $FG \doteq b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $FG' \doteq b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$, the updated information matrix Λ'_{k+l} can be written as

$$\Lambda'_{k+l} = \Lambda_{k+l} - \sum_{\substack{f \in FG \\ f \notin FG' \\ f.t \leq t_{k+l}}} \Lambda(f) + \sum_{\substack{f \in FG' \\ f \notin FG \\ f.t \leq t_{k+l}}} \Lambda(f). \tag{6.6}$$

The operator $f.t$ extracts the time instances involved with the factor f, such that the condition $f.t \leq t_{k+l}$ enforces *causality*, i.e. we do not consider factors involving states at times greater than $k + l$. The corresponding steps are summarized in Alg. 6.3 that is invoked by Alg. 6.2. We assume existence of the function `getFactorsCausal` that takes as input a factor graph and time t, and outputs only factors involving variables up to that time. Given these factors, Alg. 6.3 extracts the corresponding information matrices and adds or substracts these matrices as in Eq. (6.6). This process involves linearizing the corresponding nonlinear functions, where the linearization point is either the graph vertices V or, in case states from X_k are involved, the corresponding MAP estimate $\hat{X}_k$ of $\mathbb{P}(X_k|\mathcal{H}_k)$, which is known at time k.

We note that, similar to Eq. (5.6), the information matrices in Eq. (6.6) should be appropriately augmented: for example, the matrices Λ_{k+l} and Λ'_{k+l} represent uncertainty over two partially overlapping joint states $\{X^r_{k+l}(\mathcal{P}^r), X^{r'}_{k+l}(\mathcal{P}^{r'})\}$ and

$\{X^r_{k+l}(\mathcal{P}^r), X^{r'}_{k+l}(\mathcal{P}^{r'}_{new})\}$, respectively.

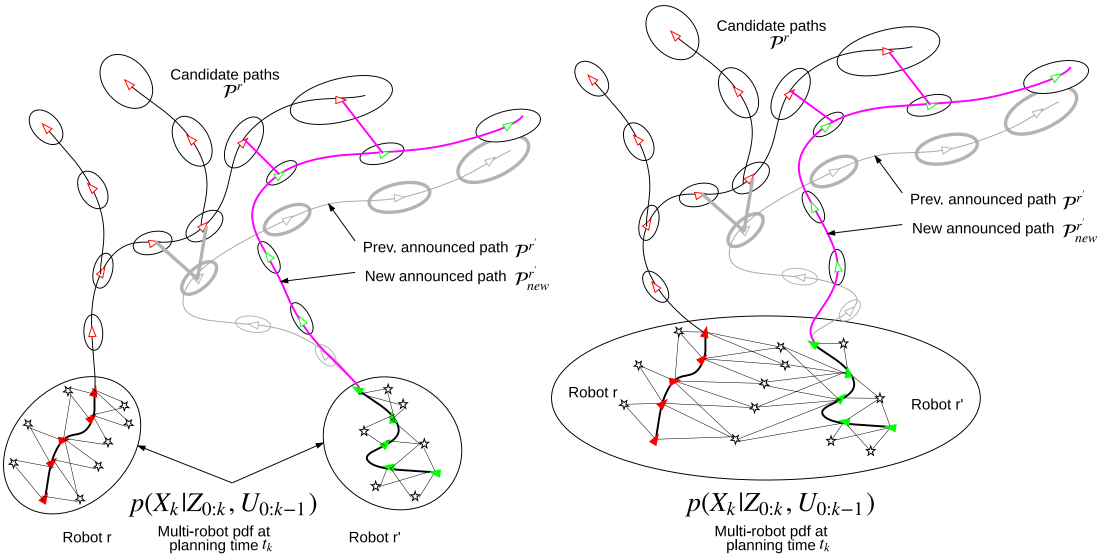
One can go further, and perform the calculation in Eq. (6.6) *incrementally*, by updating Λ'_{k+l+1} based on Λ'_{k+l} while adding and subtracting information from appropriate factors that involve time $k+l+1$. This would provide an efficient mechanism to evaluate the belief for each look ahead step, if that is required by the objective function J. We leave further investigation of this direction to future research and formulate Alg. 6.2 according to 'batch' version (Eq. (6.6)).

Illustrative Example Figure 6.1 illustrates key aspects of our approach. The figure indicates the set V_{inv} of involved vertices in multi-robot factors in either $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ by bold circle marks. As seen there are three such vertices (v_i, v_{i+1} and v_{i+2}) and four multi-robot factors (f_1, f_2, f_3 and f_4). As detailed in Alg. 6.1, each vertex $v \in V_{inv}$ includes the *changed* multi-robot factors that have to be either added or removed. In this example, for v_i there are no changed factors, since although originating from different paths, f_1 and f_2 are actually identical factors. On the other hand, v_{i+1} includes the factor f_3 to be removed, while v_{i+2} includes the factor f_4 to be added. All the candidate paths $\mathcal{P}^r$ that go through some vertex $v \in V_{inv}$ should be updated with the multi-robot factors included in v.

6.4 More than 2 Robots

The presented approach is not limited to 2 robots and naturally supports any number R of robots, with the objective function specified in Eq. (4.3). In this section we briefly specify the changes in each of the algorithmic steps to accommodate this general setting.

Section 6.1: Change in local information (Section 6.1), should be calculated with respect to all R robots, excluding current robot r. One can go further and also incorporate within $\Delta J^{r'}$ and $\Delta J^{r''}$ the impact of changed multi-robot factors between any two robots r' and r''. This direction is left to future research. *Section 6.2:* No modification is needed. *Section 6.3:* Algorithm 6.2 remains the same, however the input to the algorithm is now $J(\mathcal{P}^r, \{\mathcal{P}^{r'}\}_{r' \in \{1,\ldots,r-1,r+1,\ldots,R\}})$ instead of $J(\mathcal{P}^r, \mathcal{P}^{r'})$.


Candidate paths $\mathcal{P}^r$
Prev. announced path $\mathcal{P}^{r'}$
New announced path $\mathcal{P}^{r'}_{new}$
$p(X_k|Z_{0:k}, U_{0:k-1})$
Multi-robot pdf at planning time t_k
Robot r
Robot r'
Candidate paths $\mathcal{P}^r$
Prev. announced path $\mathcal{P}^{r'}$
New announced path $\mathcal{P}^{r'}_{new}$
Robot r
Robot r'
$p(X_k|Z_{0:k}, U_{0:k-1})$
Multi-robot pdf at planning time t_k

(a) (b)

Figure 6.2: (a) Illustration of the proposed concept without correlation as show before in Figure 1.1 compare with (b) Illustration of the proposed concept with correlation between robot states at planning time k (referred also as prior correlation in the text) due to mutual past landmark observations.

6.5 Prior Correlation

In this section we revisit the case where at planning time k, robot states are already correlated, e.g. due to observation of a common scene, see illustration in Figure 6.2. In other words Eq. (6.4) holds:

$$\mathbb{P}(X_k|\mathcal{H}_k) \neq \mathbb{P}(X_k^r|\mathcal{H}_k)\mathbb{P}(X_k^{r'}|\mathcal{H}_k). \tag{6.7}$$

As will be seen, our approach is applicable also in such a case with minor changes. Thus, the proposed approach also supports more realistic, unrolling scenarios where robot states become correlated at some point and the candidate paths can go through unknown or previously mapped areas, or a combination of both.

Prior correlation at planning time k can be expressed as

$$\mathbb{P}(x_k^r, x_k^{r'}|\mathcal{H}_k) = \int_{\neg x_k^r, x_k^{r'}} \mathbb{P}(X_k|\mathcal{H}_k) \doteq \mathcal{N}(\times, \Sigma(x_k^r, x_k^{r'})), \tag{6.8}$$

where × denotes some entry that is not of interest in the current context, and

$$\Sigma(x_k^r, x_k^{r'}) \doteq \begin{bmatrix} \Sigma_{x_k^r, x_k^r} & \Sigma_{x_k^r, x_k^{r'}} \\ \Sigma_{x_k^{r'}, x_k^r} & \Sigma_{x_k^{r'}, x_k^{r'}} \end{bmatrix}. \tag{6.9}$$

The correlation (or cross-covariance) term $\Sigma_{x_k^r, x_k^{r'}}$ will be non-zero because of Eq. (6.7).

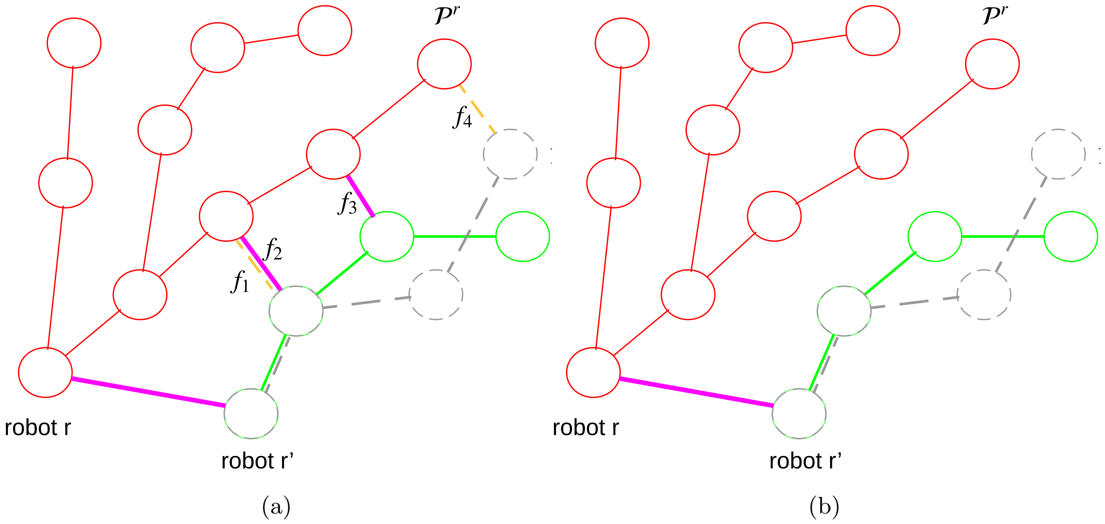


Figure 6.3: Prior correlation: (a) with and (b) without multi-robot factors.

Conceptually, marginalizing past robot poses and observed landmarks, the pdf (6.8) induces a multi-robot factor between the variables x_k^r and $x_k^{r'}$, as depicted in Figure 6.3.

Now, considering some candidate path $\mathcal{P}^r$ of robot r and previous and new announced paths $\mathcal{P}^r$ and $\mathcal{P}^{r'}_{new}$ from robot r', there are two possible cases (see Figure 6.3): (a) there are some multi-robot factors between $\mathcal{P}^r$ and $\mathcal{P}^{r'}$, and/or between $\mathcal{P}^r$ and $\mathcal{P}^{r'}_{new}$; or (b) there are no multi-robot factors between $\mathcal{P}^r$ and $\mathcal{P}^{r'}$ and also no multi-robot factors between $\mathcal{P}^r$ and $\mathcal{P}^{r'}_{new}$.

In the first case, prior correlation can be considered just as an additional multi-robot factor that is treated similarly to other multi-robot factors by our approach (see Chapter 6).

The second case (no multi-robot factors but with prior correlation) deserves further analysis. Figure 6.3b shows a diagram of such a scenario. Since there are no multi-robot factors, the posterior beliefs from Eqs. (6.1) and (6.2) turn into

$$b[\mathcal{P}^r,\mathcal{P}^{r'}] = \mathbb{P}(X_k|\mathcal{H}_k)\mathbb{P}(\mathcal{P}^r|U^r(\mathcal{P}^r),Z^r(\mathcal{P}^r))\mathbb{P}(\mathcal{P}^{r'}|U^{r'}(\mathcal{P}^{r'}),Z^{r'}(\mathcal{P}^{r'}))$$
$$b[\mathcal{P}^r,\mathcal{P}^{r'}_{new}] = \mathbb{P}(X_k|\mathcal{H}_k)\mathbb{P}(\mathcal{P}^r|U^r(\mathcal{P}^r),Z^r(\mathcal{P}^r))\underline{\mathbb{P}(\mathcal{P}^{r'}_{new}|U^{r'}(\mathcal{P}^{r'}_{new}),Z^{r'}(\mathcal{P}^{r'}_{new})),}$$

where the changed entries are underlined and shown in red color.

In the above case, if at planning time there was no prior correlation (i.e. x_k^r and $x_k^{r'}$ are not correlated), as considered in Section 6.2, then $\mathcal{P}^r$ would not be impacted by the change in the announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$. In other words (see also Eq. (6.3)):

$$b[\mathcal{P}^r] \doteq \int b[\mathcal{P}^r,\mathcal{P}^{r'}]d\mathcal{P}^{r'} \equiv b'[\mathcal{P}^r] \doteq \int b[\mathcal{P}^r,\mathcal{P}^{r'}_{new}]d\mathcal{P}^{r'}_{new}, \tag{6.10}$$

and thus, $\mathcal{P}^r$ would not be marked. However, this does not hold in general in the presence of prior correlation.

Interestingly, however, despite having prior correlation, we observe that as long as the paths $\mathcal{P}^r$, $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ go through unknown areas with no sources of absolute information (such as GPS or known landmarks), any such candidate path $\mathcal{P}^r$ of robot r is *not* impacted due to change in the announced path (from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$) and thus should *not* be marked, thereby saving calculations.

We now illustrate this observation in a simple example, and then discuss in Section 6.7 the more general case, where the above conditions are not met, and discuss a slight modification to our approach.

6.6 Simple Example

We consider a simple example where paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ only include a single look ahead step. Both paths pass through unknown areas and thus we assume existence only of visual odometry measurements z^{VO} that provide relative information between consecutive states. The posterior over $x^{r'}_k$ and $x^{r'}_{k+1}$, given z^{VO} from either $\mathcal{P}^{r'}$ or $\mathcal{P}^{r'}_{new}$ is

$$\mathbb{P}(x^{r'}_k, x^{r'}_{k+1}|\mathcal{H}_k, z^{VO}) \propto \mathbb{P}(x^{r'}_k|\mathcal{H}_k)\mathbb{P}(z^{VO}|x^{r'}_k, x^{r'}_{k+1}), \tag{6.11}$$

where $\mathbb{P}(x^{r'}_k|\mathcal{H}_k) = \mathcal{N}(\times, \Sigma_k)$ describes the posterior over $x^{r'}_k$ at planning time k, which could be obtained e.g. via $\mathbb{P}(x^{r'}_k|\mathcal{H}_k) = \int_{x^r_k} \mathbb{P}(x^r_k, x^{r'}_k|\mathcal{H}_k)$.

We now show the posterior over $x^{r'}_k$ is not influenced by the new information (measurement z^{VO}), e.g. the covariance does not change. Performing standard maximum a posteriori (MAP) inference yields the following least-squares expression:

$$x^{r'\star}_k, x^{r'\star}_{k+1} = \underset{x^{r'}_k, x^{r'}_{k+1}}{\arg\min} \|x^{r'}_k - \hat{x}^{r'}_k\|^2_{\Sigma_k} + \|z^{VO} - h^{VO}(x^{r'}_k, x^{r'}_{k+1})\|^2_{\Sigma_{VO}}, \tag{6.12}$$

where h^{VO} and Σ_{VO} are the corresponding measurement function and measurement noise covariance for visual odometry (see e.g. [22]). Linearizing and augmenting the Jacobians we get

$$\Delta x^{r'\star}_k, \Delta x^{r'\star}_{k+1} = \underset{\Delta x^{r'}_k, \Delta x^{r'}_{k+1}}{\arg\min} \|A \begin{pmatrix} \Delta x^{r'}_k \\ \Delta x^{r'}_{k+1} \end{pmatrix} - b\|^2, \tag{6.13}$$

where b is an appropriate right hand side (rhs) vector and

$$A \doteq \begin{bmatrix} \Sigma_k^{-1/2} & 0 \\ \Sigma_{VO}^{-1/2} H_{VO} & -\Sigma_{VO}^{-1/2} \end{bmatrix}. \tag{6.14}$$

The posterior covariance over $x^{r'}_k$ and $x^{r'}_{k+1}$ is $\left(A^T A\right)^{-1}$, from which we will now extract the entry that corresponds to $x^{r'}_k$ and show it is equal to Σ_k, despite the new information (measurement z^{VO}).

First, the information matrix is calculated as

$$A^TA = \begin{bmatrix} \Sigma_k^{-1} + H_{VO}^T\Sigma_{VO}^{-1}H_{VO} & H_{VO}^T\Sigma_{VO}^{-1} \\ \Sigma_{VO}^{-1}F & \Sigma_{VO}^{-1} \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix}. \tag{6.15}$$

The covariance entry that corresponds to $x_k^{r'}$ is the top left block matrix of $\left(A^TA\right)^{-1}$. Using block matrix inversion this entry can be calculated as

$$\left(A^TA\right)^{-1} = \begin{bmatrix} \left(\mathcal{A} - \mathcal{B}\mathcal{D}^{-1}\mathcal{C}\right)^{-1} & \times \\ \times & \times \end{bmatrix} \tag{6.16}$$

Substituting matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ from Eq. (6.15) and performing basic algebraic manipulation we get

$$\left(\mathcal{A} - \mathcal{B}\mathcal{D}^{-1}\mathcal{C}\right)^{-1} = \left(\Sigma_k^{-1} + F^T\Sigma_w^{-1}F - F^T\Sigma_w^{-1}\Sigma_\omega^1\Sigma_w^{-1}F\right)^{-1} = \Sigma_k, \tag{6.17}$$

as claimed. In other words, adding new relative information does not impact the state $x_k^{r'}$. Hence, it does not matter whether this new information is added due to path $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ - in both cases, the state x_k^r of robot r is not impacted despite the existence of prior correlation between x_k^r and $x_k^{r'}$. This means, in turn, that all candidate paths $\mathcal{P}^r$ of robot r that do not have multi-robot factors with $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$, can remain unmarked and should not be recalculated.

This concludes the simple example; we now proceed to discuss a more general case, where the covariance over $x_k^{r'}$ does change as a result of incorporating new information along a candidate path, and we outline a slight modification of our algorithm to also handle this case.

6.7 A More General Case

When the conditions mentioned toward the end of Section 6.5 are not met, e.g. at least one of the paths $\mathcal{P}^r$, $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ go through previously mapped areas, or when along $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ there are a priori known landmarks or available GPS signal, then Eq. (6.10) does not necessarily hold. Intuitively, a substantial update along $\mathcal{P}^{r'}$ (or $\mathcal{P}_{new}^{r'}$), e.g. due to GPS measurement, will impact the posterior over $x_k^{r'}$:

$$\mathbb{P}(x_k^{r'}|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \neq \mathbb{P}(x_k^{r'}|\mathcal{H}_k). \tag{6.18}$$

Due to prior correlation, that couples x_k^r with $x_k^{r'}$, the new information will also pass, to some degree, onward to robot r, impacting the posterior over x_k^r. If the information along the previous and new announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$ is substantially different, then the impact on the posterior of x_k^r can also be different, i.e.:

$$\begin{aligned}
\mathbb{P}(x_k^r|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) &= \int_{x_k^{r'}} \mathbb{P}(x_k^r, x_k^{r'}|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) && (6.19)\\
&\doteq \mathcal{N}(\times, \Sigma_k(U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) && (6.20)\\
\mathbb{P}(x_k^r|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})) &= \int_{x_k^{r'}} \mathbb{P}(x_k^r, x_k^{r'}|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})) && (6.21)\\
&\doteq \mathcal{N}(\times, \Sigma_k(U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})) && (6.22)
\end{aligned}$$

and

$$\mathbb{P}(x_k^r|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \neq \mathbb{P}(x_k^r|\mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})). \qquad (6.23)$$

Hence, the posteriors over candidate path $\mathcal{P}^r$, $b[\mathcal{P}^r]$ and $b'[\mathcal{P}^r]$ will change (Eq. (6.10) will not hold). It would thus seem that $\mathcal{P}^r$ should be necessarily marked, to trigger belief evolution recalculation.

However, it is often the case that while the posteriors (6.19) and (6.21), and therefore $b[\mathcal{P}^r]$ and $b'[\mathcal{P}^r]$, are not identical, in practice the difference is small and can be considered negligible given some threshold. In such a case, there is no need in recalculating belief evolution along path $\mathcal{P}^r$, and thus the latter should *not* be marked.

Based on the above observation, we propose the following slight modification to our approach. First, we evaluate the posteriors (6.19) and (6.21) - this is a one-time calculation for given previous and new announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$, which is valid to *all* candidate paths $\mathcal{P}^r$ of robot r. Then, we decide if the two posteriors are sufficiently similar given a user-defined threshold th: different information-theoretic costs can be used for this purpose (e.g. KL-divergence and relative entropy). A simple alternative, for example, is to calculate the difference in the determinant (or trace) of the posterior covariance in each case. More specifically, recalling Eqs. (6.20) and (6.22), the candidate path $\mathcal{P}^r$ is marked only if

$$\det\left(\Sigma_k(U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})\right) - \det\left(\Sigma_k(U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})\right) > th. \qquad (6.24)$$

In our current implementation we use the above criteria with the threshold th set to 10.

Chapter 7

Results

We demonstrate our approach in simulation considering scenario involving two and four robots operating in Largeunknown and GPS-deprived environments that need to navigate to different goals in minimum time but also with highest accuracy. In this basic evaluation we use a prototype implementation in Matlab and GTSAM [14] to investigate key aspects of the proposed approach. The objective function (4.3) is $J = \sum_{r=1}^{R} \left[\kappa_{goal}^{r} t_{goal}^{r} + \kappa_{\Sigma}^{r} tr \left(\Sigma_{goal}^{r} \right) \right]$, where Σ_{goal}^{r} and t_{goal}^{r} represent, respectively, the covariance upon reaching the goal and time of travel (or path length) for robot r. The parameters κ_{goal}^{r} and κ_{Σ}^{r} weight the importance of each term (we use $\kappa_{path}^{r} = 0.1$ and $\kappa_{uncert}^{r} = 1$). A probabilistic roadmap (PRM) [26] is used, to discretize the (partially unknown) environment and generate candidate paths over the roadmap. Figures 7.1 and 7.2 show the considered scenarios for two and four robots and the generated 25 candidate paths for each robot. In this and all figures to follow, we use the notation $\star$ to indicate the starting position of each robot.

We compare our approach to a standard approach that re-evaluates from scratch belief evolution and objective function for *each* candidate path of each robot r given announced paths from other robots (e.g. [8, 18, 31]). This comparison has two merits: (a) verify our approach correctly recovers the underlying pdf while identifying and re-evaluating only the impacted paths; and (b) has computational benefits.

7.1 Basic Scenarios

Figure 7.5 shows, for the two-robot scenario, one of the candidate paths of robot r, an announced path of robot r', and the generated multi-robot factors (cyan color); see also concept illustration in Figure 1.1. The corresponding belief evolution (covariance ellipses) is displayed in black. Robot r determines its best path, and announces it to other robots, which do the same; the process is repeated until convergence. Similar to [18], we use a simple heuristic for the function $\mathbf{cr_{MR}}(v_i, v_j)$ (line 4 of Alg. 6.1) to determine if two poses admit a multi-robot constraint: these constraints, possibly involving different future time instances, are formulated between any two poses with relative distance closer

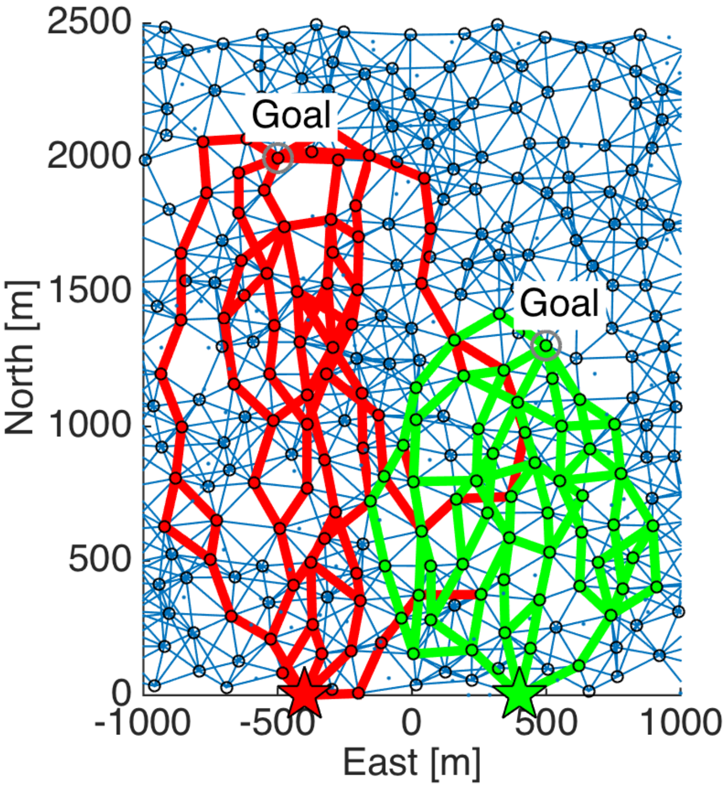


Figure 7.1: Candidate paths shown on PRM. Robot starting positions are denoted by $\star$.

than $d = 300$ meters. More advanced methods could be implemented, e.g. considering also statistical knowledge.

The set of involved vertices in PRM, V_{inv}, depicted conceptually in Figure 6.1, is shown for robot r in Figure 7.6 for the two-robot scenario. The figure shows marked (impacted) candidate paths of robot r, as a result of an update in the announced path of robot r' from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$ in one of the iterations. To reduce clutter, only the impacted (marked) candidate paths of robot r are shown. The corresponding multi-robot factors are color-coded: cyan indicates unchanged multi-robot factors (associated with both $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$), and yellow and magenta indicate multi-robot factors that are associated, respectively, only with $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$. These factors are appropriately then included with in the corresponding vertices in V_{inv} and are used for calculating belief evolution, following Algorithms 6.1 and 6.2.

In the specific situation shown in Figure 7.6, only some of the candidate paths are impacted. Our approach correctly identifies, marks and consequently re-evaluates the belief over only these impacted paths. This is in contrast to the Standard approach that re-evaluates the belief from scratch over all candidate paths and recalculates the objective function for each. As a consequence, our approach exhibits substantially reduced running time, compared to the Standard approach, while producing identical results.

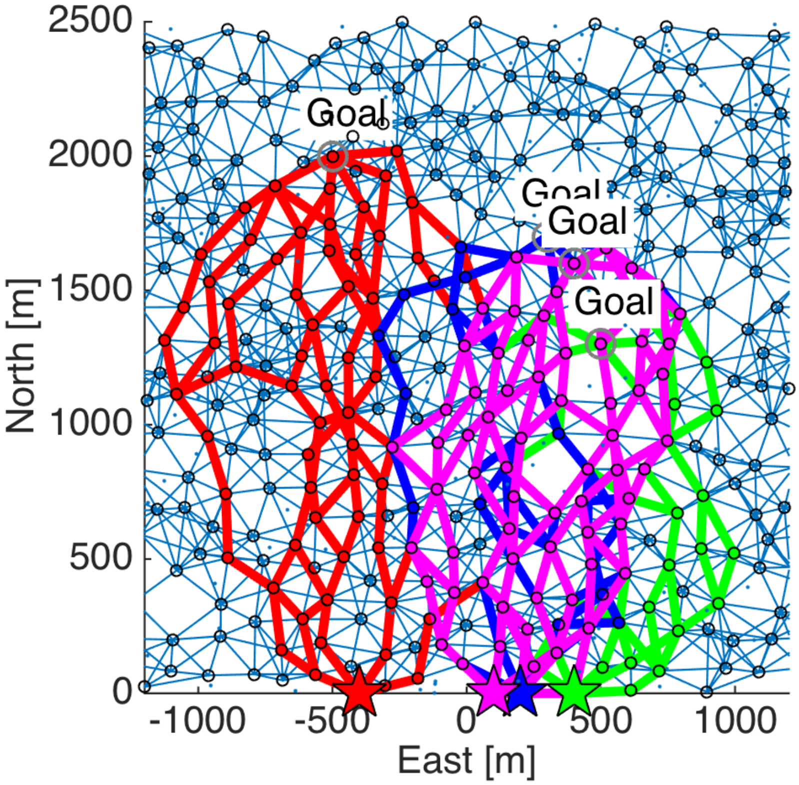


Figure 7.2: Candidate paths shown on PRM. Robot starting positions are denoted by ⋆.

Figures 7.3 and 7.4 reports statistical timing results as a function of number of candidate paths N_{cand} for each robot, considering the two-robot and four-robot scenarios from Figures 7.1 and 7.2. These results were obtained by running each approach 50 times, for each considered N_{cand}. In each such run, the scenario remains the same (goals, starting locations), while the candidate paths randomly change. As seen, as N_{cand} increases the ratio between running time of the two approaches increases, in favor of our approach. In particular, for 50 candidates and two robots, our approach is 2.5 times faster compared to the standard approach (35 versus 85 seconds); A similar trend can be seen also for four robots. In all cases, identical results were obtained, compared with the Standard approach.

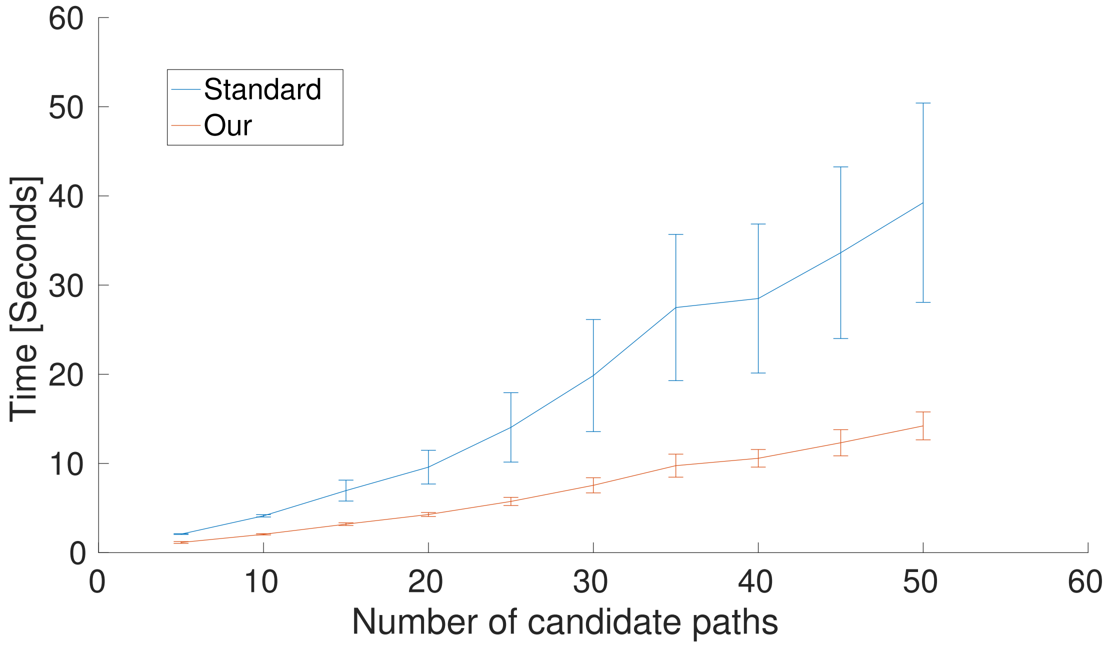


Figure 7.3: (Statistics for running time as a function of number of candidate paths for each robot, considering groups of 2 robots.

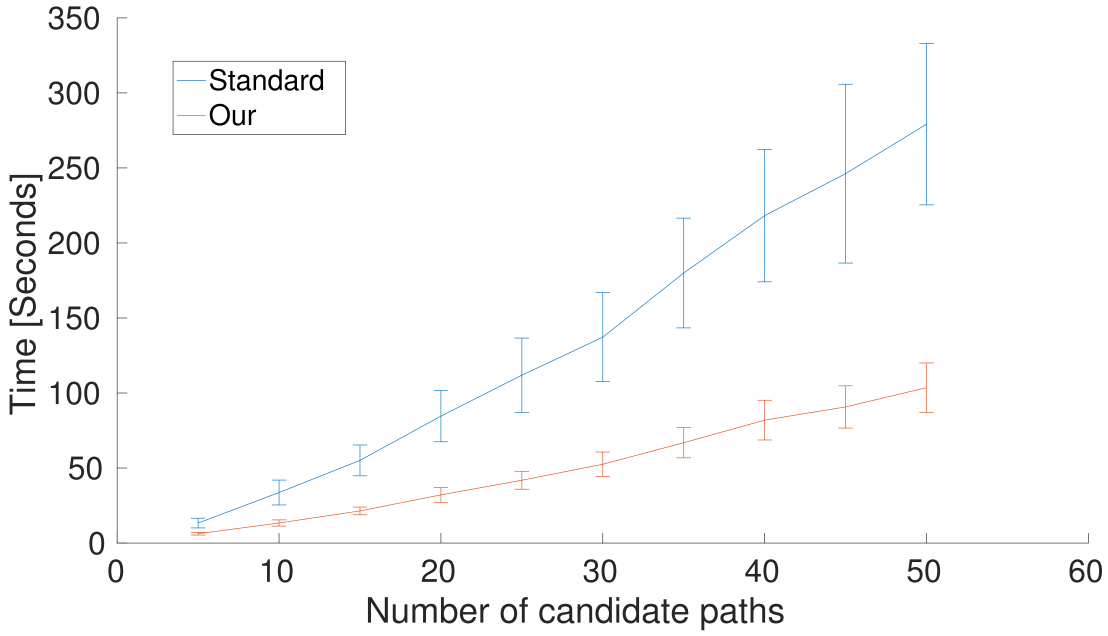


Figure 7.4: Statistics for running time as a function of number of candidate paths for each robot, considering groups of 4 robots.

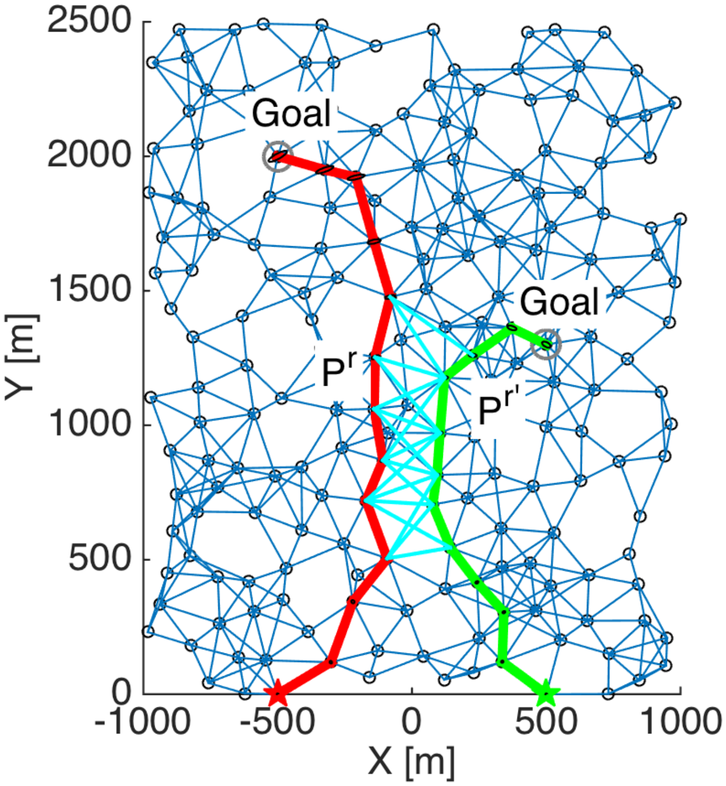
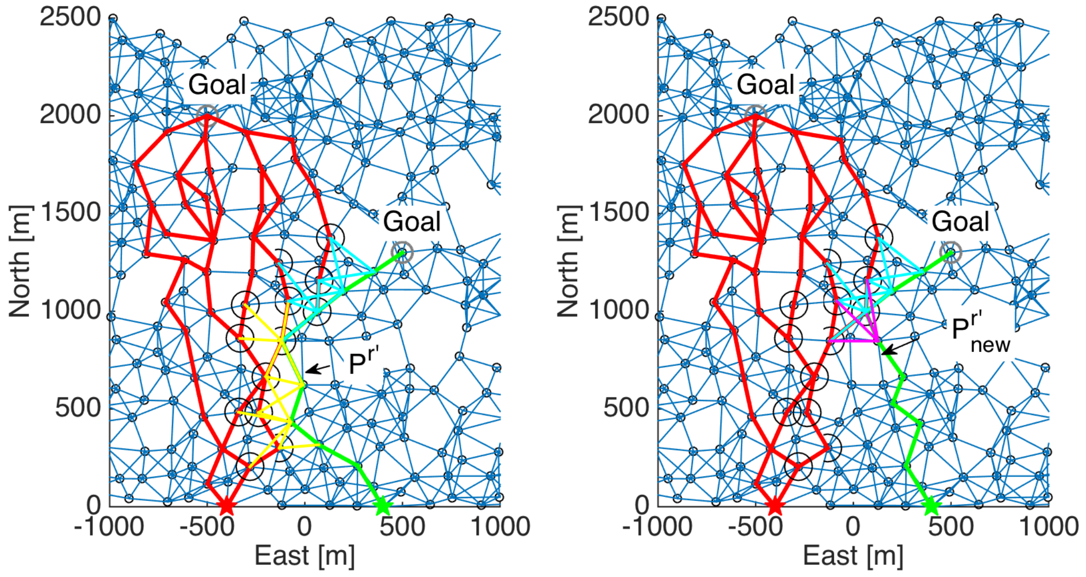


Figure 7.5: Multi-robot factors (cyan color) and belief evolution (covariance ellipses) for one of the candidate paths from Figure 7.1, considering an announced path $\mathcal{P}^{r'}$.



(a)

(b)

Figure 7.6: Illustration of the proposed approach considering a group of two robots (see Figure 6.1). Vertices in V_{inv} for robot r given a (a) previous and (b) new announced path of robot r' are shown as circles. Unchanged multi-robot factors are shown in cyan. Changed multi-robot factors associated with $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ are shown in yellow and magenta, respectively. Only impacted candidate paths of robot r are shown.

7.2 Large Scale Scenarios

We also examine our approach in a larger scenario, where each robot has to reach multiple pre-defined goals while operating in unknown environments. Such a scenario involves multiple planning sessions and multi-robot SLAM - reaching each goal triggers a new planning session during which the robots update theirs best paths. These paths are then translated into commands, in our case, the change in heading angle. In our simulative framework, the robots execute these commands and acquire new bearing and range observations of landmarks. Note that the latter can be either previously seen landmarks, that correspond to already mapped areas, and new landmarks. Considering perfect association of the landmark observations, the robots then calculate a multi-robot SLAM solution, i.e. the term $\mathbb{P}(X_k|\mathcal{H}_k)$ in Eq. (5.5).

Figures 7.7-7.14 show the results of each of the planning and SLAM sessions, while the running time is reported in Figure 7.15. Goals are indicated in these figures using both numbers and colors, with the former denoting sequence (i.e. goal 1 should be visited before goal 2), and colors indicating different robots. As seen, four robots are considered (red, green, blue and purple), and each robot has a sequence of three goals. We intentionally scattered the goals in such a way that both planning in unknown and previously-mapped environments is examined.

We show, for each planning session, the candidate paths for all robots and the best paths identified by the proposed approach, see Figures 7.7, 7.9, 7.11 and 7.13. Multi-robot factors involving future poses of different robots (along the chosen paths), and factors involving a future pose of robot $r \in [1, \ldots, R]$ and a landmark, previously observed by robot r or by any other robot in the group, are indicated in cyan color. See, e.g. Figure 7.9b for combination of both of these factors. As in the basic study (Section 7.1), covariances along the chosen paths are also shown.

At the first planning session (Figure 7.7), the robot start operating with only prior information on their initial poses (we use 1e$-$6 $[m]$, meaning robots know their exact start locations) - in other words, there is no correlation between the robot states. Using the proposed approach, the best path for each robot in the group is determined and executed until one of the robots reaches a goal. In particular, the chosen paths of the red and green robots admit a single multi-robot factor within planning. Figure 7.8a shows the corresponding SLAM solution, while Figure 7.8b shows position covariance evolution (from SLAM). While not explicitly shown, the states of red and green robots, and of blue and purple robots become correlated towards the end of this phase due to mutual landmark observations.

From this moment onward, thus, the states of these robots are (somewhat) correlated and the discussion from Section 6.5 regarding prior correlation becomes relevant. In the second planning session (Figures 7.9 and 7.10), the goals are scattered such that vast majority of the candidate paths still go through unknown areas (see Figure 7.9a). Looking at the determined best paths (Figure 7.9b), one can observe the planned multi-robot collaboration between two robot pairs (red-green and blue-purple), which is exhibited either in terms of multi-robot factors or observations of landmarks previously observed by another robot.

Despite prior correlation, however, our approach is capable of significantly reducing running time (by a factor of two, see Figure 7.15) while yielding the same results in terms of the chosen paths. This goes in hand with the observation from Section 6.5 that the belief along path of any robot r is not impacted by change in the announced path of other robots if these paths go through unknown areas and without sources of absolute information, which is the case here (recall also the example from Section 6.6).

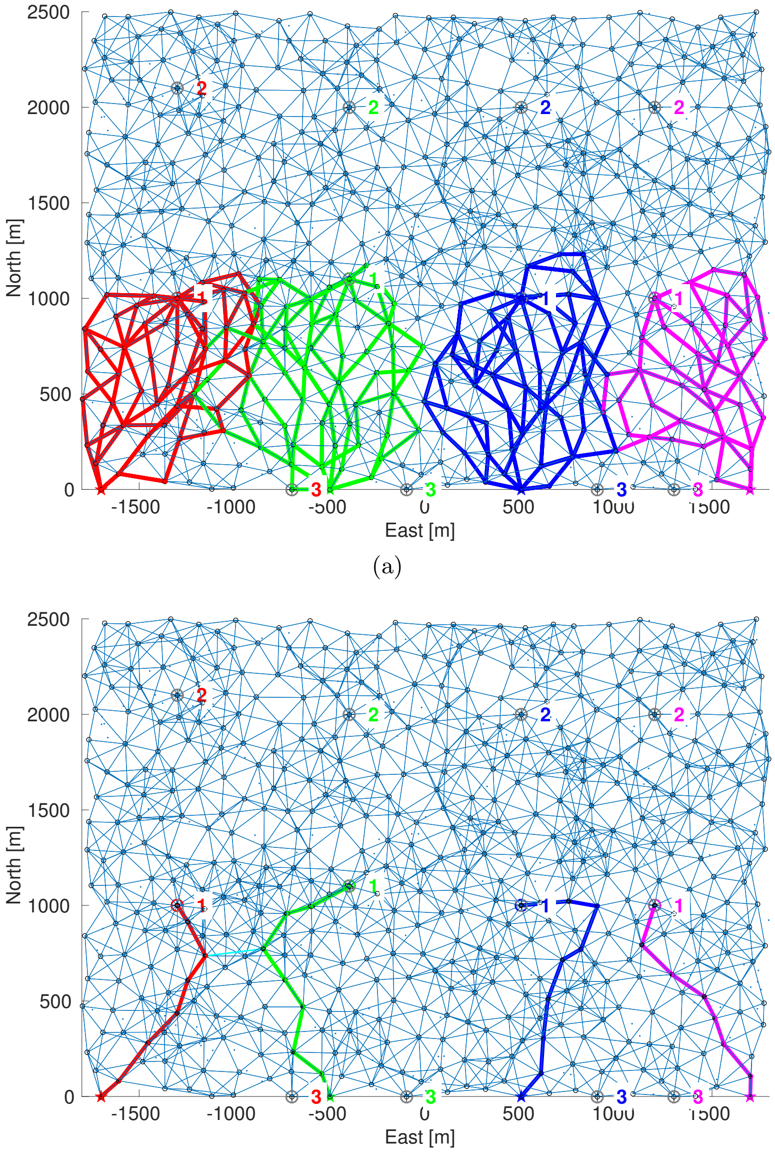
In the third planning session, the red robot still has not reached its second goal, while all the other robots already consider their next goals. After the red robot reaches its second goal, another planning session is triggered. We note that in practice, only the red robot could actually generate new candidate paths while the rest of the robots could remain with candidate paths from the previous planning session.

The third goal of each robot was intentionally chosen to force the robots to re-visit previously mapped environments (see e.g. Figures 7.13 and 7.14). As in the previous planning sessions, the states of the two robot pairs red-green and blue-purple are correlated. However, here, in addition the robots consider impact of loop closure observations within planning. These are often mutual multi-robot observations, i.e. the same previously-observed landmarks are planned to be observed by multiple robots - see the cyan lines in Figure 7.13b. Given the corresponding best paths, which were determined as such (mainly) due to these multi-robot constraints that allow significant uncertainty reduction, a multi-robot SLAM session is performed. As evident from Figure 7.14, the robots indeed reach the goals with small uncertainty, which roughly corresponds to the prior uncertainty (due to loop closures).

Finally, Figure 7.15 depicts running time for each of the planning sessions, comparing the proposed approach with the Standard approach (that does not attempt to re-use calculations). It can be clearly seen that our approach is substantially faster in all planning sessions. In particular, it is faster by a factor of two and seven in the second and third planning sessions, respectively. We emphasize this significant reduction in running time comes with no sacrifice in performance, i.e. the same paths were chosen by our and Standard approach in all planning sessions.

We also can see the benefit of MR factors compared with `Scenario3` without MR factors. The peak of the covariance in this scenario is much lower as figures 7.14b and 7.16b show.

In addition to the above-described scenario, we examined our approach in two other related scenarios. Table 7.1 provides a description of these scenarios, while the results are given in Appendix A, to avoid clutter.



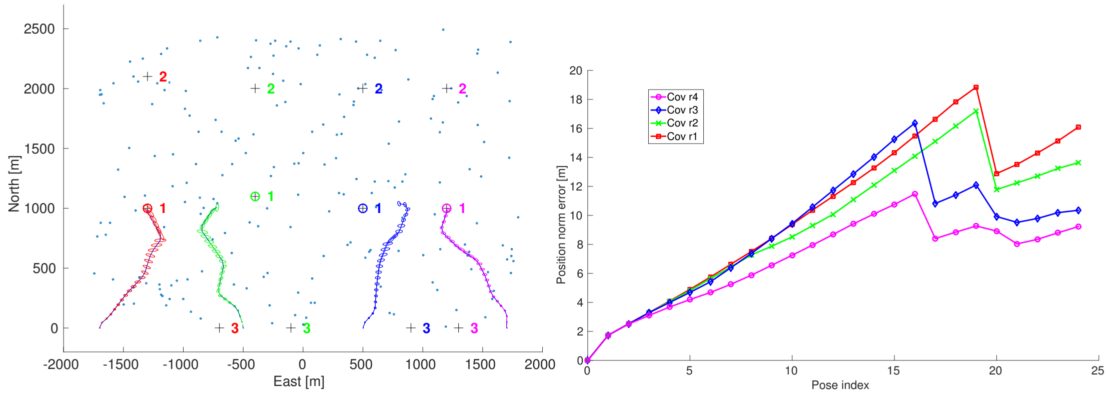
(a)

(b)

Figure 7.7: `Scenario1`. First planning session. States of different robots are not correlated. (a) Candidate paths to the first goal of each robot; (b) Chosen paths by the planning approach.

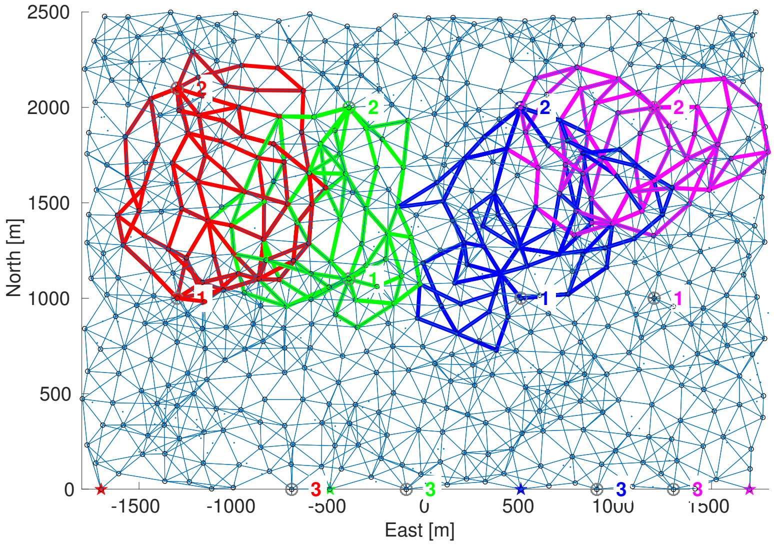
Scenario	Description	Big covariance	MR factors
`Scenario1`	All robots have small and identical covariances at the beginning. MR factors are used within planning.	no	yes
`Scenario2`	Red and green robots have large uncertainty covariances at the beginning. All robots use MR factors within planning.	yes	yes
`Scenario3`	All robots have small and identical covariances at the beginning. MR factors are *not* used within planning.	no	no

Table 7.1: Scenarios description

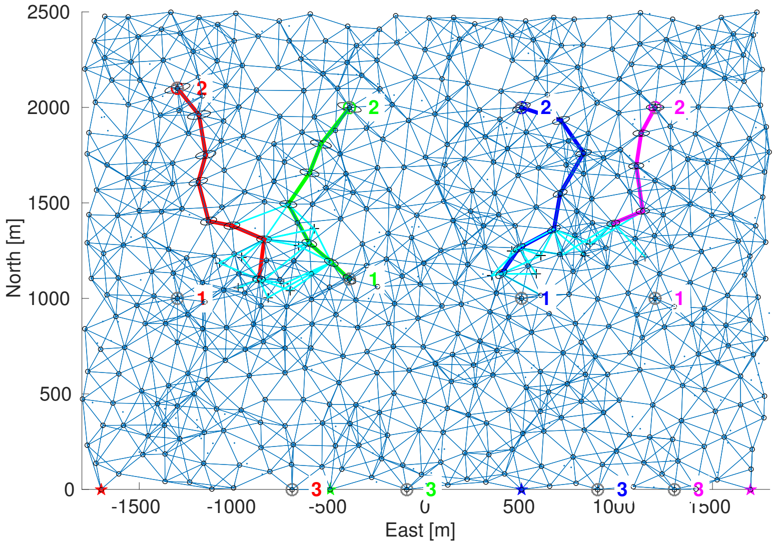


(a) (b)

Figure 7.8: `Scenario1`. (a) Multi-robot SLAM given paths determined in the first planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution.

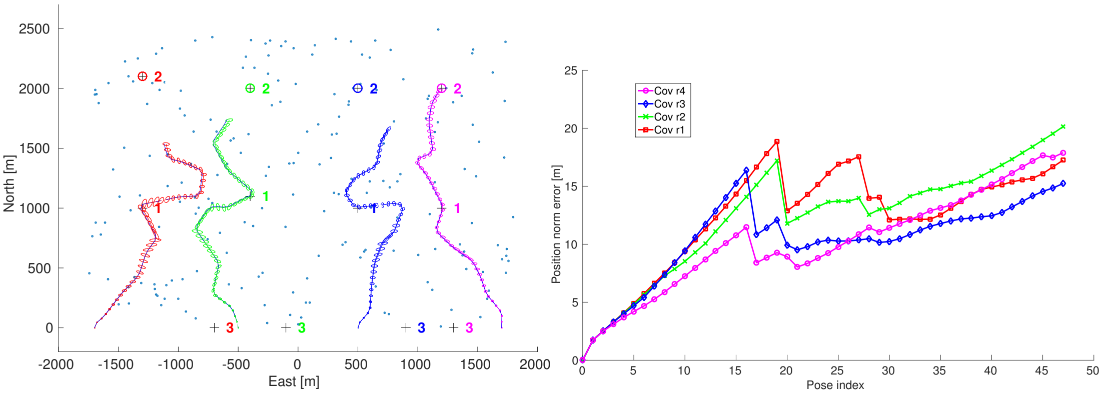


(a)



(b)

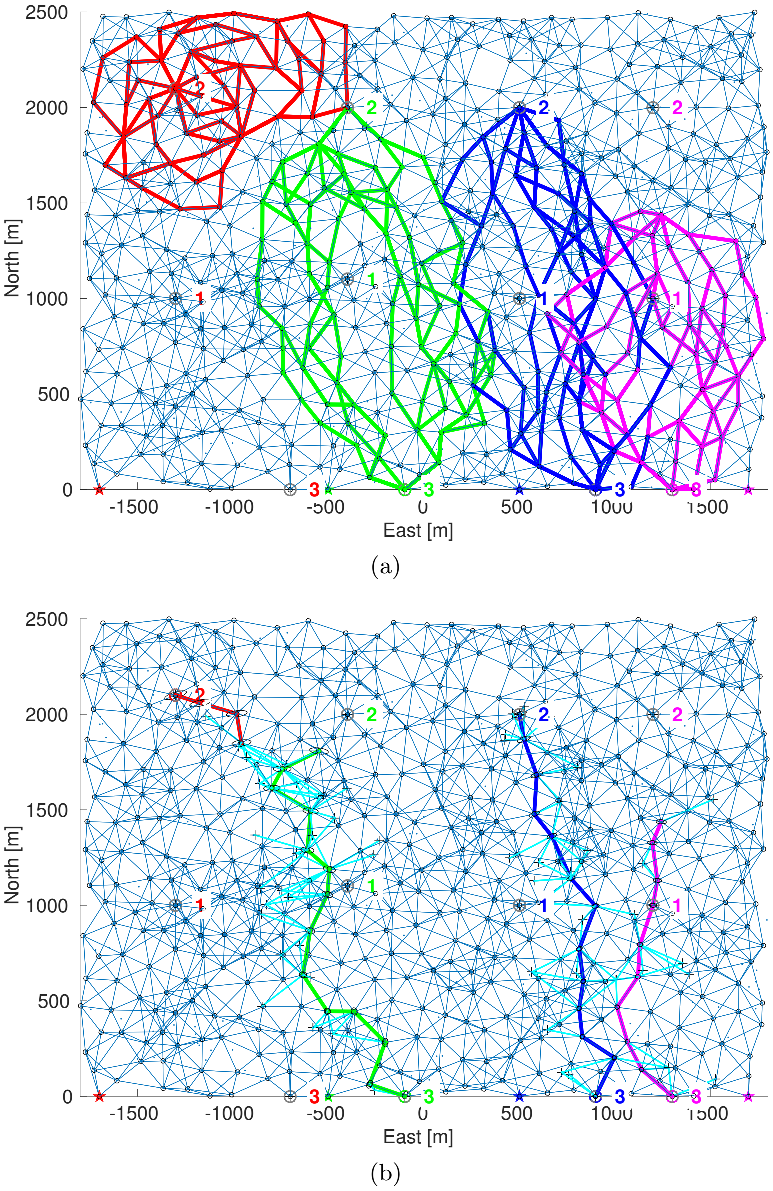
Figure 7.9: `Scenario1`. Second planning session. (a) Candidate paths to the second goal of each robot; (b) Chosen paths by the planning approach.



(a)

(b)

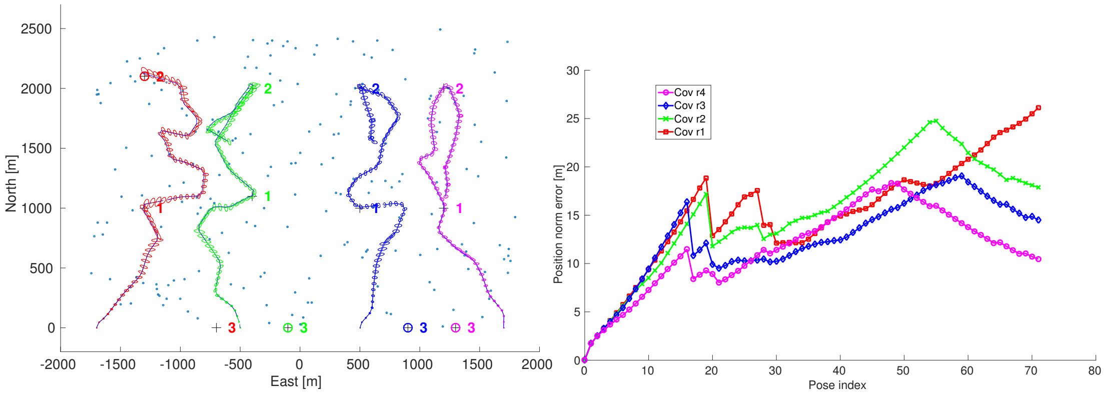
Figure 7.10: `Scenario1`. (a) Multi-robot SLAM given paths determined in the second planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution.



(a)

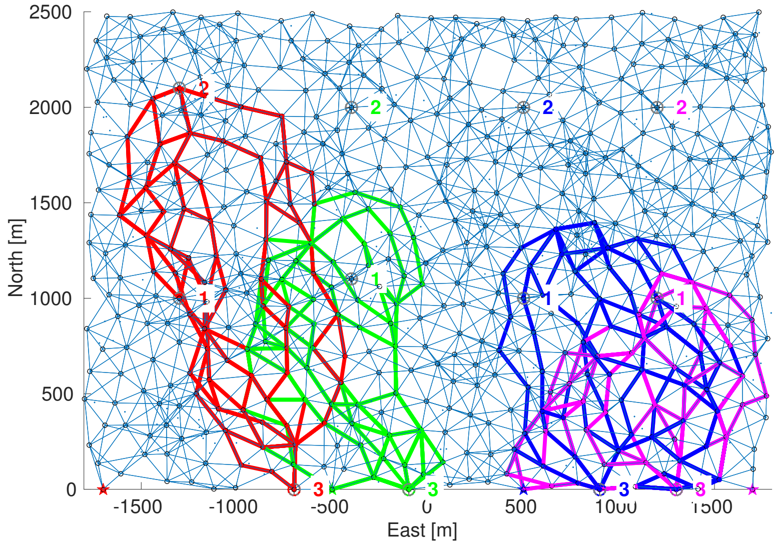
(b)

Figure 7.11: `Scenario1`. Third planning session. (a) Candidate paths to the second goal for the red robot, and to the third goal of each other robot; (b) Chosen paths by the planning approach.

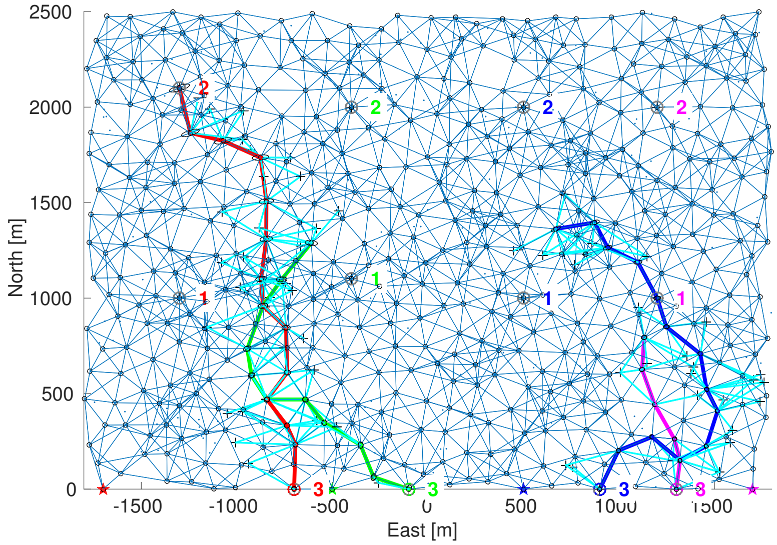


(a) (b)

Figure 7.12: `Scenario1`. (a) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution.

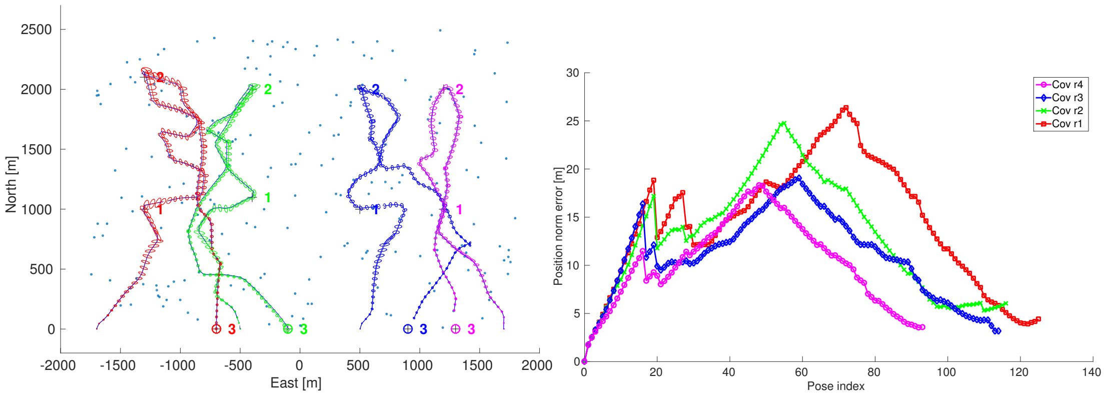


(a)



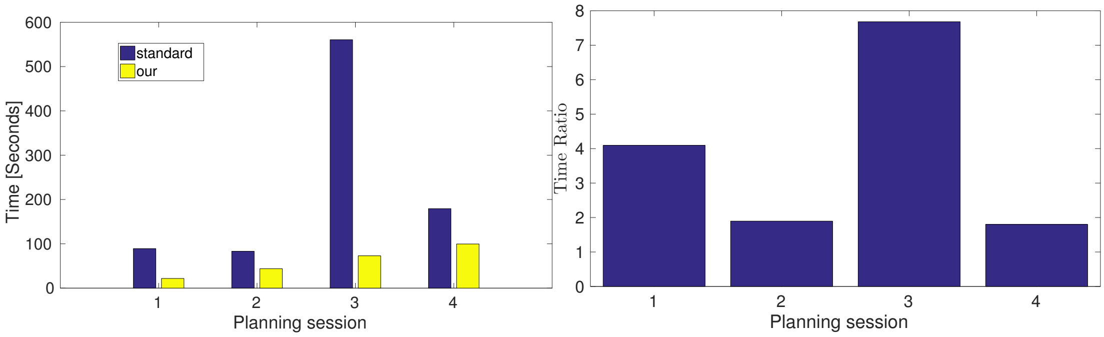
(b)

Figure 7.13: Scenario1. Fourth planning session. (a) Candidate paths to the third goal of each robot; (b) Chosen paths by the planning approach.



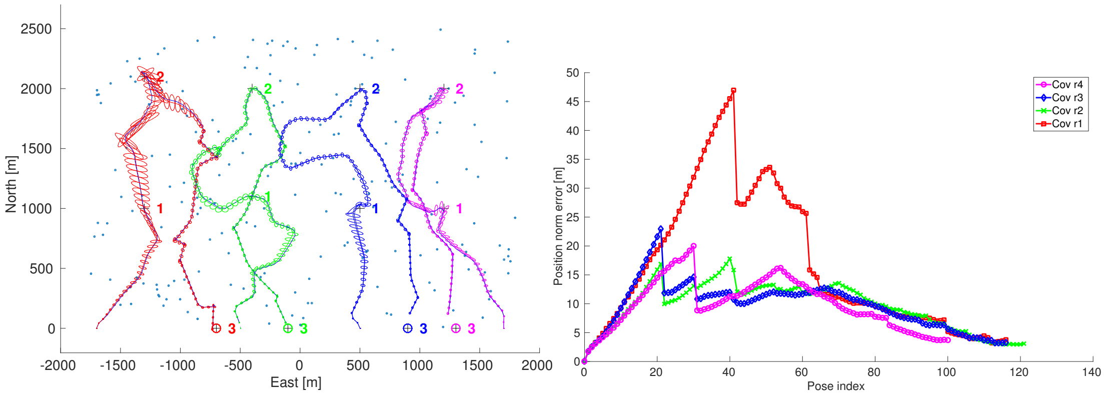
(a) (b)

Figure 7.14: `Scenario1`. (a) Multi-robot SLAM given paths determined in the fourth planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution.



(a) (b)

Figure 7.15: `Scenario1`. Running time comparison between the proposed and the Standard approach. (a) running time for each planning session. (b) Time ratio.



(a) (b)

Figure 7.16: `Scenario3`. (a) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (b) Corresponding position covariance evolution.

Chapter 8

Conclusion

We addressed the problem of decentralized belief space planning over high-dimensional state spaces while operating in unknown environments. Since exact solution is computationally intractable, a common approach is to address this problem within a sampling based motion planning paradigm, where each robot repeatedly considers its own candidate paths given the best paths (announced paths) transmitted by other robots. The process is typically repeated numerous times by each robot either until convergence or on a constant basis, with each time involving belief propagation along *all* candidate paths. In this thesis we developed an approach that identifies and efficiently re-evaluates the belief over *only* those candidate paths that are impacted upon an update in the announced path transmitted by another robot. Determining the best path can therefore be performed without re-evaluating the utility function for each candidate path from scratch. We demonstrated in simulation our approach is capable of correctly identifying and calculating belief evolution over impacted paths, and significantly reduces computation time without any degradation in performance. In future work, we are planning to evaluate the developed approach in real world and realistic synthetic experiments.

Appendix A

Additional Large Scale Scenarios

In addition to the large scale scenario from Section 7.2, we examined our approach in two other related scenarios. Table 7.1 provides a description of these scenarios.

A.1 Scenario 2

In `Scenario2`, red and green robots have large uncertainty covariances at the beginning, while the other two robots (blue and purple) have a small initial covariance as in `Scenario1`. In this scenario all robots use MR factors within planning. Figures A.1-A.4 provide the results for each planning session, while Figure A.5 shows running time. As seen, the large uncertainties of the red and green robots are reduced due to mutual observations with the blue robot, observations that were planned by the proposed approach. As earlier, running time of the latter is significantly smaller than of the Standard approach, while in both cases the same results are obtained.

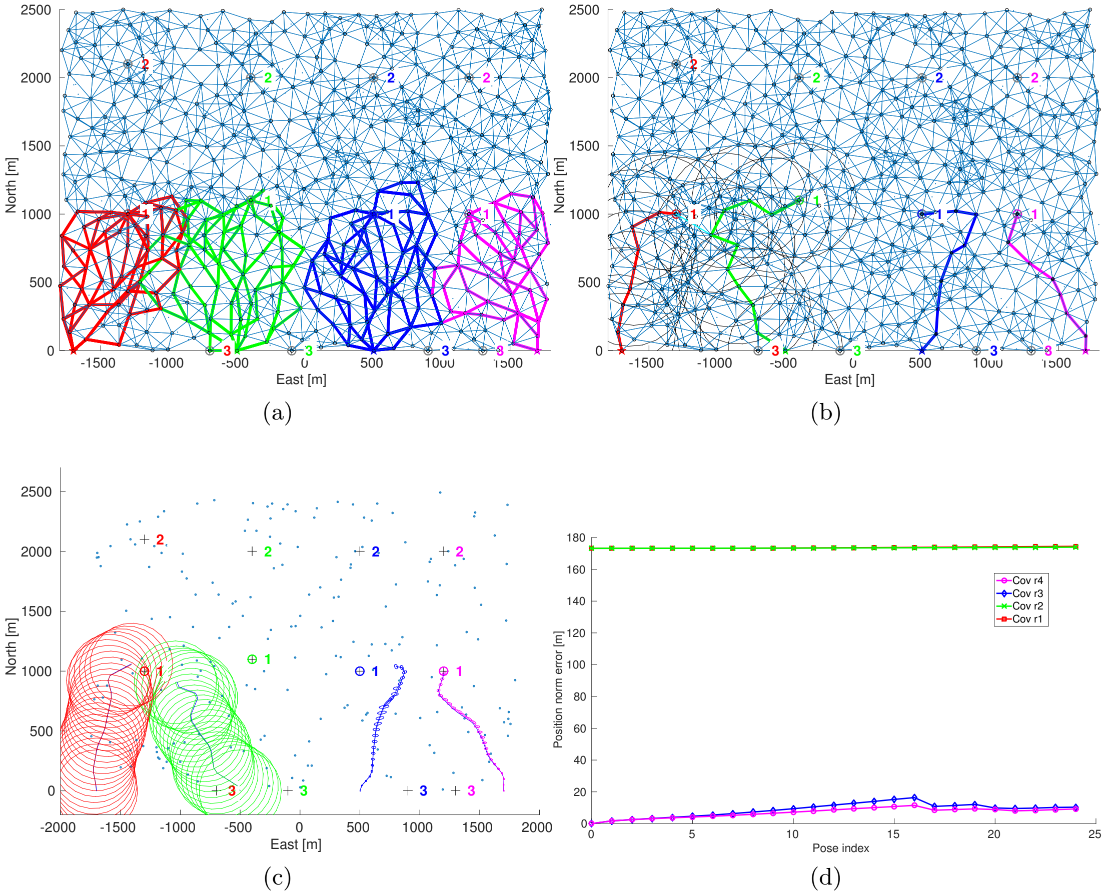


Figure A.1: `Scenario2`. First planning session. States of different robots are not correlated. (a) Candidate paths to the first goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the first planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.

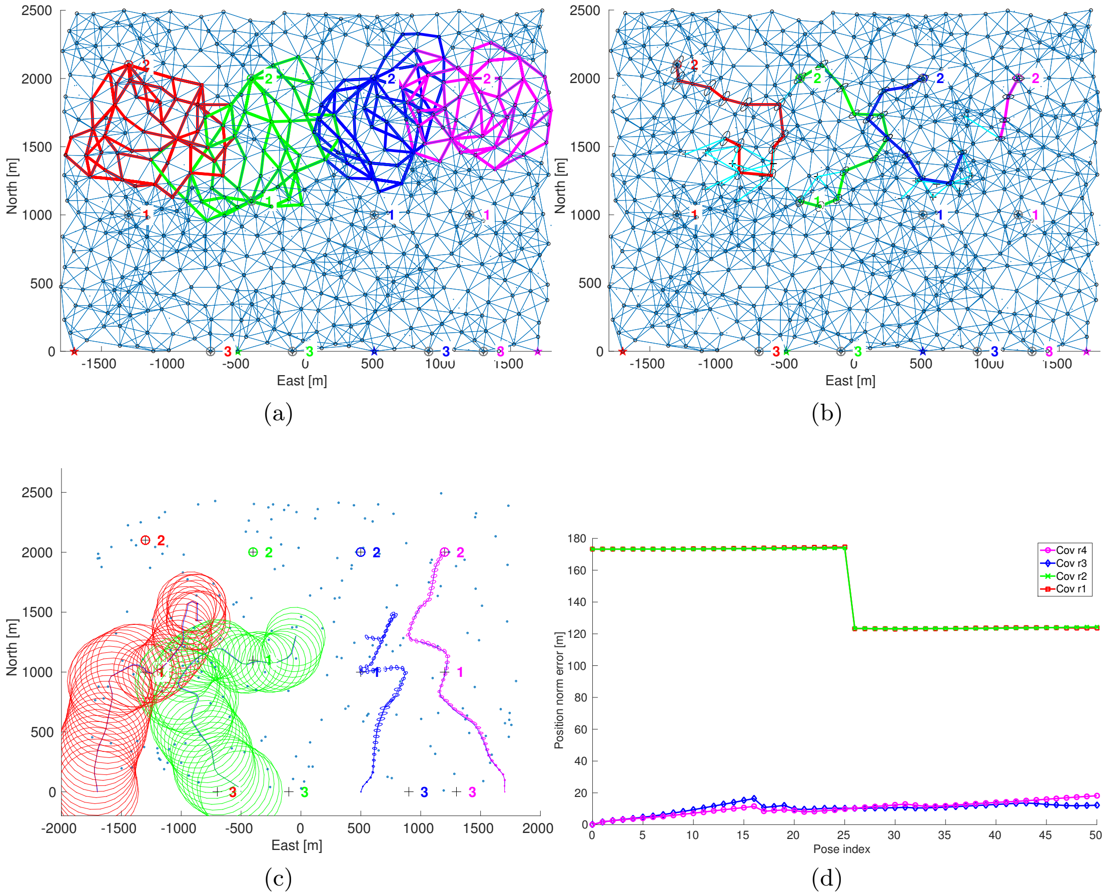


Figure A.2: `Scenario2`. Second planning session. (a) Candidate paths to the second goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the second planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.

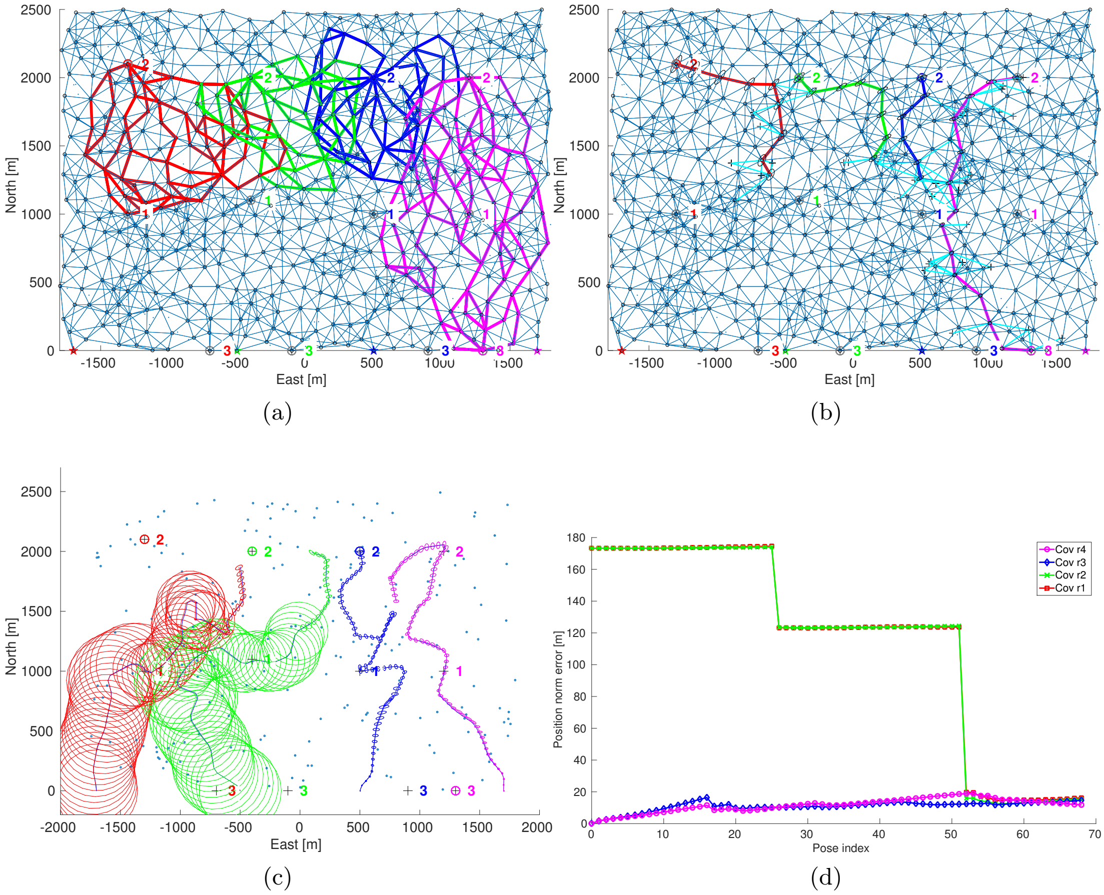
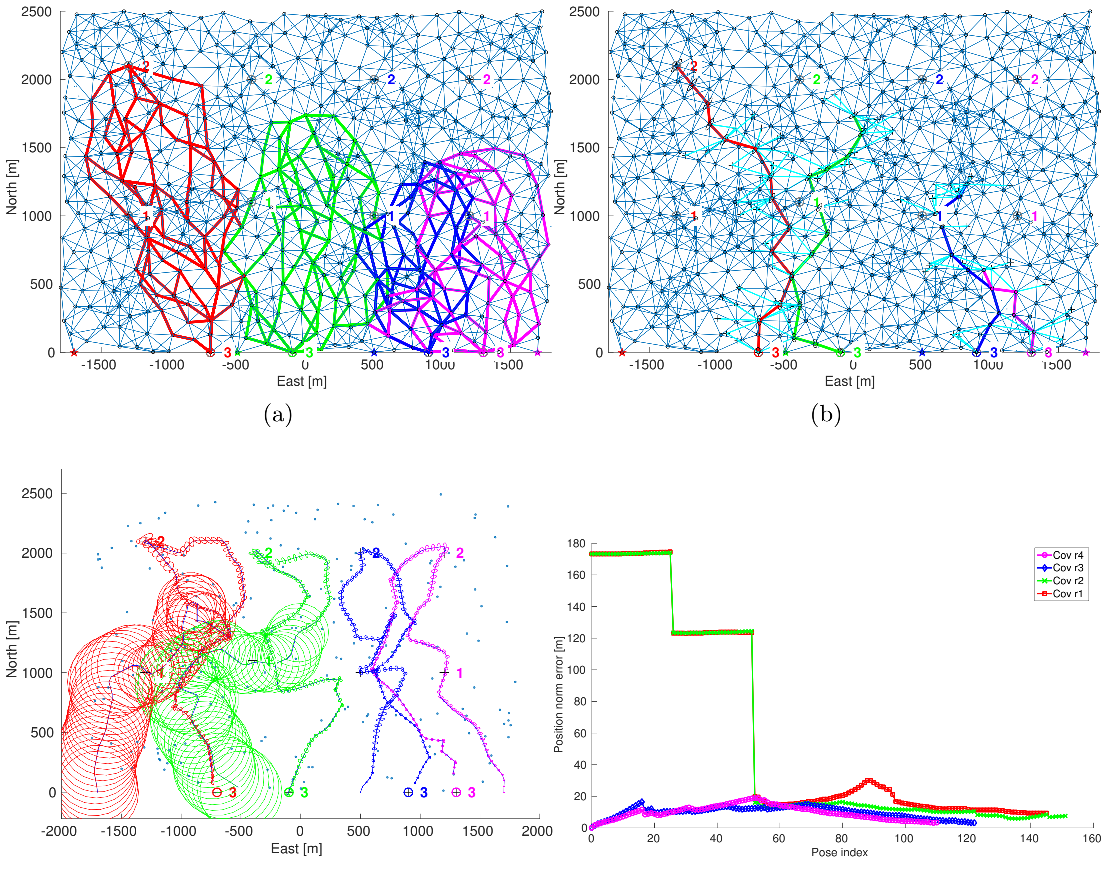


Figure A.3: Scenario2. Third planning session. (a) Candidate paths to the third goal for the purple robot, and to the second goal of each other robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.



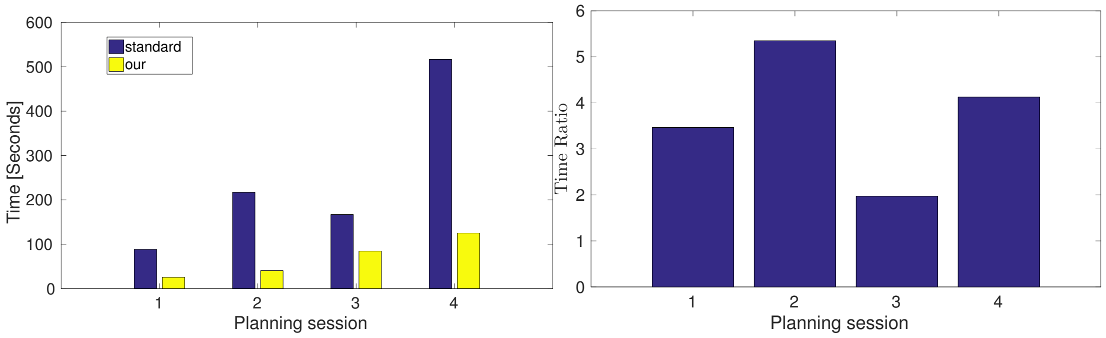
(a)

(b)

(c)

(d)

Figure A.4: `Scenario2`. Fourth planning session. (a) Candidate paths to the third goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the fourth planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.



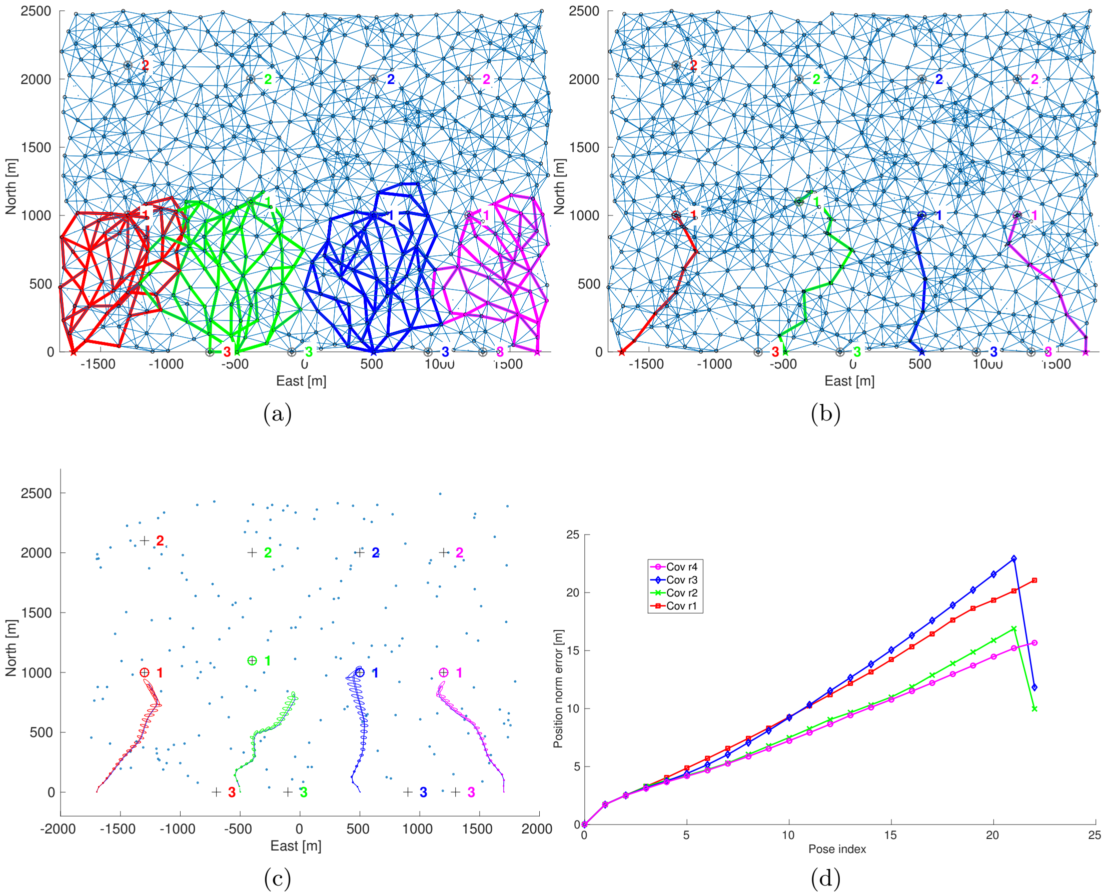
(a)

(b)

Figure A.5: `Scenario2`. Running time comparison between the proposed and the Standard approach. (a) running time for each planning session. (b) Time ratio.

A.2 Scenario 3

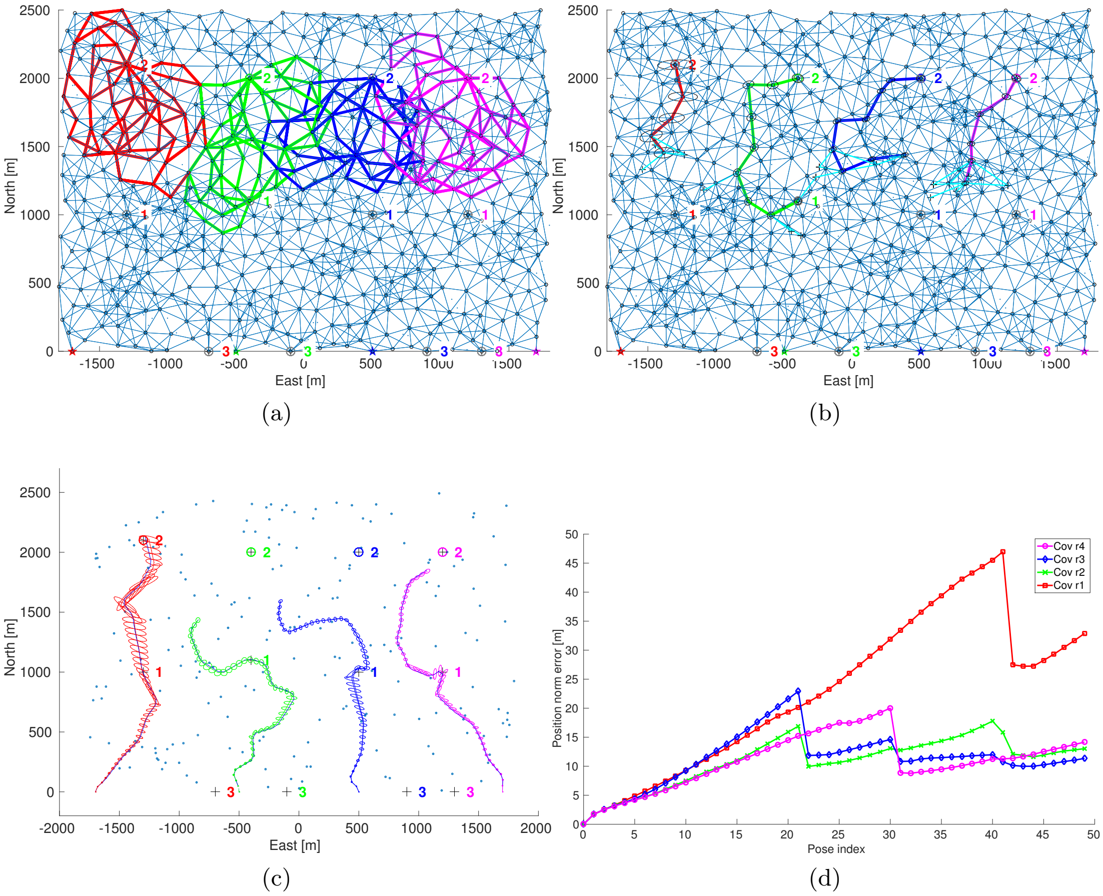
In `Scenario3`, all robots have small and identical covariances at the beginning, as in `Scenario1`. However, in this scenario the robots do *not* use MR factors within planning. Figures A.6-A.8 provide the results for each planning session, while Figure A.9 shows running time.



(a) (b)

(c) (d)

Figure A.6: `Scenario3`. First planning session. States of different robots are not correlated. (a) Candidate paths to the first goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the first planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.



(a)

(b)

(c)

(d)

Figure A.7: `Scenario3`. Second planning session. (a) Candidate paths to the second goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the second planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.

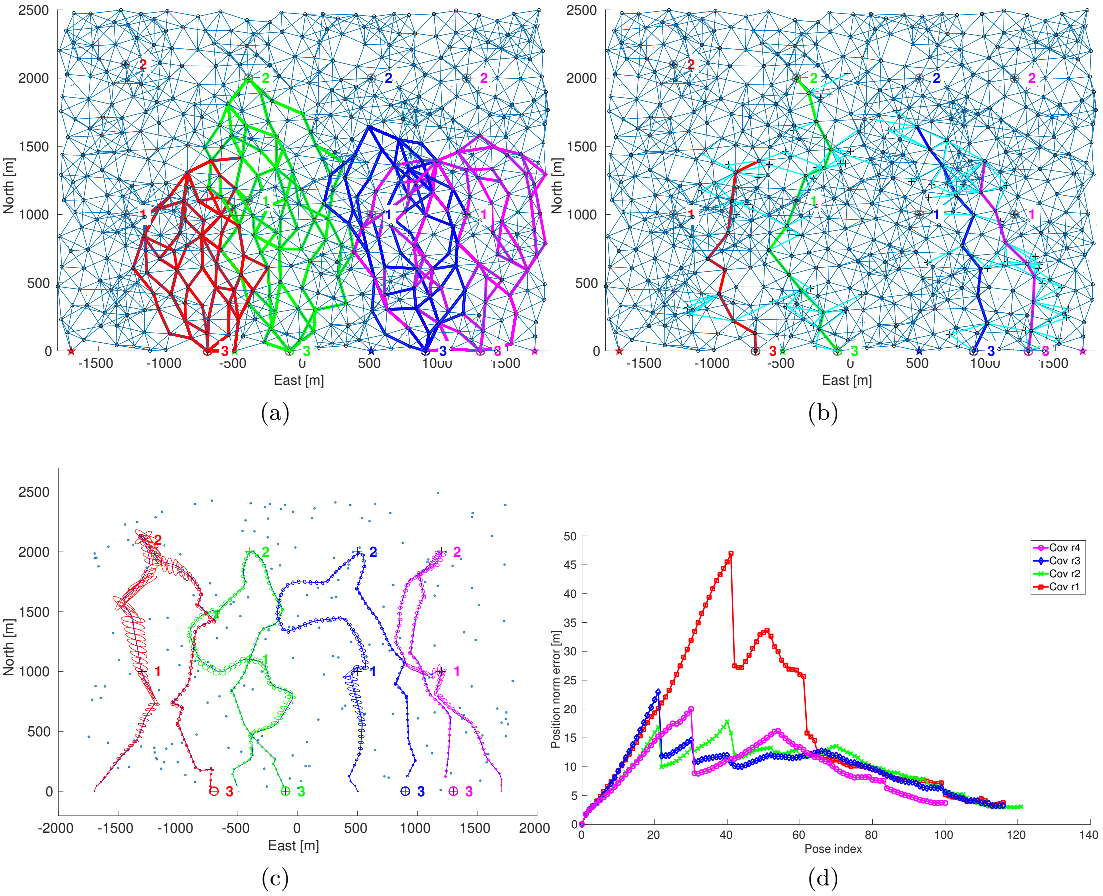


Figure A.8: `Scenario3`. Third planning session. (a) Candidate paths to the third goal of each robot; (b) Chosen paths by the planning approach. (c) Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. (d) Corresponding position covariance evolution.

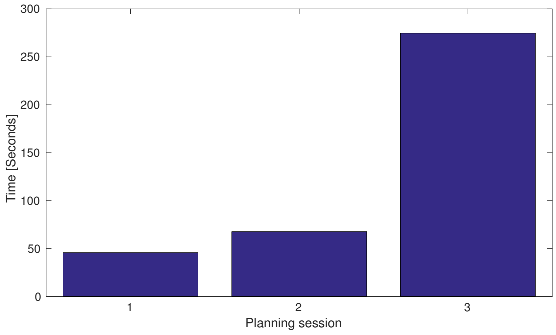


Figure A.9: `Scenario3`. Running time for each planning session.

List of my Publications

[1] T. Regev and V. Indelman. Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. In *Israel Annual Conference on Aerospace Sciences*, 2016.

[2] T. Regev and V. Indelman. Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. In *Israel Robotics Conference*, 2016.

[3] T. Regev and V. Indelman. Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016. Accepted.

[4] T. Regev and V. Indelman. Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. *Autonomous Robots*, 2016. To be submitted.

Bibliography

[5] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato. Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *Intl. J. of Robotics Research*, 2014.

[6] C. Amato, G. D. Konidaris, A. Anders, G. Cruz, J. P. How, and L. P. Kaelbling. Policy search for multi-robot coordination under uncertainty. In *Robotics: Science and Systems (RSS)*, 2015.

[7] Nancy M Amato and Yan Wu. A randomized roadmap method for path and manipulation planning. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 113–120. IEEE, 1996.

[8] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas. Decentralized active information acquisition: Theory and application to multi-robot slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[9] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[10] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 723–730, 2011.

[11] S. M. Chaves, A. Kim, and R. M. Eustice. Opportunistic sampling-based planning for active visual slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3073–3080. IEEE, 2014.

[12] A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[13] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1403–1410. IEEE, 2003.

[14] F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012.

[15] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.

[16] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robotics*, 22(6):1100–1114, Dec 2006.

[17] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *Intl. J. of Robotics Research*, pages 1271–1287, 2014.

[18] V. Indelman. Towards cooperative multi-robot belief space planning in unknown environments. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, September 2015.

[19] V. Indelman. Towards multi-robot active collaborative state estimation via belief space planning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, September 2015.

[20] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015.

[21] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Graph-based distributed cooperative navigation for a general multi-robot measurement model. *Intl. J. of Robotics Research*, 31(9), August 2012.

[22] V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738, August 2013.

[23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.

[24] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[25] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Intl. J. of Robotics Research*, 30(7):846–894, 2011.

[26] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, 1996.

[27] Lydia Kavraki and J-C Latombe. Randomized preprocessing of configuration for fast path planning. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2138–2145. IEEE, 1994.

[28] A. Kim and R. M. Eustice. Active visual slam for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 2014.

[29] H. Kurniawati, D. Hsu, and W. S. Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems (RSS)*, volume 2008, 2008.

[30] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *Intl. J. of Robotics Research*, 20(5):378–400, 2001.

[31] D. Levine, B. Luders, and J. P. How. Information-theoretic motion planning for constrained sensor networks. *Journal of Aerospace Information Systems*, 10(10):476–496, 2013.

[32] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *Intl. J. of Robotics Research*, 32(7):826–840, 2013.

[33] C. Papadimitriou and J. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[34] S. Pathak, A. Thomas, A. Feniger, and V. Indelman. Robust active perception via data-association aware belief space planning. *arXiv preprint: 1606.05124*, 2016.

[35] R. Platt, R. Tedrake, L.P. Kaelbling, and T. Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, pages 587–593, Zaragoza, Spain, 2010.

[36] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 2009.

[37] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, pages 65–72, 2005.

[38] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.

נקרא המסלול הנבחר (Announced path). חישוב המרחב ההסתברותי מתבצע עבור המסלולים המועמדים של הרובוט בהינתן המסלולים הנבחרים של שאר הרובוטים, וזאת בניגוד לשיטה הממצה, בה החישובים מתבצעים עבור כל המסלולים המועמדים של שאר הרובוטים. למרות שהשיטה הנפוצה יותר חסכונית חישובית מהשיטה הממצה, שיטה זו יכולה להיות יקרה חישובית, במיוחד עבור מקרים כך שכל רובוט מייצר הרבה מסלולים מועמדים עבורם מתבצע חישוב של המרחב הסתברותי, על פי פונקצית המטרה, אחרי שאחד הרובוטים מעדכן את המסלול הנבחר שלו. שיטות מסוג זה כמו השיטה הנפוצה, כל איטרציה הרובוט מחשב את המסלולים המועמדים מחדש.

בתיזה זו פיתחנו שיטה לייעול את שיטת המסלול הנבחר על ידי זיהוי ועדכון רק את המסלולים שמושפעים כתוצאה מההחלפה של המסלול הנבחר, כלומר החלפת מסלול נבחר חדש במסלול נבחר ישן (קודם). אנו מסיירים רק על הצמתים של המסלולים הנבחרים, הישן והחדש. עבור כל צומת, אנו מסמני את הצמתים המושפעים שנמצאים במסלולים לבחירה, שנקראים צמתים מעורבים (V_{inv}) ומחשבים עבורם את האילוצים שצריך להוסיף ולהוריד בהתאמה. עבור כל אילוץ אנו מחשבים רק את העדכון שיש להוסיף ולחסר מהאיטרציה הקודמת, ללא חישוב מחדש. אנו מחשבים את העדכון שנובע כתוצאה משינוי המסלול הנבחר. בנוסף למסלולים שלא מושפעים, כלומר המסלולים ללא אילוצים, באותה איטרציה, אנו מחשבים חישוב אחד בלבד בשביל כל אותם מסלולים. השיטה שלנו מבוססת על כך שניתן לפעפע את עדכון המידע של המסלולים שהושפעו בעזרת הצמתים המעורבים, כפי שהוסבר קודם. בעולם הרובוטיקה פותחה לאחרונה שיטה לאופטימזציה שהיא יעילה חישובית iSam. אנו משתמשים בשיטה הזאת עם היכולות של מודל גרפי, פקטור גרף בשביל חישוב יעיל מאד של המרחב הסתברותי על פי פונקצית המחיר של המסלולים. המודל גרפי הוא כלי הנדרש לחישוב העדכון לכל מסלול שהושפע.

בחלק הראשון של התרחישים, הנחנו שלרובוטים אין קורלציה התחלתית ביניהם, כאשר הם עושים תכנון וחישוב של המסלולים, אך השיטה שלנו עדיין מתמודדת עם מקרים בהם קיימת קורלציה. במקרים בהם יש קורלציה, בתנאים מסויימים, האלגוריתם מפעיל קירוב לעדכון כל המסלולים. הגישה הנאיבית הייתה לסמן כל מקרה בו יש קורלציה את כל המסלולים הנבחרים כמושפעים, ולחשב את מרחב הסתברותי שלהם. לעומת זאת, בשיטה שלנו, רק כאשר יש השפעה חזקה של הקורלציה על המסלולים הנבחרים, האלגוריתם מסמן את כל המסלולים הנבחרים כמושפעים, ומחשב את המרחב הסתברותי. כאשר ההשפעה חלשה, האלגוריתם לא יכליל את הקורלציה בחישובים, ומכאן הקירוב, ונחשב את המסלולים הנבחרים ללא שינוי כמו בפסקה הקודמת.

אנו מציגים את הגישה שלנו בסימולציה סינטטית. יצרנו תרחישים בהם נמצאים בין 2 ל 4 רובוטים, בסביבה לא ידועה. לכל רובוט הייתה נקודת התחלה ומטרות אשר הוא צריך לנווט אליהם. בסביבה עצמה היו עצמים אשר לא היו ידועים לרובוט בהתחלה, וכן אחרי תכנון מסלול הרובוטים היו נעים לעבר המטרות תוך כדי ביצוע SLAM.

ביצענו חקר ביצועים מעמיק אשר כלל מספר תרחישים. השוונו זמנים בין השיטה שלנו לבין השיטה הסטנדרתית. עבור כל התרחישים השונים השיטה שלנו יותר מהירה ביותר מפי שניים, ובמקרים מסויימים עד פי 8.

תקציר

בתיזה זו אנחנו מפתחים גישה חדשה של תכנון מרחב הסתברותי לניווט אקטיבי במערכת מבוזרת מרובת פלטפורמות בסביבה לא ידועה. כאשר הרובוטים נמצאים בסביבה לא ידועה, יש צורך לפתור שתי בעיות בבת אחת: שיערוך המיקום של הרובוט ומיפוי. יש צורך לפתור את הבעיות במקביל מכיוון שהן תלויות הדדית. כדי לשערך את מיקום הרובוט, יש צורך לדעת את מיקום העצמים במפה יותר מדוייק, וכדי לשער את העצמים במפה יש לדעת את מיקום הרובוט יותר מדוייק, ולכן אנו פותרים אותן בו־זמנית. בעיה זו ידועה כ Simultaneous Localization and Mapping (SLAM). אנו משתמשים במודל גרפי הנקרא פקטור גרף (Factor Graph), כדי למדל את הבעיה. במודל גרפי הצמתים מסמלים את הנעלמים האקראיים בבעיה, למשל מיקום הרובוט ומיקום העצמים בסביבה. הקשתות מסמלות את האילוצים בין נעלמים אקראיים הללו. לדוגמא, אילוץ של מרחק בין שני מצבים של הרובוט, (לדוגמא אודומטריה). דוגמא נוספת היא אילוץ של חיישן מרחק עם משתנה מיקום הרובוט ומשתנה מיקום של העצמים בסביבה.

בתיזה זו אנו עוסקים בתכנון מסלולים אקטיבי בסביבה לא ידועה, ולכן הרובוט צריך לתכנן מסלול באופן אוטונומי. מספר המסלולים בהם הרובוט יכול לנוע בסביבה הוא אינסופי. אנו רוצים להפוך את הסביבה לבדידה כך שהרובוט יוכל לנוע במספר סופי של אפשרויות בסביבה, כלומר הרובוט יוכל לנוע רק במקומות נבחרים. אנו משתמשים בשיטת דגימה בשם PRM, אשר דוגמת את הנקודות בסביבה הלא ידועה באופן רנדומלי ומחברת נקודות קרובות אחת לשנייה. נוצרת מפה עם צמתים אשר מסמלים את המיקום אשר הרובוט יכול להמצא בהם, והקשתות מסמלות את המסלול בו הרובוט יכול לנוע מצומת לצומת. בהינתן נקודת התחלה ונקודת סיום, הרובוט יכול לייצר לעצמו מספר מסלולים ומתוכם לבחור מסלול הטוב ביותר על סמך פונקצית המטרה.

במחקר זה אנו מתמקדים בשיתוף פעולה בין מספר רובוטים, כאשר היתרון הוא ניווט יותר רובסטי, שיערוך מיקום עצמי ומיקום עצמים בסביבה יותר מדוייק. הגישה שלנו מבוססת על שיטת דגימה PRM שמוסברת בפסקה הקודמת, כאשר רובוטים מייצרים מספר מסלולים, אשר נקראים מסלולים מועמדים, ובוחרים מהם את המסלול הכי טוב על סמך פונקציית המטרה.

השיטה האופטימלית לבחור מסלולים אלו נקראת השיטה הממצה (Exhaustive). בשיטה זו, כל רובוט מחשב את כל האפשרויות בין המסלולים בשונים בין הרובוטים השונים. שיטה זו יקרה מאד חישובית. שיטה נוספת, מקובלת יותר היא פתרון תת־אופטימלי לשיטה הקודמת, אשר נקראת שיטת המסלול הנבחר (Announced path). שיטה זו איטרטיבית, כך שבכל איטרציה, כל רובוט, מחשב את המרחב הסתברותי (Belief), כלומר את הפילוג הסתברותי עבור הנעלמים האקראיים של הרובוט על פי פונקצית מטרה. לאחר החישוב באיטרציה הנוכחית, בוחר הרובוט את המסלול הטוב ביותר שמצא עד כה על פי פונקצית המטרה ושולח מסלול זה לשאר הרובוטים. מסלול זה

המחקר בוצע בהנחייתו של פרופ"מ ואדים אינדלמן מפקולטה להנדסת אווירונאוטיקה, בפקולטה למדעי המחשב בטכניון - מכון טכנולוגי לישראל.

תודות

אני רוצה להודות למנחה שלי, שלא רק הדריך והסביר על נושאים רלוונטיים, אלא גם הציע כיוונים חדשים במחקר. למשפחתי ולהורי אשר שתמכו בי לכל אורך הדרך, ועזרו לי רבות בהגעה לתיזה. אני רוצה להודות בנוסף לחברי המעבדה, אשר באופן קבוע נתנו לי רעיונות מעניינים והצעות, אשר חלק מהם שינו את כיוון החשיבה שלי, במיוחד בעיצוב ושיפור בתוכן של המצגות. ולבסוף אני רוצה להודות למגוון חברים אשר רכשתי במהלך השנים במשך לימודי.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תכנון מרחב הסתברותי לניווט אקטיבי במערכת מבוזרת מרובת פלטפורמות בסביבה לא ידועה

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים במדעי המחשב

טל רגב

הוגש לסנט הטכניון — מכון טכנולוגי לישראל
אלול התשע"ו   חיפה   ספטמבר 2016

תכנון מרחב הסתברותי לניווט אקטיבי במערכת מבוזרת מרובת פלטפורמות בסביבה לא ידועה

טל רגב