TESIS DE LA UNIVERSIDAD DE ZARAGOZA

2015

25

María Teresa Lázaro Grañón

Map building, localization and exploration for multi-robot systems

Departamento

Informática e Ingeniería de Sistemas

Director/es

Castellanos Gómez, José Ángel
Grisetti, Giorgio

ISSN 2254-7606

Prensas de la Universidad Universidad Zaragoza

Universidad Zaragoza
1542

Tesis Doctoral

MAP BUILDING, LOCALIZATION AND EXPLORATION FOR MULTI-ROBOT SYSTEMS

Autor

María Teresa Lázaro Grañón

Director/es

Castellanos Gómez, José Ángel
Grisetti, Giorgio

UNIVERSIDAD DE ZARAGOZA

Informática e Ingeniería de Sistemas

Universidad Zaragoza

1542

Instituto Universitario de Investigación en Ingeniería de Aragón
Universidad Zaragoza

PH.D. THESIS

Map Building, Localization and Exploration for Multi-Robot Systems

MARÍA TERESA LÁZARO GRAÑÓN

Supervised by
José Ángel Castellanos Gómez
Giorgio Grisetti

Robotics, Perception and Real Time Group (RoPeRT)
Instituto de Investigación en Ingeniería de Aragón (I3A)
Escuela de Ingeniería y Arquitectura (EINA)
Universidad de Zaragoza (UZ)

January 2015

PH.D. THESIS

Map Building, Localization and Exploration for Multi-Robot Systems

María Teresa Lázaro Grañón

January 2015

Thesis defended under the *Programa de Doctorado en Ingeniería de Sistemas e Informática* with International Ph.D. Mention (*Mención Internacional*)

Supervisors

José Ángel Castellanos Gómez	Universidad de Zaragoza, Spain
Giorgio Grisetti	Università degli Studi di Roma "La Sapienza", Italy

Composition of the Thesis Commitee

Luis Enrique Montano Gella	Universidad de Zaragoza, Spain
Alberto Sanfeliu Cortés	Universitat Politècnica de Catalunya, Spain
José Santos-Victor	Instituto Superior Técnico, Lisboa, Portugal
José Neira Parra	Universidad de Zaragoza, Spain
Javier González Jiménez	Universidad de Málaga, Spain

International Reviewers

Shoudong Huang	University of Technology, Sydney, Australia
Luca Iocchi	Università degli Studi di Roma "La Sapienza", Italy

Resumen

La idea de tener robots desempeñando la tarea para la cual han sido diseñados de forma completamente autónoma e interactuando con su entorno ha sido el principal objetivo desde los inicios de la robótica móvil. Para conseguir tal grado de autonomía, es indispensable que el robot disponga de un mapa del entorno y conocer su localización en él, además de ser capaz de resolver otros problemas como el control del movimiento y la planificación de trayectorias hacia su objetivo.

En el desempeño de ciertas misiones sin conocimiento previo de su entorno, el robot debe utilizar la información imprecisa proporcionada por sus sensores para construir un mapa al mismo tiempo que se localiza en él, lo que da lugar al problema de localización y construcción de mapas de forma simultánea (o con las iniciales de su denominación anglosajona, SLAM) ampliamente estudiado en robótica móvil.

En los últimos años, ha habido un creciente interés por la utilización de equipos de robots debido a los múltiples beneficios que ofrecen respecto a sistemas de un solo robot, tales como una mayor robustez, precisión, eficiencia y la posibilidad de cooperar para realizar una tarea o cubrir entornos más grandes en menos tiempo. En este ámbito de los robots cooperativos encontramos también el caso específico de las formaciones de robots, donde deben adoptar una estructura concreta mientras navegan por el entorno.

A pesar de sus ventajas, la complejidad de los sistemas multi-robot autónomos aumenta con el número de robots ya que es mayor la cantidad de información que ha de ser manejada, almacenada y transmitida a través de la red de comunicaciones. Así pues, el desarrollo de estos sistemas presenta nuevas dificultades a la hora de dar solución a los problemas anteriormente mencionados que, en lugar de abordarlos de manera individual para cada robot, han de ser resueltos de forma cooperativa para aprovechar de forma eficiente la información recogida por los todos los miembros del equipo. El diseño de los algoritmos en este contexto multi-robot ha de orientarse a obtener la mayor escalabilidad y rendimiento posible que permitan su ejecución online.

Esta tesis está enmarcada en el ámbito de los sistemas multi-robot, proponiendo soluciones a cada uno de los procesos de navegación, localización, construcción de mapas y planificación de caminos que conforman un sistema autónomo. Una primera parte de las contribuciones presentadas en esta tesis está desarrollada en el contexto

de las formaciones de robots, que exigen una mayor cooperación y sincronización del equipo, aunque pueden ser extendidas a sistemas sin esa restricción en la navegación. Asumiendo que la formación dispone de un mapa del entorno, posiblemente parcial e imperfecto, en el que debe realizar la misión encomendada proponemos técnicas para la localización, mejora del mapa previo y exploración del mismo. En una segunda parte, proponemos un método de SLAM multi-robot sin ninguna asunción en cuanto al conocimiento previo del mapa o de las relaciones entre robots en el que utilizamos técnicas pertenecientes al estado del arte para gestionar eficientemente los recursos disponibles en el sistema. A lo largo de la tesis se ha validado el rendimiento y eficacia de los sistemas de robots en formación y de SLAM multi-robot mediante su implementación y puesta a punto tanto en simulaciones como en equipos de robots reales.

Abstract

The idea of having robots performing the task for which they have been designed completely autonomously and interacting with the environment has been the main objective since the beginning of mobile robotics. In order to achieve such a degree of autonomy, it is indispensable for the robot to have a map of the environment and to know its location in it, in addition to being able to solve other problems such as motion control and path planning towards its goal.

During the fulfillment of certain missions without a prior knowledge of its environment, the robot must use the inaccurate information provided by its on-board sensors to build a map at the same time it is located in it, arising the problem of Simultaneous Localization and Mapping (SLAM) extensively studied in mobile robotics.

In recent years, there has been a growing interest in the use of robot teams due to their multiple benefits with respect to single-robot systems such as higher robustness, accuracy, efficiency and the possibility to cooperate to perform a task or to cover larger environments in less time. Robot formations also belongs to this field of cooperative robots, where they have to maintain a predefined structure while navigating in the environment.

Despite their advantages, the complexity of autonomous multi-robot systems increases with the number of robots as a consequence of the larger amount of information available that must be handled, stored and transmitted through the communications network. Therefore, the development of these systems presents new difficulties when solving the aforementioned problems which, instead of being addressed individually for each robot, must be solved cooperatively to efficiently exploit all the information collected by the team. The design of algorithms in this multi-robot context should be directed to obtain greater scalability and performance to allow their online execution.

This thesis is developed in the field of multi-robot systems and proposes solutions to the navigation, localization, mapping and path planning processes which form an autonomous system. The first part of contributions presented in this thesis is developed in the context of robot formations, which require greater team cooperation and synchronization, although they can be extended to systems without this navigation constraint. We propose localization, map refinement and exploration techniques under the assump-

tion that the formation is provided with a map of the environment, possibly partial and inaccurate, wherein it has to carry out its commanded mission. In a second part, we propose a multi-robot SLAM approach without any assumption about the prior knowledge of a map nor the relationships between robots in which we make use of state of the art methodologies to efficiently manage the resources available in the system. The performance and efficiency of the proposed robot formation and multi-robot SLAM systems have been demonstrated through their implementation and testing both in simulations and with real robots.

Acknowledgements

It would not be fair to conclude this Thesis without a few words of thanks to the people who helped me along these years.

I would like to thank my supervisor José A. Castellanos who trusted in me from the beginning and gave me the opportunity to carry out this project. His knowledge and expertise were a valuable reference in all these years.

Many thanks also to my co-supervisor Giorgio Grisetti for his generous involvement and guidance since my first stay at Rome. His enthusiasm and dedication were motivational in the last stage of this Thesis.

My sincere thanks also to Lina M. Paz for her support and encouragement during the period of time we worked together and for the fruitful discussions we also had jointly with Pedro Piniés.

My thanks to Luis Montano with whom I took my first steps in robotics research during my Bachelor's Degree Final Project and encouraged me to go on and start a PhD.

I would also like to thank my colleagues of our robotics group for the good moments shared together and specially to Pablo Urcola for his collaboration along these years and for his willingness each time I needed his help.

Finally, my deepest thanks are to my family and friends for their constant support and patience and particularly to my parents and brother who always were there in the good and in the hard moments.

List of Figures

List of Acronyms

BN	Bayesian Network
CI	Conditional Independence
DBN	Dynamic Bayesian Network
EIF	Extended Information Filter
EKF	Extended Kalman Filter
GMRF	Gaussian Markov Random Field
KF	Kalman Filter
KLD	Kullback-Leiber Divergence
MAP	Maximum a Posteriori
MD-EKF	Measurement-Differencing based EKF
MRF	Markov Random Field
ND	Nearness Diagram
NEES	Normalized Estimation Error Squared
NTP	Network Time Protocol
ROS	Robotic Operating System
RT-WMP	Real Time Wireless Multihop Protocol
SCGP	Single-Cluster Graph Partitioning
SLAM	Simultaneous Localization and Mapping

Contents

Chapter 1

Introduction

1.1 Motivation

The development of robots accomplishing their commanded task completely autonomously has been the main objective in mobile robotics during the last decades. This is motivated by the wide variety of applications either in the social sphere, such as rescue missions, surveillance, guidance of people in emergency situations or exploration in hostile environments of hard access to humans as well as in the industrial field such as in maintenance, inspection or transportation of goods. Autonomous robots such as vacuum cleaners and lawn mowers are already present at our homes, freeing us from these time consuming tasks, whereas more advanced robotics such as the Mars Curiosity rover and the Google self-driving car have also been successfully developed.

To fulfill its mission completely autonomously in its surrounding environment, the robot has to perform a number of tasks which can be grouped in *motion control*, *path planning*, *localization* and *mapping*, closely interrelated as shown in Fig. 1.1. The motion control is the problem of computing the actions or control inputs (usually velocity commands) to move the robot towards a desired location from its current position. The path planning seeks the optimal trajectory (sequence of waypoints or sub-goals) the robot has to follow to reach the global goal of the mission. Path planning usually includes obstacle avoidance to cope with unexpected situations which may modify the initially computed trajectory. To reach a desired location or to compute a trajectory, the robot first needs to know its current position in the environment. The localization system provides the position with respect to a map of the environment which can be known *a priori* or acquired simultaneously to the performance of the mission. In this latter case, the mapping process can not be decoupled from the localization process leading to the problem of *Simultaneous Localization and Mapping* (SLAM). Due to the difficulty of integration of the uncertain information provided by the robot's on-board sensors, the SLAM problem became one of the main research topics in mobile robotics, considering

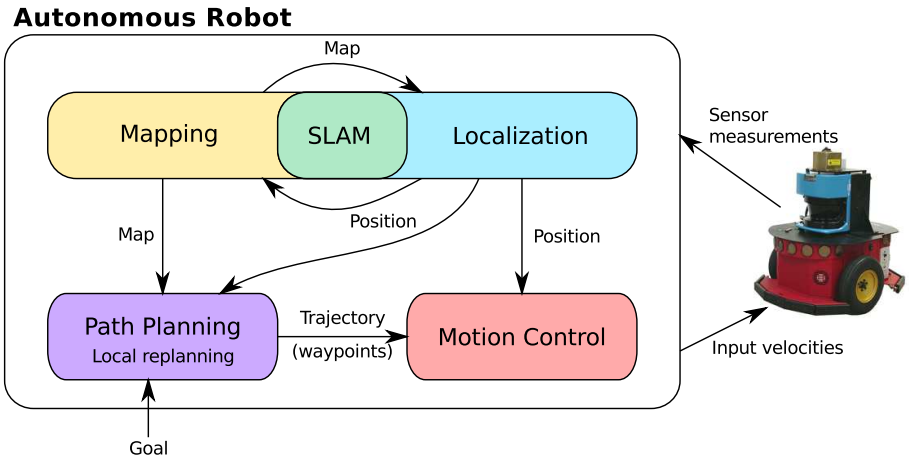


Figure 1.1: Tasks to be solved by an autonomous robot.

the concurrent learning of the environment and the robot localization in a probabilistic manner.

As we can see, these problems can not be addressed independently due to their interdependencies and solving all them together is a challenge in mobile robotics. Then, it is usual to solve individual problems by assumming certain previous knowledge. For example, having a closer look to Fig. 1.1, we can solve localization and path planning problems if the map of the environment is given. Also, we can focus on the SLAM problem without concerning of path planning and motion control by manually teleoperating the robot.

As the techniques developed for single robots matured, researchers started to show interest in the use of multiple robots due to their attractive advantages with respect to the single-robot case. For example, in surveillance and reconnaissance missions, multiple robots cover the environment in less time and offer increased robustness since, if one of the robots fails, it does not necessarily imply aborting the mission. However, the development of an autonomous multi-robot system can not be addressed by simply replicating the scheme shown previously on each single entity. That would be an inefficient implementation without taking advantage of the team cooperation. For example, robot localization can be improved if they are able to detect each other. Also, the integration of information coming from multiple robots results in more accurate maps and expanded fields of view to obtain better paths to the goal.

Then, to develop an autonomous multi-robot system, we need to extend the abovementioned tasks to exploit as much as possible all the information available in the system. In this thesis we focus on providing solutions to the localization and SLAM problems in the context of multi-robot sytems which are also integrated with path planning and multi-robot motion control tasks to obtain a complete system. Besides, part of this thesis aims at giving a practical implementation of the proposed techniques for the case

of robot formations. A robot formation is a specific case of multi-robot system where the robots are constrained to navigate following a predefined structure. Tipically, robots in the formation are organized in such a way that there is a robot *leader* in charge of following a precomputed path towards a goal whereas the rest of robots, the *followers*, try to maintain the geometric inter-vehicle restrictions even though these can be ocassionally adapted to the dynamic environment. These kind of multi-robot systems can be particularly relevant to fulfill missions where, whether due to environmental restrictions or to exploit a set of heterogeneous robots carrying different sensors, the team is forced to adopt a specific configuration. In other applications, the main task is achieved by the leader and the rest of robots follow him closely to act as suppliers of computational resources.

1.1.1 SLAM and Multi-Robot Issues

For many years, the research efforts were focused on solving the SLAM problem by using *filtering* techniques, being the most prominent those based on the Extended Kalman Filter (EKF). Under the Bayesian filtering context, the current state of the vehicle and the map are estimated by recursively integrating the robot motion and sensor measurements as they are available. However, as these techniques progressed, the eagerness of solving the SLAM problem in more challenging scenarios, made some of their weaknesses such as unbounded computational requirements, scaling and long-term map maintainance apparent. Besides, due to the inherent nonlinear nature of the SLAM problem, the EKF framework offers no guarantee of convergence due to linearization errors that cannot be reverted. These convergence problems make it difficult to obtain consistent and accurate maps and affects the robot's ability to determine when it is entering into an already explored area, the so-called loop closing problem. Despite these drawbacks, the filtering techniques have also been shown to be very efficient in situations where the dimension of the problem is bounded such as in small scale scenarios or robot localization once the map is provided.

In the last years, the so-called *smoothing* approaches have attracted the interest in this field due to their enhanced performance. These alternative methods require solving the also called *full* SLAM as a least squares optimization problem and allow to recover from wrong linearization choices since the problem is re-evaluated around the current estimate using all history of measurements and robot positions.

We have discussed the multiple advantages the use of a multi-robot team offers with respect to the single-robot case, such as robustness, scalability, accuracy, eficiency and the possibility of cooperation to perform a task or to cover larger environments. However, the development of multi-robot systems requires dealing with new issues that are not present in the single-robot case. We could group them into three main topics:

Computational issues One of the main problems a multi-robot system has to deal with

is how to combine the local map information acquired by each robot, usually expressed with respect to its own reference frame. It requires solving a data association problem dependent on the knowledge about their relative positions which can be easily computed if the robots work with respect the same reference frame. If this is not the case, the relative positions can be obtained directly through inter-robot measurements or indirectly by local map matching. Additionally, the use of multiple robots increases the overall information available, which has to be efficiently managed and stored according to the processing capabilities and memory resources.

Communication issues The way in which the communication network is designed depends principally on the task to be accomplished and on the infrastructure, being wireless Ad-Hoc networks the common choice in mobile multi-robot systems. These type of networks have limited communication range, therefore its topology can be dynamically reconfigured based on the proximity of the robots. Regarding the kind of data to be transmitted, it is important to adapt the message size to the bandwidth requirements to avoid overloading the network. This aspect may be particularly relevant in the case of handling large groups of robots.

Motion coordination Certain tasks such as area coverage or joint manipulation of goods demand strong coordination to perform them optimally. Coordination is indispensable in order to avoid robots interfering with each other and to minimize the task execution time.

Obviously, these issues are highly related, since there is no coordination without information sharing, which has to be exchanged and managed efficiently to avoid redundancies in the network. In order to address these issues we can adopt two different schemes depending on the task commanded to the multi-robot system.

Centralized scheme In centralized schemes, the computational payload and coordination decisions rely on a single robot upon reception of all information available in the network. Produces optimal results although its processing complexity is exponential in the number of robots. They are suitable in tasks where the communication and synchronization between robots can be guaranteed.

Distributed scheme In distributed schemes, the computational payload is shared by all robots and the system is more fault-tolerant since the completion of the task is not compromised by the failure of a single robot. However, they require more coordination (and therefore communication) on decisions about how to perform a task since it is more difficult to have a global vision of the whole system.

1.2 Contributions of the Thesis

This thesis presents several contributions in the field of multi-robot localization and mapping, where both filtering and smoothing approaches are used, adapting them to the specific requirements of the problems that are going to be addressed. Some of the techniques are particularized to the case of robot formations although they can be extended to more general multi-robot systems. Additionally, we present an integrated robot formation system considering also path planning and formation control. These contributions and their organization in the different chapters are summarized below:

- First, in Chapter 3, we address the problem of multi-robot cooperative navigation in formation. We achieve formation control by means of a virtual spring-damper system which generates a set of forces on the robots, later transformed into input velocities by a motion controller. This contribution was published in [85] and constitutes the starting point of this thesis. Additionally, in this thesis we set out the problem of maintaining the formation shape and reaching its commanded goal when we apply velocities corrupted by noise which affect the robots knowledge about their true position in the environment.

- In order to deal with the robot formation localization problem presented in the previous chapter, in Chapter 4 we propose an EKF-based algorithm to compute the robots' positions with respect to a given stochastic map of the navigation area. We show how a direct implementation of the EKF algorithm leads to inconsistency in the estimated localization which makes it unreliable for autonomous navigation. We justify the origin of the anomalous behaviour of the filter in the time-correlated nature of the measurement noise sequence. Then, a novel solution based on the measurement differencing technique is proposed to drive the solution of the EKF towards consistency. We presented this contribution in [49] and in this thesis we give further analysis of the problem and statistical results.

- In Chapter 5 we give a step forward towards achieving the goal of developing a complete working system for robot formations. Then, we integrate the formation control and localization approaches presented in previous chapters with global and online path planning techniques which take into account the uncertainty of the given map of the environment. Both feature-based and grid-based techniques are combined under a probabilistic perspective to obtain paths of bounded risk of collision towards the goal of the formation. Additionally, we incorporate the sensor observations in the estimation algorithm presented in the previous chapter to obtain a joint and unified cooperative view of the formation surroundings which benefits the online replanning process. We presented in [52] an overview of our robot formation system and in this thesis we give a more detailed formulation of the algorithms and provide additional simulated and real world experiments.

- In Chapter 6 we propose a filter-based SLAM algorithm not only to localize the formation in an *a priori* known stochastic map of its navigation environment but also to jointly refine it. Additionally, we adopt a distributed scheme which contributes to an increased robustness of the system. We achieve a significant reduction of the computational cost through the application of conditional independence properties over the probabilistic representation of the system state. Each robot maintains its own local and global maps which are improved with the information received when communications among robots take place. The contributions presented in this chapter were published in [50].

- Last but not least, Chapter 7 presents a multi-robot SLAM approach to model the environment without any prior map information. In this case, the method is not constrained to robot formations but works with general multi-robot systems where robots only share the navigation area occasionally. Our approach addresses the SLAM problem via optimization techniques and is specifically designed to deal with the communication and computational issues that tipically affect multi-robot systems. The use of condensed measurements during the map information exchange process among the robots allows to effectively compress relevant portions of a map in a few data. This results in a substantial reduction of both the data to be transmitted and processed, that renders the system more robust and efficient. Additionally, we propose a robust map alignment algorithm to solve the loop closing and intra-robot data association problems. Our multi-robot SLAM approach was published in [51] and in this thesis we provide further experiments to test the performance of our map alignment algorithm.

1.2.1 Publications

The novelty and originality of the approaches presented in this thesis are supported by the following peer-reviewed international conferences and workshops:

- M. T. Lázaro, L. M. Paz, P. Piniés, J. A. Castellanos and G. Grisetti. Multi-Robot SLAM using Condensed Measurements. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo Big Sight, Japan, Nov 3-8, 2013.

- M. T. Lázaro, L. M. Paz, P. Piniés and J. A. Castellanos, Distributed Localization and Submapping for Robot Formations using a prior map, *The 2013 IFAC Intelligent Autonomous Vehicles Symposium*, Gold Coast, Australia, June 26-28, 2013.

- M. T. Lázaro, P. Urcola, L. Montano, J. A. Castellanos, Position Tracking and Path Planning in Uncertain Maps for Robot Formations, *The 2nd IFAC Workshop on Multivehicle Systems*, Espoo, Finland, pp. 7-12, Oct 3-4, 2012.

- M. T. Lázaro and J. A. Castellanos, Localization of Probabilistic Robot Formations in SLAM, *2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, pp. 3179-3184, May 3-8, 2010.
- P. Urcola, L. Riazuelo, M. T. Lázaro, L. Montano, Cooperative Navigation using environment compliant robot formations, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 2789-2794, Sep 22-26, 2008.

1.2.2 Open Source Contributions

The multi-robot SLAM approach described in Chapter 7 is implemented as a ROS[1] package and is publicly available at

`https://github.com/mtlazaro/cg_mrslam`.

It provides a multi-robot graph-based 2D SLAM with any assumption about data association or initial relative positions between robots. It handles communication among robots working in an Ad-Hoc network where the map information is exchanged by using condensed maps.

[1] Robotic Operating System (`http://www.ros.org`)

Chapter 2

Basic Fundamentals of SLAM

A robot navigating autonomously in an environment requires the availability of a map and its location within it. If the environment is unknown, the robot needs to construct its own map and, at the same time, determine its position in it by using the information gathered by its own sensors. However, these sensors provide noisy measurements and are subject to errors. The Simultaneous Localization and Mapping (SLAM) is a well known research topic in robotics which takes into account these uncertainties by using probabilistic techniques. The aim of this chapter is to introduce the SLAM problem from the point of view of its probabilistic formulation, together with the basic methodologies of the different approaches that have been addressed in this thesis.

2.1 SLAM Problem Definition

Let the trajectory of a robot be represented by a discretized set of poses $\mathbf{x}_{R_{0:K}} = \{\mathbf{x}_{R_0}, \mathbf{x}_{R_1}, ..., \mathbf{x}_{R_K}\}$ where $\mathbf{x}_{R_k}$ is the 2D pose and orientation of the robot $(x, y, \theta)^T$ at time step k. The transition between two consecutive robot states $\mathbf{x}_{R_{k-1}}$ and $\mathbf{x}_{R_k}$ is governed by the control input $\mathbf{u}_k$, being $\mathbf{u}_{1:K} = \{\mathbf{u}_1, ..., \mathbf{u}_K\}$ the history of all control inputs along the trajectory. As the robot moves, it observes the environment with its on-board exteroceptive sensors, obtaining a set of measurements $\mathbf{z}_{1:K} = \{\mathbf{z}_1, ..., \mathbf{z}_K\}$.

The inherent inaccuracies in the application of the motion controls and the noisy sensor readings introduce uncertainties in the system which require the use of a probabilistic framework to deal with the mapping and localization problems. These two problems can be addressed separately in the following way:

1. If the robot positions are known, a map $\mathbf{m}$ can be inferred from them and the set of observations. This mapping problem is formulated in a probabilistic way represented by the posterior probability,

$$p(\mathbf{m}|\mathbf{x}_{R_{0:K}}, \mathbf{z}_{1:K}) \tag{2.1}$$

2. On the other hand, if the robot is provided with a map of the environment, it can use this prior knowledge to compute its trajectory with respect to the map's reference frame. This represents a robot localization problem which is formally expressed as,

$$p(\mathbf{x}_{R_{0:K}}|\mathbf{z}_{1:K}, \mathbf{u}_{1:K}, \mathbf{m}) \tag{2.2}$$

Equations 2.1 and 2.2 highlight the existing interdependency between both mapping and localization problems. In order to obtain the map of the environment $\mathbf{m}$, the robot requires an accurate estimation of its trajectory $\mathbf{x}_{R_{0:K}}$ which must be determined with respect to the map. Therefore, if both the map and the robot positions are unknown, these problems can not be solved separately. This is a well known "chicken or egg" problem, subject of thorough investigation in the last decades by the Simultaneous Localization and Mapping community. Then, the SLAM consists in estimating the joint posterior probability of the robot trajectory and the map given the history of control inputs and measurements,

$$p(\mathbf{x}_{R_{0:K}}, \mathbf{m}|\mathbf{z}_{1:K}, \mathbf{u}_{1:K}) \tag{2.3}$$

which is also known as the *full* SLAM problem, tipically solved by least squares optimization techniques. On the contrary, the *online* SLAM computes the current robot position and the map using the measurements up to certain time step k,

$$p(\mathbf{x}_{R_k}, \mathbf{m}|\mathbf{z}_{1:k}, \mathbf{u}_{1:k}) \tag{2.4}$$

which is addressed by filtering approaches. Both kind of approaches have been used along this thesis and they are explained in the rest of this chapter. The formulation for the individual problems of localization and mapping can be obtained following a similar derivation, which is detailed in [81]. The derivation of their solutions is based on the application of probabilistic rules over the variables involved in the SLAM problem, whose dependencies are represented by the Bayesian network shown in Fig. 2.1. Along this document we will make use of other graphical representations, therefore, we give a brief review of them in the next section. Subsequently, we will describe different types of map representations and the fundamentals of the filtering and optimization techniques used in this thesis to construct them.

2.2 Graphical Models for the SLAM Problem

The SLAM problem can be represented by a variety of graphical models to intuitively visualize its structure and dependencies between the variables involved. The three most commonly used graph representations are briefly introduced in this section. Further and general details about the useful properties of these probabilistic graphical models can be found in [7, 45] and a review of their application in the SLAM context can be found in [23].

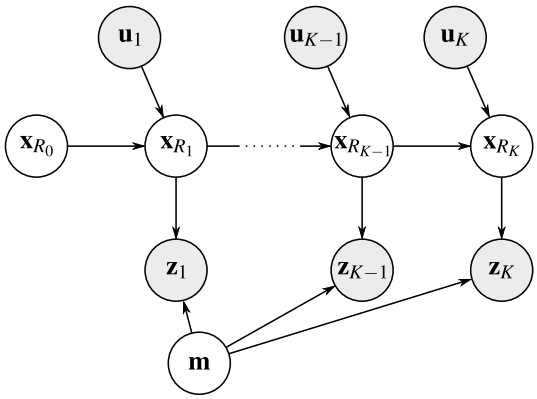


Figure 2.1: SLAM problem represented as a Bayesian network.

2.2.1 Dynamic Bayesian Networks

A Bayesian Network (BN) is a directed acyclic graph where nodes represent either hidden (i.e., unknown) or observable variables and directed links (arrows) are used to represent conditional dependencies among them. The directed link establishes a *parent-child* relation between the connected nodes where the parent is the node from which the edge leaves and the child is the node where the edge leads to. Additionally, Dynamic Bayesian Networks (DBNs) allow to model how these variables are related over time.

Figure 2.2 illustrates a short example of a SLAM problem where the variables to be estimated are the robot poses $\mathbf{x}_{R_{1:4}}$ and the map $\mathbf{m}$, represented by a set of *features* $\mathcal{F} = \{F_{1:4}\}$. The observed variables are the odometry measurements $\mathbf{u}_{1:4}$ and sensor observations where $\mathbf{z}_k^i$ is the i^{th} measurement at time step k.

Through this kind of graph we can directly represent some conditional distributions present in the SLAM problem such as $p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k), \forall k > 0$ and $p(\mathbf{z}_k^i|\mathbf{x}_{R_k}, \mathbf{x}_{F_{j_{ik}}})$ where $F_{j_{ik}}$ is the feature associated to measurement $\mathbf{z}_k^i$. These probabilities are called *transition* (or motion) *model* and *measurement* (or observation) *model*. We can also deduce the full joint posterior over the set of variables factorized as,

$$p(\mathbf{x}_{R_{0:K}}, \mathbf{m}, \mathbf{z}_{1:K}, \mathbf{u}_{1:K}) = p(\mathbf{x}_{R_0}) \prod_k p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) \prod_i p(\mathbf{z}_k^i|\mathbf{x}_k, \mathbf{x}_{F_{j_{ik}}}) \tag{2.5}$$

2.2.2 Markov Random Fields

A Markov Random Field (MRF) is an undirected graphical model and may contain cycles. In MRFs only the hidden variables are represented and measurements are implicitly encoded on the edges between the unknown the variables. The undirected graph representation from a directed graph can be found by a process called *moralization*

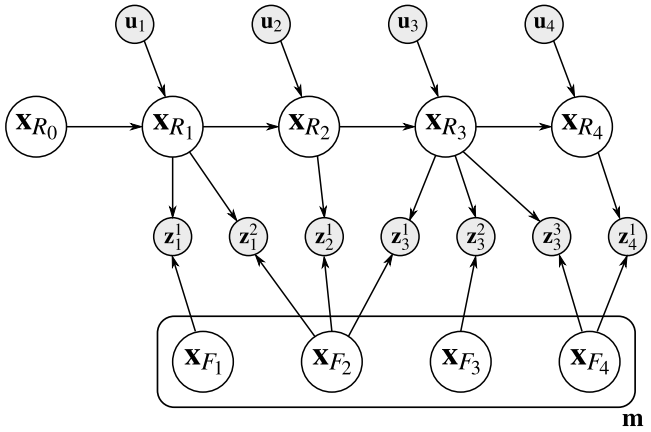


Figure 2.2: Example of a feature-based SLAM problem represented as a Bayesian network.

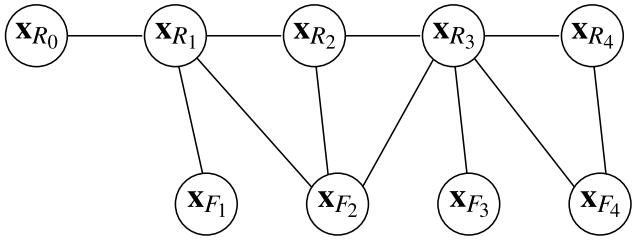


Figure 2.3: SLAM problem represented as a MRF.

where arrows are replaced by undirected links and parents of a common child must be linked each other. As a example, Fig. 2.3 shows the equivalent MRF representation of the BN from Fig. 2.2.

One of the benefits of this representation is that MRFs offer an easy way of testing the conditional independence properties between variables by simple graph separation.

The Conditional Independence Property

We will refer to the example in Fig. 2.4 where a MRF is used to show the conditional independence (CI) property of a set of random variables $\mathbf{x}_A$, $\mathbf{x}_B$ and $\mathbf{x}_C$. Suppose that we want to search for a path connecting any node in $\mathbf{x}_A$ to any node in $\mathbf{x}_B$ when the common node subset $\mathbf{x}_C$ is removed from the graph. Since no such a path exists, we can assert that subsets $\mathbf{x}_A$ and $\mathbf{x}_B$ are conditionally independent if we know the subset $\mathbf{x}_C$. Subset $\mathbf{x}_C$ is called a *vertex separator* since it partitions the graph in two disconnected subgraphs $\mathbf{x}_A$ and $\mathbf{x}_B$. Then, in MRFs, the CI property is determined by simple graph separation ([7]).

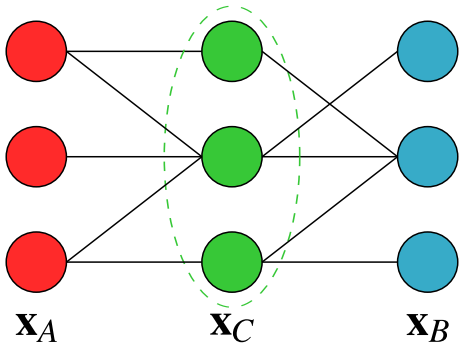


Figure 2.4: This example shows a MRF with no direct links from subset $\mathbf{x}_A$ to subset $\mathbf{x}_B$. This means that the conditional independence property $p(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_C)p(\mathbf{x}_B|\mathbf{x}_C)$ or any of its equivalents holds for any probability distribution described by this graph. Notice that we do not make any assumption about the nature of the distribution. Common separator nodes are surrounded by a dash dark line and depicted in green; nodes in $\mathbf{x}_A$ and $\mathbf{x}_B$ sets are shown in red and blue respectively.

Formally this is expressed by any of the following equivalent expressions:

$$p(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_C)p(\mathbf{x}_B|\mathbf{x}_C) \quad (2.6)$$

$$p(\mathbf{x}_A|\mathbf{x}_B, \mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_C) \quad (2.7)$$

$$p(\mathbf{x}_B|\mathbf{x}_A, \mathbf{x}_C) = p(\mathbf{x}_B|\mathbf{x}_C) \quad (2.8)$$

These equations will be applied in the explanation of the following sections and we will make a explicit use of this property in the distributed multi-robot SLAM algorithm presented in Chapter 6.

2.2.3 Factor Graphs

A factor graph [46] is a bipartite undirected graph which allows to represent factorized probability distributions. A factor graph has two kind of nodes, *variable nodes* which, in our context, can be either robot or landmark poses and *factor nodes* containing probabilistic relations between them. Concretely, each factor node contains one factor of Eq. 2.5 and is connected to all the variables that such factor depends on. The factor graph representation of the previous SLAM example is shown in Fig. 2.5. Note how the prior factor over the pose $\mathbf{x}_{R_0}$ is also included.

2.2.4 Pose Graphs

The pose graphs are a specific type of graphical model to represent the so-called *pose* SLAM problem. In contrast to the general approach to SLAM, the goal of pose SLAM

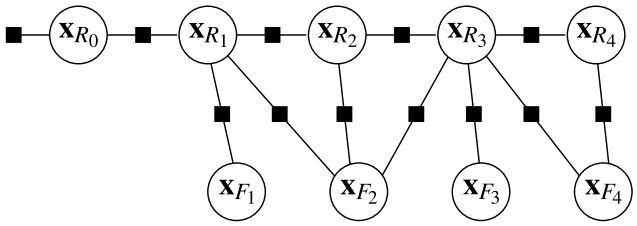


Figure 2.5: SLAM problem represented as a factor graph. Circles and squares correspond to variable nodes and factor nodes respectively.

is to compute the optimal robot trajectory whereas the map is maintained implicitly. As in the case of the general SLAM problem, the pose SLAM problem can be equally represented by MRFs or factor graphs. The pose graph contains only the robot poses and each edge/factor between two nodes encodes a spatial constraint between them. These constraints can be obtained directly from odometry measurements or indirectly when the robot observes the same part of the environment from two different poses and, based on this common observation, a "virtual" relative measurement between both positions can be determined. Once the optimal path is computed, a map estimate can be recovered by solving an instance of mapping with known poses (Eq. 2.1).

Figure 2.6 shows the pose graph simplification of our current example. Note how the edge/factor between non consecutive poses is derived from the observation of a common landmark.

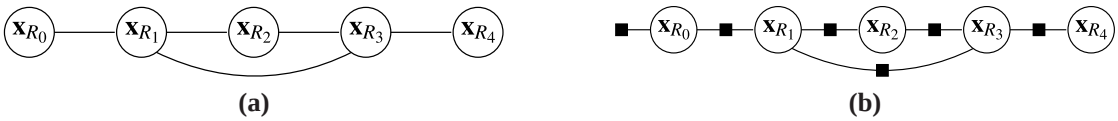


Figure 2.6: Pose graph SLAM problem represented as a MRF (a) and a factor graph (b).

2.3 Map Representations

A variety of map representations have been proposed over the years, which can be classified according to different criteria. We classify the kind of map depending on how the sensorial information is handled and integrated in the SLAM process. This leads to feature based maps which consider unique and distinguishable elements of the environment and dense maps which use raw data. The choice of the map representation is related to the task to be performed and to the algorithms and methods used to address the problem.

2.3.1 Feature Maps

In feature-based SLAM, the map of the environment is characterized by a set of *features* or *landmarks*. They are representative elements of the environment and its extraction depends on the kind of sensor used and the structure of the environment. For example, in the 2D case, usual features obtained from laser scans are segments, corners or even simpler elements, like points. In the 3D case, some common visual features extracted from images are the Speeded Up Robust Features (SURF), the Scale Invariant Feature Transform (SIFT) and the Features from Accelerated Segment Test (FAST). As downsides, feature detection can be computationally expensive and assumes the presence of outstanding features in the environment. Furthermore, this abstraction of the environment may produce a loss of the information available which difficults the data association process (i.e., determining correspondences between observations and previously mapped landmarks) in unstructured environments.

The state of a feature map can be described by a vector $\mathbf{x}_{\mathcal{F}} = \{\mathbf{x}_{F_1}, ..., \mathbf{x}_{F_n}\}$ containing the location of the n map features $\mathcal{F} = \{F_1, ..., F_n\}$. Furthermore, the assumption of the map being affected by Gaussian error requires the maintainance of the uncertainty estimation over time by means of a covariance matrix. Then, the uncertainty is maintained not only for each individual feature but also about their correlations and inter-dependencies, an important aspect to obtain consistent maps [17].

The cost of updating the map covariance matrix is quadratic in the size of the state vector which grows each time a new feature is detected and added in it. This makes the map maintainance intractable in large-scale scenarios. In order to cope with this issue, submapping techniques were proposed [89, 25, 40, 70], where the mapping problem is solved for local maps of bounded size which are later fused to create a global map.

As example, Fig. 2.7 shows a 2D segment-based map of our Robotics Lab at the Aragon Institute of Engineering Research of University of Zaragoza.

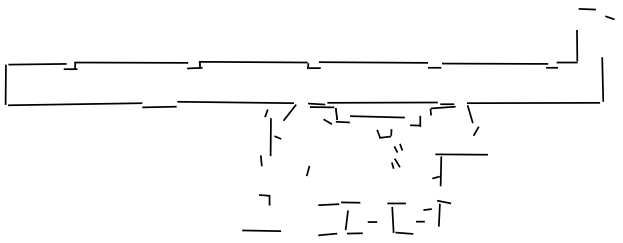


Figure 2.7: Example of segment-based feature map representation of the Robotics Lab at University of Zaragoza.

2.3.2 Dense Maps

In contrast to feature-based maps, dense maps are constructed using all raw data gathered from sensors, thus, offering a more detailed reconstruction of the environment. This fact makes dense maps suitable for both structured and unstructured environments. Data association is carried out by *registering* (aligning, also referred as scan matching in the 2D case) the raw measurements. The two common types of dense maps are grid maps and pose graphs, which are introduced below:

Grid maps

An occupancy grid map [59] partitions the environment into cells where each cell holds the probability of being occupied. The number of cells in the grid depends on the desired accuracy which is achieved by setting the size of the cells. The size of the grid affects the updating cost and memory consumption, then, grid-based algorithms should trade-off accuracy and online performance.

Since most widely used path planning and obstacle avoidance techniques are grid-based, these maps allow the use of a unique representation for SLAM and navigation purposes.

Pose graphs

Pose graphs have experienced a considerable attention in the last years thanks to their simplicity to represent the full SLAM problem.

As we introduced in the previous section, pose graphs offer a compact representation of the environment, where the map is summarized to the robot trajectory. Each node contains a robot location together with a measurement obtained at that position. Then, once the optimal trajectory is obtained, the measurements are represented with respect to the computed poses. As a drawback, raw measurements have to be stored at each node, then, this information has to be efficiently managed to avoid high memory consumption. This aspect can be particularly relevant when dealing with point clouds in dense visual SLAM.

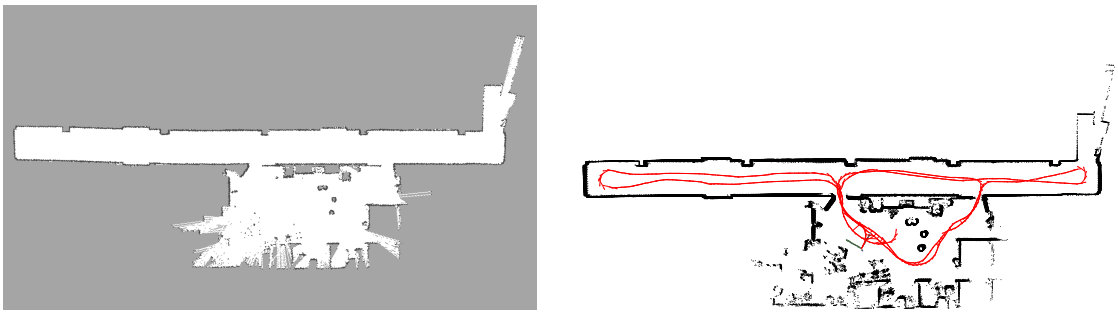
Figure 2.8 shows dense representations of the same feature map of Fig. 2.7.

2.4 Filtering Approaches

2.4.1 SLAM as a Recursive Bayesian Filtering Problem

In the filtering context, the current robot position $\mathbf{x}_{R_k}$ and the map $\mathbf{m}$ is computed given the history of sensor observations $\mathbf{z}_{1:k}$ and control inputs $\mathbf{u}_{1:k}$ up to time step k.

$$p(\mathbf{x}_{R_k}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) \tag{2.9}$$



(a) **(b)**

Figure 2.8: Example of dense map representations of the Robotics Lab at University of Zaragoza. (a) Grid map representation of the environment. Here, white cells are free of obstacles, black are occupied cells and grey cells denote unexplored area. (b) Pose graph obtained using the same raw data. The optimized robot trajectory is depicted in red where black dots are the laser scans acquired at each robot position

This probability is obtained recursively, based on the previously computed state at time $k-1$, that is $p(\mathbf{x}_{R_{k-1}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1})$.

Equation 2.9 is factorized by using Bayes' rule as,

$$p(\mathbf{x}_{R_k}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \eta\, p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}, \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) p(\mathbf{x}_{R_k}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) \tag{2.10}$$

where η is a normalizing factor that ensures Eq. 2.10 represents a valid probability distribution.

Under the Markov assumption, the current measurement $\mathbf{z}_k$ is conditionally independent of the past measurements $\mathbf{z}_{1:k-1}$ and control inputs $\mathbf{u}_{1:k}$ given the current state $\mathbf{x}_{R_k}$ and the map $\mathbf{m}$. This property can be deduced from Fig. 2.1, where $\mathbf{z}_k$ is only directly related to $\mathbf{x}_{R_k}$ and $\mathbf{m}$, therefore, the first term on Eq. 2.10 is simplified as,

$$p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}, \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) = p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}) \tag{2.11}$$

Following with the derivation, the law of total probability is applied in the second term of Eq. 2.10 to consider the state at previous step $k-1$. Then,

$$p(\mathbf{x}_{R_k}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) = \int_{\mathbf{x}_{R_{k-1}}} p(\mathbf{x}_{R_k} | \mathbf{x}_{R_{k-1}}, \mathbf{m}, \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) p(\mathbf{x}_{R_{k-1}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) d\mathbf{x}_{R_{k-1}} \tag{2.12}$$

which, again, using the Markov property, is reduced to

$$p(\mathbf{x}_{R_k}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) = \int_{\mathbf{x}_{R_{k-1}}} p(\mathbf{x}_{R_k} | \mathbf{x}_{R_{k-1}}, \mathbf{u}_k) p(\mathbf{x}_{R_{k-1}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_{R_{k-1}} \tag{2.13}$$

Finally, applying Eqs. 2.11 and 2.13 on Eq. 2.10 we obtain,

$$p(\mathbf{x}_{R_k}, \mathbf{m}|\mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \eta\, p(\mathbf{z}_k|\mathbf{x}_{R_k}, \mathbf{m}) \cdot \int_{\mathbf{x}_{R_{k-1}}} p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) p(\mathbf{x}_{R_{k-1}}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_{R_{k-1}} \tag{2.14}$$

As it can be observed, the problem is formulated in terms of the *measurement model* $p(\mathbf{z}_k|\mathbf{x}_{R_k}, \mathbf{m})$, the *transition model* $p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k)$ and the previous knowledge $p(\mathbf{x}_{R_{k-1}}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1})$.

Tipically, these models are approximated by Gaussians,

$$p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) \propto \exp\left\{-\frac{1}{2}(\mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k})^T \mathbf{Q}_{k-1}^{-1} (\mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k})\right\} \tag{2.15}$$

$$p(\mathbf{z}_k|\mathbf{x}_{R_k}, \mathbf{m}) \propto \exp\left\{-\frac{1}{2}(\mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k)^T \mathbf{R}_k^{-1} (\mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k)\right\} \tag{2.16}$$

where $\mathbf{f}$ and $\mathbf{h}$ are the transition and measurement functions, affected by additive zero-mean Gaussian noises with covariances $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$ respectively.

Usually, Eq. 2.14 is solved in two steps, the prediction and the update step. In the prediction step we compute:

$$p(\mathbf{x}_{R_k}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) = \int_{\mathbf{x}_{R_{k-1}}} p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) p(\mathbf{x}_{R_{k-1}}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_{R_{k-1}} \tag{2.17}$$

accounting for the transition model $p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}}, \mathbf{u}_k)$ and the estimate at previous step $p(\mathbf{x}_{R_{k-1}}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1})$. Notice how, obeying the filtering paradigm, previous robot position $\mathbf{x}_{R_{k-1}}$ is *marginalized* out. At the update step, the probability distribution $p(\mathbf{x}_{R_k}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k})$ is updated taking into account the sensor model described by Eq. 2.16 and normalized with the constant η to ensure that the result is a valid probability distribution.

$$p(\mathbf{x}_{R_k}, \mathbf{m}|\mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \eta\, p(\mathbf{z}_k|\mathbf{x}_{R_k}, \mathbf{m}) p(\mathbf{x}_{R_k}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) \tag{2.18}$$

2.4.2 Extended Kalman Filter

The Kalman Filter (KF) is the optimal estimator to recursively solve Eq. 2.14. It is based on the assumptions that the system is linear with zero-mean Gaussian noise sequences. However, the SLAM motion and measurement processes are nonlinear. The Extended Kalman Filter (EKF) is the KF nonlinear extension that allows to apply its equations to nonlinear models.

The state of the system $\mathbf{x}_k = (\mathbf{x}_{R_k}, \mathbf{m})^T$ is described by a normally distributed random vector with mean $\hat{\mathbf{x}}_k$ and covariance matrix $\mathbf{P}_k$,

$$\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, \mathbf{P}_k)$$

The transition and measurement models (Eqs. 2.15 and 2.16) let us formulate the SLAM problem as the nonlinear system,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_{k-1} \quad ; \quad \mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \tag{2.19}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad ; \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \tag{2.20}$$

where $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are the motion and observation noises respectively.

The application of the EKF equations requires a linearization of the nonlinear process and measurement functions around the best estimate available. Then, the EKF SLAM prediction and update steps are:

Prediction

A linearization of Eq. 2.19 around the estimated $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{u}}_k$ give us,

$$\mathbf{x}_k \simeq \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_k) + \mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{w}_{k-1} \tag{2.21}$$

where

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_k} \tag{2.22}$$

Then, the EKF prediction equations are,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_k) \tag{2.23}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.24}$$

Update

Similarly, Eq. 2.20 is linearized around the predicted estimate $\hat{\mathbf{x}}_{k|k-1}$,

$$\mathbf{z}_k \simeq \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) + \mathbf{v}_k \tag{2.25}$$

where

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}} \tag{2.26}$$

It allows to compute the *innovation* ν_k, its covariance $\mathbf{S}_k$ and the filter gain $\mathbf{K}_k$ based on the current observations,

$$\nu_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) \tag{2.27}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{2.28}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{2.29}$$

Finally, the state of the system is updated by,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\nu_k \tag{2.30}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{2.31}$$

2.4.3 Extended Information Filter

The Extended Information Filter (EIF) is the algebraic equivalent of the EKF, also known as its *inverse covariance* form where the state of the system is represented by the information vector $\mathbf{i}_k$ and matrix $\mathbf{I}_k$, which are defined in terms of the mean $\hat{\mathbf{x}}_k$ and covariance matrix $\mathbf{P}_k$ as,

$$\mathbf{I}_k = \mathbf{P}_k^{-1} \tag{2.32}$$

$$\mathbf{i}_k = \mathbf{P}_k^{-1}\hat{\mathbf{x}}_k = \mathbf{I}_k\hat{\mathbf{x}}_k \tag{2.33}$$

The prediction and update equations of the EIF can be derived algebraically from the EKF equations in the following way:

Prediction

The prediction of the state in the information form is obtained from the application of the EKF prediction equations on the definitions of the information vector and matrix. Then,

$$\mathbf{I}_{k|k-1} = (\mathbf{F}_{k-1}\mathbf{I}_{k-1}^{-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1})^{-1} \tag{2.34}$$

$$\mathbf{i}_{k|k-1} = \mathbf{I}_{k|k-1}\hat{\mathbf{x}}_{k|k-1} \tag{2.35}$$

Note that to obtain $\mathbf{i}_{k|k-1}$ we need to recover $\hat{\mathbf{x}}_{k-1}$ by solving the linear system,

$$\mathbf{I}_{k-1}\hat{\mathbf{x}}_{k-1} = \mathbf{i}_{k-1} \tag{2.36}$$

to propagate the state through the nonlinear function,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_k) \tag{2.37}$$

Update

The information matrix update equation is computed from the inverse of the expression to compute the covariance matrix in Eq. 2.31 and using the matrix inversion lemma $(A+BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B+C^{-1})^{-1}DA^{-1}$.

$$\mathbf{I}_k = \mathbf{P}_k^{-1} \tag{2.38}$$

$$= (\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\mathbf{H}_k\mathbf{P}_{k|k-1})^{-1} \tag{2.39}$$

$$= \mathbf{I}_{k|k-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k \tag{2.40}$$

The derivation of the information vector update equation makes use of the alternative form of the Kalman gain [6],

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}^T\mathbf{R}^{-1} \tag{2.41}$$

Then, applying the EKF state update equation 2.30 on the definition,

$$\begin{aligned}
\mathbf{i}_k &= \mathbf{P}_k^{-1}\hat{\mathbf{x}}_k && (2.42)\\
&= \mathbf{P}_k^{-1}(\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}))) && (2.43)\\
&= \mathbf{P}_k^{-1}\hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_k^{-1}\mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})) && (2.44)\\
&= (\mathbf{I}_{k|k-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k)\hat{\mathbf{x}}_{k|k-1} + \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})) && (2.45)\\
&= \mathbf{i}_{k|k-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) && (2.46)
\end{aligned}$$

Summing up, the EIF update equations are,

$$\begin{aligned}
\mathbf{I}_k &= \mathbf{I}_{k|k-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k && (2.47)\\
\mathbf{i}_k &= \mathbf{i}_{k|k-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) && (2.48)
\end{aligned}$$

2.4.4 EKF vs. EIF

Although both EKF and EIF provide identical results assuming the measurement and motion jacobians are evaluated at the same linearization points [61] (except for numerical and rounding errors in the operations involved) their prediction and update stages are not computationally equivalent.

On the one hand, EKF prediction is efficient whereas the update is more expensive. On the other hand, the prediction step of the EIF involves solving a linear system to recover the mean (Eq. 2.36). Fortunately, the information matrix is approximately sparse (and exactly sparse when robot positions are not marginalized out) which allows the use of numerical techniques to speed up the computations. Aditionally, its update step is carried out by simply adding the new measurement information which makes the filter inherent to decouple multi-robot processes.

Furthermore, each filter offers different possibilities in terms of their initialization. Whereas we can represent the absence of uncertainty in the EKF by setting $\mathbf{P}_0 = \mathbf{0}$, the EIF allows to initialize the estimation process in the case of lack of prior information with $\mathbf{I}_0 = \mathbf{0}$.

2.5 Maximum Likelihood Approaches

Maximum Likelihood approaches address the SLAM problem as a whole, maximizing the posterior probability of the entire robot trajectory and the map based on the history of measurements and control inputs,

$$p(\mathbf{x}_{R_{0:K}}, \mathbf{m} | \mathbf{z}_{1:K}, \mathbf{u}_{1:K}) \quad (2.49)$$

Following a similar derivation as in the filtering case we obtain:

$$
\begin{aligned}
&p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \\
&= \eta\, p(\mathbf{z}_k | \mathbf{x}_{R_{0:k}}, \mathbf{m}, \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) && (2.50)\\
&= \eta\, p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}) p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) && (2.51)\\
&= \eta\, p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}) p(\mathbf{x}_{R_k} | \mathbf{x}_{R_{0:k-1}}, \mathbf{m}, \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) p(\mathbf{x}_{R_{0:k-1}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) && (2.52)\\
&= \eta\, p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}) p(\mathbf{x}_{R_k} | \mathbf{x}_{R_{k-1}}, \mathbf{u}_k) p(\mathbf{x}_{R_{0:k-1}}, \mathbf{m} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}) && (2.53)
\end{aligned}
$$

This is a recursive solution, which can also be expressed in closed-form as

$$
p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \eta\, p(\mathbf{x}_{R_0}) p(\mathbf{m}) \prod_k p(\mathbf{x}_{R_k} | \mathbf{x}_{R_{k-1}}, \mathbf{u}_k) p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}) \tag{2.54}
$$

where $p(\mathbf{x}_{R_0})$ is the prior over the first pose $\mathbf{x}_{R_0} = (0,0,0)^T$, fixed at the origin of the global reference frame

$$
p(\mathbf{x}_{R_0}) \propto \exp\left\{ -\frac{1}{2} \mathbf{x}_{R_0}^T \mathbf{\Omega}_{R_0} \mathbf{x}_{R_0} \right\} \tag{2.55}
$$

and $p(\mathbf{m})$, the prior knowledge about the map, which is usually considered unknown and subsumed into the normalizer η. As a result, we obtain a factorized posterior of the full SLAM problem which is usually represented by a factor graph.

The Maximum a Posteriori (MAP) estimation of the robot positions $\mathbf{x}_{R_{0:k}}$ and the map $\mathbf{m}$ is given by:

$$
\begin{aligned}
\mathbf{x}^*_{R_{0:k}}, \mathbf{m}^* &= \underset{\mathbf{x}_R, \mathbf{m}}{\operatorname{argmax}} \log p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) && (2.56)\\
&= \underset{\mathbf{x}_R, \mathbf{m}}{\operatorname{argmin}} -\log p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) && (2.57)
\end{aligned}
$$

Then, taking the logarithm in Eq. 2.54,

$$
\begin{aligned}
\log p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \text{const} &+ \log p(\mathbf{x}_{R_0}) \\
&+ \sum_k \left(\log p(\mathbf{x}_{R_k} | \mathbf{x}_{R_{k-1}}, \mathbf{u}_k) + \log p(\mathbf{z}_k | \mathbf{x}_{R_k}, \mathbf{m}) \right)
\end{aligned} \tag{2.58}
$$

Now, applying Eqs. 2.55, 2.15 and 2.16 we obtain,

$$
\begin{aligned}
\log p(\mathbf{x}_{R_{0:k}}, \mathbf{m} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \text{const} &- \frac{1}{2} \mathbf{x}_{R_0}^T \mathbf{\Omega}_{R_0} \mathbf{x}_{R_0}^T \\
&- \frac{1}{2} \sum_k \{ (\mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k})^T \mathbf{Q}_{k-1}^{-1} (\mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k}) \\
&+ (\mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k)^T \mathbf{R}_k^{-1} (\mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k) \}
\end{aligned} \tag{2.59}
$$

Finally, we substitute this expresion into Eq. 2.57 which results in,

$$\mathbf{x}^*_{R_{0:k}}, \mathbf{m}^* = \underset{\mathbf{x}_R, \mathbf{m}}{\operatorname{argmin}} \sum_k \{(\mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k})^T \mathbf{Q}_{k-1}^{-1} (\mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k}) + (\mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k)^T \mathbf{R}_k^{-1} (\mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k)\} \tag{2.60}$$

which can be simplified by introducing the error functions $\mathbf{e}_k^{\text{odom}} = \mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_k) - \mathbf{x}_{R_k}$ and $\mathbf{e}_k^{\text{obs}} = \mathbf{h}(\mathbf{x}_{R_k}, \mathbf{m}) - \mathbf{z}_k$ as,

$$\mathbf{x}^*_{R_{0:k}}, \mathbf{m}^* = \underset{\mathbf{x}_R, \mathbf{m}}{\operatorname{argmin}} \sum_k \{(\mathbf{e}_k^{\text{odom}})^T \mathbf{Q}_{k-1}^{-1} \mathbf{e}_k^{\text{odom}} + (\mathbf{e}_k^{\text{obs}})^T \mathbf{R}_k^{-1} \mathbf{e}_k^{\text{obs}}\} \tag{2.61}$$

Then, SLAM is formulated as a nonlinear weighted least squares problem which can be solved by minimization methods such as Gauss-Newton or Levenberg-Marquardt. In a pose SLAM problem, the minimization problem of Eq. 2.61 is solved only for the trajectory $\mathbf{x}_{R_{0:k}}$, afterwards, the map $\mathbf{m}$ can be reconstructed from the set of poses based on Eq. 2.1. From this point we turn our attention to how this optimization can be carried out for a pose SLAM problem using pose graphs.

2.5.1 SLAM as a nonlinear Least Squares Minimization Problem

Equation 2.57 formulates the full SLAM problem as a MAP estimation problem which can be reduced to determine the positions of a robot along its trajectory $\mathbf{x}_{R_{0:K}} = (\mathbf{x}_{R_0}, ..., \mathbf{x}_{R_K})^T$ and postponing the estimation of the map $\mathbf{m}$ to when the optimal trajectory is computed. This pose SLAM problem can be represented by a pose-graph as shown in Fig. 2.6. Let us consider a pose-graph where the robot trajectory $\mathbf{x}_{R_{0:K}}$ is represented by a set of nodes $\mathbf{x} = (\mathbf{x}_0, ..., \mathbf{x}_K)$, containing a robot position together with a measurement (image or laser scan) acquired at each position and each edge encodes a measurement $\mathbf{z}$ about the relative transformation of its two connected nodes. These measurements can be computed directly from the odometry or indirectly by computing relative transformations from the observations e.g. by using a visual place recognition system or scan matching. During the process of graph construction, usually referred as *front-end*, constraints between consecutive nodes and loop closures (constraints between non-consecutive nodes introduced during place revisiting) are determined. In other words, in a pose-graph SLAM approach, we assume that our robot is equipped with a simple sensor capable of measuring the transformation between robot locations in the trajectory when they are either temporally or spatially close.

Assuming that the measurements are affected by Gaussian noise, a constraint between two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ is characterized by its mean $\mathbf{z}_{ij}$ and its information matrix $\mathbf{\Omega}_{ij}$. Then, given a pair of nodes $\mathbf{x}_i, \mathbf{x}_j$ and a measurement $\mathbf{z}_{ij}$ connecting both nodes, it is possible to compute the error committed in the estimation:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \hat{\mathbf{z}}_{ij} - \mathbf{z}_{ij} \tag{2.62}$$

where $\hat{\mathbf{z}}_{ij} = \mathbf{g}(\mathbf{x}_i, \mathbf{x}_j)$ is the expected measurement given the current configuration of nodes $\mathbf{x}_i, \mathbf{x}_j$. In the 2D case $\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j)$ computes the position and the orientation of $\mathbf{x}_j$ in the frame of $\mathbf{x}_i$.

Let $\mathcal{C} = \{\langle i, j \rangle\}$ be the set of pairs of nodes for which a measurement $\mathbf{z}_{ij}$ exists. Note that this measurement represents indistinctly an odometry or sensor observation with inverse covariance $\mathbf{\Omega}_{ij}$. Based on Eq. 2.61, the goal of the pose-graph approach is to find the configuration of nodes $\mathbf{x}^*$ which minimizes the overall error:

$$F(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \underbrace{\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)}_{F_{ij}} \tag{2.63}$$

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} F(\mathbf{x}) \tag{2.64}$$

This constitutes a nonlinear least-squares minimization problem which does not admit a closed solution. Instead, if a good initial guess is available, the problem can be addressed by numerical methods such as Gauss-Newton or Levenberg-Marquardt which iteratively approximate the solution by carrying out local linearizations. Recent implementations (also called *back-ends*) to solve this graph optimization problem like **g^{2}o** [47] or iSAM [42] require a time that depends on the number of edges, and their success in finding the correct solution is affected by the initial guess available to the system.

A general procedure to solve Eq. 2.64 is presented in [32] and described below. Let us first denote, for simplicity's sake, the sensor model and error function as

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{e}_{ij}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \mathbf{z}_{ij} \tag{2.65}$$

The sensor model is, in general, a nonlinear function. However, it can be approximated around an initial guess $\check{\mathbf{x}}$ of the robot's trajectory by a first-order Taylor expansion:

$$\mathbf{g}(\mathbf{x}) \simeq \mathbf{g}(\check{\mathbf{x}}) + \left. \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\check{\mathbf{x}}} (\mathbf{x} - \check{\mathbf{x}}) \tag{2.66}$$

which, alternatively, can be expressed in terms of the *increments* $\Delta\mathbf{x}$:

$$\mathbf{g}(\check{\mathbf{x}} + \Delta\mathbf{x}) \simeq \mathbf{g}(\check{\mathbf{x}}) + \mathbf{J}_{ij} \Delta\mathbf{x} \tag{2.67}$$

with

$$\mathbf{J}_{ij} = \left. \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\check{\mathbf{x}}} \tag{2.68}$$

Applying 2.67 in 2.65, we obtain a linearized version of the error function:

$$\mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \mathbf{z}_{ij} \tag{2.69}$$

$$\simeq \mathbf{g}(\check{\mathbf{x}}) + \mathbf{J}_{ij} \Delta\mathbf{x} - \mathbf{z}_{ij} \tag{2.70}$$

$$= \mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta\mathbf{x} \tag{2.71}$$

where $\mathbf{e}_{ij} = \mathbf{e}_{ij}(\check{\mathbf{x}}_i, \check{\mathbf{x}}_j)$.

Substituting Eq. 2.71 in the error terms F_{ij} of Eq. 2.63 we obtain,

$$F_{ij}(\check{\mathbf{x}} + \Delta\mathbf{x}) = \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta\mathbf{x})^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta\mathbf{x}) \tag{2.72}$$

$$\simeq (\mathbf{e}_{ij} + \mathbf{J}_{ij}\Delta\mathbf{x})^T \mathbf{\Omega}_{ij} (\mathbf{e}_{ij} + \mathbf{J}_{ij}\Delta\mathbf{x}) \tag{2.73}$$

$$= \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij} + 2 \underbrace{\mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}^T} \Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\mathbf{J}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}} \Delta\mathbf{x} \tag{2.74}$$

Now, we can write a linear approximation of the function $F(\mathbf{x})$ in Eq. 2.63 as,

$$F(\check{\mathbf{x}} + \Delta\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} F_{ij}(\check{\mathbf{x}} + \Delta\mathbf{x}) \tag{2.75}$$

$$\simeq \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij} + 2\mathbf{b}_{ij}\Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H}_{ij} \Delta\mathbf{x} \tag{2.76}$$

$$= \underbrace{\sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}}_{c} + 2 \underbrace{\sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{b}_{ij}^T}_{\mathbf{b}^T} \Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{H}_{ij}}_{\mathbf{H}} \Delta\mathbf{x} \tag{2.77}$$

With this procedure we have obtained a quadratic form in $\Delta\mathbf{x}$ of the function $F(\mathbf{x})$. We get the values for $\Delta\mathbf{x}$ that minimizes function 2.77 by setting its first derivative equal to zero:

$$\frac{\partial(\Delta\mathbf{x}^T \mathbf{H} \Delta\mathbf{x} + 2\mathbf{b}^T \Delta\mathbf{x} + c)}{\partial \Delta\mathbf{x}} = 2\mathbf{H}\Delta\mathbf{x} + 2\mathbf{b} = \mathbf{0} \tag{2.78}$$

Then, the problem is reduced to find a solution of the linear system:

$$\mathbf{H}\Delta\mathbf{x}^* = -\mathbf{b} \tag{2.79}$$

where the computed increments are applied to the initial guess as,

$$\mathbf{x}^* = \check{\mathbf{x}} + \Delta\mathbf{x}^* \tag{2.80}$$

The matrix $\mathbf{H}$ is the information matrix of the system, since it is obtained by projecting the measurement error in the space of the trajectories via the Jacobians. It is sparse by construction, having non-zeros only between poses connected by a constraint. Its number of non-zero blocks is twice the number of constraints plus the number of nodes. This allows us to solve the linear system in Eq. 2.79 with efficient approaches like sparse Cholesky factorization.

The Gauss-Newton algorithm iteratively apply previous equations until some convergence criterion is reached. The current solution can be used as initial guess of the next iteration. However, the convergence is not guaranteed and it might happen that one iteration of Gauss-Newton results in a worse solution than the previous one. The

Levenberg-Marquardt algorithm tries to overcome this problem by introducing a damping factor λ which allows to recover from wrong steps. Then, instead of Eq. 2.79, the Levenberg-Marquardt method solves a damped version of it:

$$(\mathbf{H} + \lambda \mathbf{I})\Delta\mathbf{x}^* = -\mathbf{b} \tag{2.81}$$

Intuitively, if $\lambda \to \inf$, $\Delta\mathbf{x}^* \to \mathbf{0}$. So, the larger is λ, the smaller are the increments. This parameter is adjusted at each iteration based on the error of the new configuration. If the error is reduced, the damping factor is decreased. Otherwise, the solution is reverted and λ is increased.

Least Squares on a Manifold

Previous section describes a general solution to the least squares minimization problem defined in equation 2.64. However, the derivations and operations involved are only valid under the assumption that the state space is Euclidean which, in general, is not the case in the SLAM context.

For example, in the 2D case, the robot pose (x, y) spans over a Euclidean domain whereas its orientation spans over the non-Euclidean rotation group SO(2). A common approach is to represent the orientation using an angle $\theta \in [-\pi, \pi)$; however, this minimal representation suffers from singularities when operating in the vicinity of $\pm\pi$ and requires angle renormalization. A different approach is, for example, the use of rotation matrices, which is an over-parameterized representation (i.e. it is a 2x2 matrix to represent one angle). However, directly applying Eq. 2.80 to rotation matrices can break the orthogonality constraint, leading to non-valid solutions. This problem becomes even more complex in the 3D case, where orientations are in SO(3) and can be described using an over-parameterization (e.g, rotation matrices or quaternions) or with a minimal representation (e.g, Euler angles, which present a singularity known as gimbal lock).

In order to take advantage of the benefits of both representations, we can consider the state space as a *smooth manifold* [53] which is introduced in the SLAM context [36, 33] to increase optimization robustness. A manifold is a topological space which locally behaves like a Euclidean space. Then, the intuition behind manifolds is to work globally with an over-parameterized representation and locally with its minimal form. As proposed by [37], we can define the following two operators to obtain a mapping between both representations:

$$\boxplus : M \times \mathbb{R}^m \to M \tag{2.82}$$

$$\boxminus : M \times M \to \mathbb{R}^m \tag{2.83}$$

Here, m is the dimension of the manifold M. The operator $\boxplus$ applies a small perturbation in the Euclidean space to the manifold, whereas $\boxminus$ computes the perturbation between to states in the manifold.

In order to apply the concept of manifolds into our optimization problem, we can use an over-parameterized representation for the global state $\mathbf{x}$ and a minimal representation for the increments $\Delta\mathbf{x}$. Then, given the current estimate $\check{\mathbf{x}}$ we can move in the manifold to approximate the solution by iteratively mapping the Euclidean increments into the global state using the $\boxplus$ operator: $\Delta\mathbf{x} \mapsto \mathbf{x} \boxplus \Delta\mathbf{x}$. Moreover, since we assume these increments are small, they are far from singularities.

Using this operator, we can define a new error function as

$$\tilde{\mathbf{e}}_{ij}(\check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}) = \hat{\mathbf{z}}_{ij} \boxminus \mathbf{z}_{ij} \tag{2.84}$$

$$= \mathbf{g}(\check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}) \boxminus \mathbf{z}_{ij} \tag{2.85}$$

As before, we can compute the Taylor expression of the new error funtion under a small perturbation of the state variables around a linearization point $\check{\mathbf{x}}$ as

$$\tilde{\mathbf{e}}_{ij}(\check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}) \simeq \tilde{\mathbf{e}}_{ij} + \tilde{\mathbf{J}}_{ij}\Delta\tilde{\mathbf{x}} \tag{2.86}$$

with

$$\tilde{\mathbf{J}}_{ij} = \left.\frac{\partial(\mathbf{g}(\check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}) \boxminus \mathbf{z}_{ij})}{\partial\Delta\tilde{\mathbf{x}}}\right|_{\Delta\tilde{\mathbf{x}}=\mathbf{0}} \tag{2.87}$$

With a straightforward extension of the notation, we can insert Eq. 2.86 in Eq. 2.72 leading to the linear system:

$$\tilde{\mathbf{H}}\Delta\tilde{\mathbf{x}}^* = -\tilde{\mathbf{b}} \tag{2.88}$$

Note that the Hessian $\tilde{\mathbf{H}}$ of the manifold problem no longer represents the information matrix of the trajectories but of the trajectory increments $\Delta\tilde{\mathbf{x}}$.

Once we find a solution for the increments $\Delta\tilde{\mathbf{x}}^*$ computed in the local Euclidean surroundings of the initial guess $\check{\mathbf{x}}$, they are re-mapped into the original over-parameterized space as

$$\mathbf{x}^* = \check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}^* \tag{2.89}$$

Robust optimization using condensed measurements

We have seen that the least squares SLAM problem can be solved by iterative approaches like Gauss-Newton or Levenberg-Marquardt. However, their success in finding the global minimum heavily depends on how good is the initial guess and the smoothness of the sensor model.

In this section we briefly describe the optimization approach presented in [31] which provides a procedure to solve a factor graph even if the initial guess is poor or in the case of highly nonlinear measurement functions. The approach exploits the spatial and temporal locality properties of factor graphs present in the SLAM problem. This methodology is the basis of the multi-robot SLAM approach that will be presented in Chapter 7.

The overall procedure is illustrated in Fig. 2.9 applied on a simple landmark-based SLAM problem. The original graph is partitioned in local sub-graphs. These local maps are related to each other through *shared* variables belonging to more than one local map, which are vertex separators. These smaller local maps are easier to solve even under hard conditions and a reasonable solution can be obtained using a direct method. To determine a unique solution we need to "fix" a *origin* (gauge) for each local map. After solving the local maps we obtain a Gaussian approximation of each variable within the local map, relative to its origin.

Then, we seek for a global alignment of the local maps that satisfies all the equality constraints induced by the shared variables. In order to preserve the structures of the local maps computed before, we replace each sub-graph with a simpler one which approximates the original local solution. This reduced problem is constructed by considering the origin and the shared variables of each local map and computing a set of *condensed* factors relating the origin to each shared variable, therefore obtaining a star-like topology of the original subproblem. These condensed factors summarize the relationship between a variable and the origin of the local map by considering all the measurements when optimizing the local map.

The procedure to determine the condensed factors is outlined below. Given the set of separators $\{\mathbf{x}_i\}$ and the origin $\mathbf{x}_g$ we can define a family of measurement functions

$$\mathbf{h}^{\text{typeOf}(\mathbf{x}_i)}(\mathbf{x}_g, \mathbf{x}_i) \overset{\text{def.}}{=} \mathbf{h}(\mathbf{x}_g, \mathbf{x}_i) \tag{2.90}$$

that depend on the *type* of the separators which can be either landmark poses or robot poses. The origin of the local maps is always selected to be a robot pose. Once we know the type of a factor, we need to compute the mean $\mathbf{z}_i$ and information matrix $\mathbf{\Omega}_i$ of the corresponding measurement. To this end, we recall (2.86) that relates measurement function and error vector through the $\boxminus$ operator: $\mathbf{e}_i(\mathbf{x}_g, \mathbf{x}_i) = \mathbf{h}(\mathbf{x}_g, \mathbf{x}_i) \boxminus \mathbf{z}_i$. The error function depends on the (known) measurement function $\mathbf{h}_i(\cdot)$ and on the unknown measurement $\mathbf{z}_i$. Since the error of the local solution is small, the measurement vector at the equilibrium can be approximated as:

$$\mathbf{z}_i = \mathbf{h}(\check{\mathbf{x}}_g, \check{\mathbf{x}}_i), \tag{2.91}$$

where $\check{\mathbf{x}}_g$ and $\check{\mathbf{x}}_i$ are the actual values of the origin and of a separator after solving the sub-problem.

To qualify the factors we still have to compute the information matrix $\mathbf{\Omega}_i$. Since the origin node is fixed, its covariance matrix is zero. Thus, only the marginal covariance of $\mathbf{x}_i$ contributes in determining $\mathbf{\Omega}_i$. Since we are working with a manifold formulation, we can extract the covariance matrices of the increments $\Delta\mathbf{x}_i$ from the corresponding blocks of the Hessian $\check{\mathbf{H}}$ (Eq. 2.88) of the system and remap them through the error function:

$$\mathbf{e}_i(\mathbf{x}_g, \mathbf{x}_i \boxplus \Delta\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_g, \mathbf{x}_i \boxplus \Delta\mathbf{x}_i) \boxminus \mathbf{z}_i. \tag{2.92}$$

We then remap the marginal covariance of $\Delta\mathbf{x}_i$ by using the unscented transform [41]. We extract a set of sigma points $\{\sigma^k_{\Delta\mathbf{x}_i}\}$ from the marginal covariance $\Sigma_{\Delta\mathbf{x}_i}$ of the increments $\Delta\mathbf{x}_i$ and we remap them through (2.92) as follows:

$$\sigma^k = \mathbf{e}_i(\mathbf{x}^*_g, \mathbf{x}^*_i \boxplus \sigma^k_{\Delta\mathbf{x}_i}). \tag{2.93}$$

We then compute $\mathbf{\Omega}_i$ by inverting the covariance matrix reconstructed from the projected sigma points.

Once we "condensed" the local maps, we assemble an approximation of the original global factor graph by combining all the newly computed factor graphs into a new sparser factor graph, whose solution is a global configuration of the origins and of the shared variables. Then, we can determine a good initial guess by arranging the local maps computed at the beginning of the procedure accordingly. At this point, an optimization which considers the original factors can further refine the approximated solution.

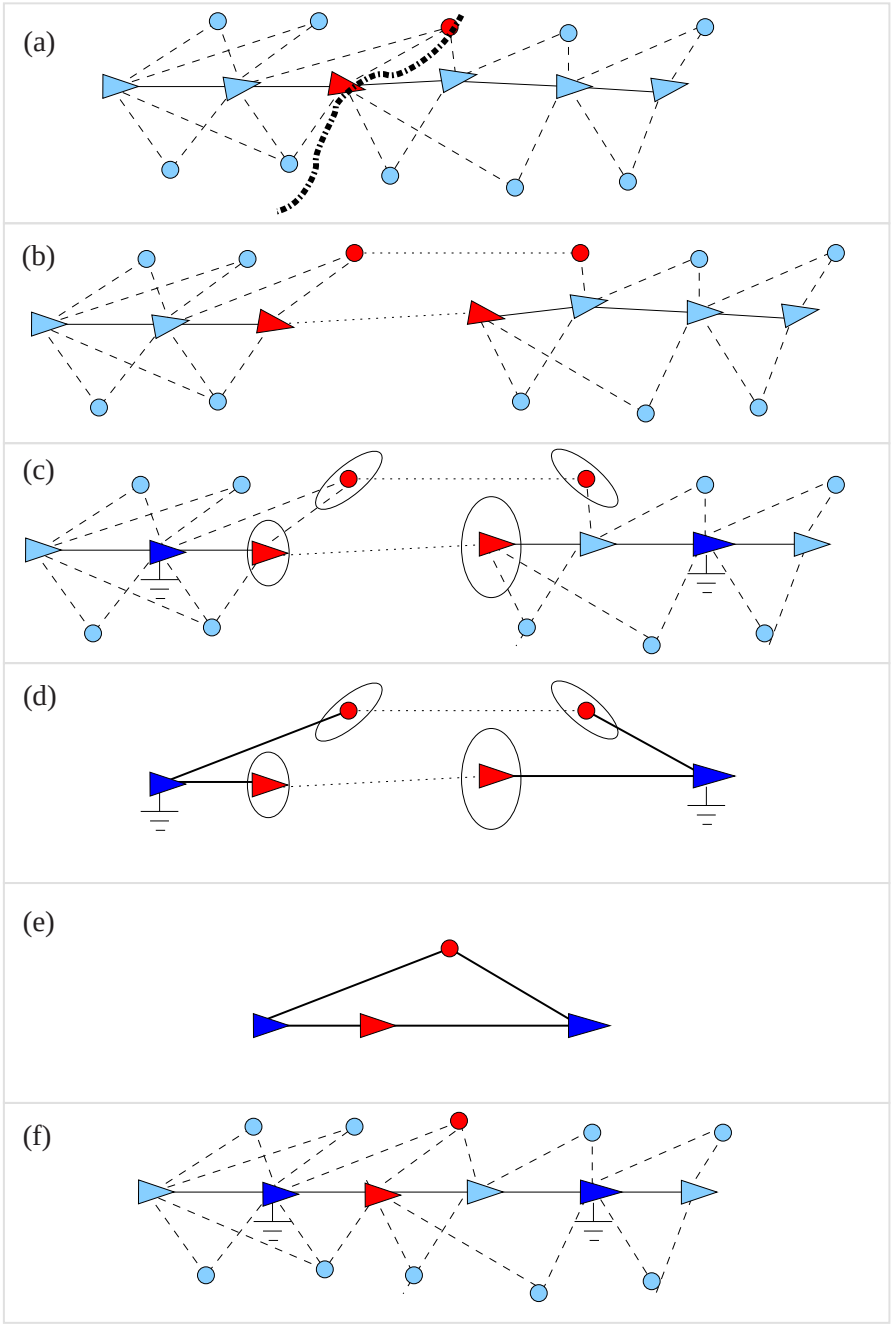


Figure 2.9: The overview of the optimization procedure on a simple landmark-based SLAM problem. Here we illustrate the factor graph by highlighting only the variables. The factors denote binary measurements and are encoded in the edges. (a),(b) The problem is partitioned into sub-graphs. The shared variables are in red, and dotted lines show the corresponding variables in different partitions. (c) We solve these problems independently with respect to their origins (dark blue), and we determine the marginal covariances of the shared variables. (d) We compute condensed factors connecting each shared variable to its origin. (e) We solve the complete problem on the condensed factors to determine the layout of the local maps (f).

Part I

SLAM for Robot Formations using Prior Information

Chapter 3

Robot Formations

The first step towards achieving an autonomous multi-robot system is the design of the motion control strategy to enable the robots to navigate in the environment and reach their goal. In this chapter, we first address the motion control of each vehicle by means of a motion generator which provides the input velocities to be applied to the robot. Then, we address the cooperative navigation for robot formations based on a spring-damper analogy which allows the formation to comply with the structure of the environment. We point out the difficulties that appear in formation control in the presence of noise which motivates the use of the probabilistic localization and mapping techniques presented along this thesis.

3.1 Motivation

The main distinguishing feature between a robot formation and a general multi-robot system is the navigation of the group maintaining a specific shape. As a consequence, formations require stronger coordination and communication among the robots.

There are a variety of applications which could benefit from the use of robot formations, i.e., situations in which the robots have to meet certain requirements regarding their relative distances and angles. Fig. 3.1 shows two applications of robot formations in emergency situations developed at the University of Zaragoza [74, 84]. For example, in case of an accident inside a tunnel (Fig. 3.1a), we can deploy a team of robots to inspect the area and identify potential hazards to humans. Under this circumstance, where there is no communications infrastructure, the distance the robots have to maintain is oriented to guarantee the connectivity with a base station, outside the tunnel. Then, the chain formation is deployed in such a way that each robot acts as a relay for the next one till the dangerous area is reached. Another possible application is the guidance of people towards the exit during a building evacuation (Fig. 3.1b). In this situation, the robots adapt their motion and shape to the people behaviour which can move inside a

safe area delimited by the formation.

(a) **(b)**

Figure 3.1: Example of applications of robot formations developed at University of Zaragoza. (a) Formation deployment inside a road tunnel. (b) A triangle-shaped robot formation guides a group of people towards the exit during a building evacuation.

Along the years, many approaches have been proposed to reach the formation control where the most commonly used are those based on virtual structures and leader-following behaviors.

In [55] the concept of virtual structure is introduced to maintain the geometric relationships among the robots. However, in this model, the formation moves quite rigidly along the individual trajectories computed for each robot and is not easily reconfigurable in case of obstacle presence. In leader-following approaches, the robot designated as the *leader* guides the formation to the goal while the other robots, i.e., the *followers*, try to keep a desired distance and angle with respect to the leader. This strategy simplifies the path planning problem which is addressed by the leader and the followers maintain the formation whenever possible, otherwise it is adapted to the structure of the scenario. In the worst case, the formation becomes a chain and the path computed by the leader is also feasible for the followers. Common implementations of leader-following approaches are vision-based leader tracking [22], [87] and spring-based systems [34],[56].

In this thesis we make use of our previous work [85] to control the robot formation based on a spring-damper system. It provides a flexible, smooth and environment compliant navigation by dynamically adapting the velocities of the robots and the formation's shape to the environment restrictions. In the rest of this chapter we describe this formation control strategy for the sake of completeness. We begin in section 3.2 by describing how to control each single robot and continue in section 3.3 by explaning the cooperative control of the formation. Additionally, we perform along this chapter a brief analysis of the formation behaviour under the influence of noise, (e.g., comming from

sensor readings or in the application of the input controls), pointing out the importance of a reliable localization to maintain the formation and to reach its goal. This motivates the localization and mapping techniques presented in subsequent chapters.

3.2 Single Robot Motion Control

The first step in order to achieve the formation control is to model how a single robot can move around smoothly and stably in the environment. By following a physical analogy, we can consider the robot movement originates from the application of a virtual force **F** on the robot. The source of such force can be, for example, the combination of an attractive force generated by the robot's goal and repulsive forces coming from obstacles.

The force **F** gives a magnitude F and direction of movement θ which need to be transformed into the desired control input $\mathbf{u} = (v\ \omega)^T$, the linear and angular velocities of the robot. To this end, we use the motion controller described in [2], which allows us to compute linear and angular velocities physically feasible for the robot. It is based on geometric and dynamic constraints of the robot, characterized by the differential equation:

$$\dot{\mathbf{u}} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{F} \tag{3.1}$$

where

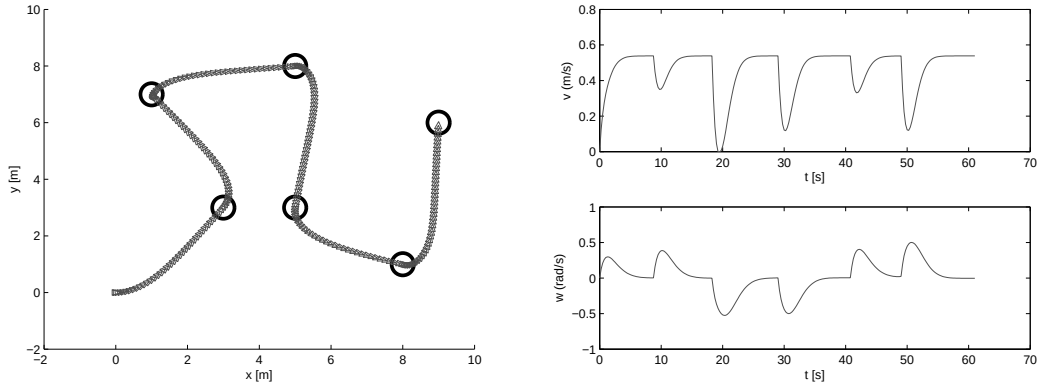
$$\mathbf{A} = -2b\begin{bmatrix}1 & 0\\ 0 & k_i\end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix}1 & 0\\ 0 & k_i h\end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix}F cos\theta\\ F sin\theta\end{bmatrix} \tag{3.2}$$

and b (viscous friction coefficient), k_i (inertial coefficient) and h (moment arm) are the controller parameters which must be tuned to obtain the desired robot behaviour as explained in [2]. We solve Eq. 3.1 by using finite difference approximation:

$$\frac{v_k - v_{k-1}}{\Delta t} = -2bv_k + F cos\theta; \qquad \frac{\omega_k - \omega_{k-1}}{\Delta t} = -2bk_i\omega_k + k_i h F sin\theta \tag{3.3}$$

$$v_k = \frac{F\Delta t cos\theta + bv_{k-1}}{1 + 2b\Delta t}; \qquad \omega_k = \frac{k_i h F \Delta t sin\theta + \omega_{k-1}}{1 + 2bk_i\Delta t} \tag{3.4}$$

From Eq. 3.4 we can observe that the new velocities provided by the controller depend on the the previous velocities and the force, which is computed based on the relative position of the robot with respect to the goal. In an ideal situation, the computed velocities are effectively applied on the robot and it is moved to the desired position. Figure 3.2 shows a simulation in a noise-free environment where a robot is commanded to reach six consecutive goals (black circles). Each goal exerts an attractive force on the robot which provides a direction of movement based on its current position. At each time step, the motion controller provides smooth linear and angular velocities to comply with the desired robot behaviour (Fig. 3.2b).



(a) **(b)**

Figure 3.2: Single robot motion in a noise-free scenario. (a) Ground-truth trajectory followed by the robot. (b) Linear and angular velocities computed by the motion controller.

However, in a real situation, different sources of errors may affect the controller behaviour. For example, if the controller is not well modeled and the robot is not able to reach the commanded velocities, we can read the actual velocities from the robot sensors to provide a feedback on v_{k-1}, however, these sensor readings are also subject to errors. Additionally, the application of the velocities itself is corrupted by noise which affects the robot's knowledge about its actual position and therefore, the proper direction of movement to reach its goal. To visualize these problems, we performed the same experiment shown in Fig. 3.2 where the velocities and localization readings were corrupted by noise. The result is visualized in Fig. 3.3, where the corrupted velocities make the robot follow an undesired trajectory (blue path) despite thinking it has reached its goal (red path). However, despite the presence of noise, the motion controller adapts smoothly and provides input velocities close to the desired ones (Fig. 3.3b). This suggests the use of the velocities provided by the controller to estimate the robot's displacement.

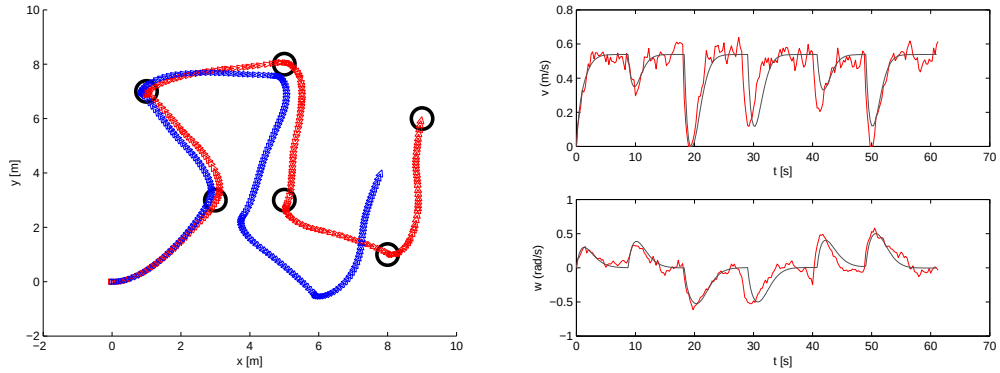
We take account of the difference between the commanded and actual velocities by establishing an error model for the linear and angular velocities:

$$v = \hat{v} + \varepsilon_v;\ \varepsilon_v \sim \mathcal{N}(0, \sigma_v^2) \tag{3.5}$$

$$\omega = \hat{\omega} + \varepsilon_\omega;\ \varepsilon_\omega \sim \mathcal{N}(0, \sigma_\omega^2) \tag{3.6}$$

Then, the state vector of the velocities $\mathbf{u}$ is modelled as a Gaussian, with its mean $\hat{\mathbf{u}}$ and covariance $\mathbf{P_u}$:

$$\mathbf{u} = \begin{pmatrix} v \\ \omega \end{pmatrix} \sim \mathcal{N}\left(\hat{\mathbf{u}} = \begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix}, \mathbf{P_u} = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{pmatrix}\right) \tag{3.7}$$



(a) **(b)**

Figure 3.3: Motion controller behaviour in the presence of random noise. (a) Ground-truth (blue) and odometry-based (red) trajectory. The attractive force towards the goal is computed based on its inaccurate position knowledge, and therefore, the robot considers it has reached the goal. (b) Comparison between the velocities computed in a situation with noisy sensor readings (red) and the ideal ones (grey).

Given this velocity error model, we can estimate the robot motion $\mathbf{x}_{R_k}^{R_{k-1}}$ between two consecutive steps $k-1$ and k with elapsed time Δt in terms of the robot's state space representation $(x, y, \theta)^T$. Again, under the Gaussian assumption, the displacement is characterized by its mean $\hat{\mathbf{x}}_{R_k}^{R_{k-1}}$ and covariance matrix $\mathbf{P}_{R_k}^{R_{k-1}}$. Then, the velocities state vector is transformed into the robot cartesian coordinates using the following transformation:

$$\mathbf{x}_{R_k}^{R_{k-1}} = \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \frac{v}{\omega}\sin\omega\Delta t \\ \frac{v}{\omega}(1-\cos\omega\Delta t) \\ \omega\Delta t \end{pmatrix} \tag{3.8}$$

Note that this transformation is nonlinear. Thus, we perform a first-order linearization around the estimated velocities $\hat{\mathbf{u}}$:

$$\mathbf{x}_{R_k}^{R_{k-1}} \simeq \mathbf{f}(\hat{\mathbf{u}}) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\hat{\mathbf{u}}} (\mathbf{u} - \hat{\mathbf{u}}) \tag{3.9}$$

with,

$$\left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\hat{\mathbf{u}}} = \begin{pmatrix} \frac{\sin\omega\Delta t}{\omega} & \frac{v}{\omega}\left(\Delta t\cos\omega\Delta t - \frac{\sin\omega\Delta t}{\omega}\right) \\ \frac{(1-\cos\omega\Delta t)}{\omega} & \frac{v}{\omega}\left(\Delta t\sin\omega\Delta t - \frac{(1-\cos\omega\Delta t)}{\omega}\right) \\ 0 & \Delta t \end{pmatrix} \tag{3.10}$$

Finally the estimate of the relative displacement is calculated as:

$$\hat{\mathbf{x}}_{R_k}^{R_{k-1}} = \begin{pmatrix} \frac{\hat{v}}{\hat{\omega}} \sin \hat{\omega}\Delta t \\ \frac{\hat{v}}{\hat{\omega}}(1-\cos \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t \end{pmatrix} \tag{3.11}$$

and its covariance matrix is mapped from the velocities space into the robot coordinates space by:

$$\mathbf{P}_{R_k}^{R_{k-1}} \simeq \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\hat{\mathbf{u}}} \mathbf{P}_{\mathbf{u}} \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\hat{\mathbf{u}}}^{T} \tag{3.12}$$

3.3 Robot Formation Control

In the spring-damper model, the robots are virtually connected to each other by a set of linear and torsional springs and dampers (fig. 3.4). Linear springs allow the robots to maintain a relative established distance whereas torsional springs are used to maintain a desired angle with respect to their leader.

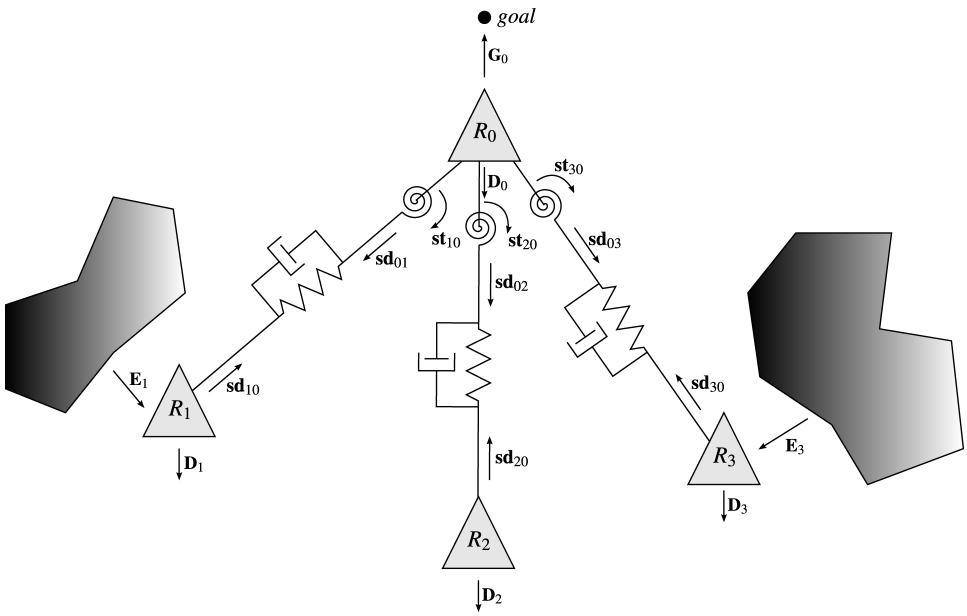


Figure 3.4: Spring-damper model

Due to the interconnections, a set of virtual internal forces are generated between the robots which affect the dynamics of the system. The internal force generated by this

structure that influences robot R_i can be computed as follows:

$$\mathbf{SD}_i = \sum_{j=1}^{N} a_{ij}\mathbf{sd}_{ij} + \sum_{j=1}^{N} b_{ij}\mathbf{st}_{ij} \tag{3.13}$$

where $\mathbf{sd}_{ij}$ is the force generated between robots R_i and R_j by the linear spring and damper and $\mathbf{st}_{ij}$ is the one generated by the torsional spring. The elements $a_{ij}, b_{ij} \in \{0,1\} \forall i, j \in \{0...r\}$ represents the influence of the linear spring-damper link and torsional spring link between robots R_i and R_j respectively.

Additionaly, a damping term due to the friction $\mathbf{D}_i$ is introduced to simulate a real system,

$$\mathbf{D}_i = f_d \mathbf{v}_i \tag{3.14}$$

where f_d is the damping coefficient and $\mathbf{v}_i$ the velocity vector of the robot.

The influence of the environment can also be included in the dynamics of the system as a repulsive force $\mathbf{E}_i$ coming from the obstacles and therefore, providing the formation with obstacle avoidance capabilities. In the case of the robot leader, it is assumed to navigate in a free space, following a safe path computed by a planner. For this reason, this external force is applied only to the robot followers. Instead of that, the leader of the formation experiences an attractive force $\mathbf{G}_0$ generated by its goal.

To sum up, the resultant force $\mathbf{F}_i$ applied to each robot is, for a robot follower:

$$\mathbf{F}_i = \mathbf{SD}_i + \mathbf{D}_i + \mathbf{E}_i \tag{3.15}$$

and for the robot leader:

$$\mathbf{F}_0 = \mathbf{G}_0 + \mathbf{SD}_0 + \mathbf{D}_0 \tag{3.16}$$

This force is transformed into linear and angular velocities with the motion controller explained in the previous section. For any further details about how to compute these forces, the reader is referenced to [85]. An extended approach for cooperative formation control is presented in [83] where the Nearness Diagram (ND) method [57] for obstacle avoidance is combined with different navigation strategies depending on the environment complexity.

Figure 3.5 shows the control of a triangle-shaped formation navigating through the environment. Thanks to the interaction of the set of virtual springs and dampers the robots always try to maintain their relative distances and orientations. We can see this effect in Fig. 3.5a, where the leader "waits" for the formation reconfiguration each time it reaches a subgoal and heads towards the next one.

As in the case of single robot control, there are some sources of errors affecting directly to the formation control, such as the location readings used to compute the forces or the velocity readings to feedback the controller. The problem is more evident since it does not only affect the reachability of the goal but also the formation keeping.

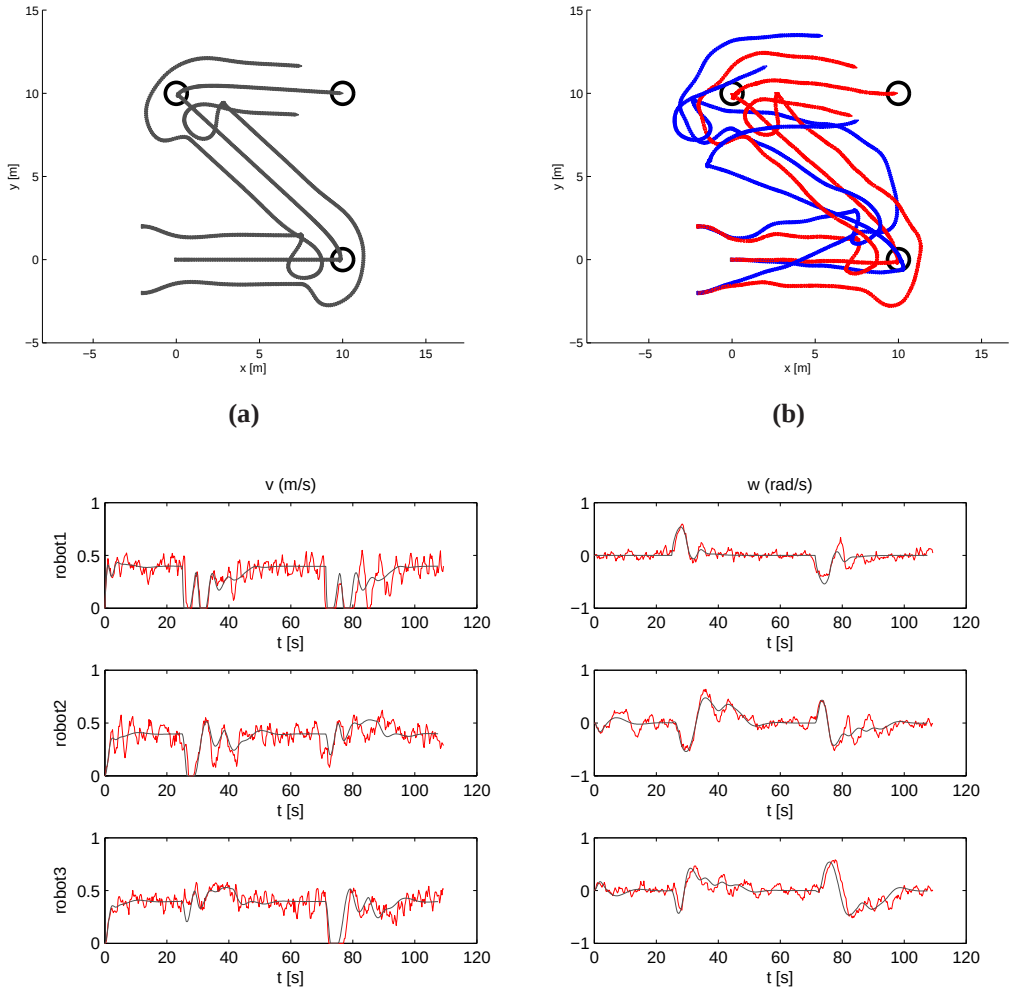
In order to address this problem, in the next chapter we propose a probabilistic formulation of the robot formation to deal with the different sources of uncertainty and we will provide the formation with a localization system based on a prior map to improve the robot formation behaviour.

3.4 Conclusions

In this chapter, we have presented a navigation system for robot formations which makes use of a virtual structure of springs and dampers that allows to maintain the relative distances and orientations imposed to the robots. This virtual structure originates a set of forces among the robots which can be later transformed into input velocities for each individual robot by means of a motion generator.

The motion controller provides velocities physically feasible for the robot and adapts smoothly in the presence of noise thus we use these velocities to estimate the robot displacement.

Despite the stable behaviour of the controller, we have evidenced that the accumulation of errors results in an inaccurate localization of the robots which prevents the formation to maintain its shape and reach its goal. This constitutes the motivational starting point for the development of improved multi-robot localization and SLAM techniques presented in the rest of this thesis.



(a) **(b)** **(c)**

Figure 3.5: Motion control of a 3-robot formation. (a) Trajectory followed in a noise-free situation. Notice how torsional springs oblige the robot followers to maintain their relative orientation with respect to the leader at any time. (b) Estimated (red) and ground-truth trajectory in the presence of noise. (c) Comparison between the commanded velocities in the "ideal" error-free (grey) and noisy scenario (red).

Chapter 4

Localization of Robot Formations in SLAM

In the previous chapter, we have seen that, in order to achieve the navigation in formation and to reach their goal, robots need to be accurately localized within the environment. This chapter presents an EKF-based approach which makes use of a previously built feature-based stochastic map to solve the robot formation pose tracking problem. We show how a direct implementation of the EKF algorithm leads to inconsistency in the estimated localization. We justify the origin of the anomalous behaviour of the filter in the time-correlated nature of the measurement noise sequence. A novel solution based on the measurement differencing technique is proposed to drive the solution of the EKF towards consistency.

4.1 Introduction

Traditionally, SLAM research has focussed on how to efficiently build accurate and reliable maps and, to this end, a variety of representations have been proposed such as feature, grid and graph-based maps. One of the many potential applications of these maps is their posterior use to localize a robot while it navigates through the environment.

Recalling Chapter 2, the robot localization problem consists in estimating the robot's pose $\mathbf{x}_{R_k}$ at the current time step k with respect to the map's reference frame using the information obtained from its on-board sensors. It is stated from a probabilistic point of view as the problem of estimating the posterior distribution $p(\mathbf{x}_{R_k}|\mathbf{z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{m})$ where $\mathbf{z}_{1:k}$ are the set of partial measurements of the environment provided by its perception sensors, $\mathbf{u}_{1:k}$, the history of control inputs and $\mathbf{m}$, the prior map.

Given a map of the environment, we can distinguish two main localization problems depending on the robot's prior knowledge about its initial position: the *global localization* tries to determine the robot's position within the given map without any

information about its initial pose and the *pose tracking* which, once the initial pose is known, aims to maintain the robot's pose estimate along time.

This problem has been principally addressed by Bayesian filters, due to their efficiency and real-time performance. Then, we can follow a similar derivation of the localization problem as we explained for the filtering-based SLAM problem in Chapter 2, leading to,

$$p(\mathbf{x}_{R_k}|\mathbf{z}_{1:k},\mathbf{u}_{1:k},\mathbf{m}) = \eta\, p(\mathbf{z}_k|\mathbf{x}_{R_k},\mathbf{m}) \int p(\mathbf{x}_{R_k}|\mathbf{x}_{R_{k-1}},\mathbf{u}_k)\, p(\mathbf{x}_{R_{k-1}}|\mathbf{z}_{1:k-1},\mathbf{u}_{1:k-1},\mathbf{m})\, d\mathbf{x}_{R_{k-1}} \tag{4.1}$$

which is solved in two phases, prediction and update steps.

Different filtering methods have been proposed over the years, some of them developed using a specific map representation. For example, grid-based continous localization was addressed in [78] by using local to global map registration (i.e., map matching). In [12, 28], the authors presented a grid-based Markov localization to solve both global and pose tracking problems. The robustness and efficiency of this approach was later increased by using particle filters [82] or by combining grid-based and topological maps [80]. From a feature-based perspective, the use of previously built maps to globally localize individual vehicles has been reported in [62] and [69]. Usually, robot localization in feature or landmark-based maps is implemented by using Kalman filters. However, although these maps have been adquired by probabilistic techniques, and therefore, there is a map uncertainty, these approaches normally consider the position of features and landmarks fixed and known.

One straightforward extension of any of the aforementioned techniques to multi-robot systems would be to address the team localization problem individually for each robot and to replicate in each vehicle the current implementation of the localization system. However, this methodology is against the team cooperation and coordination philosophy to obtain mutual benefit between robots.

For example, in certain situations a robot could lose its localization capability (e.g., due to the presence of obstacles occluding its vision of the environment). If another teammate is able to detect it and communicate their relative position, the first robot can use this information to improve its localization. Besides, if robots share and join all observations they can obtain a broader and extended view of the environment which can benefit the whole group.

First works in multi-robot cooperative localization like [48] and [73] divided the team of robots into two groups: while one group moved, the other group remained stationary and acted as landmark. This strategy is useful in the absence of landmarks or in uncharted environments but the robots must maintain visual contact which constraints the robot displacements and requires a robot identification system. Inter-robot position measurements are also used in [77] within an EKF framework to improve the estimates of the group and as well as in [75] where the Kalman Filter equations are decomposed

and distributed among the robots. This latter work also pointed out the benefits of maintaining the cross-correlations between the estimates of the robots. When two robots meet and share their inter-robot measurements to update their positions, the estimates of rest of the group are also improved due to these inter-dependencies. However, none of these works make a explicit use of an environmental model to improve robot localization which is only updated when they detect each other. Additionally, in [27] inter-robot detections are combined with environment measurements using probabilistic techniques.

Furthermore, and from a state estimation perspective, consistency issues are of paramount importance to assure convergence of the solution provided by the estimation algorithm. Intuitively, inconsistency appears when the uncertainty about the estimated robot position does not correctly represent the true error. In such case, the robot is unaware of how good its localization is and can not trust in it for other navigation or planning purposes. This issue has been discussed in the EKF framework [16, 5, 39] and several factors affecting the filter consistency have been identified such as linearization errors, measurement noise, number of measurements in the filter or re-observation times of a feature, among others. All these factors also affect multi-robot systems, however, consistency issues have not received as much attention in this context. In [38] the consistency of multi-robot cooperative localization was investigated from an observability perspective. Also, the effect of using repeated measurements on the consistency of the algorithm has been reported in [60] by taking into account the correlations between consecutive relative-state measurements, and in [3] by keeping track of the origins of measurements and preventing them from being used more than once.

In the rest of this chapter, we turn our attention towards how to use a previously built feature-based stochastic map to localize a team of autonomous vehicles while they navigate coordinately in formation using the motion control approach described in Chapter 3. Instead of using direct robot-to-robot measurements to strengthen relations among robot locations, their estimates get correlated by sensing the same portion of the environment.

The contributions are two-fold. First, the work emphasizes the inherent difficulties of using stochastic maps to localize a robot formation due to the existence of time-correlated measurement sequences. Whenever these statistical dependencies are not properly considered we show that the algorithms lead to inconsistent estimation of the robot formation localization. Second, we originally formulate the problem of robot formation localization in SLAM within the EKF framework by using the *measurement differencing* technique [11, 6, 71] which allows the elimination of colored measurement sequences within the update step of the filtering algorithm.

4.2 Probabilistic Representation of the Robot Formation

Let a robot formation be composed of $r+1$ heterogeneous vehicles $R_0, R_1, ..., R_r$, where R_0 is the robot leader and $R_j, j = 1, ..., r$ are the robot followers. A certain geometric shape, e.g. equilateral triangle, regular pentagon, etc, is imposed to the team depending on the number of vehicles and the task commanded to the formation. From a probabilistic view-point, the location of the robot formation can be represented by a discrete-time state vector $\mathbf{x}_{\mathcal{R}}$ formed by the location of the robot leader R_0 with respect to (wrt) a base reference frame B and the location of each robot follower R_j wrt the robot leader R_0 , and by its associated covariance matrix $\mathbf{P}_{\mathcal{R}}$ which stores the statistical dependencies between those estimated locations. Following the Gaussinity assumption, $\mathbf{x}_{\mathcal{R}} \sim \mathcal{N}(\hat{\mathbf{x}}_{\mathcal{R}}, \mathbf{P}_{\mathcal{R}})$ with,

$$\hat{\mathbf{x}}_{\mathcal{R}} = \begin{pmatrix} \hat{\mathbf{x}}^{B}_{R_0} \\ \hat{\mathbf{x}}^{R_0}_{R_1} \\ \vdots \\ \hat{\mathbf{x}}^{R_0}_{R_r} \end{pmatrix}; \mathbf{P}_{\mathcal{R}} = \begin{pmatrix} \mathbf{P}_{R_0} & \cdots & \mathbf{P}_{R_0 R_r} \\ & \ddots & \\ \mathbf{P}_{R_r R_0} & \cdots & \mathbf{P}_{R_r} \end{pmatrix} \tag{4.2}$$

This leader-centric representation reduces the volume of uncertainty, i.e., the determinant of the covariance matrix $\mathbf{P}_{\mathcal{R}}$, in comparison with an absolute representation wrt the base frame B of each robot location vector and, therefore, linearization errors due to large uncertainty values are minimized [15].

4.3 EKF-based Localization of the Robot Formation

From a Bayesian view-point the pose of the robot formation is given by the recursively estimated conditional probability density function $p(\mathbf{x}_{\mathcal{R}_k} | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{y}_{\mathcal{F}})$ where $\mathbf{z}_{1:k}$ represent the set of sensor readings gathered by the sensors mounted on the robots from environmental features from time step 1 up to time step k, $\mathbf{u}_{1:k}$ the sequence of input controls and $\mathbf{y}_{\mathcal{F}} \sim \mathcal{N}(\hat{\mathbf{y}}_{\mathcal{F}}, \mathbf{P}_{\mathcal{F}})$ represents the stochastic map of the previously mapped navigation area consisting of a set of features $\mathcal{F} = \{F_1, F_2, ..., F_n\}$. In the sequel, the EKF algorithm [6] is used as the core estimation technique.

4.3.1 EKF Prediction Step

The method described in section 3.2 provides a displacement estimate $\mathbf{x}^{R_{i,k-1}}_{R_{i,k}}$ for each robot R_i from time step $k-1$ to k based on the input velocities to the motion controller.

Then, we represent the formation displacement by the random variable $\mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}$:

$$\mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}} = (\mathbf{x}_{R_{0,k}}^{R_{0,k-1}} \mathbf{x}_{R_{1,k}}^{R_{1,k-1}} \dots \mathbf{x}_{R_{r,k}}^{R_{r,k-1}})^T \tag{4.3}$$

with covariance matrix,

$$\mathbf{P}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}} = \begin{pmatrix} \mathbf{P}_{R_{0,k}}^{R_{0,k-1}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{R_{1,k}}^{R_{1,k-1}} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{P}_{R_{r,k}}^{R_{r,k-1}} \end{pmatrix} \tag{4.4}$$

The EKF-prediction step propagates the state of the robot formation from time step $k-1$ to time step k by using the estimated displacements through the following nonlinear function,

$$\mathbf{x}_{\mathcal{R}_k} = \mathbf{f}(\mathbf{x}_{\mathcal{R}_{k-1}}, \mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}) \tag{4.5}$$

The use of the EKF requires a linearization of the nonlinear model around the best current estimates $(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}})$ using a Taylor series expansion,

$$\mathbf{x}_{\mathcal{R}_k} = \mathbf{f}(\mathbf{x}_{\mathcal{R}_{k-1}}, \mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}) \tag{4.6}$$

$$\simeq \mathbf{f}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}) + \mathbf{F}_{k-1}(\mathbf{x}_{\mathcal{R}_{k-1}} - \hat{\mathbf{x}}_{\mathcal{R}_{k-1}}) + \mathbf{G}_{k-1}(\mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}} - \hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}) \tag{4.7}$$

where $\mathbf{F}_{k-1}$ and $\mathbf{G}_{k-1}$ are the jacobians of $\mathbf{f}$ with respect $\mathbf{x}_{\mathcal{R}_{k-1}}$ and $\mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}$respectively,

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_{\mathcal{R}_{k-1}}}\right|_{(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}})} ; \quad \mathbf{G}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}}\right|_{(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}})} \tag{4.8}$$

The linearized equation 4.7 can also be expressed in terms of the errors denoted by $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ as

$$\tilde{\mathbf{x}}_{\mathcal{R}_k} \simeq \mathbf{F}_{k-1}\tilde{\mathbf{x}}_{\mathcal{R}_{k-1}} + \mathbf{v}_{k-1}; \quad \mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{G}_{k-1}\mathbf{P}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}\mathbf{G}_{k-1}^T) \tag{4.9}$$

Then, the estimates for the state vector $\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}$ and its associated covariance matrix $\mathbf{P}_{\mathcal{R}_{k|k-1}}$ are given by,

$$\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}} = \mathbf{f}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}) \tag{4.10}$$

$$\mathbf{P}_{\mathcal{R}_{k|k-1}} = \mathbf{F}_{k-1}\mathbf{P}_{\mathcal{R}_{k-1}}\mathbf{F}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{P}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}\mathbf{G}_{k-1}^T \tag{4.11}$$

Due to our leader-centric representation, function 4.5 differs depending on whether the robot is the leader or a robot follower. A more detailed description of this function and the derivation of its jacobians can be followed in appendix A.

4.3.2 EKF Update Step

At time step k observations $\mathbf{z}_k$ are related with map features through the following nonlinear measurement model,

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{F}_k}) + \mathbf{w}_k \tag{4.12}$$

being $\mathbf{w}_k$ a zero-mean Gaussian noise with covariance matrix $\mathbf{R}_k$.

Due to the inherent nonlinearities, a linearized equation around the current predicted position $\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}$ is used within the EKF-update step,

$$\begin{aligned} \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{F}_k}) + \mathbf{w}_k & (4.13)\\ &\simeq \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{H}_k(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}) + \mathbf{G}_{\mathcal{F}_k}(\mathbf{y}_{\mathcal{F}_k} - \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{w}_k & (4.14) \end{aligned}$$

where $\mathbf{H}_k$ and $\mathbf{G}_{\mathcal{F}_k}$ are the jacobian matrices (see appendix A) of the linearized measurement equations with respect to the state vector $\mathbf{x}_{\mathcal{R}_k}$ and with respect to the subset of features $\mathbf{y}_{\mathcal{F}_k}$ observed at time k from the a priori stochastic map respectively,

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_{\mathcal{R}_k}}\right|_{(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k})} \;;\; \mathbf{G}_{\mathcal{F}_k} = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{y}_{\mathcal{F}_k}}\right|_{(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k})} \tag{4.15}$$

The classical EKF update equations provide estimates for the state vector $\hat{\mathbf{x}}_{\mathcal{R}_{k|k}}$ and its associated covariance matrix $\mathbf{P}_{\mathcal{R}_{k|k}}$:

$$\begin{aligned} \hat{\mathbf{x}}_{\mathcal{R}_{k|k}} &= \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}))\\ \mathbf{P}_{\mathcal{R}_{k|k}} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{\mathcal{R}_{k|k-1}} \end{aligned} \tag{4.16}$$

using the filter gain obtained as,

$$\mathbf{K}_k = \mathbf{P}_{\mathcal{R}_{k|k-1}}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{\mathcal{R}_{k|k-1}}\mathbf{H}_k^T + \mathbf{G}_{\mathcal{F}_k}\mathbf{P}_{\mathcal{F}_k}\mathbf{G}_{\mathcal{F}_k}^T + \mathbf{R}_k)^{-1} \tag{4.17}$$

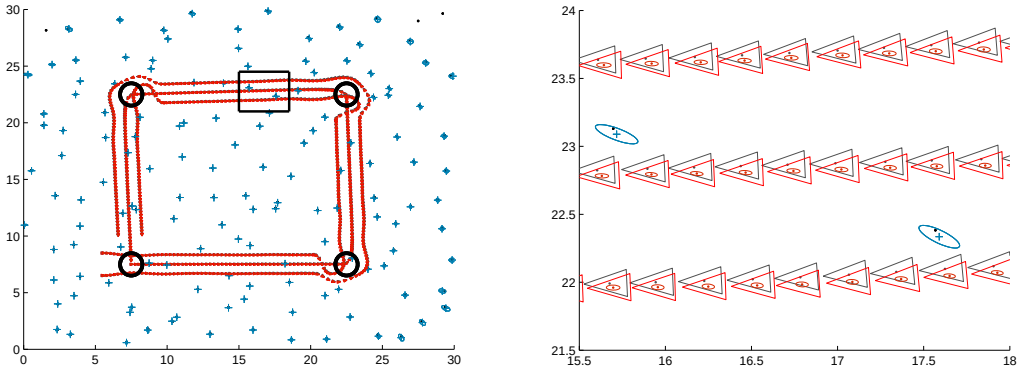
4.3.3 Simulation Results: Inconsistency

A set of simulation experiments were conducted to analyze the consistency of a direct implementation of the EKF algorithm based on the aforementioned problem formulation. The single robot localization problem formulation can be seen as the particular case of considering only the robot leader in the robot formation formulation. Then, a single robot and two formations composed of 3 and 5 robots were commanded to reach 4 consecutive goals arranged in a loop-trajectory within a previously mapped (2D point-based stochastic map) navigation area, depicted in Fig. 4.1. Each vehicle was equipped with a range-bearing sensor capable of observing the available environmental features.

For these simulation experiments, we assume known data association. As a measure of consistency [6] a statistical test based on the Normalized Estimation Error Squared (NEES) was used,

$$\text{NEES} = (\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k})^T \, \mathbf{P}_{\mathcal{R}_k}^{-1} (\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k}) \leq \chi^2_{r,1-\alpha} \tag{4.18}$$

where $\chi^2_{r,1-\alpha}$ is a threshold obtained from the χ^2 distribution with $r = \dim(\mathbf{x}_{\mathcal{R}_k})$ degrees of freedom, and α the desired significance level (usually 0.05).



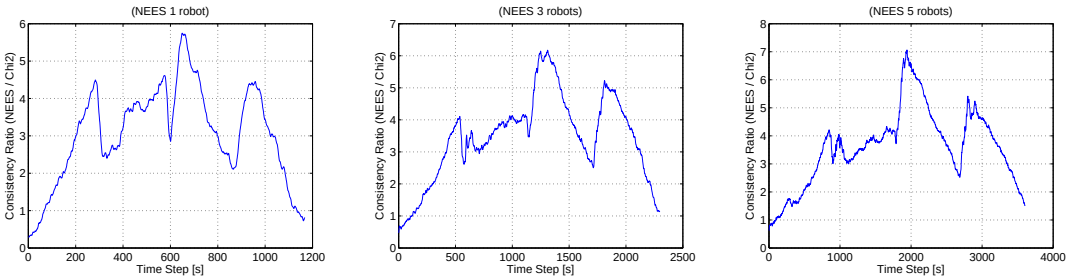
(a) **(b)**

Figure 4.1: Map of the environment used in the simulation experiments to evaluate the robot formation localization performance. (a) 2D Point-based stochastic map (blue) and trajectory followed by the 3-robot formation. (b) Zoomed part of the formation trajectory where the robots' ground-truth location is depicted in black and its estimate in red. The small robot covariance ellipses demonstrate the overconfidence in the error estimation using the classical EKF implementation.

Figure 4.2 plots the consistency ratio ($\text{NEES}/\chi^2_{r,1-\alpha}$) for the sequence of time steps of the experiments illustrating a problem of inconsistency in the estimated solution of this direct implementation of the EKF algorithm. It can also be observed how this inconsistency issue gets worse with the number of robots. This fact is not unexpected since the factors which may influence in the filter consistency such as number of observations or feature re-observations increase with the number of robots in the formation.

Additionally, Fig. 4.3 shows the errors and covariance bounds obtained in one of the simulations with the 3-robot formation. It can be observed how the covariances do not correctly match the true estimation errors.

In the following section we take a closer view of the results and a modified EKF formulation is proposed to drive the robot formation estimated localization towards consistency.



(a) **(b)** **(c)**

Figure 4.2: Inconsistency of the estimated robot formation localization within a stochastic map with a direct implementation of the EKF algorithm. (a) shows the consistency ratio $\text{NEES}/\chi^2_{r,1-\alpha}$ for the single-robot implementation whereas (b) and (c) show the results for a 3 and 5-robot formation respectively. The estimation is considered consistent when the consistency ratio is less than one. For each experiment, the average of 50 Monte Carlo runs is depicted.

4.4 Measurement-Differencing EKF-based Localization of the Formation

A close view of the robot formation localization problem within an a priori stochastic map, supported by the simulation results obtained in the previous section, suggests a re-formulation of the EKF algorithm taking into account the statistical dependencies between the map features $\mathbf{y}_{\mathcal{F}_{k-1}}$ and $\mathbf{y}_{\mathcal{F}_k}$ (both subsets of $\mathbf{y}_{\mathcal{F}}$) used within the EKF update step in two consecutive time steps $k-1$ and k.

Given the set of matched map features of two consecutive time instants $\mathbf{y}_{\mathcal{F}_{k-1}}$ and $\mathbf{y}_{\mathcal{F}_k}$, their statistical dependencies are expressed by a linear transformation,

$$\mathbf{y}_{\mathcal{F}_k} = \mathbf{F}_{C_k}\mathbf{y}_{\mathcal{F}_{k-1}} + \mathbf{n}_k \tag{4.19}$$

where [1]

$$\mathbf{F}_{C_k} = \mathbf{P}_{\mathcal{F}_k\mathcal{F}_{k-1}}\mathbf{P}^{-1}_{\mathcal{F}_{k-1}} \tag{4.20}$$

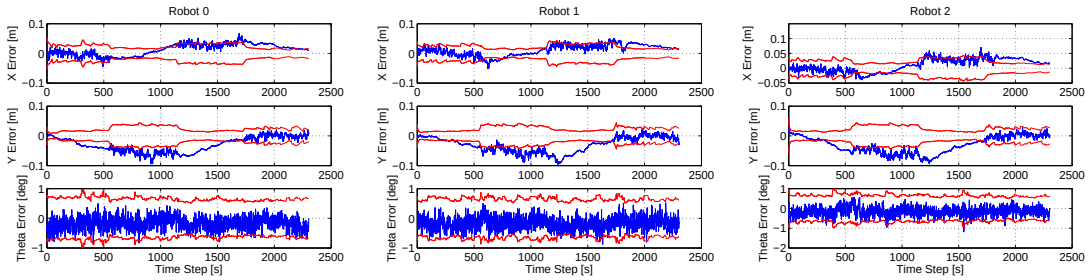
and $\mathbf{n}_k$ is a white noise measurement sequence with covariance matrix $\mathbf{P}_{\mathbf{n}_k}$,

$$\mathbf{P}_{\mathbf{n}_k} = \mathbf{P}_{\mathcal{F}_k} - \mathbf{P}_{\mathcal{F}_k\mathcal{F}_{k-1}}\mathbf{P}^{-1}_{\mathcal{F}_{k-1}}\mathbf{P}_{\mathcal{F}_{k-1}\mathcal{F}_k} \tag{4.21}$$

[1] Given two jointly Gaussian random variables $\mathbf{x}$ and $\mathbf{y}$:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left\{ \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}; \begin{pmatrix} \mathbf{P_x} & \mathbf{P_{xy}} \\ \mathbf{P_{yx}} & \mathbf{P_y} \end{pmatrix} \right\}$$

a linear relation of the form $\mathbf{y} = \mathbf{Ax} + \mathbf{b}$ can be formulated, with $\mathbf{A} = \mathbf{P_{xy}}\mathbf{P_x}^{-1}$ and $\mathbf{P_b} = \mathbf{cov}(\mathbf{b}) = \mathbf{P_y} - \mathbf{P_{yx}}\mathbf{P_x}^{-1}\mathbf{P_{xy}}$.



(a) **(b)** **(c)**

Figure 4.3: Errors and 2σ-covariance bounds obtained in the 3-robot formation simulation using the classical EKF implementation. The errors in each position component (x, y, θ) are depicted for all robots.

Equation (4.19) defines a colored measurement noise sequence, which together with eqs. (4.5) and (4.12) completely reformulates the problem at hand.

The first approach reported in the literature which considers the existence of a colored measurement noise sequence within the EKF framework dates back to the works of Bryson et al. [11] where the state vector was augmented with the colored error terms. Later work pointed out relevant numerical problems of this approach mainly due to null-uncertainty observations and ill-conditioned transition matrices. Current practical approaches [6, 71] concern the so-called measurement differencing technique, which provides an efficient and mathematically sound method to remove the time-correlated portion of the measurement errors. We extend previous work by others in the field of filtering theory by formulating the robot formation localization problem in SLAM as a measurement differencing based EKF algorithm to whiten the originally colored measurement noise sequence defined in eq. (4.19).

4.4.1 Whitening the Measurement Equation

Let $\mathbf{r}_k$ represent the measurement considered within the EKF-update step at time k, derived from the real measurements $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$ obtained at two consecutive time instants as,

$$\mathbf{r}_k \triangleq \mathbf{z}_k - \mathbf{\Lambda}_k \mathbf{z}_{k-1} \tag{4.22}$$

where matrix $\mathbf{\Lambda}_k$ is chosen such that $\{\mathbf{r}_k, 0 < k < \infty\}$ approaches a discrete-time white-noise driven stochastic process [6, 71]. Following the derivation of the appendix B measurement $\mathbf{r}_k$ can be rewritten as,

$$\mathbf{r}_k \simeq \mathbf{h}_k^* + \mathbf{H}_k^* (\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}) + \mathbf{w}_{\mathbf{r}_k} \tag{4.23}$$

where,

$$\mathbf{h}_k^* = \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}}) \tag{4.24}$$

$$\mathbf{H}_k^* = \mathbf{H}_k - \mathbf{\Lambda}_k \mathbf{H}_{k-1} \mathbf{F}_{k-1}^{-1} \tag{4.25}$$

and the white noise sequence $\mathbf{w}_{\mathbf{r}_k}$, with covariance matrix $\mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}}$, is given by,

$$\mathbf{w}_{\mathbf{r}_k} = \mathbf{\Lambda}_k \mathbf{H}_{k-1} \mathbf{F}_{k-1}^{-1} \mathbf{v}_{k-1} + \mathbf{w}_k - \mathbf{\Lambda}_k \mathbf{w}_{k-1} + \mathbf{G}_{\mathcal{F}_k} \mathbf{n}_k \tag{4.26}$$

Note from Eq. 4.24 that, at time step k, jacobians corresponding to the linearized measurement equation of $\mathbf{z}_{k-1}$ are re-evaluated at the best state estimate at time $k-1$, that is $\hat{\mathbf{x}}_{\mathcal{R}_{k-1|k-1}}$. Matrix $\mathbf{\Lambda}_k$ is computed (see appendix B) such that the time-correlated components of the measurement $\mathbf{r}_k$ are removed,

$$\mathbf{\Lambda}_k \simeq \mathbf{G}_{\mathcal{F}_k} \mathbf{F}_{C_k} \mathbf{G}_{\mathcal{F}_{k-1}}^T (\mathbf{G}_{\mathcal{F}_{k-1}} \mathbf{G}_{\mathcal{F}_{k-1}}^T)^{-1} \tag{4.27}$$

Note that previous works [6, 71], under the linearity assumption both in the motion and measurement equations, reported that $\mathbf{\Lambda}_k = \mathbf{F}_{C_k}$, being a particular case of the more general result provided in this thesis. In our case, the existence of matrix $\mathbf{\Lambda}_k$ has been verified for the cases of 2D-point and 2D-segment based stochastic maps.

Equation (4.26) introduces a cross-correlation term between $\mathbf{w}_{\mathbf{r}_k}$ and $\mathbf{v}_{k-1}$, namely,

$$\mathbf{C}_k = E[\mathbf{v}_{k-1} \mathbf{w}_{\mathbf{r}_k}^T] = \mathbf{Q}_{k-1} (\mathbf{\Lambda}_k \mathbf{H}_{k-1} \mathbf{F}_{k-1}^{-1})^T \tag{4.28}$$

which is introduced in the EKF algorithm following [10] using the filter gain,

$$\mathbf{K}_k = (\mathbf{P}_{\mathcal{R}_{k|k-1}} \mathbf{H}_k^{*T} + \mathbf{C}_k)(\mathbf{H}_k^* \mathbf{P}_{\mathcal{R}_{k|k-1}} \mathbf{H}_k^{*T} + \mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} + \mathbf{H}_k^* \mathbf{C}_k + \mathbf{C}_k^T \mathbf{H}_k^{*T})^{-1} \tag{4.29}$$

Finally, the Measurement-Differencing EKF-based update (hereinafter referred to as MD-EKF) equations are given by,

$$\hat{\mathbf{x}}_{\mathcal{R}_{k|k}} = \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}} + \mathbf{K}_k (\mathbf{r}_k - \mathbf{h}_k^*) \tag{4.30}$$

$$\mathbf{P}_{\mathcal{R}_{k|k}} = \mathbf{P}_{\mathcal{R}_{k|k-1}} - \mathbf{K}_k (\mathbf{H}_k^* \mathbf{P}_{\mathcal{R}_{k|k-1}} \mathbf{H}_k^{*T} + \mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} + \mathbf{H}_k^* \mathbf{C}_k + \mathbf{C}_k^T \mathbf{H}_k^{*T}) \mathbf{K}_k^T \tag{4.31}$$

As we can observe from Eq. 4.30, the redefinition of the measurement equation leads to a modified innovation $(\mathbf{r}_k - \mathbf{h}_k^*)$ in the filter update which could be expressed following equations 4.22 and 4.24 in terms of the difference of the innovations from consecutive time steps as,

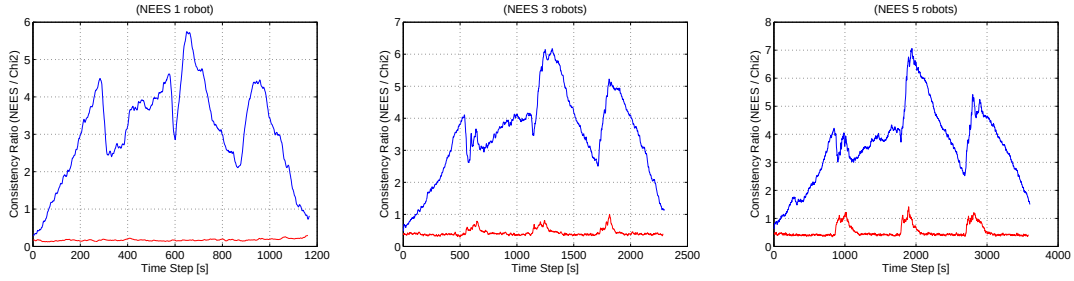
$$\mathbf{r}_k - \mathbf{h}_k^* = \mathbf{z}_k - \mathbf{\Lambda}_k \mathbf{z}_{k-1} - (\mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}})) \tag{4.32}$$

$$= \mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k (\mathbf{z}_{k-1} - \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}})) \tag{4.33}$$

Intuitively, if the difference of innovations is small, the modified filter update yields more pessimistic estimations than in the classical EKF. This is what happens when features are re-observed and contribute to very little additional information. More formally, this difference is regulated by $\mathbf{\Lambda}_k$ which depends on $\mathbf{F}_{C_k}$ and $\mathbf{P}_{\mathcal{F}_k\mathcal{F}_{k-1}}$, i.e., the degree of correlation between features observed in consecutive time steps (see equations 4.20 and 4.27). Conversely, if $\mathbf{P}_{\mathcal{F}_k\mathcal{F}_{k-1}} = \mathbf{0}$, then $\mathbf{\Lambda}_k = \mathbf{0}$ and the filter will behave as the standard EKF implementation.

Simulation results

As in the previous section, we conducted a set of experiments to validate the proposed reformulation of the robot formation localization problem. Figure 4.4 plots the consistency ratio (NEES$/\chi^2_{r,1-\alpha}$) for the sequence of time steps of the experiments for the implementation of the measurement differencing algorithm proposed in this section. In this case, the modified EKF algorithm provides a consistent solution for the robot formation estimated localization.

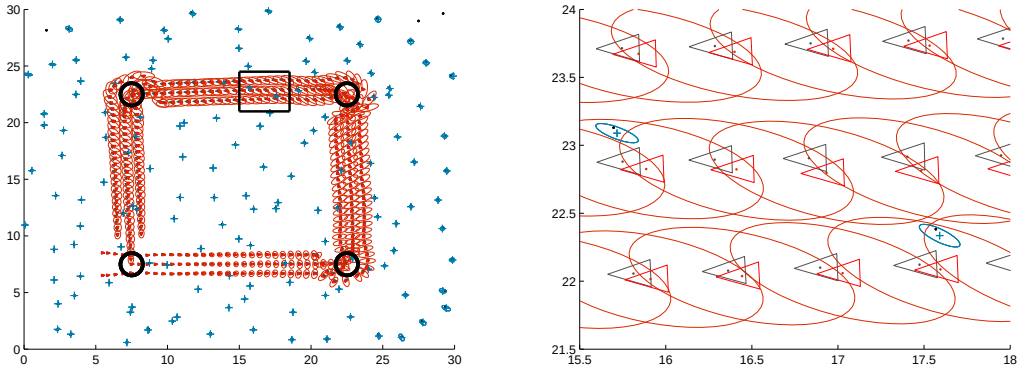


(a) **(b)** **(c)**

Figure 4.4: Improvement in the consistency of the single-robot ((a)) and robot formation ((b) and (c)) estimated localization with a measurement-differencing based EKF algorithm. The consistency ratio remains under the threshold (NEES$/\chi^2_{r,1-\alpha} \leq 1$) during most of the simulation. However, it is shown less consistent for short periods of time (e.g., around time steps 1000, 2000 and 3000 in the 5-robot formation case) due to formation turning which is a highly nonlinear situation and increases with the number of robots. The average of 50 Monte Carlo runs is depicted.

Figure 4.5 shows the estimated and ground-truth trajectory followed by the 3-robot formation. Although there is no significant difference with respect to the trajectory shown in Fig. 4.1, now the covariances match the ground-truth position. This can also be seen in Fig. 4.6, where the errors and uncertainty bounds are depicted.

It is worth to point out that, although in Fig. 4.5b, the estimates seem to be less accurate than the estimates in Fig. 4.1b, the latter are inconsistent and unreliable which



(a) **(b)**

Figure 4.5: Ground-truth and estimated trajectory followed by a 3-robot formation with the modified measurement differencing based EKF. Now, the uncertainty ellipses correctly include the actual robot position.

can derive in wrong data association and filter divergence. This fact was also observed and discussed in [5].

The computational cost of the MD-EKF algorithm does not appreciably increase over the direct implementation because the dimensions of the matrices involved are the same.

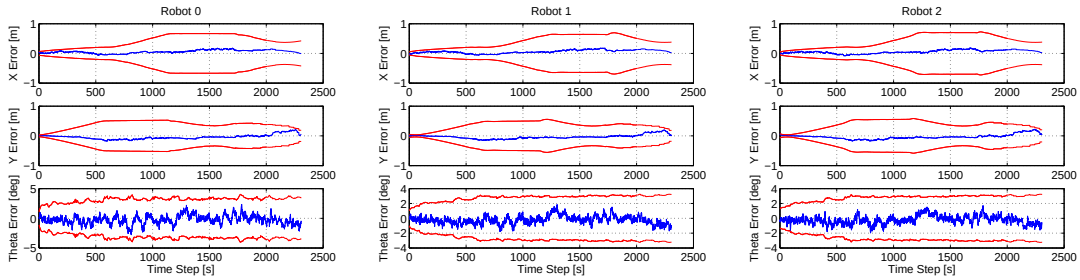
4.5 Experimental Results

The measurement differencing EKF-based localization algorithm has been tested both in the multi-robot simulation platform Player/Stage [29] and in real experiments with a 3-robot Pioneer 3-AT team in a triangular-shaped formation scheme.

4.5.1 Experiments in Player/Stage

A first set of experiments have been conducted within the rapid prototyping tool Player/Stage which allows code development and testing in similar conditions as to those subsequently faced in the real scenario but constraint to the information provided by simulated motion control and data acquisition.

The formation was commanded a 100-m loop trajectory within a previously available segment-based stochastic map. Thanks to the availability of ground-truth, the consistency of the proposed algorithm in this quasi-real scenario was verified as the consistency ratio plot of Fig. (4.7) highlights. Also, in Figs. 4.8 the frontal, lateral and angular



(a) **(b)** **(c)**

Figure 4.6: Errors and 2σ-covariance bounds obtained in the 3-robot formation simulation using the measurement differencing based EKF implementation. The errors in each position component (x, y, θ) are depicted for all robots.

errors for each robot in the formation are displayed together with their associated 2-σ uncertainty bounds. In all the cases the estimated errors are within the computed bounds.

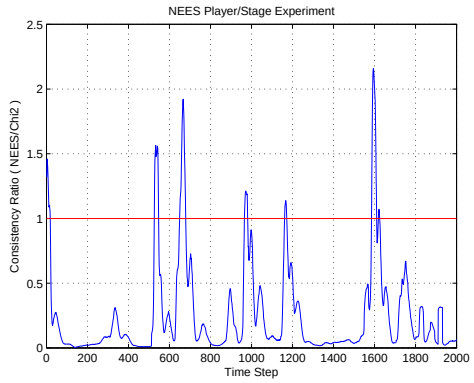
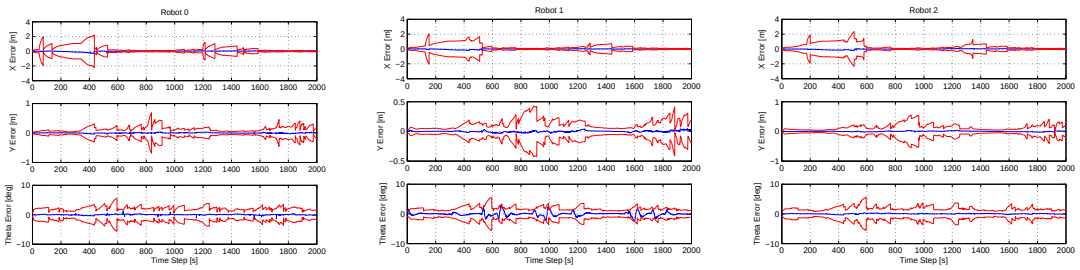


Figure 4.7: Consistency of the estimated robot formation localization with a measurement-differencing based EKF algorithm in the Player/Stage experiments.

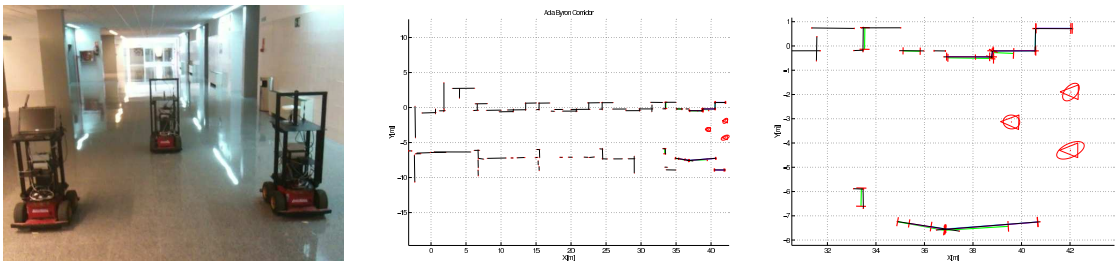
4.5.2 Experiments with the Pioneer 3-AT Robots

Real experiments have been conducted by using a 3-robot triangular-shaped formation of Pioneer 3-AT vehicles. Figure (4.9) depicts the initial localization of the vehicles within an indoor environment. The formation was commanded to reach a distant goal location (about 40-m from the starting location) while avoiding obstacles and adapting its shape to the environment. The robot leader plans a safe path to the goal destination and the robot slaves follow the leader while maintaining the desired formation topology.



(a) **(b)** **(c)**

Figure 4.8: Experimental results obtained in the Player/Stage setting. Frontal, lateral and angular errors for each vehicle in the 3-robot triangular-shaped formation and their associated 2-σ uncertainty bounds are shown.



(a) **(b)** **(c)**

Figure 4.9: Initial setting of the robot formation in a real experiment: (a) 3-robot triangular-shaped formation, (b) segment-based stochastic map of the navigation area, and (c) EIF-based initial localization of the robot formation within the map (with location uncertainty magnified x5).

Communication among the vehicles of the formation is provided by a real-time wireless multihop protocol implemented in a centralized mode where robot slaves sent both the sensors observations and the commanded velocities to the robot leader. The robot leader executes the localization algorithm and it communicates the estimated poses to each robot slave ([85] and reference therein).

Initially, a segment-based stochastic map of the navigation area (Fig. 4.9b) is obtained by using the information provided by the 2-D laser scanned mounted in one of the vehicle (in our case, and without lose of generality, the robot leader) which previously had explored the environment. Then, the global localization of the vehicles (Fig. 4.9c) is computed by the algorithms reported in [69].

Figure (4.10) shows the estimated localization of each of the vehicles of the for-

mation at four different time steps along the planned trajectory towards the goal destination. Even though ground-truth was not available during the execution of the real experiments, the figure highlights the compatibility between the previously available stochastic map and the segmented sensor readings plotted wrt the estimated vehicles localization.

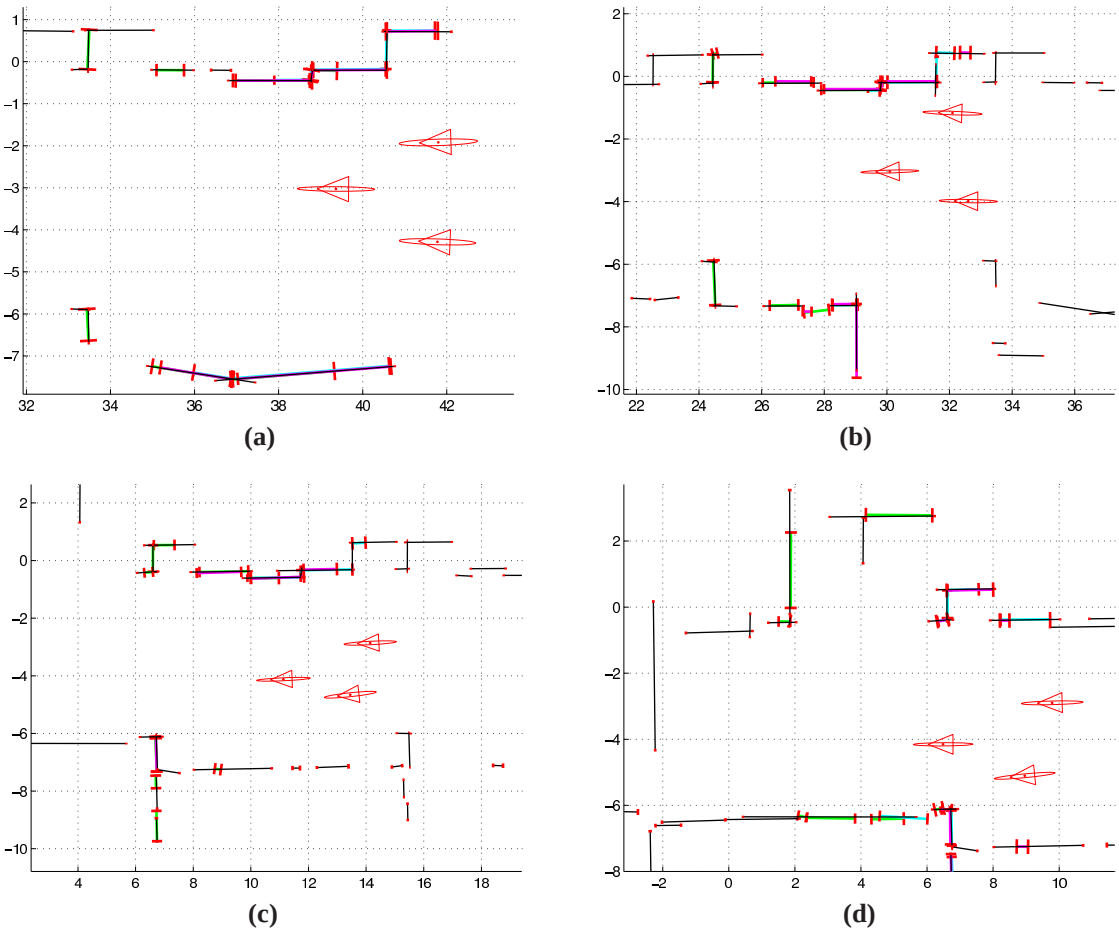


Figure 4.10: Four snapshots of the estimated localization of the members of the robot formation within the stochastic map (plotted in black). The segmented sensor observations are plotted in green, blue and magenta and red lines represent the segments uncertainties. Robot uncertainties' ellipses are magnified by x5.

4.6 Conclusions

In this chapter we have addressed the problem of utilization of feature-based stochastic maps for robot formation localization. From the Bayesian perspective, the classical EKF algorithm was initially formulated by considering the motion models of each vehicle within the formation and the environmental observations gathered by the exteroceptive sensors mounted on the vehicles. We have reported the inconsistency of the direct implementation of the EKF prediction and update equations for the problem at hand, producing unreliable position estimates. Having a closer view to algorithm hypotheses we have suggested the time-correlated nature of the sequence of measurements considered in the previous, direct implementation. We have adapted a solution based on the measurement differencing technique, already reported in the filtering literature, to the robot formation localization problem in SLAM and its implementation shows how the modified EKF estimation is driven towards consistency.

Chapter 5

Robot Formations in Partially *a priori* Known Environments

In this chapter we present a complete working system for robot formations where, in addition to the navigation and localization techniques presented in previous chapters, cooperative perception and path planning tasks under environment uncertainty are also considered. Feature-based and grid-based mapping strategies are combined in a probabilistic way to compute an obstacle-free and of bounded-risk plan towards the goal. The formation benefits from the cooperative perception to obtain a joint vision of the environment, used for online replanning purposes. The system is tested and validated by means of a set of simulations as well as in real experiments.

5.1 Introduction

In Chapter 3 we presented an approach to control the formation shape while navigating towards a designated goal. Then, in the previous chapter we proposed a technique which provides a reliable localization of the robots in an environment known in advance.

Following with the context presented in previous chapters and to achieve greater autonomy, we must provide the system with a path planner to compute the best path towards the final goal. However, in our case, the path planning strategy is constrained by the presence of uncertainty in the prior map available to the system. This is a problem of great interest since, when planning through risky areas, the formation may reach a dead-end, therefore requiring a global replanning which would increase the final cost of the mission.

The problem of planning under uncertainty is treated in different ways in the literature depending on the source of the error. Some works take into account the uncertainty of the environment like [58], who searches the trajectory that minimizes the expected cost of collision by using probabilistic roadmaps under the assumption of independence

in the probability distribution of the obstacles. In [86], motion and sensing uncertainties are considered to obtain prior probability distributions of the states and the control inputs of the robot for a previously computed path. Other kind of works focus on the uncertainty of the path, like [18] who seeks the path which minimizes the uncertainty at the goal, leading to paths that are no optimal in terms of Euclidean distance to the goal. The work presented in this chapter considers environment uncertainty for global path planning, whereas localization and sensor uncertainty is taken into account during the online execution of the global path.

Most of the successful path planning techniques [35, 44] use a grid representation of the environment whereas the map is available to the formation as a segment-based stochastic map. In order to use grid-based path planning techniques, we project the initial feature-based map into a grid where each cell represents the risk level for traversing a certain area of the environment. Then, the overall goal is to guarantee the mission success by planning through areas with low probability of obstacle presence or, in other words, with less risk of traversability. Unlike the works which minimize the overall accumulated risk of collision, the path planning technique presented in this chapter aims at computing the best path where the probability of failure at each cell of the path is lower than a threshold. This concept is also used in [8], although we propose a rigorous modeling of the obstacle location uncertainty, used as a measure of collision risk in a cell.

Once a global path is obtained, we can extract a set of waypoints or sub-goals the leader (and consequently the rest of robots) has to reach sequentially. However, the final execution of the planned path may be affected by the presence of unexpected obstacles which either were not considered in the prior map or were known with uncertainty. At the level of each robot and in the event of finally finding an obstacle, the reactive obstacle avoidance system would prevent from the collision. However, at the level of the formation, is necessary the maintainance of a local map for online path planning between sub-goals. This local planning task can benefit from the joint and wider vision of the environment provided by the robot formation. When building this joint local map from the observations shared by the formation, sensor and robot localization uncertainties have to be taken into account. In order to obtain a set of integrated lower uncertainty observations we propose a modification of the localization algorithm presented in previous chapter.

5.2 Path planning under Uncertainty

Path planning for robot formations can be stated globally for all the robots of the team. But that approach has a high computational burden. In this work, we propose a flexible alternative approach, computationally lighter. The path planning is achieved by the leader of the formation, and the followers maintain the formation adapting to the

environment by means of the spring-damper analogy (Chapter 3). The formation is maintained while possible, otherwise it adapts its shape to the scenario structure. In the worst case, the formation will become a chain, so the path planned by the leader will also feasible for the followers.

5.2.1 Definition of Risk Maps

Let a stochastic feature-based representation (e.g. built by a SLAM algorithm), maybe partial and incomplete, either of the navigation area or of the local cooperative perception of the members of the robot formation, be represented by a set of geometric features $\mathbf{y}_{\mathcal{F}} = \{\mathbf{y}_{F_1}, \mathbf{y}_{F_2}, ..., \mathbf{y}_{F_n}\}$ known wrt a certain reference frame (not superscripted for simplicity). A *risk map* $\Gamma = \{\gamma_{ij}; i = 1, ..., N_i; j = 1, ..., N_j\}$ is defined as the projection of the stochastic feature-based representation into an $N_i \times N_j$ grid-based representation (Fig. 5.1), with common base reference frame, where γ_{ij} is the *risk value* associated to the cell (i, j) and intuitively defined as,

$$\gamma_{ij} \triangleq \text{Prob}\left(\bigcup_{n=1}^{N} \mathbf{y}_{F_n} \text{ is projected on cell } (i, j)\right) \tag{5.1}$$

where high risk values suggest the likely presence of either map features, clutter or dynamic obstacles and therefore the existence of non-traversable areas for the robot formation.

Real time constraints refrain from the exact computation of the risk values given by Eq. 5.1 even for medium-scale environments. Thus, an approximate sample-based strategy is adopted in subsequent paragraphs where the selected number of samples should trade-off real-time constraints and the precision of the sample distribution in approximating the real distribution of the stochastic feature-based representation for a user-defined cell size for the grid-based representation. In our proposal, we use segments as features for the stochastic map and thus efficient algorithms such as [9] reduce the computational requirements of the proposed projection.

5.2.2 Global Paths of Bounded Cell Risk

Let $\mathbf{x}_{R_{0,t_0}}$ and $\mathbf{x}_{R_{0,t_f}}$ be the probabilistic representation of the current location of the robot leader at time t_0 and its desired goal location at time t_f respectively. Also, let $\mathbf{y}_{\mathcal{F}} \sim \mathcal{N}(\hat{\mathbf{y}}_{\mathcal{F}}, \mathbf{P}_{\mathcal{F}})$ be the stochastic feature-based representation of the navigation area, where $\hat{\mathbf{y}}_{\mathcal{F}}$ is a vector containing the estimated location of the map features $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$, and $\mathbf{P}_{\mathcal{F}}$ its associated covariance matrix.

A rough estimate of a safe (or traversable) path π (a set of reachable waypoints) for the robot leader connecting locations $\mathbf{x}_{R_{0,t_0}}$ and $\mathbf{x}_{R_{0,t_f}}$ is computed by an A^*-type method

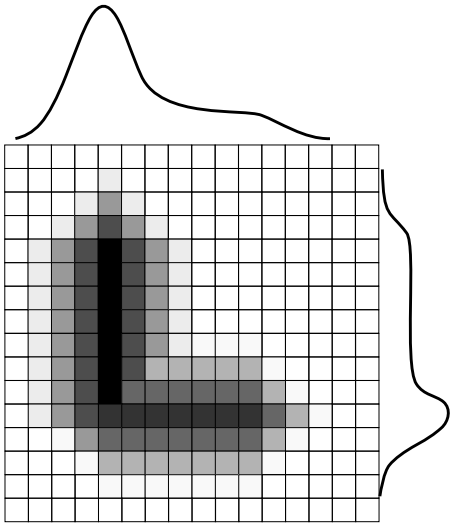


Figure 5.1: Sample risk map obtained as the projection of a stochastic segment-based representation into a grid-based representation. The marginal probability density functions of two features (top and right) are shown together with the associated gray-scaled risk values.

as detailed in Alg. 1. From the feature-based representation of the environment $\mathbf{y}_{\mathcal{F}}$, a risk map Γ is computed (function *riskMap*) as described in section 5.2.1. Then, the algorithm iteratively increases the risk level γ from null risk ($\gamma = 0$) up to maximum risk ($\gamma = 1$) searching for safe paths in the binary risk map Γ_{th} , thresholded at the risk level γ (function *computePath*).

The computation of a global path for the robot leader connecting two locations in the navigation area could be performed off-line in critical situations (e.g. initialization of the robot formation at unknown locations, loss of localization information due to kidnapping, etc.), once recovered from the loss of an estimated location for the robot formation[1]. Also, for on-line computation, the risk level γ used for the projection of the feature-based stochastic map into the grid-based representation could be user defined. Under both approaches, the computed safe path π returned by Alg. 1 constitutes an open-loop (using control theory terminology) solution to the path planning with uncertainty problem requiring frequent replanning in real navigation tasks due to unmodelled effects in the a priori stochastic map or the presence of dynamic objects.

[1]The absence of previous information suggests the formulation of this subproblem from an Extended Information Filter (EIF) perspective. In our work, we generalize the previously reported works [62, 69] to the team $\{R_0, R_1, ..., R_r\}$ of $r+1$ vehicles

Algorithm 1 Global path planning in risk maps

Require: $\mathbf{x}_{R_{0,t_0}}, \mathbf{x}_{R_{0,t_f}}, \mathbf{y}_{\mathcal{F}}$

Ensure: π is a safe path from $\mathbf{x}_{R_{0,t_0}}$ to $\mathbf{x}_{R_{0,t_f}}$ of bounded risk level γ

$\Gamma \leftarrow riskmap(\mathbf{y}_{\mathcal{F}})$
$\gamma \leftarrow 0$ {Initial risk level}
while $\gamma \leq 1$ **do**
  {Decide traversability depending on the risk level}
  $\Gamma_{th} \leftarrow threshold(\Gamma, \gamma)$
  {Find the shortest path at this risk level}
  $\pi \leftarrow computePath(\mathbf{x}_{R_{0,t_0}}, \mathbf{x}_{R_{0,t_f}}, \Gamma_{th})$
  if $\pi \neq \emptyset$ **then**
    {A plan of bounded risk γ is found}
    return $\langle \pi, \gamma \rangle$
  else
    {Goal is not reachable. Increase the risk.}
    $\gamma \leftarrow \gamma + \Delta\gamma$
  end if
end while
{The goal is not reachable at all.}
{Return an empty path and the maximum risk}
return $\langle \emptyset, 1 \rangle$

5.3 Cooperative On-line Replanning

In wide-opened and uncluttered environments once a global path for the robot leader is obtained, the robot followers would execute their paths, from origin t_0 to destination t_f, thanks to the links derived from the virtual spring-damper analogy of the robot formation and its commanded geometrical shape. However, in real settings frequent real-time replanning would be mandatory to avoid unmodeled or dynamic objects in the environment while flexibly and adaptively maintaining the formation structure. Therefore, local safe paths for the robot formation should be computed based on their common understanding of the environment and a sufficiently precise localization of the robots.

5.3.1 Pose Tracking with Observation Improvement

Real time localization of the robot formation within the a priori stochastic map while it executes the computed global path towards the goal can be achieved through the filter-based pose tracking approach proposed in Chapter 4. Recalling from that chapter, after each prediction step (section 4.3.1), robots gather a set of observations $\mathbf{z}_k$ with measurement noise $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. Data association provides a set of pairings $(\mathbf{y}_{\mathcal{F}_k}, \mathbf{z}_k)$ of map features and sensor observations related by the linearized measurement equation 4.14.

As the robots in the formation usually navigate in the same area, a number of redundant observations and with different levels of uncertainty could be found. This issue is illustrated in Fig. 5.2 for a 2D point-based feature map. In order to obtain a unified and lower uncertainty set of observations $\mathbf{z}'_k$, in this section we propose a modified update phase of the MD-EKF algorithm presented in Chapter 4. This observation improvement results from the inclusion of the observations in the state vector and their subsequent association with the features of the prior map within the measurement update of the algorithm.

Then, at time step k, the predicted state of the robot formation $\mathbf{x}_{\mathcal{R}_{k|k-1}}$ is augmented with the set of sensor observations $\mathbf{z}_k$ gathered from all robots,

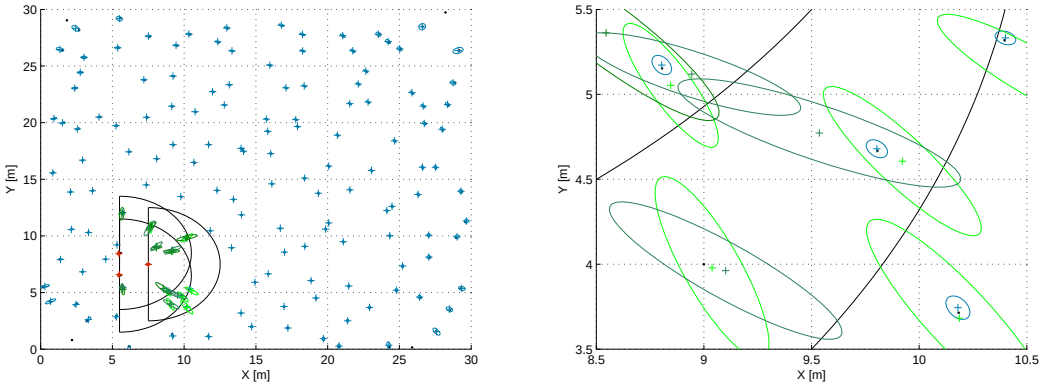
$$\mathbf{x}^a_k = \begin{pmatrix} \mathbf{x}_{\mathcal{R}_k} \\ \mathbf{z}'_k \end{pmatrix}; \hat{\mathbf{x}}^a_{k|k-1} = \begin{pmatrix} \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}} \\ \mathbf{z}_k \end{pmatrix}; \mathbf{P}^a_{k|k-1} = \begin{pmatrix} \mathbf{P}_{\mathcal{R}_{k|k-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{pmatrix} \tag{5.2}$$

We formulate the measurement equation implicitly so that the new observations satisfy the following constraint:

$$\mathbf{f}_k(\mathbf{x}^a_k, \mathbf{y}_{\mathcal{F}_k}) = \mathbf{h}_k(\mathbf{x}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{F}_k}) - \mathbf{z}'_k = \mathbf{0} \tag{5.3}$$

As usual, we approximate the measurement equation by its first order Taylor expansion:

$$\mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{z}_k + \mathbf{H}_{\mathcal{R}_k}(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}) + \mathbf{G}_{\mathcal{F}_k}(\mathbf{y}_{\mathcal{F}_k} - \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{H}_{z_k}(\mathbf{z}'_k - \mathbf{z}_k) = \mathbf{0} \tag{5.4}$$



(a) **(b)**

Figure 5.2: Redundant observations obtained by a robot formation. The zoomed area shows map features (blue) and robot observations (green). Observations from features not contained in the a priori map are also considered (left-down corner). Features are observed either by one, two or the three robots.

where

$$\mathbf{H}_{\mathcal{R}_k} = \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_{\mathcal{R}_k}} = \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_{\mathcal{R}_k}}\bigg|_{(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k})} \tag{5.5}$$

$$\mathbf{G}_{\mathcal{F}_k} = \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}_{\mathcal{F}_k}} = \frac{\partial \mathbf{h}_k}{\partial \mathbf{y}_{\mathcal{F}_k}}\bigg|_{(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k})} \tag{5.6}$$

$$\mathbf{H}_{z_k} = \frac{\partial \mathbf{f}_k}{\partial \mathbf{z}'_k} = -\mathbf{I} \tag{5.7}$$

Note that setting $\mathbf{z}'_k = \mathbf{z}_k - \mathbf{w}_k$ and $\mathbf{H}_{z_k} = -\mathbf{I}$ equation 5.4 derives in the explicit linearized equation 4.14.

Following a similar derivation as in Chapter 4 we obtain the pseudo-measurement $\mathbf{r}_k \triangleq \mathbf{z}_k - \mathbf{\Lambda}_k \mathbf{z}_{k-1}$ considering the augmented state vector,

$$\mathbf{r}_k \simeq \mathbf{h}^*_k + \mathbf{H}^a_k(\mathbf{x}^a_k - \hat{\mathbf{x}}^a_{k|k-1}) + \mathbf{w}_{\mathbf{r}_k} \tag{5.8}$$

with

$$\mathbf{h}^*_k = \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}}) \tag{5.9}$$

$$\mathbf{H}^a_k = (\mathbf{H}_{\mathcal{R}_k} - \mathbf{\Lambda}_k \mathbf{H}_{\mathcal{R}_{k-1}} \mathbf{F}^{-1}_{k-1} \quad \mathbf{H}_{z_k}) \tag{5.10}$$

and the measurement noise $\mathbf{w}_{\mathbf{r}_k}$ with covariance matrix $\mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}}$ is given by,

$$\mathbf{w}_{\mathbf{r}_k} = \mathbf{\Lambda}_k \mathbf{H}_{\mathcal{R}_{k-1}} \mathbf{F}_{k-1}^{-1} \mathbf{v}_{k-1} - \mathbf{\Lambda}_k \mathbf{w}_{k-1} + \mathbf{G}_{\mathcal{F}_k} \mathbf{n}_k \tag{5.11}$$

Finally, the augmented state vector is updated by

$$\hat{\mathbf{x}}_{k|k}^a = \hat{\mathbf{x}}_{k|k-1}^a + \mathbf{K}_k(\mathbf{r}_k - \mathbf{h}_k^*) \tag{5.12}$$

$$\mathbf{P}_{k|k}^a = \mathbf{P}_{k|k-1}^a - \mathbf{K}_k(\mathbf{H}_k^a \mathbf{P}_{k|k-1}^a \mathbf{H}_k^{aT} + \mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} + \mathbf{H}_k^a \mathbf{C}_k + \mathbf{C}_k^T \mathbf{H}_k^{aT}) \mathbf{K}_k^T \tag{5.13}$$

with the filter gain given by

$$\mathbf{K}_k = (\mathbf{P}_{k|k-1}^a \mathbf{H}_k^{aT} + \mathbf{C}_k)(\mathbf{H}_k^a \mathbf{P}_{k|k-1}^a \mathbf{H}_k^{aT} + \mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} + \mathbf{H}_k^a \mathbf{C}_k + \mathbf{C}_k^T \mathbf{H}^{aT})^{-1} \tag{5.14}$$

and the cross-correlation term between $\mathbf{w}_{\mathbf{r}_k}$ and $\mathbf{v}_{k-1}$, namely,

$$\mathbf{C}_k = E[\mathbf{v}_{k-1} \mathbf{w}_{\mathbf{r}_k}^T] = \mathbf{Q}_{k-1}(\mathbf{\Lambda}_k \mathbf{H}_{\mathcal{R}_{k-1}} \mathbf{F}_{k-1}^{-1})^T \tag{5.15}$$

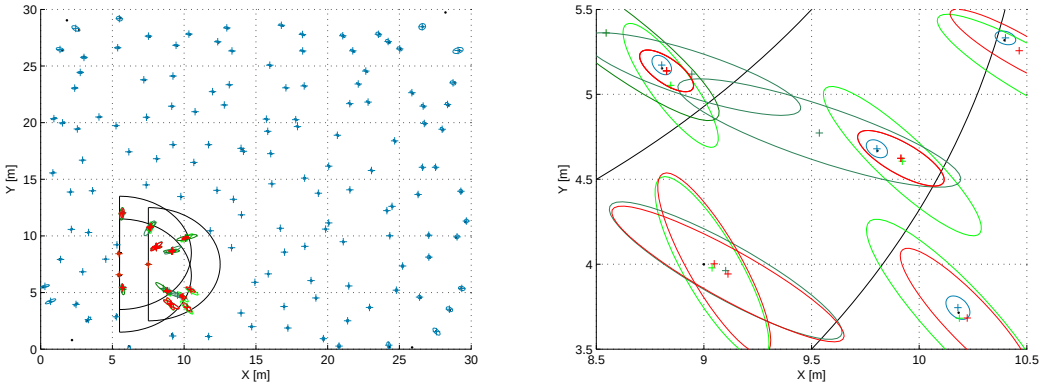
which is augmented with zeros to fit the dimensions of $\mathbf{H}_k^a$.

After this filter stage, the robot localization is updated as in the original formulation of the measurement differencing based EKF presented in Chapter 4 whereas, at the same time, the uncertainty of the observations included in the state vector and matched against the map features is decreased. Additionally, the result obtained for different observations of the same feature is identical as shown in Fig. 5.3. This eases the duplicate detection and fusion process explained in the next section.

5.3.2 Cooperative Local Planning

The performance (e.g., precision, reliability and robustness) of the robot formation to complete the commanded task profits from the close cooperation among the different vehicles of the team emphasizing the importance of a centralized approach (at least at selected synchronization time steps) to consistently and coherently joint the views of the different vehicles. Of paramount importance is the availability of this jointly coherent understanding of the navigation environment, with wider and integrated fields of views, to increase the efficiency and optimality of the on-line replanning process. For example, as illustrated in the experimental section, the best available path towards the next waypoint may be hidden to the robot leader (due to its limited sensorial field of view) but visible to one of the followers, thus profiting the performance of robot formation replanning.

The pose tracking technique of the previous section estimates a set of improved observations $\mathbf{z}_k'$. Nevertheless, the augmented state vector $\mathbf{x}_k^a$ still contains a number of redundant observations of the environment. Therefore, a duplicate detection and observation fusion procedure is performed in the leader reference frame. The geometrical



(a) **(b)**

Figure 5.3: Observation improvement after the filter update. The original measurement covariances (green) are reduced to the red ones which overlap in the case of redundant observations of the same feature (e.g., left-upper feature). Non matched observations are not improved after this process (e.g., left-down observations).

constraints between observed features namely, $\mathbf{z}_l^{R_i}$ and $\mathbf{z}_m^{R_j}$ expressed wrt the reference frame of robot R_i and R_j respectively, could be formulated as,

$$\mathbf{h}(\mathbf{x}_{R_i}^{R_0}, \mathbf{z}_l^{R_i}) = \mathbf{h}(\mathbf{x}_{R_j}^{R_0}, \mathbf{z}_m^{R_j}) \tag{5.16}$$

Subsequently, a statistical test based on the Mahalanobis distance provides jointly consistent matchings and duplicated observations are removed from the augmented state vector. This process is straightforward for 2D point features, since, as shown in previous section (Fig. 5.3b), observations matched against the same map features are identical at this stage. In the case of higher level features, like line segments, a merging step would take place. Furthermore, common observations from non-mapped features (and thus, not updated in the filtering process of previous section) included in the state vector can also be constrained within an EKF-update step, where the linearization of Eq. 5.16 would be used. As a result, a reduced, reliable, lower-uncertainty set of integrated sensor observations $\mathbf{z}_k''$ characterizing the common understanding of the formation environment is obtained. Figure 5.4 shows the result obtained with this procedure in the 2D point feature based example.

Afterwards, a local risk map Γ_{R_0} in the reference frame of the robot leader (thus leader-centric) is obtained by the projection of $\mathbf{z}_k''$ into a grid-based representation spanning the joint field of view of the robot formation.

The motion planning is similar to the one outlined in section 5.2 and described in more detail in [83]. It computes a local safe path π_{local} for the robot leader in Γ_{R_0} for

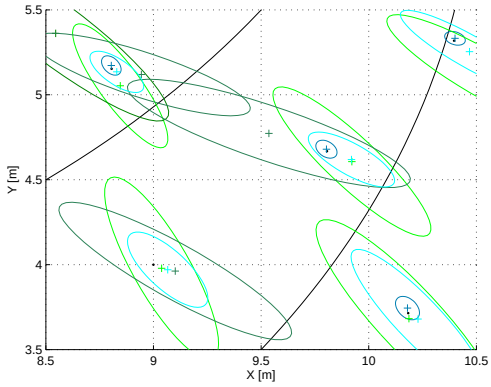


Figure 5.4: Final set of reduced-uncertainty integrated observations (cyan).

a risk level γ. But in this case, the cost function for the A* algorithm is computed in a different way. As a multi-leg journey, the path π_{local} is formed by an ordered sequence of cells to be visited $\{c_1,...,c_n\}$ that minimizes the total weighted distance from the initial to the destination locations, where the weighting coefficient for traveling from cell c_l to cell c_{l+1} is related to the risk value of the cell c_{l+1} and computed as $\alpha_{l+1} = 1/(1-\gamma_{l+1})$. Thus, the local safe path π_{local} results from the minimization of,

$$\pi_{local} = \text{argmin} \sum_{l=1}^{n-1} \alpha_{l+1}\, dist(c_l, c_{l+1}) \tag{5.17}$$

So, the algorithm takes into account both the distance to the goal and the accumulated risk along the path.

5.4 System Overview

Figure 5.5 depicts the hybrid centralized-distributed architecture of our integrated system with indication of the different modules, data flows and execution threads in the presence of wireless RT-WMP communication. Centralized execution refers to the robot leader thread whilst distributed execution refers to the robot followers threads.

Additionally, algorithms 2 and 3 present the pseudo-code of the algorithms executed by the robot leader and the robot followers respectively. The *leaderNavigationStrategy* and *moveLeader* functions track the local safe path resulting from Eq. 5.17 thanks to the control strategy described in Chapter 3. Furthermore, functions *followerNavigationStrategy* and *moveFollower* compute, distributively, the motion command for each robot follower in the formation. These two functions are directly adapted from the strategy selection and the leader and follower strategies defined in [83]. As mentioned above,

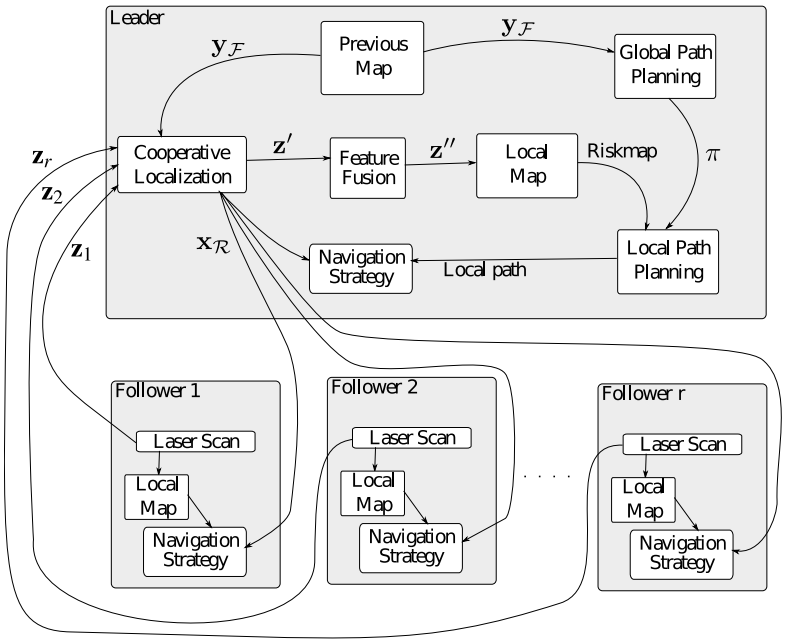


Figure 5.5: Overview of the hybrid centralized-distributed architecture of the reported integrated system. The schema depicts modules, relations and data flows, and execution threads (grey-shaded boxes). Data shared between different robots require the use of a communication protocol.

the trajectories of each of the members of the robot formation adapt to the structure (e.g. confined, wide-open, etc) and dynamics (e.g. moving objects, unexpected clutter, etc) of the navigation area maintaining the user-commanded geometry for the robot formation.

5.5 Experimental Results

In this section we report both simulated and real experiments obtained by a robot formation in the presence of sensing, localization and mapping uncertainties as described along the previous sections. We first illustrate, from a simulation-based perspective (Player-Stage simulation environment): (i) the computation of global plans; (ii) the adaptability of the geometric structure of the robot formation to the dynamics of the environment; and (iii) the benefits of cooperative perception in replanning tasks. Then, in a real setting, we show the performance of the complete integration scheme in a navigation task commanded to a three-robot formation in an indoor scenario.

Algorithm 2 Leader algorithm

{Algorithm for the robot leader R_0}
Require: $\mathbf{x}_{R_0,t_0}, \mathbf{x}_{R_0,t_f}, \mathbf{y}_{\mathcal{F}}$
$\pi \leftarrow computeGlobalPath(\mathbf{x}_{R_0,t_0}, \mathbf{x}_{R_0,t_f}, \mathbf{y}_{\mathcal{F}})$
while last waypoint of π not reached **do**
$\mathbf{z} \leftarrow gatherObservationsFromRobots(R_0, ..., R_r)$
$\langle \mathbf{x}_{\mathcal{R}}, \mathbf{z}' \rangle \leftarrow poseTracking(\mathbf{y}_{\mathcal{F}}, \mathbf{z})$
$\mathbf{z}'' \leftarrow cooperativePerception(\mathbf{z}')$
$\Gamma \leftarrow buildLocalRiskMap(\mathbf{z}'')$
$sendInfoToFollowers(\mathbf{x}_{\mathcal{R}})$
$\pi_{local} \leftarrow computeLocalPath(\Gamma, \pi)$
$S_{leader} \leftarrow leaderNavigationStrategy(\mathbf{x}_{\mathcal{R}}, \pi_{local})$
$moveLeader(S_{leader})$
end while

Algorithm 3 Followers algorithm

{Algorithm for robot follower R_i}
while R_0 keeps moving **do**
$\mathbf{z}_i \leftarrow gatherObservationsFromSensors(R_i)$
$sendObservationsToLeader(\mathbf{z}_i)$
$\mathbf{x}_{\mathcal{R}} \leftarrow getLocFromLeader(R_0)$
$S_{follower_i} \leftarrow followerNavigationStrategy(\mathbf{x}_{\mathcal{R}}, \mathbf{z}_i)$
$moveFollower(S_{follower_i})$
end while

5.5.1 Global Path Planning

Figure 5.6a displays a stochastic map of an office-like environment (note the sharp edges at the top left part of the figure and blurred edges at the bottom right due to the greater distance to the base reference frame). This map was constructed previously to the navigation execution. Two possible global paths connect the initial location of the robot formation (labelled *Start*) and its commanded final destination (labelled *Goal*): (i) a snake-like path through a long curved corridor, or (ii) a smooth path traversing a door (labeled in the figure) unobservable from the initial location.

As shown in figures 5.6c and 5.6d the algorithm has a significant sensitivity to the selected risk level (0.1 and 0.05 respectively) that dramatically changes the planned path. By setting the risk level to 0.1 the path planner selects a shorter but more risky path (figure 5.6c) traversing the door, unobservable from the initial location and with a significant location uncertainty computed during the prior mapping stage. Conversely, setting the risk level to 0.05 results in a more conservative strategy, that avoids traversing the uncertain door, with a longer but safer path (figure 5.6d) to the goal destination.

Planning a risky path from a global perspective (figure 5.6e red line traversing the uncertain door) may lead to replanning during the execution, and therefore longer than expected paths to the goal destination, suggested by the local planner due to the observations gathered by on-board sensors. Once the leader of the formation has reached the top right entrance of figure 5.6f (a zoomed view of figure 5.6e), the suggested path turns into a dead-end forcing a replanning action and therefore a much longer path.

Finally, it is worth mentioning that, given two different stochastic maps (figures 5.6a and 5.6b) a closely related risk map (figure 5.6d) could be computed from the correct selection of risk levels. Therefore, in a real setting, it would be difficult to differentiate between highly uncertain open spaces and non-traversable spaces.

5.5.2 Formation Adaptability

Figure 5.7 shows a typical office-like indoor environment where a five-robot formation is commanded to navigate from an initial location (labelled *Star*t) towards a final destination (labelled *Goal*). Thanks to the robot formation integrated control scheme, the robots compliantly maintained the user-defined geometric structure (pentagon-shape in this case) along the way.

The system computes the safest global path (in the risk map associated to the navigation area) between the start and the goal locations that is dynamically updated by cooperative replanning thanks to the different observations of the individual vehicles. Note that vehicles align to traverse the different doors whilst they tend towards the commanded geometric shape in wide-open areas.

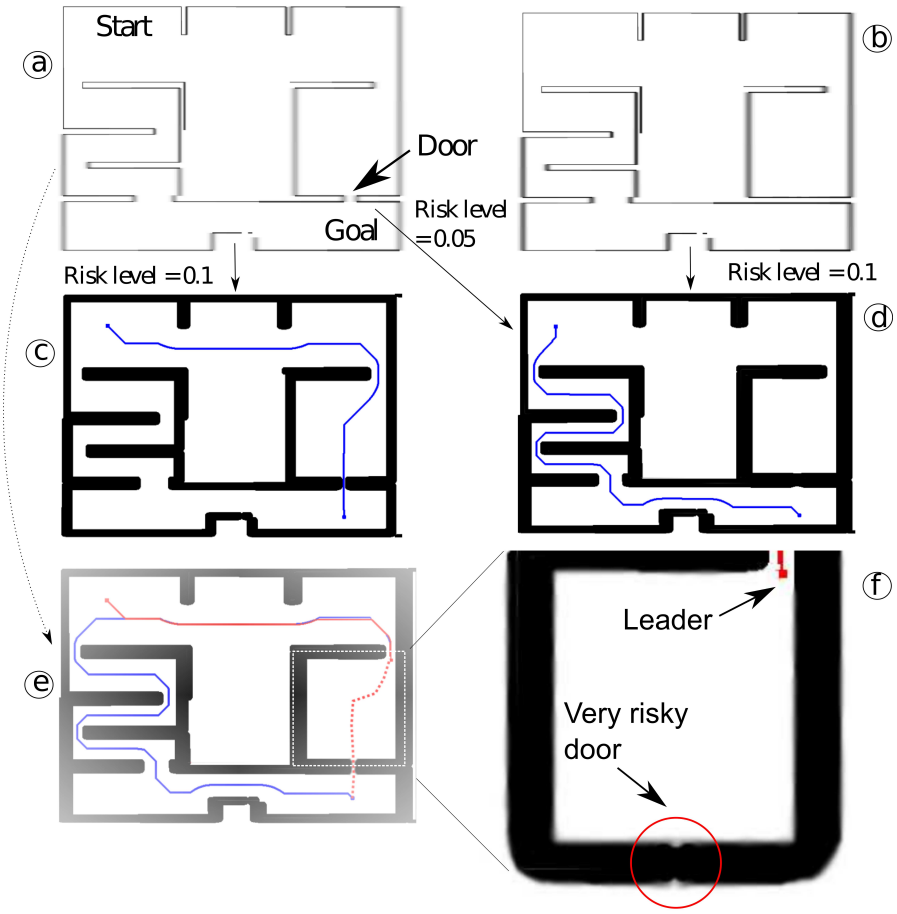


Figure 5.6: Influence of uncertainty in risk-based global path planning for the robot leader of the formation.

5.5.3 Cooperative Online Replanning

Figure 5.8 reports the behavior of the five-robot formation in an open-door event detected by one of the robot followers and transmitted to the robot leader thanks to the cooperative perception strategy reported above. Initially, the global path planning algorithm drives the robot formation from the initial location to the final location (Fig. 5.7) through the way-point $\mathbf{x}_{G_1}$. Then, while navigating towards $\mathbf{x}_{G_1}$ a shortest path to the goal destination appears when the labelled door (Fig. 5.7) becomes wide open but, unfortunately, this event occurs outside the field of view of the robot leader. Nevertheless, cooperative replanning thanks to the common understanding of the navigation area (Fig. 5.8a) allows the correction of the previously planned path profiting the robot formation from the short-cut to the goal.

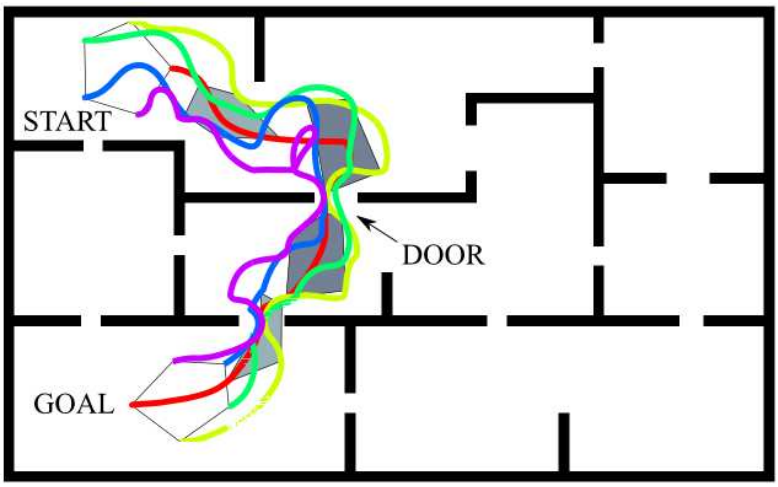
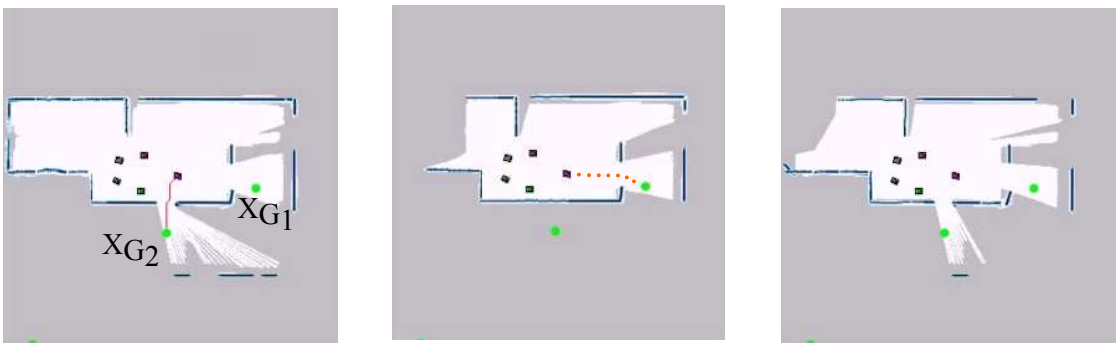


Figure 5.7: Five-robot formation adaptability to a simulated indoor environments. From the individual traces of the different vehicles we observed that they compliantly maintained the user-defined geometric structure (pentagon-shape in this case) whilst navigating through narrow and wide open spaces towards the commanded goal destination.

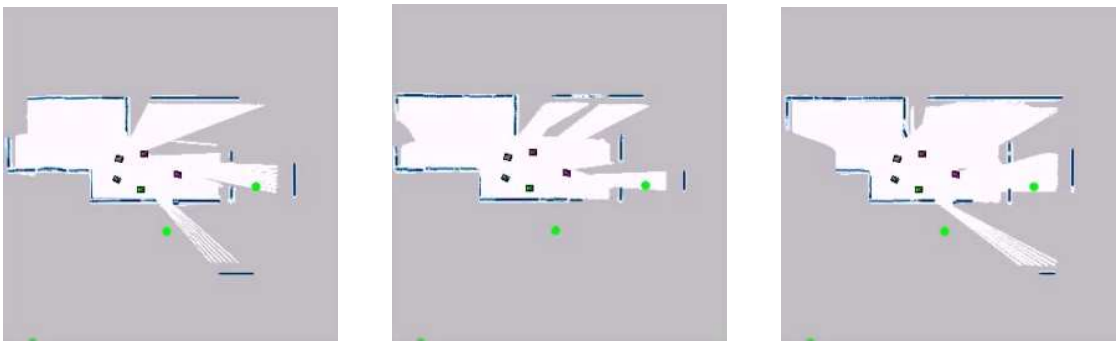
5.5.4 Experiments in Real Scenarios

We have experimented with a three-robot Pioneer 3-AT formation equipped with on-wheel encoders and SICK LMS-200 laser rangefinders with a 180-deg field of view. Communication between the vehicles is achieved by a dedicated channel using a real time wireless multi-hop protocol [79], which includes communication queues to avoid synchronization failures. The hybrid centralized-distributed implementation of the complete system reported computation times of the order of the laser scan cycle (aprox. 4.5 Hz) where a maximum of 100 samples have been used to compute the risk maps during cooperative replanning. The similarity between the real and the sampled distributions has been measured in terms of both the Kullback-Leibler Divergence (KLD) [19] and the Normalized Estimation Error Squared (NEES) [6]: In the case of 100 samples, the KLD equal 34,7 nats that represents a 26% improvement over 10 samples and it is only improved by 3% in the case of 1000 samples. Also, the consistency of the approximation (χ^2 -test based on NEES) amounts for 98,4% for 100 samples, with an improvement of 33,6% with respect to 10 samples, and only improved by the use of 1000 samples in a roughly 1,5%.

Figure 5.9 illustrates the stochastic feature-based representation of a medium-size office-like environment (aprox. 30m$\times$40m) where the global path planner have been executed to compute a safe path from the initial location of the robot formation (labelled *Start*) towards the final destination (labelled *Goal*). A longer but safer path (in solid-red) is selected by the algorithm instead of the shortest path (in dashed-blue) due to the high risk area (risk map enlarged on the left of the figure) due to the presence of clutter in the



(a) Cooperative view **(b)** Leader view **(c)** Follower 1 view



(d) Follower 2 view **(e)** Follower 3 view **(f)** Follower 4 view

Figure 5.8: Cooperative replanning towards the goal destination $\mathbf{x}_{G_2}$. The figure shows the integrated field of view in the reference frame of the leader (a) and the different individual perspective of each robot in the formation. Figure (b) shows the erroneous path planned from the perspective of the robot leader in the absence of cooperation with other team members.

corridor.

Furthermore, Fig. 5.10 illustrates the real-time capabilities of the reported strategy for efficient cooperative replanning under the event of a door-opening. As shown, a three-robot triangular-shaped formation is commanded from $\mathbf{x}_{G_1}$ towards $\mathbf{x}_{G_5}$ via the sequence of ordered waypoints. During the execution, a door, mapped as closed in the a priori map and outside of the field of view of the robot leader is opened and observed by the robot followers, therefore the reported cooperative strategy replans the path towards the goal destination skipping the waypoint $\mathbf{x}_{G_3}$ and $\mathbf{x}_{G_4}$.

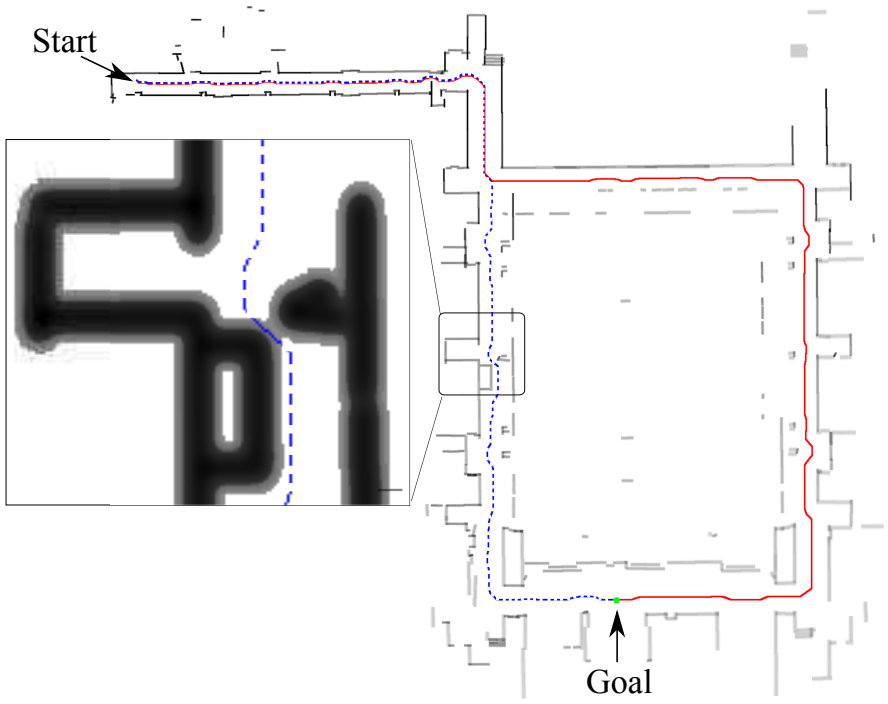
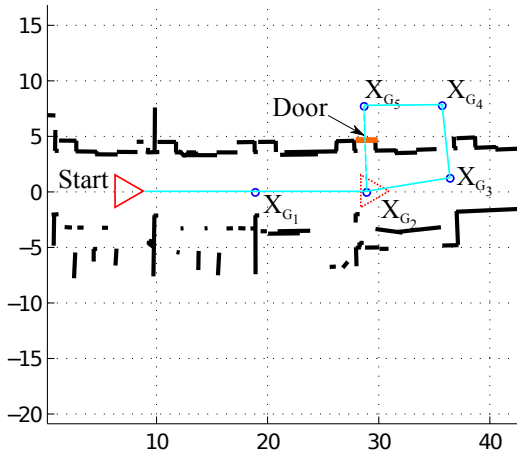


Figure 5.9: Safety enhanced path planning. The algorithm favors the larger but safer path (red) against the shortest path (red) due to the unsafe area on the middle of the corridor (risk map enlarged on the left of the figure).

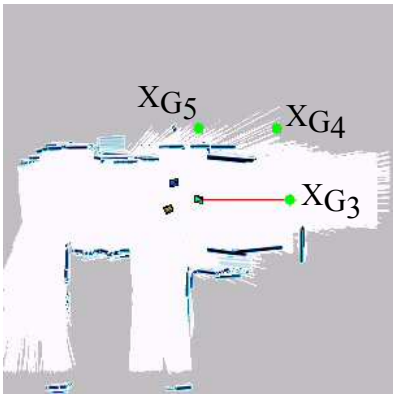
5.6 Conclusions

This chapter has reported an integrated system for robot formations working in partially known environments.

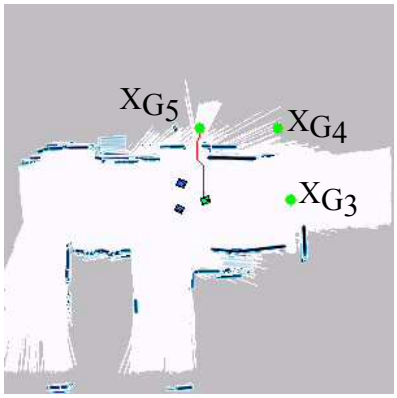
Following the framework established in previous chapters, where we addressed the cooperative navigation in formation and its localization within an uncertain prior map, in this chapter we have turned our attention to path planning tasks with the goal of achieving an autonomous robot formation system.



(a) Cooperative replanning scenario.



(b) Planning towards $\mathbf{x}_{G_3}$.



(c) Planning towards $\mathbf{x}_{G_5}$.

(d) Moving along the corridor.

(e) Heading to the opened door.

Figure 5.10: Cooperative replanning in a real-world scenario. (a) A three-robot triangular-shaped formation (solid red) is commanded from $\mathbf{x}_{G_1}$ towards $\mathbf{x}_{G_5}$ via the sequence of ordered waypoints (solid blue line). During the execution, a door (labelled) outside of the field of view of the robot leader is opened and observed by the robot followers (dotted red triangle), therefore the reported cooperative strategy replans the path towards the goal destination (dashed blue line) skipping the waypoints $\mathbf{x}_{G_3}$ and $\mathbf{x}_{G_4}$. The cooperative local maps before and after the door opening event are shown in (b) and (c) together with captures of the real experiment in (d) and (e).

We have seen the existing risk when planning through high uncertain areas, which might correspond to non-traversable areas in a real setting, forcing a complete replanning of the path to follow to reach the goal. Then, we have presented global and online path planning strategies to limit the possibility of failure in the execution of the computed plan. The cooperative perception enhances the online performance by expanding the leader field of view to new open spaces that can lead to shortening the trajectory towards the goal.

Experimental results, both in simulation and in realistic medium-size office-like settings, have illustrated the performance of the described approach by using a hybrid, centralized-distributed, architecture with wireless communication capabilities achieving a 4.5 Hz cycle-time for a three-robot triangular-shaped formation.

Chapter 6

Distributed SLAM for Robot Formations using a Prior Map

Previous chapters presented a centralized algorithm to localize the robots in the formation within an a priori map of the environment which remained unchanged during the execution of the mission. In this chapter, we turn our attention to how the robots can jointly improve this prior stochastic map in an efficient way. Additionally, we seek to enhance the system robustness by using a distributed scheme where the computational overhead is shared by all robots. Each robot maintains its own local and global maps which are improved with the information received when communications among robots take place. The use of conditional independence properties allows that, after the synchronization steps, each robot has exactly the same information about the map and about the location of the robots at its disposal.

6.1 Introduction

In previous chapters, we have proposed an EKF-based approach to localize a team of robots within an *a priori* stochastic map of the environment. The method is initially designed in a centralized way in order to optimally integrate all observations gathered by the robots and to have a global vision of the system. However, the whole system heavily relies on the robot leader, which is not exempt from failures, and demands constant communication between robots to maintain their positions updated.

In order to improve robustness and scalability of the system, in this chapter we move towards a distributed paradigm in which each robot of the formation is able to localize itself in the previous map. Additionally, in this chapter we propose the robots coordinately improve the given stochastic map turning our goal into solving a distributed multi-robot SLAM problem.

To this end, in this chapter we present a novel and efficient algorithm where each

robot updates its own copy of the prior global map by sending/receiving local information to/from the rest of the team. Our distributed estimation algorithm is described in terms of a Gaussian Markov Random Field (GMRF) which allows us to analyze the conditional independence (CI) properties of the problem to achieve high efficiency [7]. The application of the CI property in the algorithm is twofold. On the one hand, the robots in the formation only need to constantly update the local region in which they move whereas global updates can be postponed reducing computational cost [72]. On the other hand, robots do not make observations of each other in our system but get indirectly related by observing common map features. The load in the communication channel is then reduced since each robot just sends an information summary of features observed since last communication. As a result, the algorithm proposed does not rely on a central server improving flexibility and robustness and reduces the computational and communication requirements at each step. Moreover, it is shown that, in a linear filtering context, the resulting decoupled method produces the exact results than using only one filter. In this chapter, we use the Extended Information Filter (EIF) as the core of the distributed algorithm such that after all messages are sent/received each robot's estimate is equal to the centralized solution.

Distributed localization can also be addressed if a map is not available to the robots. For example, in [4], a decentralized EIF algorithm to jointly localize a team of robots is presented. Each robot integrates its own data and cooperative localization is performed through inter-robot measurements fused by a central server which can be replicated in each robot for robustness. In [76], the equations of a centralized EKF are decoupled and distributed among the robots who share their information when they see each other and in [63] the distributed localization is formulated as a Maximum a Posteriori problem. Unlike these previous works and as we did in previous chapters, we assume the robots do not have the capability of detecting each other and cooperative localization is not achieved by the use of inter-robot measurements but through observation of features from a common prior map.

One typical problem affecting distributed systems is the *double counting* of information (also known as *data incest*). This problem arises when the data sent through the network is not properly managed and is integrated more than once by the robots. To solve this problem, the Channel Filter (CF) is introduced in [30]. The CF prevents double-counting of information by using a tree communication topology (i.e., with no loops) and by keeping a record of the information transmitted over the communication channel. The transmission of information in our algorithm is similar to this filter but we avoid double-counting by synchronizing the transmission of messages. The main difference with that work is that we implement in addition an efficient algorithm to update the state vectors of the robots by postponing global updates without introducing approximations.

In the same multi-robot context, recent works also address the distributed SLAM

problem. In [54] each robot computes its own centralized-equivalent estimate of the system using the odometry and measurements that have been received from other robots. In [14] a Rao-Blackwellized Particle Filter is implemented as estimation kernel that works in simple scenarios with unknown initial correspondences. Each time a pair of robots communicate they have to calculate their relative transformation and interchange all the information gathered since the last meeting. The work presented in [64], copes with bandwidth communication requirements, in such a way that each robot selects a set of features with the greatest information gain to be sent. To avoid the double counting information problem, they combine an EIF with the Convariance Intersection algorithm, a sub-optimal filter which guarantees consistency although yielding pessimistic estimates. In contrast to these works, in our approach the robots only communicate their positions and the new local information gathered since last synchronization using small local matrices that reduce the communication bandwidth.

The rest of the chapter is organized as follows: Section 6.2 describes the main steps and notation of our distributed localization and mapping algorithm for robot formations with a prior map. Section 6.3 makes use of the CI property to reduce the computational cost for each robot when working on a local region of the global map. Section 6.4 is devoted to explain the message passing protocol which also takes into account the CI property. The complete distributed localization and mapping algorithm is described in section 6.5 together with a computational and communication complexity analysis. In section 6.6 the results of the testing experiments are presented.

6.2 Problem Statement

Given a previously built feature-based stochastic map $\mathbf{x}_{\mathcal{F}}$, our goal is to localize a formation of $r+1$ robots in it while improving the map estimate. The robot formation and the map are given by $[\mathbf{x}_{\mathcal{R}}^T \mathbf{x}_{\mathcal{F}}^T]^T$, where $\mathbf{x}_{\mathcal{R}} = \{\mathbf{x}_{R_j} | j = 0 \dots r\}$ contains the location of the team of robots.

A naive approach to update the joint estimate is to use a centralized method where a leader fuses all the odometry and sensor measurements gathered by the rest of the team. Instead, in this chapter we implement a distributed algorithm in which each robot j estimates its own pose $\mathbf{x}_{R_j}$ and updates its own estimate of the map $\mathbf{x}_{\mathcal{F}}^j$. Robots periodically broadcast the map information acquired since their last synchronization together with their position to maintain the shape of the formation while navigating towards the goal. During the periods of lack of communication, each robot predicts the position of the rest of the team based on its knowledge about the formation structure. The advantage of this distributed strategy over the centralized one is that the system becomes more robust since each robot keeps an estimate of its position and the global map and does not depend on the availability of a central server or constant communication with the rest of the team to maintain the shape of the formation.

The efficiency of the algorithm proposed in terms of computational cost and communication bandwidth is based on the following two ideas:

1. In order to reduce the computational cost, the prior map $\mathbf{x}_{\mathcal{F}}$ is divided into local working areas. While the formation traverses a local region $\mathbf{x}_{F_l}$ each robot updates the features in it using the measurements gathered at each step whereas the rest of the map $\mathbf{x}_{F_g}$ is not modified. When the formation moves to a new local area all robots update the previous unmodified features $\mathbf{x}_{F_g}$, assign the new set of local landmarks to $\mathbf{x}_{F_l}$ and repeats the same procedure. Since the number of features in a local region is bounded, this algorithm maintains a constant computational cost $\mathcal{O}(1)$ when working in a local area. In addition, we will show that the map estimate obtained is the same as if we had been working with the whole map.

2. When working in a local region, robots send messages to each other at synchronization steps to improve their own local estimates by using the information gathered by other members of the team. During these synchronization steps, we assume all-to-all communication availability. Instead of transmitting raw measurements, robots send information matrices of the features observed since last synchronization reducing the amount of information on the communication channel. After each synchornization step, each robot has the same information such that the map estimates coincide with the one obtained using a centralized version, i.e. $\mathbf{x}^j_{\mathcal{F}} \rightarrow \mathbf{x}_{\mathcal{F}} | \, j = 0 \ldots r$.

Previous ideas are based on the Conditional Independence (CI) property of the variables involved in the estimation. In section 2.2.2 we give a brief and intuitive review of this property which will be applied in the explanation of the two following sections. Along each description we will make use of a MRF similar to Fig. 2.4 to help the reader recognize the CI property.

6.3 Local Estimates and Global Updates

In this section we focus on how each robot works in a local region and is able to update the prior map. At instant k, the state vector for a robot j is given by $\mathbf{x}^T_{j,k} = [\mathbf{x}^T_{R_{j,k}} \mathbf{x}^T_{\mathcal{F}}]^T$, i.e the current robot position and the estimate of the map. During this explanation we will omit the robot index j to simplify the notation. We assume a multivariable Gaussian distribution on the state vector described by the information vector $\mathbf{i}_k$ and matrix $\mathbf{I}_k$.

In order to work in local regions with a consistent procedure for global map updates, we distinguish three operations: GlobalToLocal, Local EIF and LocalToGlobal. We will use the Gaussian Markov Random Field (GMRF) in Fig. 6.1 to show the application of the conditional independence property in these operations.

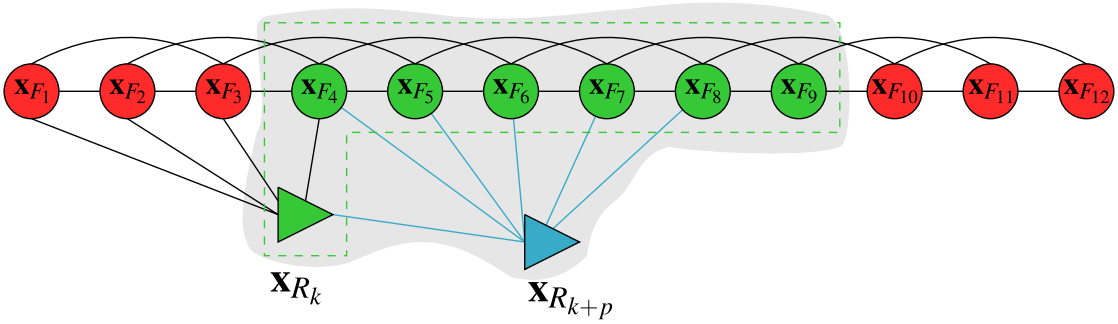


Figure 6.1: GMRF of the individual robot estimation process. The local map corresponds to elements inside the shadowed region. Since there is no link between robot $\mathbf{x}_{R_{k+p}}$ and features F_g, the initial robot position at instant k and the local features $F_l = \{F_4 \dots F_9\}$ make the robot at instant $k+p$ conditionally independent of map elements $F_g = \{F_1 \dots F_3, F_{10} \dots F_{12}\}$. To easily verify the CI property node colors have been selected to match those in Fig. 2.4. Also, the common separator is surrounded by a dash line.

6.3.1 Global to Local

The prior map of the example depicted in Fig. 6.1 is composed by feature elements $\mathbf{x}_{\mathcal{F}} = \{\mathbf{x}_{F_1} \dots \mathbf{x}_{F_{12}}\}$. Notice that features are already connected since they come from a prior map estimation. For a clear representation, the GMRF in Fig. 6.1 is simplified by dropping some inter-feature links. Instead of working with the whole map $\mathbf{x}_{\mathcal{F}}$ we want to work in the local region shadowed in the figure. At instant k, the robot $\mathbf{x}_{R_k}$ is about to enter to this local region from an already updated global map whose information matrix and vector are given by $\mathbf{I}_k$ and $\mathbf{i}_k$ respectively. The new local state vector will be $\mathbf{x}_{l,k} = [\mathbf{x}_{R_k}^T, \mathbf{x}_{F_l}^T]^T$ formed by the current robot position and the feature subset $F_l = \{F_4 \dots F_9\}$. Features that are not in the local region correspond to elements $F_g = \{F_1 \dots F_3, F_{10} \dots F_{12}\}$. To obtain the local region $\mathbf{x}_{l,k}$ we just marginalize it from the joint distribution as shown in Eq. 6.1,

$$p(\mathbf{x}_{R_k}, \mathbf{x}_{F_l}) = \int p(\mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{x}_{F_g}) d\mathbf{x}_{F_g} \tag{6.1}$$

For Gaussian distributions the marginal is given by the Schur complement [7], obtaining the local marginal information $\mathbf{i}_{l,k}^m, \mathbf{I}_{l,k}^m$ from the global state $\mathbf{i}_k, \mathbf{I}_k$ as it is shown in Algorithm 4. A copy of this marginal at instant k is stored for future use in the distributed algorithm.

6.3.2 Local EIF

While operating in the local region $\mathbf{x}_l$ the robot just carries out a standard EIF algorithm [88] in which the initial robot pose is kept in the state vector (i.e. $\mathbf{x}_{R_k}$ is not marginalized

Algorithm 4 $(\mathbf{I}^m_{l,k}, \mathbf{i}^m_{l,k}) =$ **GlobalToLocal** $(\mathbf{I}_k, \mathbf{i}_k)$

$$\mathbf{I}_k = \begin{pmatrix} \mathbf{I}_l & \mathbf{I}_{lg} \\ \mathbf{I}_{gl} & \mathbf{I}_g \end{pmatrix}, \mathbf{i}_k = \begin{pmatrix} \mathbf{i}_l \\ \mathbf{i}_g \end{pmatrix} \tag{6.2}$$

{*Marginalization of local submap elements from the global state*}

$$\mathbf{I}^m_{l,k} = \mathbf{I}_l - \mathbf{I}_{lg}\mathbf{I}_g^{-1}\mathbf{I}_{gl} \tag{6.3a}$$
$$\mathbf{i}^m_{l,k} = \mathbf{i}_l - \mathbf{I}_{lg}\mathbf{I}_g^{-1}\mathbf{i}_g \tag{6.3b}$$

{*Return Marginal distribution at instant k*}

out). Then, after p steps $\mathbf{x}_{l,k+p} = [\mathbf{x}^T_{R_{k+p}}, \mathbf{x}^T_{R_k}, \mathbf{x}^T_{F_l}]^T$ and the probability distribution of the global map can be factorized as follows:

$$p(\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{\mathcal{F}} | \mathbf{z}_{1:k+p}) = p(\mathbf{x}_{F_g} | \mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{z}_{1:k+p}) p(\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l} | \mathbf{z}_{1:k+p}) \tag{6.4}$$

The second factor $p(\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l} | \mathbf{z}_{1:k+p})$ corresponds to the probability distribution of the local region. Notice that there is no direct link between $\mathbf{x}_{F_g}$ and $\mathbf{x}_{R_{k+p}}$ in Fig. 6.1 therefore the first factor can be simplified by using the CI property as follows:

$$p(\mathbf{x}_{F_g} | \mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{z}_{1:k+p}) = p(\mathbf{x}_{F_g} | \mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{z}_{1:k}) \tag{6.5}$$

Consequently, when the robot performs the move-sense local cycle, the global part $\mathbf{x}_{F_g}$ remains conditionally independent of the current local robot $\mathbf{x}_{R_{k+p}}$ and the new observations $\mathbf{z}_{k+1:k+p}$ that have been gathered in the local region and therefore does not require continous updates.

6.3.3 Local to Global

When the robot is about to change to a new local region at step $k+p$, it first updates the elements of its total map $\mathbf{i}_k$ and $\mathbf{I}_k$. We take advantage again of the fact that the new local information acquired during last p steps only affects the elements that correspond to the local region. Therefore, features in $\mathbf{x}_{F_g}$ are conditionally independent of measurements $\mathbf{z}_{k+1:k+p}$ (Eq. 6.5). In terms of EIF, this statement allows us to easily recover the new information given by $\mathbf{I}^{new}_{l,k+1:k+p}, \mathbf{i}^{new}_{l,k+1:k+p}$ (Algorithm 5, Eq. 6.6) from the subtraction of the current local map at instant $k+p$ and the local map at instant k augmented with zeros

at the position of robot $\mathbf{x}_{R_{k+p}}$ to fit the dimensions. This information is finally added to $\mathbf{i}_k$ and $\mathbf{I}_k$, the global state at step k, to update the total estimate. Equation 6.7 shows this operation where $\boxplus$ is the operator in charge of adjusting the dimensions of the matrices and vectors for a coherent addition of information to the common local elements. Once the total estimate is updated, we marginalize out the oldest robot position $\mathbf{x}_{R_k}$. Similarly to Eq. 6.3, we use the Schur Complement to perform the marginalization.

Algorithm 5 $(\mathbf{I}_{k+p}, \mathbf{i}_{k+p}) =$ **LocalToGlobal** $(\mathbf{I}^m_{l,k}, \mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k+p}, \mathbf{i}^m_{l,k+p}, \mathbf{I}_k, \mathbf{i}_k)$

{*Information due to the measurements obtained during last p steps*}

$$\mathbf{I}^{new}_{l,k+1:k+p} = \mathbf{I}^m_{l,k+p} - \mathbf{I}^m_{l,k} \tag{6.6a}$$
$$\mathbf{i}^{new}_{l,k+1:k+p} = \mathbf{i}^m_{l,k+p} - \mathbf{i}^m_{l,k} \tag{6.6b}$$

{*Global Update: adding new information*}

$$\mathbf{I}_{k+p} = \mathbf{I}_k \boxplus \mathbf{I}^{new}_{l,k+1:k+p} \tag{6.7a}$$
$$\mathbf{i}_{k+p} = \mathbf{i}_k \boxplus \mathbf{i}^{new}_{l,k+1:k+p} \tag{6.7b}$$

{*Marginalize* $\mathbf{x}_{R_k}$ *out, the oldest position*}
$(\mathbf{I}_{k+p}, \mathbf{i}_{k+p})$ = `marginalizeOut`$(\mathbf{x}_{R_k}, \mathbf{I}_{k+p}, \mathbf{i}_{k+p})$
{*Return Global state vector at instant* $k+p$}

6.4 Passing Messages between the Robot Formation

We will make use of the CI property to efficiently send and receive update messages between the team of robots. The key idea is that as the robots do not observe each other they just get related by measurements of common map features. This indirect relation means that the robot formation is CI given the map. Formally, this insight is represented by the following equation:

$$p(\mathbf{x}_{R_{0,k}}, \ldots, \mathbf{x}_{R_{r,k}} | \mathbf{x}_{\mathcal{F}}) = p(\mathbf{x}_{R_{0,k}} | \mathbf{x}_{\mathcal{F}}) \ldots p(\mathbf{x}_{R_{r,k}} | \mathbf{x}_{\mathcal{F}}) \tag{6.8}$$

In Fig. 6.2 we can see a small example that illustrates this property. In the example (figure left) three robots use information of their individual odometry to move from instant k to $k+1$. They also get connected to some map features given the observations at both

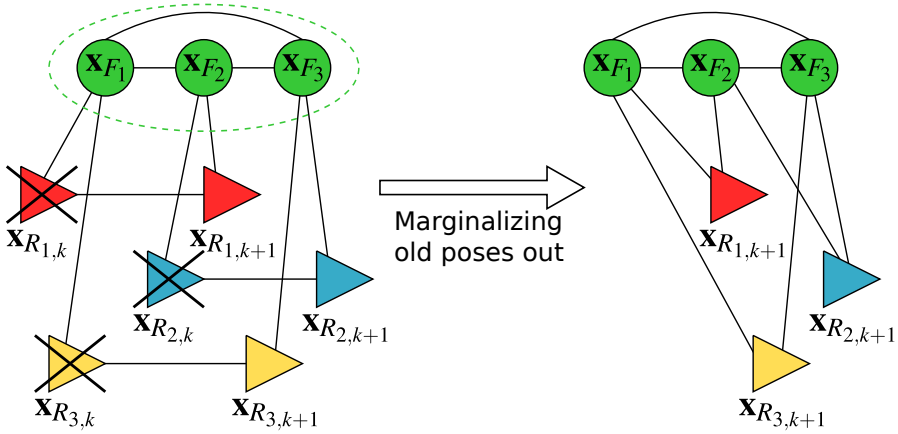


Figure 6.2: GMRF of the robot formation. The example shows three robots that make observations of some map features during two consecutive steps. Robots are only related through the features, i.e. they are conditionally independent given the map (green nodes). Previous robot positions $\mathbf{x}_{R_{j,k}}$ but the first one are marginalized out (left). Red, blue and yellow nodes are separated by green nodes and therefore they are CI as in Fig. 2.4.

steps. Since we are using a filtering paradigm, robot positions at k are marginalized out creating a new clique [26] with all the elements that were connected to them. The result is a new graph (figure right) that links the current positions with map features. Observe that there are no direct links between robots, that is, the robot formation is CI given the map. In subsequent steps this property remains.

In order to obtain the same estimation as in a centralized system, robots are synchronized periodically and broadcast the new information gathered since the last synchronization. From the point of view of a robot, the synchronization is based on two steps: first, the robot broadcasts its own information to the rest of the team; second, it receives messages from the other members of the formation. We explain these steps in the following subsections.

6.4.1 Send Messages

Algorithm 6 details the operations performed to send a message from robot j to the formation. Suppose that at time s a synchronization occurred and the estimate kept by each robot is updated and coincides with the one obtained in a centralized system. From this recently updated estimate, robot j calculates the marginal of the features $\mathbf{I}^{m,j}_{\mathcal{F},s}$ and $\mathbf{i}^{m,j}_{\mathcal{F},s}$.

When a new synchonization step takes place at $s+p$ the new map information gathered by the robot since last synchronization is calculated. This new information will be the difference between the feature marginals at $s+p$ and s. Equation 6.9 in Algorithm

Algorithm 6 sendMessages$(\mathbf{I}^{j}_{s+p}, \mathbf{i}^{j}_{s+p})$

$\mathbf{I}^{m,j}_{\mathcal{F},s}, \mathbf{i}^{m,j}_{\mathcal{F},s}$ {*Feature marginal of robot j stored from last synchronization at instant s*}
$(\mathbf{I}^{m,j}_{\mathcal{F},s+p}, \mathbf{i}^{m,j}_{\mathcal{F},s+p}) = \texttt{marginalizedOut}(\mathbf{x}_{R_{s+p}}, \mathbf{I}^{j}_{s+p}, \mathbf{i}^{j}_{s+p}))$

$$\mathbf{I}^{new,j}_{\mathcal{F},s+1:s+p} = \mathbf{I}^{m,j}_{\mathcal{F},s+p} - \mathbf{I}^{m,j}_{\mathcal{F},s} \tag{6.9a}$$
$$\mathbf{i}^{new,j}_{\mathcal{F},s+1:s+p} = \mathbf{i}^{m,j}_{\mathcal{F},s+p} - \mathbf{i}^{m,j}_{\mathcal{F},s} \tag{6.9b}$$

$\texttt{broadcast}(\mathbf{I}^{new,j}_{\mathcal{F},s+1:s+p}, \mathbf{i}^{new,j}_{\mathcal{F},s+1:s+p})$

Algorithm 7 receiveMessages()

for $r \neq j$ **do**
  $\mathbf{I}^{r,m}_{\mathcal{F},s+p}, \mathbf{i}^{r,m}_{\mathcal{F},s+p}$ {*Feature Marginal received from robot r*}
  $\mathbf{I}^{j}_{s+p} = \mathbf{I}^{j}_{s+p} \boxplus \mathbf{I}^{r,m}_{\mathcal{F},s+p}$
  $\mathbf{i}^{j}_{s+p} = \mathbf{i}^{j}_{s+p} \boxplus \mathbf{i}^{r,m}_{\mathcal{F},s+p}$
end for

6 shows this operation. The subtracted information is finally broadcasted from robot j to the rest of the team.

6.4.2 Receive Messages

Algorithm 7 details the operations carried out when messages are received to update the map of robot j. As robots are CI given the map, the information sent to robot j only affects its feature elements in the information matrix and vector and therefore can be directly added by using the operator $\boxplus$, as it was explained in subsection 6.3.3. After these operations, all robots share the same information about the map.

6.5 Distributed Localization and Mapping Algorithm for Robot Formations

In this section we combine for each robot the techniques described in sections 6.3 and 6.4. On the one hand, the formation works in a local region of the map to reduce the computational cost. As the robots navigate relatively close in the formation they are localized in the same submap. On the other hand, each robot maintains its own

Algorithm 8

$mapChange = false$
while $(k < nsteps)$ **do**
  $(\mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k})$=`GlobalToLocal`$(\mathbf{i}_k, \mathbf{I}_k)$
  while not *mapChange* **and** $(k < nsteps)$ **do**
    $(\mathbf{i}^m_{l,k+1}, \mathbf{I}^m_{l,k+1}, mapChange)$=`LocalEIF`$(\mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k})$
    if *mapChange* **or** *checkSinc*() **then**
      `sendMessages`$(\mathbf{i}^m_{l,k+1}, \mathbf{I}^m_{l,k+1})$
      `receiveMessages`()
      `sendPositionToFormation`()
      `receivePositionsFromFormation`()
    end if
    $k = k+1$
  end while
  {*After p local EIF steps, a global update takes place*}
  $(\mathbf{I}_{k+p}, \mathbf{i}_{k+p})$=`LocalToGlobal`$(\mathbf{I}^m_{l,k}, \mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k+p}, \mathbf{i}^m_{l,k+p}, \mathbf{I}_k, \mathbf{i}_k)$
end while

estimation and is in charge of its own observations to update the map. From time to time, the robots get synchronized to obtain the same map estimate. At these instants, each robot also broadcasts its best estimated position to maintain the formation structure. Notice that, after a synchronization, the estimated pose of the formation and the map coincide with that of a centralized version.

Our distributed method for each member of the robot formation is presented in Algorithm 8. First, the robot is localized in a local working region (GlobalToLocal). While it remains in the same region ($mapChange = false$), the standard EIF operations are carried out to estimate its position and features location but without marginalizing out the initial position of the robot in the submap (LocalEIF). Based on the knowledge about the formation structure and using the spring-damper approach described in chapter 3 as navigation strategy, each robot is able to predict the position of the rest so that the robots do not stop navigating in formation towards their goal in absence of communication. If a synchronization event *check_sync* is registered (e.g. after a determined period of time, or when a robot makes a request, etc.), an exchange of messages takes place to update the states of the team with the same information. Also, when the formation changes to a new local region ($mapChange = true$), the robots synchronize to update its total map and another iteration of the main execution loop is realized. The new information that each robot receives about the features will affect its own position estimation. For this reason and to update their knowledge about the formation, each time there is an exchange of messages, each robot sends its improved position to the rest of the robots (sendPositionToFormation) and receives the information from the rest (receivePositionsFromFormation).

6.5.1 Computational and Communication Complexity Analysis

While each robot is working in its own local submap, it performs the LocalEIF algorithm. The computational cost of this filter depends on the number of features in the submap. Since we are not adding new features to the map, the number of features remains constant and the cost will be $\mathcal{O}(1)$ while working in the local region.

When a change of submap is carried out, there are two operations involved, LocalToGlobal, to update the global map, and GlobalToLocal, to extract a new region of the map. As it was explained in section 6.3, the operation LocalToGlobal consists in adding the new information to the global map, therefore, its cost is $\mathcal{O}(1)$. In GlobalToLocal, we find the most costly operation of Algorithm 8, where we have to invert almost the whole map, (Eq. 6.3), leading to a cost of $\mathcal{O}(n^3)$ in the worst case, being n the fixed number of features of the prior map. Unlike other algorithms, this operation only takes place at each map changing step. As it was studied in [68], there exists a trade-off between the size of the local maps and the frequency with which the robots need to change of submaps. If the submaps are small, global updates will be more frequent, but, on the other hand, larger submaps will increase the computational cost of local updates.

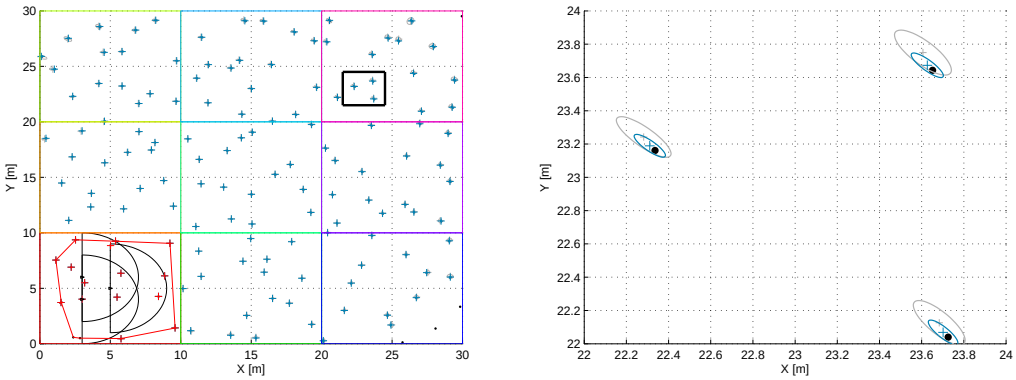
Concerning the communication, the total amount of information to be sent to other robots is bounded to the number of features in the local map. Since each robot broadcasts the new information added to the submap to the rest, the communication complexity scales with the number of robots $\mathcal{O}(r+1)$.

6.6 Results

Through the following simulation results we want to show the advantages of the distributed submapping algorithm proposed in this section. We have designed a simulation environment of 30x30m where three robots set in a triangle formation have to navigate along a 120m loop scenario (Fig. 6.3). Each robot has a prior stochastic map of the navigation area divided in submaps of 10x10m. The current local region $\mathbf{x}_{F_l}$ is common to all the robots and is selected depending on which features are being observed by the robots, thus, this local region can be composed of several submaps.

One of the main advantages of this algorithm is the improvement of the given stochastic map where uncertainties and errors of the map features decrease. This can be seen in Fig. 6.3 and in its zoomed area (Fig. 6.3b), where the previous and the final map features are depicted. Additionally, Fig. 6.4a shows the improvement in the map features errors measured in terms of the Root Mean Squared Error (RMSE).

As a direct consequence of the map improvement, the robots are localized more accurately. Figure 6.4b shows the RMSE obtained on each component of the localization of the robot formation. We also compare this result to the RMSE error obtained when the map is not refined and the formation is only localized within the prior map using the



(a) **(b)**

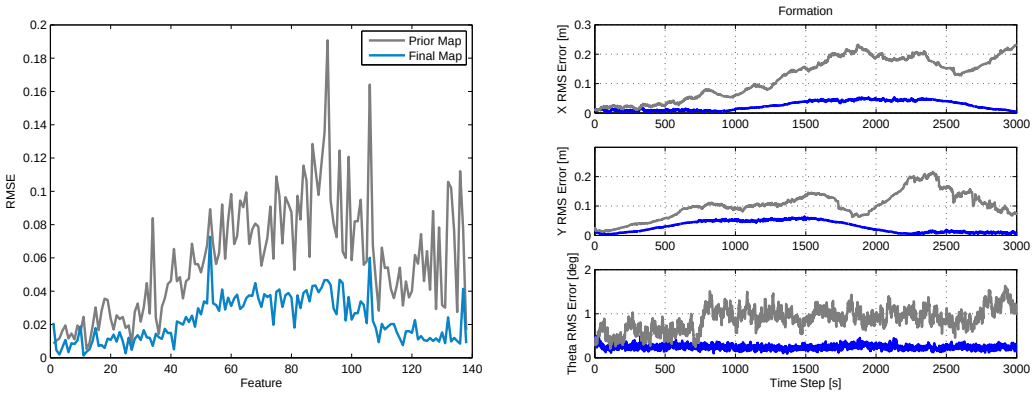
Figure 6.3: 2D Point-based stochastic map of the simulation environment. (a) Initial setting of the formation and current local region where local map features and submap bounds are depicted in red. The area inside the black box is enlarged in (b) which shows the reduction of the covariances (blue) with respect the a priori map (grey). Black dot in the zoomed in area represents the ground truth of the feature.

algorithm proposed in chapter 4.

The next advantage concerns the computational cost. Three different implementations have been compared. First, the submapping and distributed technique proposed in this section (Dist+sub) where each robot updates its own local and global maps based on its own observations and synchronizes with the other robots to obtain a better estimation. Second, a centralized version of the submapping technique (Cent+sub) based on [72], where the robot leader is the one who updates the local and global maps using the observations gathered by all the robots. Finally, a centralized EKF-based version in which the leader does not work with local maps but only with the global map (Cent+glob).

Figure 6.5a shows a comparison between the first (Dist+sub) and second version (Cent+sub). We have supposed the worst case time complexity for the first implementation in which the formation synchronizes at each time step and an additional time per step appears due to the execution of the send-receive operations. However, the cost of the map update in the distributed version (green line) is lower, resulting in a less time complexity with respect to the centralized implementation (red line). Peaks in the times are due to global updates, where Algorithms 4 and 5 are executed.

In Fig. 6.5b, the computational cost of the first (Dist+sub) and the third version (Cent+glob) is depicted. We can see how even when a global update occurs, the times of the distributed implementation are lower than the global version. We can also observe that, since we are not adding new features to the map, the cost of the global EKF

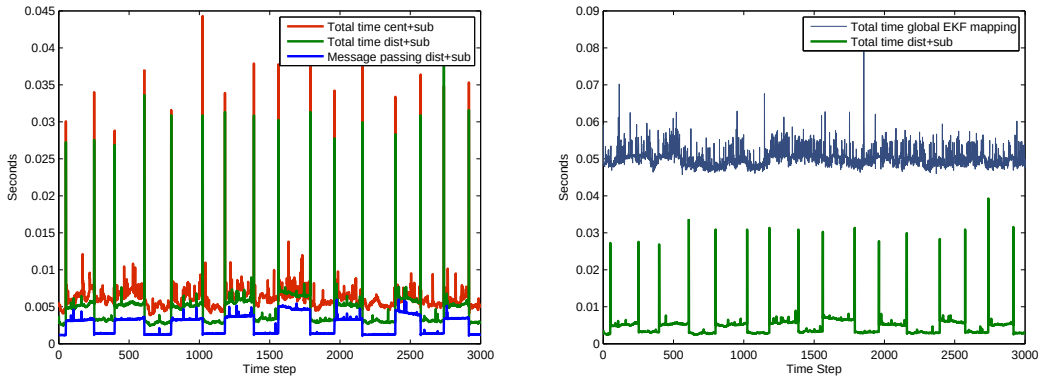


(a) **(b)**

Figure 6.4: Root Mean Squared Error of the map features and the robot formation. (a) Comparison between the RMSE per feature in the a priori map and in the final map. In the a priori map, maximum errors are due to far features wrt the base reference while the minimum errors are due to features involved during a loop closure event or those that are near to the origin. After running the proposed SLAM algorithm, the error is more balanced in the whole final map. (b) Evolution of the RMSE error of all the robots of the formation along the simulation. The errors on the position components obtained when the formation navigates while refining the prior map (blue) are lower than if the map is not updated (grey) and only the localization algorithm proposed in chapter 4 is used.

mapping remains approximately constant. In the distributed implementation, the times are also constant while working in the same submap, but these times change depending on the number of features of the submaps (e.g. time at instant 2250 and at instant 2500). Note that, in a real experiment, an additional time would have to be considered in both cases, due to the data association process. In the global centralized EKF mapping, the leader would perform this task by matching all the observations from all robots, thus, this time would be proportional to the number of robots and observations whereas in the distributed version, this time would only depend on the number of observations.

Finally, we want to emphasize that the result obtained with the distributed submapping method proposed in this section is equivalent to the result obtained in a centralized-global mapping implementation. Each time the robots perform a synchronization, they will obtain the optimal solution (i.e. equivalent to the centralized). In Fig. 6.6 we show the consistency ratio NEES$/\chi^2_{r,1-\alpha}$, $r = \dim(\mathbf{x}_{F_l})$ of the local map features when robots synchronize every 10 time steps. We can see how, while there is no synchronization, NEES for each robot's features is different, but when a synchronization occurs the solution obtained is equal to the centralized. Besides, the estimation of the map features is consistent, since the ratio NEES$/\chi^2_{r,1-\alpha} \leq 1$.



(a) **(b)**

Figure 6.5: Comparison of times between different implementations. (a) Times per step for the centralized and distributed submapping implementations. Worst case where robots synchronize at each step is shown. For that reason, synchronization time (blue), which affects to the distributed implementation, is never zero. (b) Time comparison between the global centralized and the submapping distributed implementations. Synchronization messages are transmitted at every step.

6.7 Conclusions

In this chapter we have proposed a distributed estimation algorithm within the stochastic framework presented in previous chapters for robot formations. Using a prior map of the environment, our method efficiently tackles the localization of the robot formation at the same time the map is improved with new observations.

The proposed algorithm does not rely on a central server improving flexibility and robustness. This is achieved by describing the distributed estimation problem as a GMRF, which allows us to take double advantage of the CI properties to reduce the computational and communication requirements: as first result the formation only experience constant updates whereas global updates are postponed until a new local region transition takes place; a second cost reduction is achieved due to the fact that robots only get indirectly related through the observation of common map features.

In consequence, the load in the communication channel will only scale linearly with the number of resources since each robot broadcasts only an information summary of features observed from the last robot formation communication. The algorithm results show an accuracy improvement of the a priori map, being the final result equivalent to the one obtained in a centralized implementation with a lower computational effort.

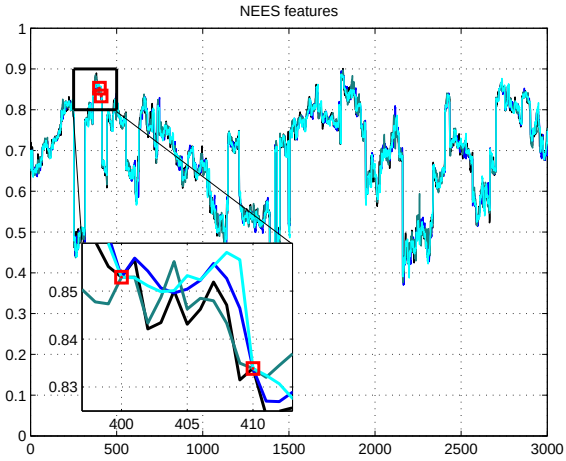


Figure 6.6: Consistency ratio NEES/$\chi^2_{r,1-\alpha}$ of the features in the centralized (black) and the distributed (the three blue lines, one for each robot of the formation) implementations. In the zoomed area, red boxes correspond to synchronization times (every 10 time steps) where centralized and distributed results are exactly the same. If synchronizations occurred at every time step, the three blue lines would coincide with the black line.

Part II

Multi-Robot Simultaneous Localization and Mapping

Chapter 7

Multi-Robot SLAM using Condensed Maps

This chapter describes a graph-based SLAM approach specifically designed to address the communication and computational issues that affect multi-robot systems. The proposed method utilizes condensed measurements to exchange map information between the robots. These measurements can effectively compress relevant portions of a map in a few data. This results in a substantial reduction of both the data to be transmitted and processed, that renders the system more robust and efficient. As documented by the simulated and real world experiments, these advantages come with a very little decrease in accuracy compared to ideal (but not realistic) methods that share the full data among all the robots.

7.1 Motivation

In previous chapters, we have presented multi-robot localization and SLAM algorithms based on filtering techniques, aiming at taking advantage of their speed properties. Under the assumption of working in a common previously built map, the system could be represented with respect to the same base reference and each robot had a good guess about the location of the rest of the team. The use of map features for a compact representation of the environment reduced the communication overload. However, frequent sychronizations were required to maintain coherency in both centralized and distributed algorithms. Developed in the context of robot formations, communications between robots could be guaranteed as they move nearby. However, this context is not suitable and can not be straightforwardly extended to more general multi-robot systems where robots explore different areas of the environment and they meet only occasionally.

In this chapter we propose a general multi-robot graph-based SLAM approach to build a dense map of the environment without prior map knowledge nor assumption on

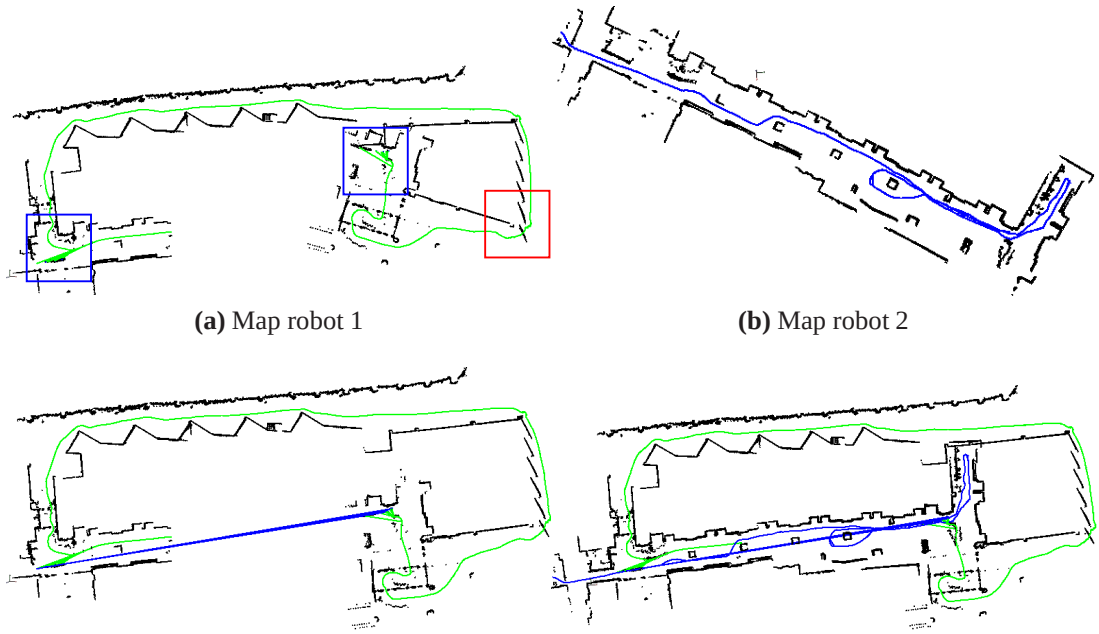
the robots connectivity. As we already introduced in chapter 2 the single-robot graph-based SLAM problem involves to construct a pose-graph whose nodes represent robot poses and in which and edge between twho nodes encodes a sensor measurement that constrains the connected poses. One such graph is constructed by a *front-end* algorithm, the estimation process is simplified to the problem of finding a configuration of nodes maximally consistent with the measurements. This requires solving a large error minimization problem which is often done by means of modern least-squares optimization approaches, also called *back-ends* in the SLAM context. This least-squares minimization problem can be re-evaluated at each time new information is incorporated into the graph, providing a enhanced performance with respect to filtering techniques.

In principle, using multiple robots to acquire the map is more robust, since the failure of a single system does not necessarily compromise the whole result. Furthermore, the parallel acquisition of data by multiple robots might result in less time needed for building the map. Despite these attractive properties, multi-robot systems for SLAM presents substantial challenges of both theoretical and practical nature. Ideally, existing algorithms for single-robot graph-based SLAM could be extended to handle the multi-robot case just by constructing and optimizing the graph based on all measurements gathered by the robots. Unfortunately, such an approach presents several challenges. First, determining constraints between pairs of robots' graphs requires a re-localization scheme without any initial guess. This might dramatically increase the chances of adding wrong edges to the graph, and would compromise the entire process. Second, assuming to have an ideal error-free front-end, the graph obtained by each robot would rapidly increase its size. In the worst case, each robot would add a set of edges to the graph with a quadratic dependency on the number of robots. Consequently it would limit the on-line performance of any state-of-the-art optimizer whose complexity roughly increases with the number of edges.

Furthermore, the above scenario assumes the robot can perfectly communicate with each other, which is typically not the case. Wireless communications in large environments are usually brittle and depend on the positions of the nodes. Furthermore they present bandwidth limitations, that would prevent the robots to share large amounts of data.

In this chapter we propose an approach for multi-robot SLAM that addresses the issues raised above. The method is designed to operate with very limited communication facilities and allows to dynamically add and remove robots from the system.

Each robot in the team computes its own map, but it refines it by integrating a set of virtual or *condensed measurements* [31] coming from the other robots. These condensed measurements can be seen as a reduced version of the graph constructed by the other robots, that contains only the information relevant to the receiver in order to refine its own map. They are easily managed by the optimization back-end and allow to substantially reduce the size of the optimization problem that each member of the team



(a) Map robot 1 **(b)** Map robot 2

(c) Map robot 1 + condensed graph from robot 2 **(d)** Final global map

Figure 7.1: This figure illustrates a motivating example of our approach. Two robots cooperate to construct a map of a building containing a loop of 250m. (a), (b) Each robot is in charge of mapping one part of the large loop. Due to the lack of enough observations robot 1 commits a big error and fails in the estimation of its part of the loop ((a), red square). However, it meets and localizes robot 2 at two points of its trajectory ((a), blue squares) who sends a compressed version of its map that contains measurements relating these two locations. (c) When robot 1 adds these measurements (blue edges) to its map, it improves its estimation. (d) Since the maps become interconnected, we are able to reconstruct the global map by merging the individual maps and optimizing them together.

has to solve, thus increasing the efficiency. In order to localize a robot in any other robot's graph with high presence of outliers, we propose a robust voting approach that substantially decreases the chances of wrong data associations and loop closings. The system has been tested both on real robots and on simulated environments. Figure 7.1 illustrates a motivating example of the approach.

7.2 Related Work

Graph-based optimization algorithms have become the most successful techniques to solve the full SLAM problem due to their improved performance which has motivated their use in the multi-robot context. In [24] we can find the first comprehensive graph-

based approach to distributed SLAM with landmarks. The algorithm uses a multifrontal QR factorization in which no measurements are communicated between robots or robots and a server. Instead, the communication is limited to QR update messages, which condense the entire measurement history on the individual robots into small upper trapezoidal matrices. The data association problem is not considered and the measurements are processed off-line for each robot. In [1] the authors present a Collaborative Smoothing and Mapping (C-SAM) algorithm to build a joint map from a team of robots without initial knowledge of their relative positions. Therefore C-SAM does not present a proper distributed SLAM solution but a centralized version of the problem. Only simulated results are provided in the paper. A recursive solution for multi-robot pose graph SLAM is presented in [43]. The main novelties of this approach reside on its incremental nature, i.e. the solution does not depend on a batch optimization after all measurements are taken, and on the introduction of anchor nodes that allow each robot to use its own reference frame whereas inter-robot measurements and graphs obtained from other robots can be easily managed in the same framework. Therefore in each encounter the robots interchange their graphs which do not need to be transformed to a common frame since can be tackled using the anchor nodes. The paper does not take into account any issues of communication bandwidth constraints between robots.

The closest approach to our proposed method is presented in [20] and [21]. The authors address in [20] the multi-robot problem with an extended Smoothing and Mapping approach called Decentralized Data Fusion (DDF) which is represented using a factor graph. Each robot optimizes its own trajectory and its landmark map and then creates a *condensed map* formed exclusively by the marginalization of *common landmarks*. These condensed maps are mutually interchanged among neighboring robots to create a simplified neighborhood graph of landmarks that is optimized by each robot. To correct the local map with the information obtained from the optimization of the neighborhood map a set of hard equality constraints are established between each neighborhood landmark and its corresponding local version. In summary, robots get mutually connected by sending graph nodes of shared features that must be hard-linked with their corresponding local representations. In [21] the work is extended with a novel multi-robot data association method for robust decentralized mapping. The data association is based on a triangulation algorithm that provides matching between maps.

Our multi-robot SLAM system is based on the concept of *condensed measurements* [31] which were introduced in section 2.5.1. During map construction, robots meet and exchange data in different parts of the environment. The messages are governed by a protocol explained in detail later in this chapter and results in each robot augmenting its pose graph with a measurement about the relative position of the encountered partner. After the first encounter, each time a pair of robots meet they additionally interchange a set of *condensed measurements*, which is just a factor graph of the shared variables obtained from an approximation of their respective global graphs at the equilibrium. The

advantages of this approach are three-fold: 1- Each robot only carries its own graph that gets minimally augmented whenever an encounter with other robot of the team takes place; 2- The mutual influence between the team of robots is easily tackled by using the *condensed measurements* since only new virtual factors (edges in the graph) between the shared nodes must be taken into account in the optimization process; Neither special constraints nor different graph representations are required. 3- The communication bandwidth is efficiently used since a summarized (condensed) representation of the required constraints is transmitted between robots. In addition, we propose a technique to robustly find alignments between local maps. This technique is used to find loop closures or alignments between local maps from different robots.

7.3 Condensed Graphs

Recall from section 2.5.1, in pose-graph based SLAM we are interested on determining the robot positions $\mathbf{x}_{R_{0:K}}$ along its whole trajectory. The problem is modelled with a graph where each node $\mathbf{x}_i$ represents a robot position $\mathbf{x}_{R_i}$ and each edge encodes a measurement $\mathbf{z}_{ij}$ relating a pair of nodes $(\mathbf{x}_i, \mathbf{x}_j)$ whose uncertainty is characterized by the information matrix $\mathbf{\Omega}_{ij}$. Then, the problem is formulated as a nonlinear least squares optimization problem to find a configuration of nodes $\mathbf{x}^*$ which minimizes the overall error:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}_{ij}^T(\mathbf{x}_i, \mathbf{x}_j) \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \tag{7.1}$$

where $\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ represents the error between a measurement $\mathbf{z}_{ij}$ and the expected measurement given the current configuration of nodes:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{g}(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{z}_{ij} \tag{7.2}$$

In our case, $\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j)$ is the measurement function that computes the position and orientation of $\mathbf{x}_j$ in the frame of $\mathbf{x}_i$:

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_j \ominus \mathbf{x}_i \tag{7.3}$$

To solve this problem, modern optimization approaches like $\mathbf{g^2o}$ [47] or iSAM [42] require a time that depends on the number of edges, and their success in finding the correct solution is affected by the initial guess available to the system. In the single robot case, this initial guess is typically good, since the robot can rely on an estimate that is constructed incrementally, and that at each point in time contains all the information acquired so far. Conversely, in the multi-robot case it might happen that when two robots meet and want to share their map, the individual estimates are affected by a large error. Furthermore, to carry out the optimization by using one of these approaches, the two robot would have to share their entire graph, which is potentially large.

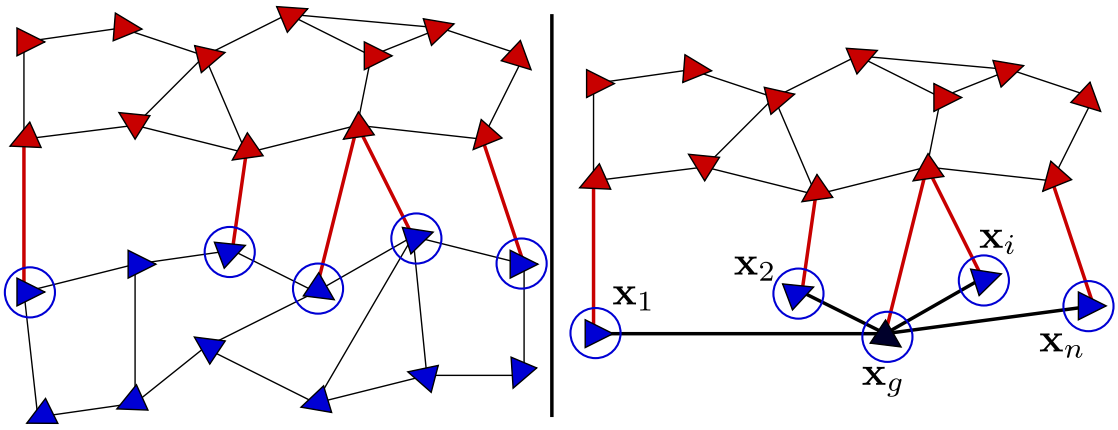


Figure 7.2: In this figure, we illustrate the use of condensed measurements to share information between two robots. The graph of robot *A* is illustrated in red and the graph of robot *B* is illustrated in blue. Red edges show the measurements between nodes of the robot *A*'s graph and the Robot *B*'s graph. Instead of sending to robot *A* all its graph, Robot *B* sends a condensed version, consisting of a central node (gauge, $\mathbf{x}_g$), and a set of condensed factors connecting the gauge with each of the nodes ($\mathbf{x}_i, i = 1...n$) seen from robot *A*. Notice that $\mathbf{x}_g$ can also be selected from the nodes already seen by robot *A*.

To lessen this problem, in this chapter we propose an alternative approach based on condensed measurements [31]. When two robots *A* and *B* meet, they share a reduced graph so that each robot receives from the other only the information needed to refine its own estimate. Figure 7.2 intuitively illustrates this process. Let us assume that robot *A* has observed a set of nodes $\mathbf{x}_i, i = 1...n$ from robot *B*'s graph. In order to optimize its own graph, by taking into account the information from *B*, robot *A* should know how these shared nodes are related in the space. This information is clearly contained in the graph of *B*, but it is too large to be sent over the network. Instead of sending the full graph, *B* sends a "condensed" version that has substantially less nodes, but that captures the information necessary to *A* to perform this optimization.

The condensed graph will be composed of the nodes $\mathbf{x}_i$ from robot *B*'s graph and a set of condensed measurements relating those nodes. The process to compute the condensed measurements is the following:

- We select an arbitrary node $\mathbf{x}_g$ from the nodes $\mathbf{x}_i$ of robot *B*'s graph. This node is *fixed* as the origin (gauge) of the graph which is optimized to obtain a local solution with respect to the gauge.

- Once we have a minimal error configuration for the graph of robot *B*, we compute a set of condensed measurements $\mathbf{z}_i$ between $\mathbf{x}_g$ and each other node $\mathbf{x}_i$. Following the procedure explained in section 2.5.1 and using the error function 7.2 we characterize each measurement by its mean and information matrix. To this end,

we regard the graph of B as a local map.

This procedure converts robot B's graph into a condensed version with star topology where each measurement $\mathbf{z}_i$ incorporates the knowledge in the original graph of robot B that $\mathbf{x}_g$ has about the position of $\mathbf{x}_i$.

7.4 Multi-Robot SLAM using Condensed Graphs

This section describes in detail the proposed multi-robot SLAM system. The approach operates on raw sensor measurements acquired by mobile robots equipped with a laser scanner. Inter-robot communication is based on a wireless Ad-Hoc network that dynamically adapts depending on the mutual locations of the robots. Section 7.4.1 presents the details of the communication model.

Each robot executes a standard laser-based SLAM pipeline: the state of the system is stored in a pose-graph which is constantly optimized by the $\mathbf{g^2o}$ optimizer. When the robot moves for a certain distance, a new node is added to the graph, and the odometry measurement is used to label the edge between the new and the previous robot positions. The laser scan acquired at the new position is matched against a set of candidate scans stored in the nodes of the graph. The candidate nodes are selected if the current robot position falls in their uncertainty ellipses. This gives a set of candidate loop closing edges between non temporally subsequent nodes, that are inserted in the graph upon validation by a voting scheme procedure described in Section 7.4.3.

To extend this single-robot SLAM algorithm to the multi-robot case, we need to augment the graph-construction method described above to handle information coming from other robots. The multi-robot front-end will be in charge of:

- robustly localizing other robots into the current robot's map, based on their raw sensor measurements.
- integrating the condensed measurements of the other robots in the current graph. This is achieved simply by including the set of condensed edges.

In the remainder of this section we describe in detail our communication model and how we address the problems outlined above, by taking into account the limitations of the communication infrastructure.

7.4.1 Communication Model

The approach is founded on the assumption that no infrastructure is present. Thus the communication between robots is point-to-point. Robots can communicate only when they are within a certain distance, and the communication graph changes dynamically based on the current configuration of the multi-robot system.

This models conservatively the behaviour of wireless Ad-Hoc networks. Wireless communication has a limited range and bandwidth which will vary depending on the protocol (WiFi, Bluetooth, ...), the IEEE standard used (e.g. 802.11b/g/n...) and also the structure of the environment. Not relying in infrastructure has substantial practical advantages.

The proposed communication model works in a robot-independent way, where the messages are transmitted asynchronously and contain the most up-to-date information available. The probability that a message sent is correctly delivered decreases with its size. To maximize the probability that the messages are correctly delivered, in our algorithm we kept the size of the single messages as small as possible, possibly fitting within an Ethernet frame (1400 bytes). Each robot periodically sends a ping and, based on the ping messages received by the other robots, it determines its neighbors. When two robots are within communication range they send two kind of messages: to transmit their local maps and to manage the condensed graphs.

Local map transmission

The local map is transmitted through a message containing the following information:

- The last measurement (laser scan) acquired, and the current Id of the node containing the laser scan in the graph.
- The up-to-date estimated locations of the last N nodes.

With this information each robot is able to reconstruct the local maps of the team mates in range. Notice that a robot sends only the most recent laser scan, which is the bulky part of the message. To determine a local map consisting of N scans we need to buffer the last N messages from each sender, and render the scans according to the most recent list of estimates of the nodes. The latter is transmitted each time a new node is added to the graph. This allows to update the local maps with minimal communication overhead, even if the graph changes its configuration. The local maps of other robots are used to localize them in the current robot's map. This is done by using a voting scheme for robust outlier rejection in combination with a correlative scan matching algorithm, which is described in detail in Section 7.4.3.

Condensed graph transmission

To manage the graph, a robot sends a message containting the following information:

- A list of nodes of the other robot's local map it has matched against its own local map.
- A condensed graph extracted by its own graph and consisting of the edges relating the nodes that have been matched by some other robot.

These messages are sent whenever a new node is added to the graph, based on the number of mates in range.

7.4.2 Multi-Robot SLAM

In this section, we illustrate how the messages defined above are used to implement our multi-robot SLAM approach. To simplify the description, we refer to Figure 7.3 and without loss of generality we assume having only two robots: *A* (red) and *B* (blue).

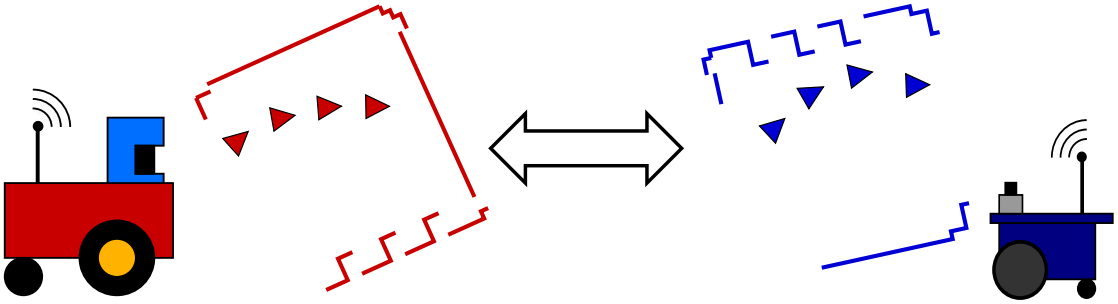
Initially (see Figure 7.3a), each robot constructs its own map with a single-robot SLAM algorithm. When a communication is available *A* starts receiving the current local map of *B*, by storing its most recent readings. A matching procedure is executed to align the two local maps, and results in a set of candidate edges connecting the map of *A* and the map of *B* (see Figure 7.3b). When reasonably confident about the correctness of these edges, robot *A* sends to *B* this list (see Figure 7.3c). *B* then computes a condensed graph containing only the nodes of its map that appear in the candidate edges found by *A* (see Figure 7.3d), and sends it to *A*. Finally *A*, includes these measurements in its own graph to get a more consistent map (see Figure 7.3e).

This algorithm can be implemented within a robot *A* in a straightforward way by maintaining the following data structures:

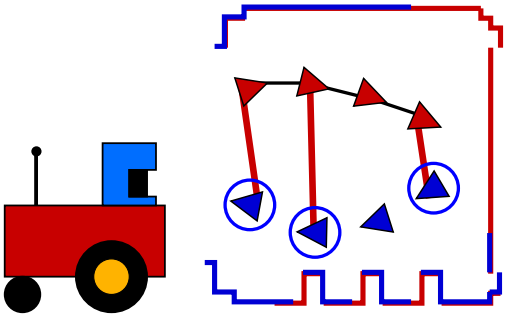
- the graph $\mathcal{G}_A$ obtained by single-robot SLAM
- for each other robot *B*:
  - the most recent local map $\mathcal{M}_B$ consisting of the last N nodes, that is used for cross-localization.
  - the list $\mathcal{E}_A^B$ of candidate edges between the map of *A* and the map of *B* that have been found by *A*.
  - the list of $\mathcal{E}_B^A$ edges received from *B*, that connect the map of *A* and the map of *B* and that have been found by *B*.
  - the condensed graph $\mathcal{G}_A^B$ sent by *B*.

Robot *A* updates the local maps of each other robot $\mathcal{M}_B$ and the list of edges $\mathcal{E}_B^A$ whenever a new message is received. Each time the single-robot SLAM algorithm running on *A* adds a new node to the graph, the estimate of the last N nodes and the last laser scan are sent to allow the other robots to construct the local map of *A*. Subsequently, Robot *A* runs a map-alignment algorithm between its local map and each $\mathcal{M}_B$, and updates the list of candidate edges by using the procedure described in the next section.

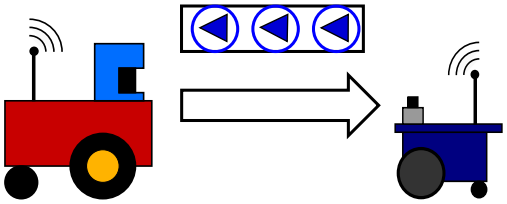
Finally, by knowing $\mathcal{E}_B^A$ Robot *A* computes which nodes of its own map are relevant for Robot *B*, and sends the corresponding condensed measurements. In computing the



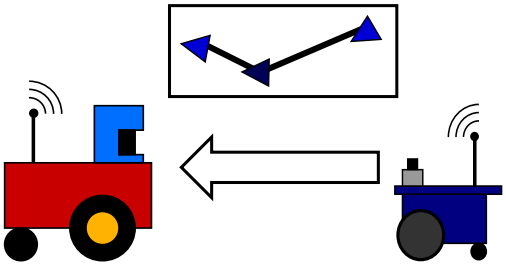
(a) Local map transmission



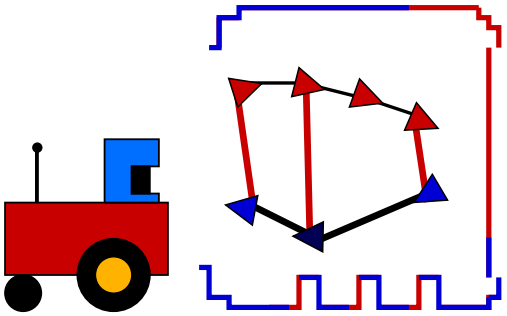
(b) Local maps alignment



(c) Matched nodes' IDs transmission



(d) Condensed graph transmission



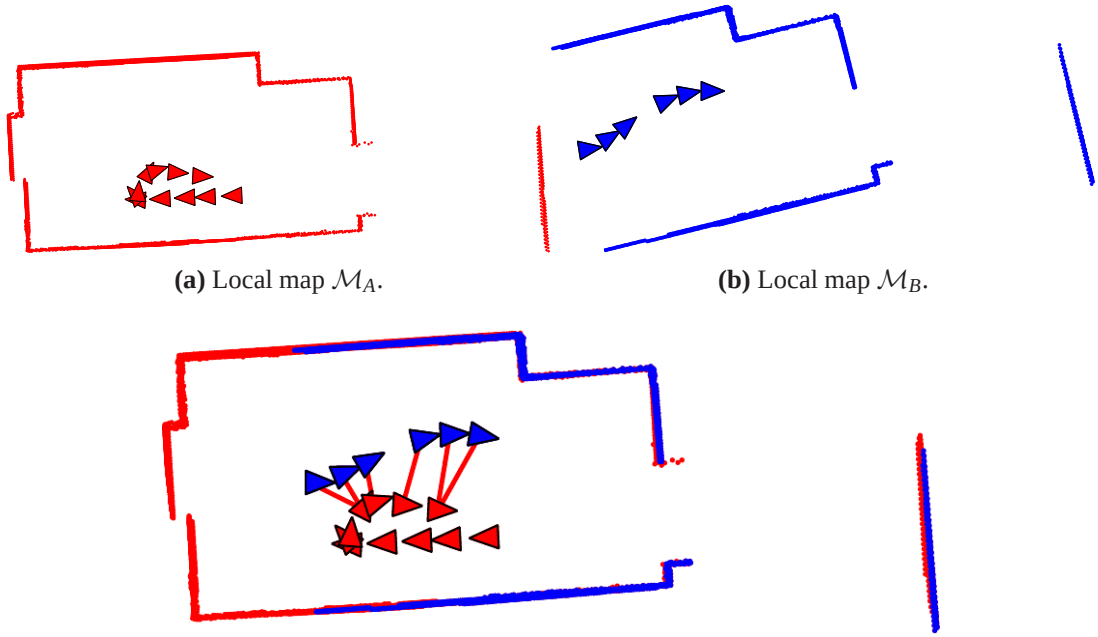
(e) Condensed graph addition

Figure 7.3: Illustration of our multi-robot SLAM algorithm in a two robots scenario. Robot *A* is depicted in red and Robot *B* in blue. Triangles represent the nodes of the graph. (a) Each robot runs a graph-based SLAM algorithm and constructs its own map. When they are within a communication range, they share their current local maps; (b) *A* localizes *B* and determines a set of candidate edges connecting the two maps; (c) *A* informs to *B* which of its nodes it has matched; (d) *B* computes condensed measurements that connect the nodes in its own map that appear in the edges found by *A*; (e) *A* includes these edges in its own graph.

condensed measurements Robot *A* considers only the portion of the graph acquired with its own sensors, thus avoiding multiple integration of information.

7.4.3 Robust Map Alignment

In this section, we describe our approach to robustly align two local maps $\mathcal{M}_A$ and $\mathcal{M}_B$ onto each other. A local map consists of a portion of the graph. We recall that each node consists of a robot pose and a laser scan acquired at that pose. Figure 7.4 illustrates the problem.



(a) Local map $\mathcal{M}_A$.

(b) Local map $\mathcal{M}_B$.

(c) Alignment of the two local maps.

Figure 7.4: Example of map alignment between two local maps after finding a set of edges jointly consistent.

Our goal is to find a set of edges $\mathcal{S}$ between the nodes of the two local maps such that they are maximally consistent, given the scans. To this end we match each scan $\mathbf{s}_j^B$ contained in node $\mathbf{x}_j^B$ of $\mathcal{M}_B$ with each scan $\mathbf{s}_i^A$ contained in node $\mathbf{x}_i^A$ of $\mathcal{M}_A$, by using a correlative scan matcher. Note that each matching can result in zero or more measurements $\mathbf{z}_k$ of $\mathbf{x}_j^B$ with respect to $\mathbf{x}_i^A$. Each of these solutions is then converted in an edge e_k relating nodes $\mathbf{x}_i^A$ and $\mathbf{x}_j^B$, and added to a pool of candidate edges $\mathcal{E}$.

Given this pool of edges, we run a voting scheme procedure to determine which of them are inliers, summarized in Algorithm 9. The idea is the following: to determine a translation between the two local maps it is sufficient to translate them so that one candidate edge e_k is satisfied (its error is $\mathbf{0}$).

Let $\mathbf{x}_i^A, \mathbf{x}_j^B$ the location of the nodes containing scans $\mathbf{s}_i^A, \mathbf{s}_j^B$ and $\mathbf{z}_k$ their computed relative measurement be represented as the isometry matrices X_i^A, X_j^B and Z_k, respectively.

The transformation matrix T_k which satisfies the constraint is:

$$T_k X_j^B = X_i^A Z_k \tag{7.4}$$

$$T_k = X_i^A Z_k (X_j^B)^{-1} \tag{7.5}$$

Applying this translation affects the error of all other candidate edges, and their error will be small if they are consistent with e_k, while it will be large otherwise. Based on these errors and using an inlier threshold τ we determine inliers and outliers. Once we obtain a set of maximally consistent edges $\mathcal{S}$ using this procedure, we decide whether to accept the match or not if a minimum number of inliers is achieved. Figure 7.5 illustrates the procedure.

The bottleneck of this schema is the scan matching routine and the computational cost of algorithm will depend on the number of candidates edges in $\mathcal{E}$. Accordingly, we need to limit the number of times we perform scan-matching. By considering that the local maps can be assumed to be consistent, and that one of the two local maps is acquired incrementally one scan at a time, we can implement the above procedure in an efficient way. Each time we receive a new scan $\mathbf{s}_j^B$, we match it against the local map constructed by the union of all $\mathbf{s}_i^A$. The scan matcher results in a set of transformations between $\mathbf{s}_j^B$ and the map $\mathcal{M}_A$. These transformations are converted in edges between $\mathbf{s}_j^B$ and the closest node in $\mathcal{M}_A$, after applying the transformation. The resulting edges are inserted in the pool. The joint consistency validation is done at every step, and the candidate edges that are marked as outliers for a certain number of times are removed from the pool.

7.5 Experiments

The multi-robot SLAM approach proposed in this chapter has been validated through simulations and real world experiments. The system is implemented in C++ as a ROS package and the simulations have been conducted with the Stage simulator.

7.5.1 Robust Map Alignment

First, we have validated the map alignment procedure explained in section 7.4.3 in a simulation environment to have the ground truth at our disposal for measuring the quality of the final map obtained. Additionally, we have compared the results with respect to Single-Cluster Graph Partitioning (SCGP) [66]. Therefore, we will follow with a brief summary of the SCGP approach and how we apply it to our problem of finding a set of maximally consistent edges that align two partial maps. Then, the comparison between both approaches will be presented.

Algorithm 9 Compute $\mathcal{S}$

Require: $\mathcal{M}_A$, $\mathcal{M}_B$, $\mathcal{E}$, τ

bestChi2 $\leftarrow$ inf, bestInliers $\leftarrow 0$, $\mathcal{S} \leftarrow \{\}$
for each edge $e_k(\mathbf{x}_i^A, \mathbf{x}_j^B) \in \mathcal{E}$ **do**
  $\mathcal{S}_k \leftarrow \{\}$
  inliers $\leftarrow 0$; totalChi2 $\leftarrow 0$;
  {Compute T_k such that error $\mathbf{e}_k = 0$}
  $T_k \leftarrow X_i^A Z_k (X_j^B)^{-1}$
  {Apply T_k to the nodes in $\mathcal{M}_B$;}
  for each node $\mathbf{x}_j^B \in \mathcal{M}_B$ **do**
    $\bar{X}_j^B \leftarrow T_k X_j^B$;
  end for
  {Compute new error $\mathbf{e}_k$ for each edge in $\mathcal{E}$}
  for each edge $e_k \in \mathcal{E}$ **do**
    totalChi2 += $\mathbf{e}_k$;
    if $\mathbf{e}_k < \tau$ **then**
      inliers++;
      $\mathcal{S}_k \leftarrow \{\mathcal{S}_k, e_k\}$
    end if
  end for
  if (inliers $>$ bestInliers **or**
  (inliers = bestInliers **and** totalChi2 $<$ bestChi2)) **then**
    bestChi2 $\leftarrow$ totalChi2
    bestInliers $\leftarrow$ inliers
    $\mathcal{S} \leftarrow \mathcal{S}_k$
  end if
end for

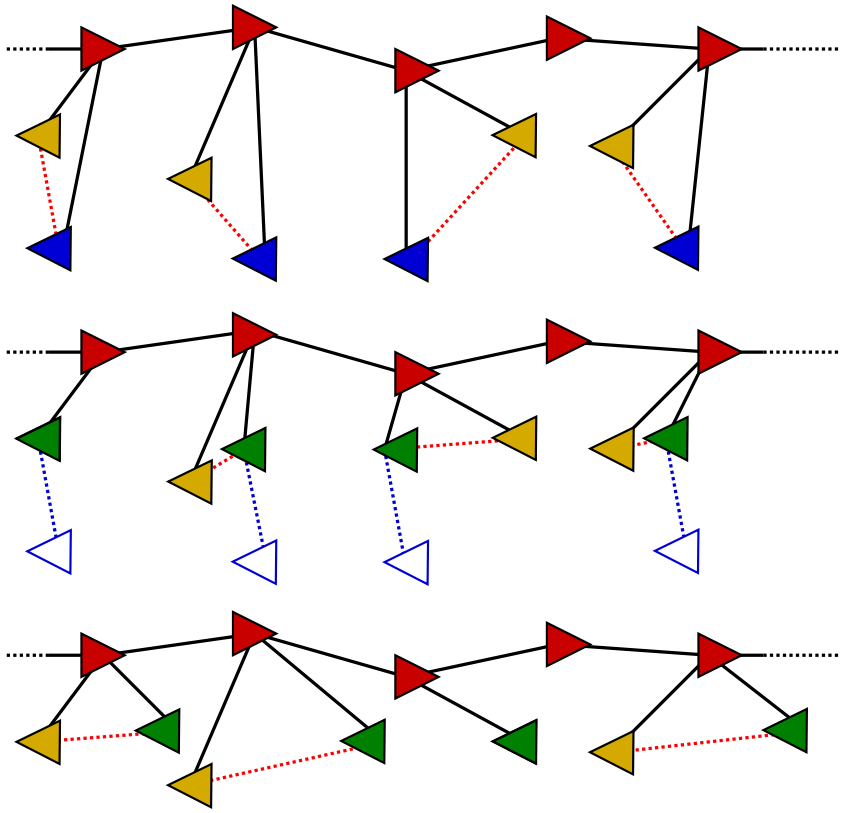


Figure 7.5: Top: In red, nodes belonging to a local map, in blue, current estimation of the received nodes and in yellow, the same nodes with respect to the candidate closure edges. Dashed red lines represent the error in the estimation for each edge. Middle: Green, position of the nodes after applying the transformation (blue dashed line) that makes the error of the first node equal to zero. With this configuration, the error in the second and fourth nodes is small (they could be selected as inliers if the error is lower than a threshold) whereas the error in the third node is large (outlier). Bottom: Configuration of the nodes if the transformation to make the error of the third node equal to zero is applied. Since it is a wrong closure, the error in the rest of nodes is large, they are selected as outliers and this configuration of nodes is rejected. Notice that this procedure can be used whether the local maps are from different robots or from the same robot trying to compute loop closing edges.

Single-Cluster Graph Partitioning for graph-based map alignment

The SCGP method computes, from a set of candidate hypotheses, the subset which is maximally consistent. It represents the problem as a graph, where each node is a candidate hypothesis and an edge between two nodes represents their mutual consistency. Then, given a set of n candidate hypotheses, the graph is encoded in a $n \times n$ consistency matrix A where each element a_{ij} contains the pairwise consistency between two nodes.

Let u be a binary indicator vector representing a subset of the hypotheses, SCGP

attempts to maximize the average consensus:

$$r(u) = \frac{u^T A u}{u^T u} \tag{7.6}$$

Assuming A being symmetric, the vector u which maximizes Eq. 7.6 satisfies:

$$Au = r(u)u \tag{7.7}$$

which is an eigenvector problem with $r(u)$ the dominant eigenvalue. The two first dominant eigenvectors v_1 and v_2 and eigenvalues λ_1 and λ_2 of A can be efficiently computed using the Power Method. A minimum ratio λ_1/λ_2 can be imposed to assess the confidence of the result, otherwise the whole set of candidates is rejected [67]. Then, the dominant eigenvector is discretized to obtain the binary indicator vector.

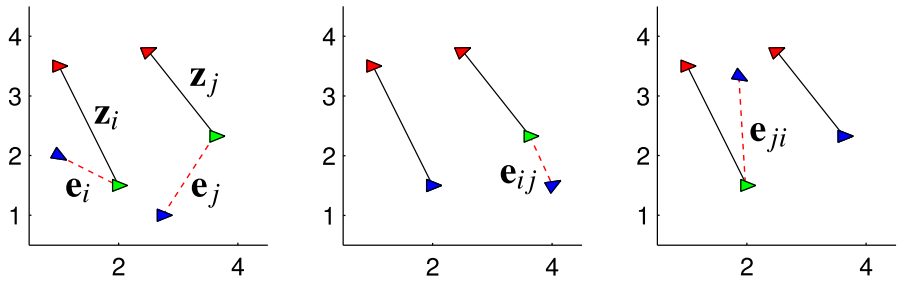


Figure 7.6: Left: Initial configuration of the nodes (red and blue) and current errors with respect to the constraints. Middle: Configuration of the nodes if the transformation which satisfies the first constraint is applied. Right: Configuration of the nodes if the transformation which satisfies the second constraint is applied.

In order to apply the SCGP method into our map alignment problem, we proceed similarly as in Algorithm 9. Given n candidate edges, we compute for each candidate edge e_i the transformation which satisfies the constraint, apply it to the nodes of the rest of edges e_j and compute the new error e_{ij} of each edge. With this procedure we have a measure of consistency of each edge e_j with respect to e_i. We can not fill matrix A directly with these values since this measure is not reciprocal (i.e. $e_{ij} \neq e_{ji}$) as it can be seen in Fig. 7.6. To obtain a symmetrix matrix A out of these values we set:

$$a_{ij} = a_{ji} = e^{-\frac{e_{ij}+e_{ji}}{2}} \tag{7.8}$$

After applying this function, two compatible edges e_i, e_j will have low error e_{ij} and large compatibility value a_{ij} and viceversa.

Simulation Results

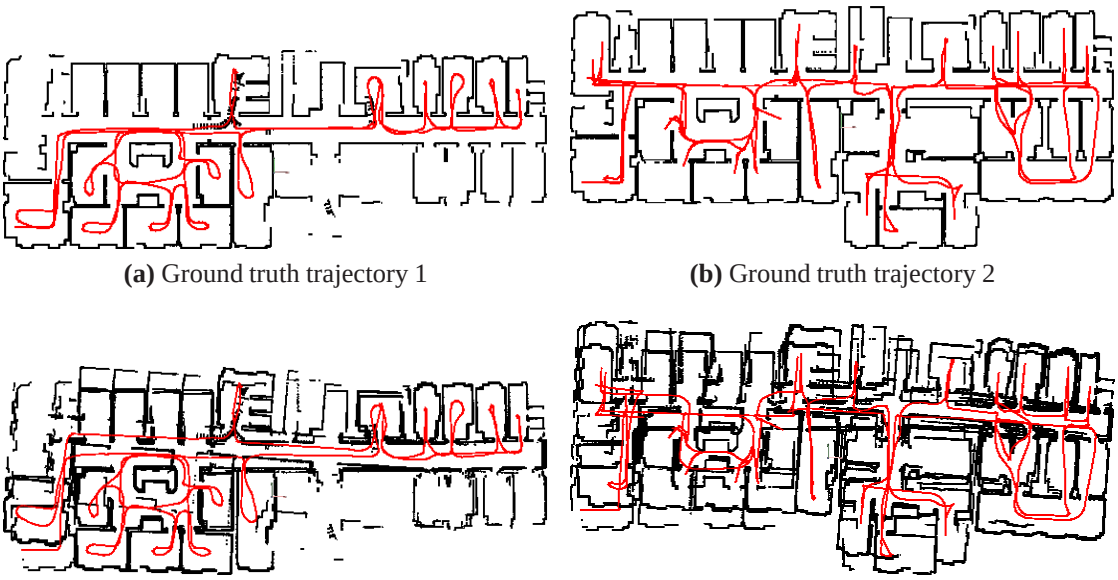
In order to compare the map alignment algorithm proposed in this chapter with the SCGP approach, we have designed two different trajectories in a simulated environment containing several loop closings as shown in Fig. 7.7. Given the initial trajectories 7.7c and 7.7d, we have executed the single-robot SLAM routine explained in section 7.4 with both approaches as loop closing detectors and varied their intrinsic parameters according to the following restrictions: 1) We check if a minimum number of edges are found jointly consistent in a window of time in order to finally approve a map alignment. In our implementation, the candidate edges are maintained in the pool for 10 time steps and we have varied this restriction in both methods from 3 to 8 minimum inliers. 2) In the case of the SCGP method, a minimum ratio λ_1/λ_2 is imposed to consider the result confident enough. This ratio depends on the properties of the consistency matrix and how it is built. In our implementation, we observed that λ_1/λ_2 ratios much higher than 2 were unusual, then we considered this value as the more restrictive one. Note that $\lambda_1/\lambda_2 > 1$ is always satisfied and, in this case, SCGP always trusts on the confidence of the consistency matrix. 3) As it is explained in Algorithm 9, the outlier rejection of our method is based on a threshold τ. We have a wider range of thresholds in this case where the lower the threshold is, the more restrictive the algorithm is.

We have measured the quality of the final maps obtained with both approaches with respect to the ground-truth trajectories in terms of the mean Chi2 error per edge. To this end, we created a ground-truth graph by extracting a set of virtual edges between neighboring nodes, by using the approach described in [13]. The results for different thresholds and minimum number of inliers are summarized in Table 7.1 where values are only shown if the map converged to a correct solution. These results are accompanied by precision and recall measures (shown in Fig. 7.8) which are defined based on the number of true/false positive/negatives by:

$$Precision = \frac{tp}{tp+fp} \quad ; \quad Recall = \frac{tp}{tp+fn} \tag{7.9}$$

From the results we can observe that our proposed algorithm obtains better final maps proved by a lower overall error. In general, the error is higher as the minimum number of inliers is increased since finding a loop closure also depends on how much portion of the trajectory overlaps and is not always possible to achieve the required inliers. Additionally, we verify that the minimum λ_1/λ_2 ratio must be higher than 1 otherwise the simulations always reach a point at which they diverge.

Regarding the precision and recall measures shown in Fig. 7.8 we observe that our approach has a slightly smaller precision than SCGP, therefore more wrong loop closing edges are added to the graph. However, it gets many more right since it has a higher recall. With the use of robust kernels in the optimization phase [65], these few outliers could be identified and their effect on the final solution minimized. Conversely,



(a) Ground truth trajectory 1 **(b)** Ground truth trajectory 2

(c) Scan matching trajectory 1 **(d)** Scan matching trajectory 2

Figure 7.7: Trajectories and laser scan data collected by the robot in a simulation environment. (a) and (b) show the ground-truth maps whereas (c) and (d) show the initial map configuration after running a sequential scan matching over the two trajectories without loop closure detections. The initial χ^2 mean error per edge is 40.412 for the first map and 63.997 for the second map.

although SCGP presents better precision values, this comes with a sacrifice on the overall number of detected inliers and, consequently, certain good loop closures are skipped by the algorithm which contributes in higher errors of the final maps obtained. These quantitative results support the overall better performance observed and confirmed by visual inspection of the final maps.

In some cases, the inclusion of outlier loop closing edges results in a complete map divergence. However, there are certain edges that can be classified as outliers from a statistical point of view although they are not so wrong from a qualitative perspective due the good performance of the scan matcher and then, their inclusion in the map do not prevent from obtaining a good final result. For this reason, we have also measured the recall values for both approaches in the situation when there are only good (statistically speaking) inliers. In this case, the absence of outliers implies there are no false positives and then, the precision is always 1. The results for both trajectories are shown in Fig. 7.9 which confirme the better performance of our method.

		Min. Inliers					
		3	4	5	6	7	8
τ	3.5	1.417	-	0.647	0.826	10.179	20.515
	3	-	-	1.105	0.682	1.119	1.441
	2.5	1.974	-	0.645	1.29	1.506	1.433
	2	0.902	0.495	1.047	0.772	2.159	2.163
	1.5	1.133	0.769	0.951	1.466	1.561	2.511

(a) Our method - Trajectory 1

		Min. Inliers					
		3	4	5	6	7	8
λ_1/λ_2	1	-	-	-	-	-	-
	1.25	18.791	35.691	0.687	0.508	1.837	4.061
	1.5	-	-	1.308	2.388	1.541	1.344
	1.75	0.465	1.305	1.841	3.554	1.572	1.344
	2	-	1.403	0.925	2.712	1.266	5.254

(b) SCGP - Trajectory 1

		Min. Inliers					
		3	4	5	6	7	8
τ	3.5	-	0.764	-	0.863	0.853	1.020
	3	-	0.647	-	0.627	0.688	0.794
	2.5	-	0.561	-	0.833	2.047	0.874
	2	1.106	0.686	-	0.72	1.174	70.585
	1.5	-	0.687	-	1.190	4.698	6.488

(c) Our method - Trajectory 2

		Min. Inliers					
		3	4	5	6	7	8
λ_1/λ_2	1	-	-	-	-	-	-
	1.25	-	-	-	1.871	16.524	73.42
	1.5	-	0.87	0.764	1.972	1.266	46.113
	1.75	-	0.723	1.197	2.941	73.821	46.114
	2	0.773	0.877	1.062	2.935	73.825	46.114

(d) SCGP - Trajectory 2

Table 7.1: Map errors obtained using both methods in the two trajectories. Rows and columns are ordered from less to more restrictive thresholds. The values are the mean χ^2 error per edge.

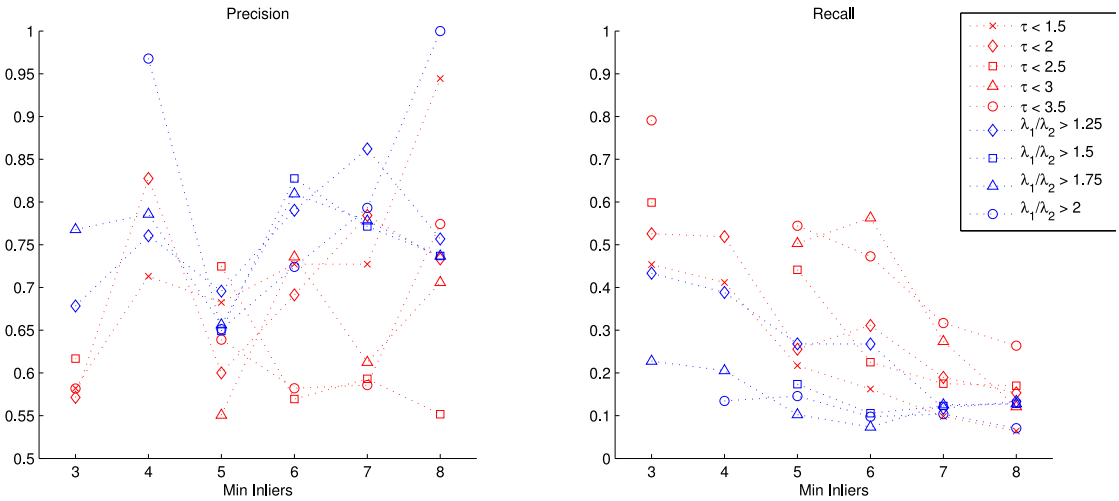
7.5.2 Multi-Robot SLAM System

Simulation results

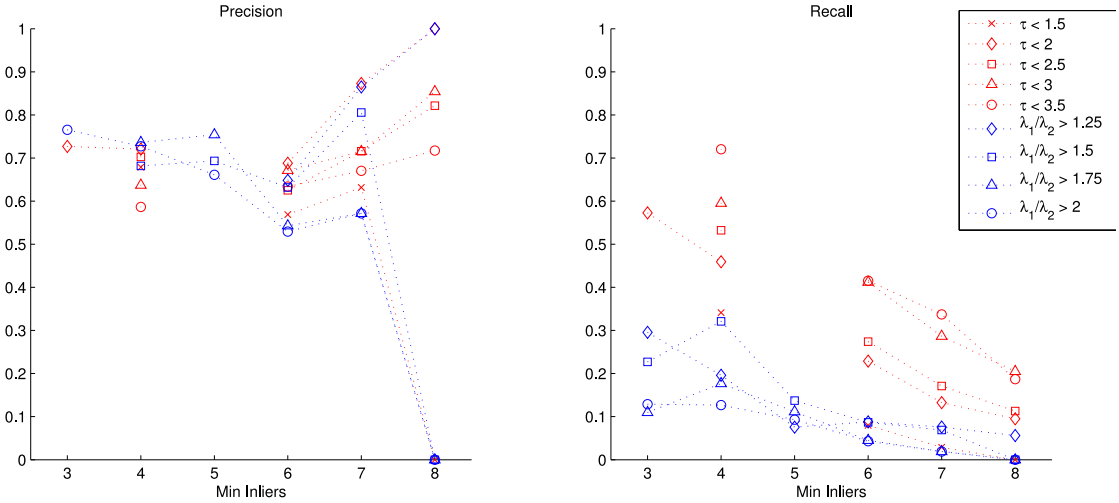
We quantitatively evaluated the performance of our system through simulation experiments. In particular, we measure how the proposed multi-robot system performs in terms of optimization time, bytes transmitted by each robot and accuracy with respect an ideal implementation in which the robots share their whole graph instead of the condensed version. Additionally, we want to analyze how these aspects scale with the number of robots and therefore we tested our approach with 2, 4 and 8 robots. The simulation environment is shown in Fig. 7.10a. We designed trajectories such that each robot met at least once with another robot. As an example, the trajectories and final map obtained in the 8 robots simulation are shown in Fig. 7.10b.

Figure 7.11 shows the results for the optimization times and communication overload obtained in the simulations. Clearly, the more robots are used for mapping the less time is needed to cover the entire environment and the smaller will be the map of each robot. Figures 7.11a–7.11c show the optimization times for both approaches. It can be seen how, in the condensed approach (green), the optimization times increase linearly as the map grows. Receiving a condensed graph implies adding a few edges to their graphs and this does not affect substantially the computation. In the ideal implementation (red), times grow also linearly with the number of edges. However, this number has a substantial increment when the robots meet and receive the whole graph from the others. This happens, for example at time 350 in the two robots simulation.

The communication overload is shown in Figures 7.11d–7.11f. As explained in



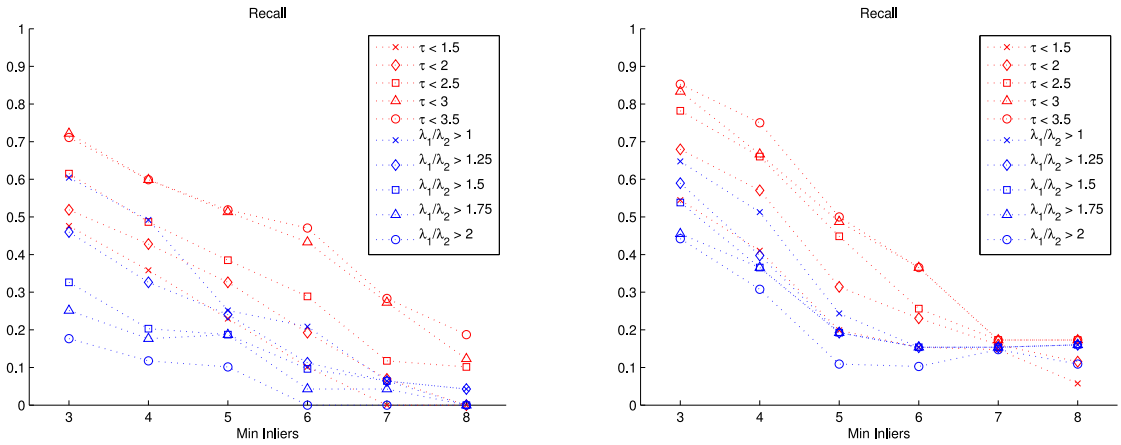
(a) Trajectory 1



(b) Trajectory 2

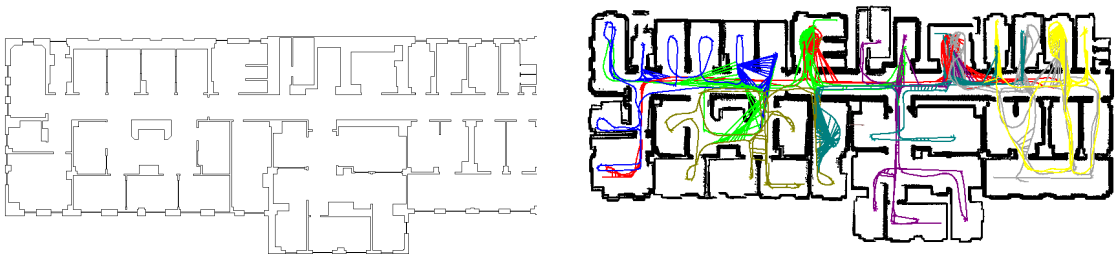
Figure 7.8: Precision and recall values for both trajectories.

section 7.4.1, two kind of messages are sent, one containing the local map and another one to send the condensed graph. Transmitting the local map has a constant size if the number of nodes to send is fixed. In our implementation, we transmit both the updated estimates and ids of the last 5 nodes plus the last laser scan obtaining a message of constant size of 1580 bytes. Since this value is the same for both condensed and ideal approaches, this type of message is not taken into account in the results. However, as it can be seen in the figures, the size of the messages to send the graph in the ideal approach differs substantially from the messages in the condensed approach, where the size of the messages stays below 1000 bytes in most of cases. The size of the message



(a) Trajectory 1 **(b)** Trajectory 2

Figure 7.9: Recall values obtained for both trajectories in the absence of outliers.

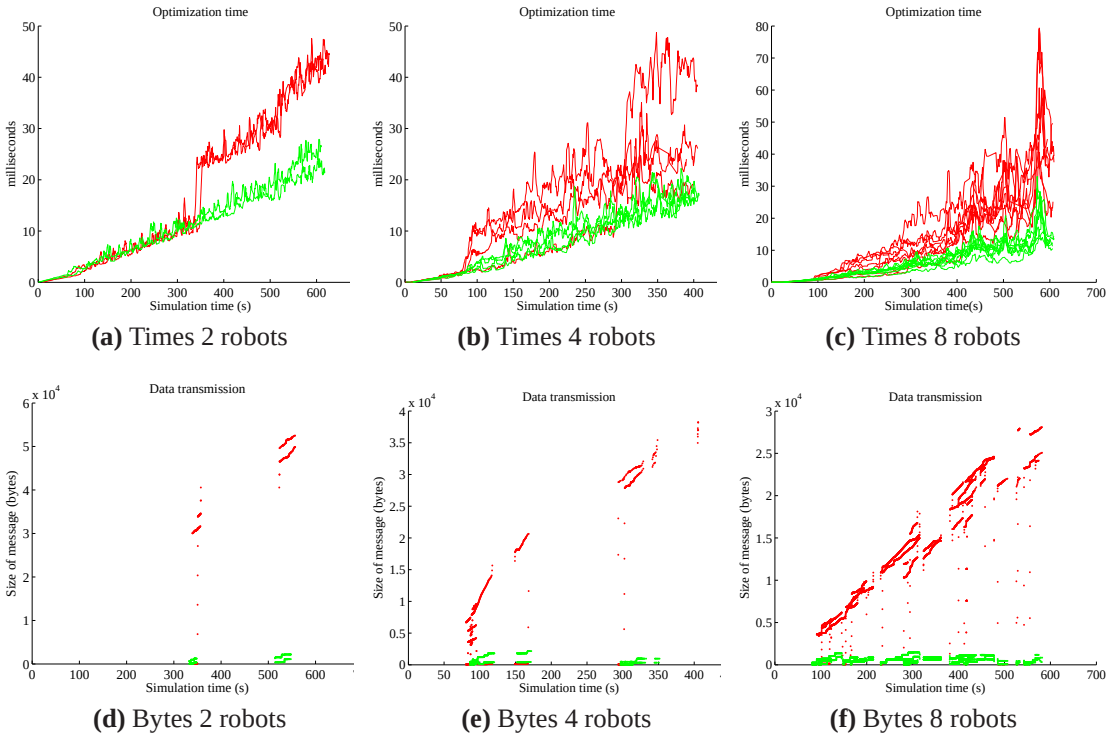


(a) **(b)**

Figure 7.10: (a) Simulation environment. (b) Trajectories and final map in a 8 robots experiment.

that a robot has to send in the condensed approach will grow with the number of nodes of its own map another robot has matched.

By using the ground truth of the simulation, we compared the accuracy of our multi-robot SLAM approach with the ideal implementation. Table 7.2 shows the overall mean Chi2 error per edge for each one of the simulations. The number of edges of each individual map varies with the simulation, and from one robot to another. For this reason we use the mean error per edge as a measure of accuracy for both approaches. As it can be seen in Table 7.2 the mean errors are very similar and therefore, we can conclude that the accuracy is not sacrificed when sharing the condensed graphs instead of the whole version. This result is confirmed by the visual inspection of the maps.



(a) Times 2 robots **(b)** Times 4 robots **(c)** Times 8 robots

(d) Bytes 2 robots **(e)** Bytes 4 robots **(f)** Bytes 8 robots

Figure 7.11: Timings for the optimization of the graph and bytes transmitted by each robot. The results are shown in green for the condensed graph approach and in red for the ideal implementation.

Real World Experiments

We conducted a real world experiment by using three Pioneer 3-AT robots, equipped with SICK laser rangefinders. The robots were simultaneously controlled by three persons that steered them manually in the environment shown in Figure 7.12. The robots communicated through an Ad-Hoc network by sending UDP packets and each of them was running the algorithm described in this paper. We previously synchronized the clocks of all robots with NTP. To be able to reproduce the experiment, we recorded a dataset containing the own measurements each robot logged its own measurements (odometry and laser), and the ping received by other robots. This allow us to reproduce off-line the connectivity of the communication network, and repeat the experiment off-board.

The results of this experiment are shown in Figure 7.12. The individual maps obtained by each robot together with the condensed graphs received from other robots are depicted in Figures 7.12a–7.12c. During their navigation, each robot was able to meet and localize some other robot into its own map. The meeting points are depicted with

	Accuracy	
	Condensed Graphs	Ideal
2 robots	1.404	1.442
4 robots	1.572	1.548
8 robots	1.884	1.899

Table 7.2: Comparison of the accuracy obtained by our condensed measurement multi-robot SLAM approach and the ideal implementation. The numbers are the χ^2 error of the edges in the ground-truth graph, evaluated with nodes placed as reported by the algorithm.

squares in the individual views. These intra-robot localizations make that all maps become interconnected which allows us to reconstruct the global map shown in Figure 7.12d.

In addition to the experiment described here, we executed additional tests with Erratic robots equipped with an Hokuyo UTM laser rangefinder with two robots. The result after merging the individual maps is shown in Figure 7.13.

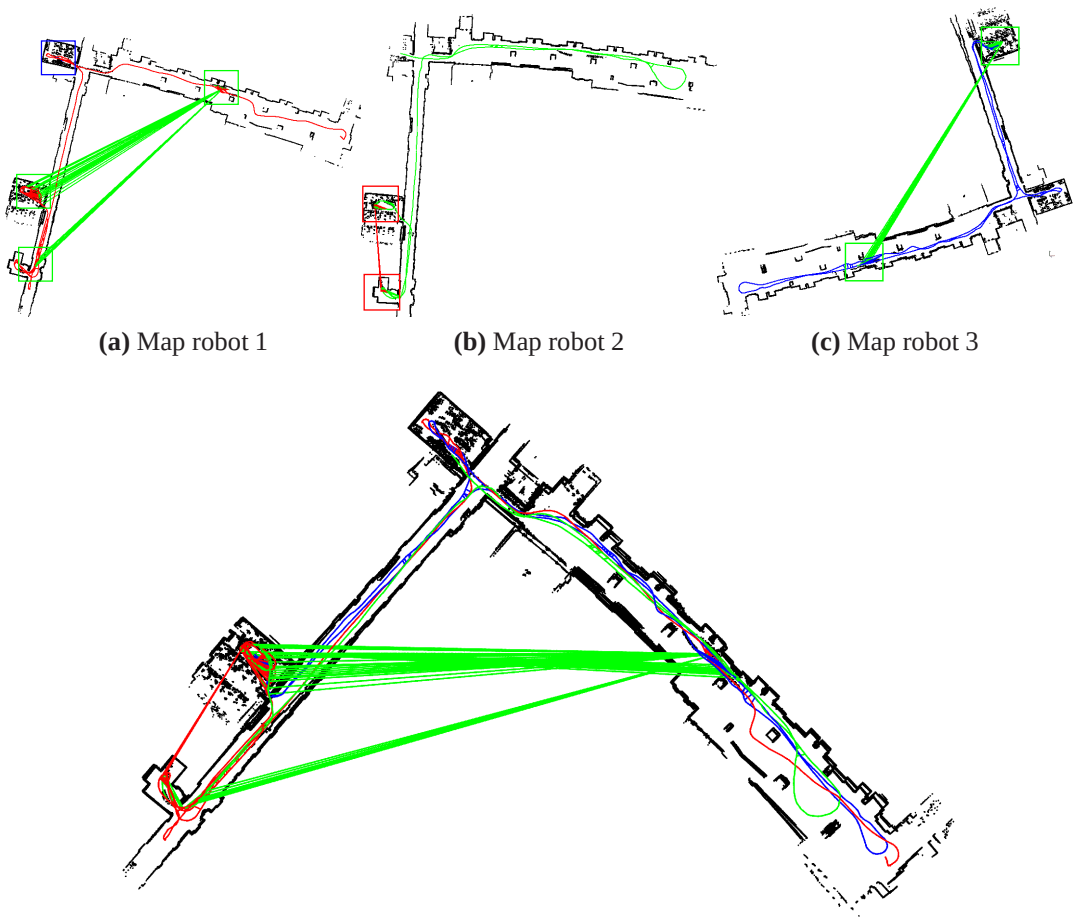
Post Processing

The procedures described above are the core of our multi-robot SLAM. Compared with a centralized approach that has access to all information of all robots, our system leads to a higher error in positions where the robots do not meet. This arises from the fact that robots only share local maps around their current position, thus they cannot relocalize. This is visible in the right hand side of Figure 7.13. Solving this problem would require transmitting substantial more information, since the robots would have to share all the measurements. Despite this limitation, our schema produces solutions that are sufficient for the robots to navigate. In a subsequent processing stage, a global accurate map can be obtained by merging the solutions of all robots and optimizing them including the condensed measurements. This aligns the map in a global frame. This map can be further improved by adding a set of constraints by matching scans between neighboring nodes. Due to the good initial guess obtained by the map alignment, this step is relatively straightforward, leading to results illustrated in Figure 7.14.

7.6 Conclusions

In this chapter we have proposed an approach for multi-robot SLAM that specifically addresses the limitations in network and computation affecting multi-robot systems.

The use of condensed measurements allows to efficiently share map information among robots which is easily incorporated into each robot's graph. The exchange of data is carried out under a communication protocol designed to maximize the probability that



(a) Map robot 1 **(b)** Map robot 2 **(c)** Map robot 3

(d) Global map

Figure 7.12: Multi-robot SLAM experiment at the Ada Byron building of the University of Zaragoza. (a), (b) and (c) show the individual maps obtained by each robot, depicted in red, green and blue respectively. The condensed graphs received from other robots are depicted in the colour of the sender robot. (d) shows the global map after all individual maps are merged and jointly optimized.

the messages are successfully delivered and to guarantee each robot receives the most up-to-date information available.

Graph optimization results in consistent and accurate map estimates also thanks to the good initial guess provided by a robust front-end where a map alignment procedure is used to detect loop closures and solve intra-robot data associations. Our method adds a relatively limited complexity to the traditional single-robot SLAM methods resulting in an overall increase of robustness and computational efficiency with respect to naive multi-robot SLAM implementations.

Figure 7.13: Experiment at the DIS building of La Sapienza University of Rome. The misalignment observed in the bottom right corner originates from the fact that the robots never meet in that region, thus they are unable to determine constraints between that part of their trajectories. This can be recovered when the two robots meet in that region, or in a post-processing phase.

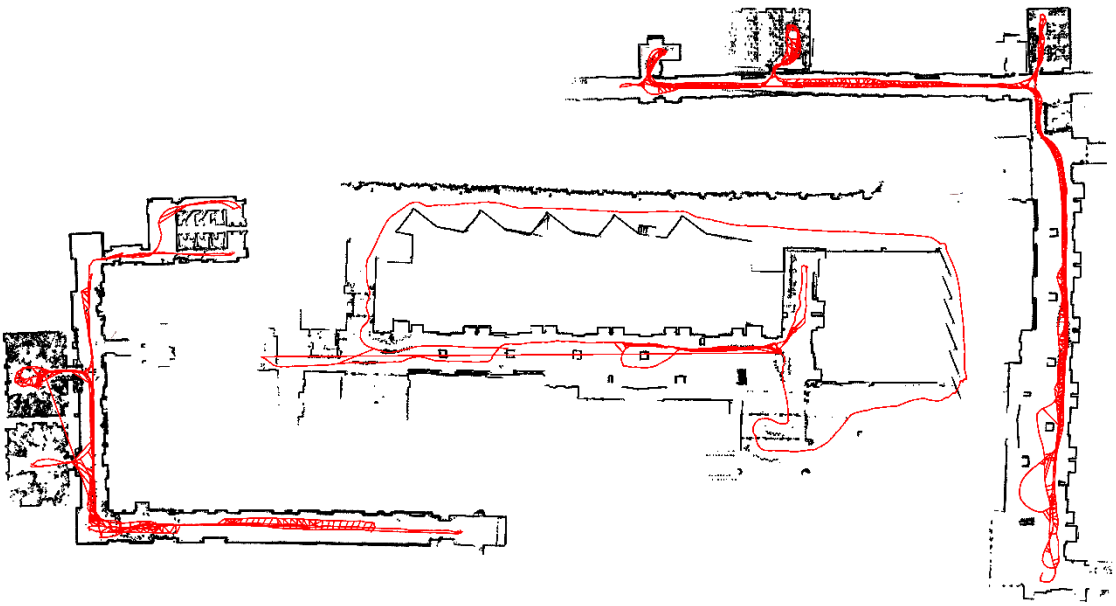


Figure 7.14: Results of the three real world experiments performed to verify the proposed approach, using a straightforward centralized processing of the joint estimates obtained by our multi-robot SLAM method.

Chapter 8

Conclusions

8.1 Conclusions

Along this thesis we presented contributions in the different tasks that must be performed to obtain an autonomous multi-robot system. We have mainly addressed the localization and SLAM problems but we have also contributed to the motion and planning problems to analyze their application and performance in a real working system.

First of all, we considered the problem of how the team of robots can move in the environment. Then, in Chapter 3 we presented a cooperative navigation system for robot formations based on a flexible virtual structure that adapts its shape to the environment. In this chapter we also analyzed the importance of having a good localization system to deal with the accumulation of errors introduced by the sensors which prevent the team from a correct navigation in formation and the reachability of its goal.

This issue motivated the design of a localization algorithm presented in Chapter 4 developed in a EKF filtering framework to take advantage of its speed properties. The state of the formation was formulated using a leader-centric probabilistic representation to reduce the effect of the linearization errors due to high levels of uncertainty. Using a given feature-based stochastic map of the environment for absolute positioning reference, we demonstrated the direct implementation of the EKF algorithm provided inconsistent, and therefore, unreliable localization estimates, a matter of importance if we want to assure the long-term performance of the method. However, although the convergence in this nonlinear filtering context can not be guaranteed we can propose alternatives to drive the estimation towards consistency. Insights into the underlying problem suggested us that cross-correlations among map features observed in consecutive time steps were not properly handled by the algorithm and we proposed a solution based on the measurement differencing technique to improve the filter consistency.

This localization algorithm was part of an integrated system for robot formations described in Chapter 5. With the overall objective of having the formation working

in partially known environments, we proposed, in addition to the navigation and localization techniques from previous chapters, a solution to the path planning problem considering the uncertainty of a given map. This is motivated since planning through imprecisely known areas may lead to reach dead-ends requiring a complete global replanning, increasing the total execution time of the mission. The global path planning is carried out by the leader of the formation based on the prior map before starting the mission, thus this is the less cooperative part of the system. It is, on the contrary, the online replanning which takes into account the unexpected elements of the environment not considered in the prior map and makes use of a local map built from the cooperative perception of the team.

Up to this point, we considered the map provided to the formation, probably imprecise or incomplete, was not modified during the execution of the mission. In Chapter 6, we proposed a distributed SLAM algorithm to jointly improve the given map while the robots navigate in the environment. The use of the Conditional Independence property opened us the possibility to work with submaps, greatly decreasing the computational cost of the mapping process while obtaining the same result as if we worked with the entire map. Besides, it allowed us to bound the amount of information shared through the network and to guarantee the robots have exactly the same information each time they synchronize.

Finally, in Chapter 7 we addressed the problem of how to use multiple robots to build a map of the environment from scratch, without any prior information about the number of robots or their initial locations. We provided a complete multi-robot SLAM system handling intra-robot data association and capable of working under computational and communication resource constraints. Concretely, the use of condensed maps enabled the robots to share portions of their own maps relevant to refine those of others. The transmission of information was conducted by means of an event-driven communication protocol oriented to minimize the possibility of packet loss. As a result of these strategies we considerably reduced the optimization problem to be solved by each robot obtaining a comparable accuracy to an ideal implementation, where robots shared their whole maps.

8.2 Future Work

In this thesis we have demonstrated the performance and efficiency of the proposed contributions, most of them working on real multi-robot systems. However, one can always progress and do things better. This section points out some possible improvements.

Regarding the cooperative navigation of the formation, we observed how in some situations (see for example those in Figs. 3.5 and 5.7) sudden changes in the leader trajectory result in forced behaviors on the followers, which always try to reconfigure to maintain their relative positions with respect to the leader. A more efficient strategy

considering the prediction of trajectories of the followers combined with the proposed formation control could result in shorter and smoother trajectories for the followers. Also, a dynamic change of the leader role inside the formation, if the system configuration allows it, would provide more efficient performance.

The proposed localization system provides consistent estimates, however, it is implemented in a centralized manner relying heavily on the leader. Future work should be oriented towards distributed approaches where the EIF, due to its inherent decoupling properties could be used. We provide in appendix C first insights into the measurement differencing EIF-based formulation.

The path planning under uncertainty algorithm proposed in this thesis seeks for the path that minimizes the risk at each of the steps. This solution considers that, once the robot reaches a step k of the planned trajectory, it does not matter which is the risk of the $k-1$ previous steps since this part of the trajectory has already successfully been traversed and it is only concerned if it will be able to traverse step $k+1$. Other kind of approaches try to minimize the accumulated risk of all path and we are already moving towards this reasearch line where the cost of choosing a wrong path and the need of rectifying is also being considered.

The multi-robot localization and map improvement algorithm presented in Chapter 6 could be extended to consider new features not present in the prior map. In the case of non identifiable features (i.e., unknown data association) this would require a consensus between robots to decide whether the new features they observed are the same or not. Besides, we would have to take care of the map maintainance for long-term performance since indefinitely updates could lead to overly optimistic map estimation. Optimal submap partitioning and asynchronous communications could also been investigated.

The multi-robot graph-based SLAM approach presented in Chapter 7 may show map misalignments in areas where robots did not coincide in time but which we resolved in a post-processing phase of the joined map. This additional step could be avoided in the ideal but inefficient implementation where robots share their whole maps. Instead, robots could also interchange condensed maps of strategic areas of the environment they visited at different times. Furthermore, the map alignment algorithm could be used not only to correctly align two local maps but also to disambiguate between local alignments obtained at different parts of the trajectory. This ambiguity may also appear when two robots meet in different but symmetric scenarios. A robot identification system could also be used to solve it but at the expense of restricting robot movements so their identifiers were visible to each other.

8.3 Conclusiones

A lo largo de esta tesis hemos presentado contribuciones en las diferentes tareas que han de llevarse a cabo para conseguir un sistema multi-robot autónomo. Aunque nos hemos centrado principalmente en los problemas de localización y SLAM, también hemos contribuido a los problemas de control del movimiento y el cálculo de trayectorias para analizar su aplicación y rendimiento en un sistema real completo.

En primer lugar, hemos abordado el problema de cómo puede un equipo de robots moverse por el entorno. Así pues, en el capítulo 3 hemos presentado un sistema de navegación cooperativa para formaciones de robots basado en un estructura virtual flexible que se adapta al entorno. En este capítulo también hemos analizado la importancia de disponer de un buen sistema de localización para hacer frente a la acumulación de errores introducidos por los sensores que impiden la correcta navegación en formación y el alcance del objetivo.

Esta cuestión ha motivado el diseño de un algoritmo de localización presentado en el capítulo 4 y desarrollado en el contexto del filtro de Kalman extendido (EKF) con el fin de aprovechar su rapidez. Se ha formulado el estado de la formación utilizando una representación probabilista centrada en el líder que permite reducir los efectos de los errores de linealización debidos a altos niveles de incertidumbre. Utilizando un mapa estocástico de características del entorno como referencia para su localización global, hemos demostrado que una implementación directa del EKF resultaba en estimaciones de localización inconsistentes y, por consiguiente, poco fiables, siendo éste un asunto de importancia para la aplicación del algoritmo a largo plazo. Sin embargo, aunque en este contexto de filtrado no lineal no se pueda garantizar la convergencia, podemos proponer alternativas para tratar de mejorar la consistencia. Así pues, un análisis más en profundidad del problema nos ha llevado a sugerir que las correlaciones entre características del mapa observadas en instantes consecutivos no estaban siendo consideradas adecuadamente en el algoritmo y hemos propuesto una modificación basada en la diferencia de medidas para mejorar la consistencia del filtro.

Este algoritmo de localización ha formado parte de un sistema integrado para formaciones de robots, descrito en el capítulo 5. Con el objetivo de tener a la formación trabajando en entornos parcialmente conocidos, hemos propuesto, además de las técnicas de navegación y localización descritas en los capítulos anteriores, una solución al problema de planificación de trayectorias considerando la incertidumbre del mapa proporcionado. Este problema viene motivado por el hecho de que planificar por zonas conocidas de modo impreciso puede llevar a caminos sin salida que requieran una completa replanificación global, incrementando el tiempo total de ejecución de la misión. Es el líder quien se encarga de realizar la planificación global antes de iniciar la misión, siendo esta por tanto la parte menos cooperativa de la tesis. Sí lo es, sin embargo, la replanificación online que tiene en cuenta los elementos imprevistos del entorno o no

considerados en el mapa previo y que hace uso de un mapa local construido a partir de la percepción cooperativa del equipo.

Hasta este punto, se había considerado que el mapa proporcionado a la formación, probablemente impreciso o incompleto, no era modificado durante la realización de la misión. En el capítulo 6 hemos propuesto un algoritmo de SLAM distribuido para mejorar conjuntamente el mapa al mismo tiempo que los robots navegan por el entorno. El uso de la propiedad de independencia condicional nos ha dado la posibilidad de trabajar con submapas, reduciendo ampliamente el coste computacional de la construcción del mapa y obteniendo el mismo resultado que si hubiéramos trabajado con el mapa completo. Además nos ha permitido limitar la cantidad de información a ser transmitida a través de la red y garantizar que los robots dispusieran de exactamente la misma información cada vez que se sincronizaran.

Finalmente, en el capítulo 7 hemos abordado el problema de cómo usar varios robots para construir un mapa del entorno desde cero, sin ninguna información previa acerca del número de robots o su localización en el mapa. Hemos propuesto un sistema de SLAM multi-robot completo que resuelve el problema de asociación de datos entre robots y es capaz de trabajar bajo recursos computacionales y de comunicación limitados. Concretamente, el uso de mapas condensados ha permitido que los robots compartan información de sus propios mapas relevante para mejorar los de otros. El intercambio de información se ha llevado a cabo por medio de un protocolo de comunicaciones orientado a minimizar la posibilidad de pérdida de paquetes. Como resultado de estas estrategias hemos reducido considerablemente el problema de optimización que cada robot debe resolver, obteniendo al mismo tiempo una precisión comparable a una implementación ideal, donde los robots comparten su mapa completo.

8.4 Trabajo Futuro

En esta tesis hemos demostrado el rendimiento y eficiencia de las contribuciones propuestas, teniendo la mayoría de ellas funcionando en equipos multi-robot reales. Sin embargo, uno siempre puede progresar y hacer mejor las cosas. En esta sección señalamos algunas de las posibles mejoras.

Respecto a la navegación cooperativa de la formación, observamos cómo, en algunas situaciones (por ejemplo, las mostradas en las figuras 3.5 y 5.7) cambios repentinos en la trayectoria del líder dan lugar a comportamientos forzados en los seguidores, que siempre tratan de reconfigurarse para mantener su posición relativa con el líder. Una estrategia más eficiente que considerara la predicción de las trayectorias de los seguidores en combinación con el control de la formación propuesto podría resultar en trayectorias más cortas y suaves para ellos. Además, si la configuración del sistema lo permite, un cambio dinámico de líder dentro de la formación podría proporcionar un mejor y más eficiente rendimiento.

El sistema de localización propuesto proporciona estimaciones consistentes, sin embargo, está implementado de forma centralizada, dependiendo excesivamente del líder. El trabajo futuro debería orientarse a obtener algoritmos distribuidos en los que el filtro de información extendido (EIF) por sus propiedades inherentes para desacoplar la estimación podría usarse. En el apéndice C ofrecemos una primera visión sobre la formulación del EIF con diferencia de medidas.

El algoritmo de planificiación de caminos bajo incertidumbre propuesto en esta tesis busca el camino que minimiza el riesgo en cada uno de sus pasos. Esta solución considera que, una vez que el robot alcanza un paso k de la trayectoria, no importa cual era el riesgo de los $k-1$ pasos anteriores ya que esta parte del recorrido ya ha sido atravesado con éxito y sólo le preocupa si será capaz de atravesar el siguiente paso $k+1$. Otro tipo de estrategias tratan de minimizar el riesgo acumulado en todo el camino y estamos actualmente estudiando esta otra línea de investigación donde también consideramos el coste de elegir un camino incorrecto y tener que rectificar.

El algoritmo de localización multi-robot y mejora de mapa presentado en el capítulo 6 podría extenderse para considerar nuevas características no presentes en el mapa previo. En el caso de características no identificables requeriría un consenso entre robots para decidir si las nuevas características observadas por varios robots son las mismas o no. Además, sería necesario prestar atención al mantenimiento del mapa para un rendimiento a largo plazo, ya que la actualización del mapa indefinidamente podría llevar a estimaciones excesivamente optimistas. La división óptima de submapas y las comunicaciones asíncronas también podrían ser investigadas.

El método de SLAM multi-robot basado en grafos presentado en el capítulo 7 puede mostrar zonas del mapa mal alineadas si los robots no coincidieron al mismo tiempo en esa zona, lo cual resolvimos procesando el mapa conjunto en una fase posterior. Este paso adicional puede evitarse en la implementación ideal pero ineficiente en la que los robots comparten sus mapas completos. En su lugar, los robots podrían intercambiar mapas condensados de áreas estratégicas del entorno que hubieran visitado en distintos instantes. Además, el algoritmo de alineación de mapas podría utilizarse no solamente para alinear mapas locales sino también para desambiguar entre alineaciones obtenidas en distintas partes de la trayectoria. Esta ambigüedad también puede aparecer cuando dos robots se encuentran en escenarios diferentes pero simétricos. Un sistema de identificación de robots podría utilizarse para resolver este problema aunque a costa de restringir los movimientos de los robots para que sus identificadores fueran visibles entre sí.

Appendix A

Equations of the EKF-based Localization of the Formation

A.1 Process model equations

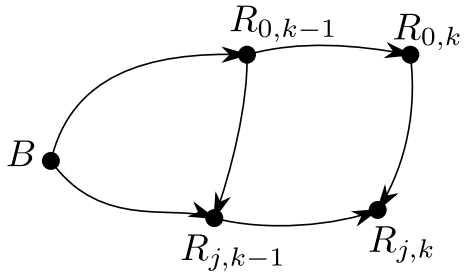


Figure A.1: Geometric relations between the robot leader R_0 and a robot follower R_j during the state transition from time step $k-1$ to time step k.

Given the geometric relations given in Fig. (A.1), the state of the robot formation is propagated from time step $k-1$ to time step k using the estimated displacements $\mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}} \sim \mathcal{N}(\hat{\mathbf{x}}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}, \mathbf{P}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}})$ through the following function:

$$\mathbf{x}_{\mathcal{R}_k} = \mathbf{f}(\mathbf{x}_{\mathcal{R}_{k-1}}, \mathbf{x}_{\mathcal{R}_k}^{\mathcal{R}_{k-1}}) \tag{A.1}$$

$$= \begin{pmatrix} \mathbf{f}_{R_0}(\mathbf{x}_{R_{0,k-1}}^{B}, \mathbf{x}_{R_{0,k}}^{R_{0,k-1}}) \\ \mathbf{f}_{R_j}(\mathbf{x}_{R_{0,k}}^{R_{0,k-1}}, \mathbf{x}_{R_{j,k-1}}^{R_{0,k-1}}, \mathbf{x}_{R_{j,k}}^{R_{j,k-1}}) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{R_{0,k-1}}^{B} \oplus \mathbf{x}_{R_{0,k}}^{R_{0,k-1}} \\ \ominus \mathbf{x}_{R_{0,k}}^{R_{0,k-1}} \oplus \mathbf{x}_{R_{j,k-1}}^{R_{0,k-1}} \oplus \mathbf{x}_{R_{j,k}}^{R_{j,k-1}} \end{pmatrix} \tag{A.2}$$

where $\mathbf{f}_{R_0}$ and $\mathbf{f}_{R_j}$ are the state transition functions for the robot leader and for a robot follower R_j respectively.

Linearization of the equations using first order Taylor series expansion give us:

$$\mathbf{f}_{R_0} \simeq \mathbf{f}_{R_0}(\hat{\mathbf{x}}^B_{R_{0,k-1}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}) + \mathbf{F}_{R_0}(\mathbf{x}^B_{R_{0,k-1}} - \hat{\mathbf{x}}^B_{R_{0,k-1}}) + \mathbf{G}_{R_0}(\mathbf{x}^{R_{0,k-1}}_{R_{0,k}} - \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}) \tag{A.3}$$

$$\begin{aligned}\mathbf{f}_{R_j} \simeq\ & \mathbf{f}_{R_j}(\hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k-1}}, \hat{\mathbf{x}}^{R_{j,k-1}}_{R_{j,k}}) \\ & + \mathbf{G}^{R_0}_{R_j}(\mathbf{x}^{R_{0,k-1}}_{R_{0,k}} - \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}) + \mathbf{F}_{R_j}(\mathbf{x}^{R_{0,k-1}}_{R_{j,k-1}} - \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k-1}}) + \mathbf{G}_{R_j}(\mathbf{x}^{R_{j,k-1}}_{R_{j,k}} - \hat{\mathbf{x}}^{R_{j,k-1}}_{R_{j,k}})\end{aligned} \tag{A.4}$$

where

$$\mathbf{F}_{R_0} = \frac{\partial \mathbf{f}_{R_0}}{\partial \mathbf{x}^B_{R_{0,k-1}}} = \mathbf{J}_{1\oplus}\{\hat{\mathbf{x}}^B_{R_{0,k-1}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}\} \tag{A.5}$$

$$\mathbf{G}_{R_0} = \frac{\partial \mathbf{f}_{R_0}}{\partial \mathbf{x}^{R_{0,k-1}}_{R_{0,k}}} = \mathbf{J}_{2\oplus}\{\hat{\mathbf{x}}^B_{R_{0,k-1}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}\} \tag{A.6}$$

$$\mathbf{G}^{R_0}_{R_j} = \frac{\partial \mathbf{f}_{R_j}}{\partial \mathbf{x}^{R_{0,k-1}}_{R_{0,k}}} = \mathbf{J}_{1\oplus}\{\ominus \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k}}\}\mathbf{J}_{\ominus}\{\hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}\} \tag{A.7}$$

$$\mathbf{F}_{R_j} = \frac{\partial \mathbf{f}_{R_j}}{\partial \mathbf{x}^{R_{0,k-1}}_{R_{j,k-1}}} = \mathbf{J}_{2\oplus}\{\ominus \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k}}\}\mathbf{J}_{1\oplus}\{\hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k-1}}, \hat{\mathbf{x}}^{R_{j,k-1}}_{R_{j,k}}\} \tag{A.8}$$

$$\mathbf{G}_{R_j} = \frac{\partial \mathbf{f}_{R_j}}{\partial \mathbf{x}^{R_{j,k-1}}_{R_{j,k}}} = \mathbf{J}_{2\oplus}\{\ominus \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{0,k}}, \hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k}}\}\mathbf{J}_{2\oplus}\{\hat{\mathbf{x}}^{R_{0,k-1}}_{R_{j,k-1}}, \hat{\mathbf{x}}^{R_{j,k-1}}_{R_{j,k}}\} \tag{A.9}$$

Then, the EKF Prediction of the robot formation is,

$$\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}} = \mathbf{f}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{x}}^{\mathcal{R}_{k-1}}_{\mathcal{R}_k}) \tag{A.10}$$

$$\mathbf{P}_{\mathcal{R}_{k|k-1}} = \mathbf{F}_{k-1}\mathbf{P}_{\mathcal{R}_{k-1}}\mathbf{F}^T_{k-1} + \mathbf{G}_{k-1}\mathbf{P}^{\mathcal{R}_{k-1}}_{\mathcal{R}_k}\mathbf{G}^T_{k-1} \tag{A.11}$$

with,

$$\mathbf{F}_{k-1} = \begin{pmatrix} \mathbf{F}_{R_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{R_j} \end{pmatrix}; \mathbf{G}_{k-1} = \begin{pmatrix} \mathbf{G}_{R_0} & \mathbf{0} \\ \mathbf{G}^{R_0}_{R_j} & \mathbf{G}_{R_j} \end{pmatrix} \tag{A.12}$$

A.2 Measurement model equations

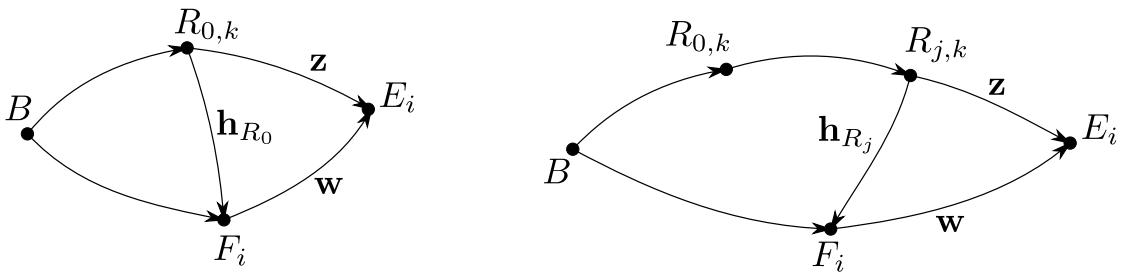


Figure A.2: Pairing between an observation E_i and a feature F_i in the case of a robot leader (R_0, left) and a robot follower (R_j, right).

Following the geometric relations represented in Fig. (A.2) a sensor observation E_i is related to the robot's position and an environmental feature F_i through the nonlinear measurement model,

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{F}_k}) + \mathbf{w}_k \tag{A.13}$$

$$= \begin{pmatrix} \mathbf{h}_{R_0}(\mathbf{x}^B_{R_{0,k}}, \mathbf{y}_{\mathcal{F}_k}) \\ \mathbf{h}_{R_j}(\mathbf{x}^B_{R_{0,k}}, \mathbf{x}^{R_{0,k}}_{R_{j,k}}, \mathbf{y}_{\mathcal{F}_k}) \end{pmatrix} + \mathbf{w}_k = \begin{pmatrix} \ominus \mathbf{x}^B_{R_{0,k}} \oplus \mathbf{y}_{F_i} \\ \ominus \mathbf{x}^{R_{0,k}}_{R_{j,k}} \ominus \mathbf{x}^B_{R_{0,k}} \oplus \mathbf{y}_{F_i} \end{pmatrix} + \mathbf{w}_k \tag{A.14}$$

where $\mathbf{w}_k$ is a zero-mean white gaussian noise with covariance matrix $\mathbf{R}_k$.

Due to our leader-centric representation of the robot formation, function $\mathbf{h}_k$ differs depending on the role of the robot, being $\mathbf{h}_{R_0}$ and $\mathbf{h}_{R_j}$ the function for the robot leader and for a robot follower R_j respectively.

The use of the EKF requires a first order linearization of the nonlinear measurement model, thus,

$$\mathbf{h}_{R_0} \simeq \mathbf{h}_{R_0}(\hat{\mathbf{x}}^B_{R_{0,k}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{H}_{R_0}(\mathbf{x}^B_{R_{0,k}} - \hat{\mathbf{x}}^B_{R_{0,k}}) + \mathbf{G}_{R_0}(\mathbf{y}_{\mathcal{F}_k} - \hat{\mathbf{y}}_{\mathcal{F}_k}) \tag{A.15}$$

$$\mathbf{h}_{R_j} \simeq \mathbf{h}_{R_j}(\hat{\mathbf{x}}^B_{R_{0,k}}, \hat{\mathbf{x}}^{R_{0,k}}_{R_{j,k}}, \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{H}^{R_0}_{R_j}(\mathbf{x}^B_{R_{0,k}} - \hat{\mathbf{x}}^B_{R_{0,k}}) + \mathbf{H}_{R_j}(\mathbf{x}^{R_{0,k}}_{R_{j,k}} - \hat{\mathbf{x}}^{R_{0,k}}_{R_{j,k}}) + \mathbf{G}_{R_j}(\mathbf{y}_{\mathcal{F}_k} - \hat{\mathbf{y}}_{\mathcal{F}_k}) \tag{A.16}$$

where

$$\mathbf{H}_{R_0} = \frac{\partial \mathbf{h}_{R_0}}{\partial \mathbf{x}^B_{R_{0,k}}} = \mathbf{J}_{1\oplus}\{\ominus\hat{\mathbf{x}}^B_{R_{0,k}}, \hat{\mathbf{y}}_{F_i}\}\mathbf{J}_{\ominus}\{\hat{\mathbf{x}}^B_{R_{0,k}}\} \tag{A.17}$$

$$\mathbf{G}_{R_0} = \frac{\partial \mathbf{h}_{R_0}}{\partial \mathbf{y}_{\mathcal{F}_k}} = \mathbf{J}_{2\oplus}\{\ominus\hat{\mathbf{x}}^B_{R_{0,k}}, \hat{\mathbf{y}}_{F_i}\} \tag{A.18}$$

$$\mathbf{H}^{R_0}_{R_j} = \frac{\partial \mathbf{h}_{R_j}}{\partial \mathbf{x}^B_{R_{0,k}}} = \mathbf{J}_{1\oplus}\{\ominus\hat{\mathbf{x}}^B_{R_{j,k}}, \hat{\mathbf{y}}_{F_i}\}\mathbf{J}_{2\oplus}\{\ominus\hat{\mathbf{x}}^{R_{0,k}}_{R_{j,k}}, \ominus\hat{\mathbf{x}}^B_{R_{0,k}}\}\mathbf{J}_{\ominus}\{\hat{\mathbf{x}}^B_{R_{0,k}}\} \tag{A.19}$$

$$\mathbf{H}_{R_j} = \frac{\partial \mathbf{h}_{R_j}}{\partial \mathbf{x}^{R_{0,k}}_{R_{j,k}}} = \mathbf{J}_{1\oplus}\{\ominus\hat{\mathbf{x}}^B_{R_{j,k}}, \hat{\mathbf{y}}_{F_i}\}\mathbf{J}_{1\oplus}\{\ominus\hat{\mathbf{x}}^{R_{0,k}}_{R_{j,k}}, \ominus\hat{\mathbf{x}}^B_{R_{0,k}}\}\mathbf{J}_{\ominus}\{\hat{\mathbf{x}}^{R_{0,k}}_{R_{j,k}}\} \tag{A.20}$$

$$\mathbf{G}_{R_j} = \frac{\partial \mathbf{h}_{R_j}}{\partial \mathbf{y}_{\mathcal{F}_k}} = \mathbf{J}_{2\oplus}\{\ominus\hat{\mathbf{x}}^B_{R_{j,k}}, \hat{\mathbf{y}}_{F_i}\} \tag{A.21}$$

The classical EKF update equations provide estimates for the state vector $\hat{\mathbf{x}}_{\mathcal{R}_{k|k}}$ and its associated covariance matrix $\mathbf{P}_{\mathcal{R}_{k|k}}$:

$$\begin{aligned}\hat{\mathbf{x}}_{\mathcal{R}_{k|k}} &= \hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_{k|k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_k}))\\ \mathbf{P}_{\mathcal{R}_{k|k}} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{\mathcal{R}_{k|k-1}}\end{aligned} \tag{A.22}$$

using the filter gain obtained as,

$$\mathbf{K}_k = \mathbf{P}_{\mathcal{R}_{k|k-1}}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{\mathcal{R}_{k|k-1}}\mathbf{H}_k^T + \mathbf{G}_{\mathcal{F}_k}\mathbf{P}_{\mathcal{F}_k}\mathbf{G}_{\mathcal{F}_k}^T + \mathbf{R}_k)^{-1} \tag{A.23}$$

where matrices $\mathbf{H}_k$ and $\mathbf{G}_{\mathcal{F}_k}$ are formed by,

$$\mathbf{H}_k = \begin{pmatrix} \mathbf{H}_{R_0} & \mathbf{0} \\ \mathbf{H}^{R_0}_{R_j} & \mathbf{H}_{R_j} \end{pmatrix}; \mathbf{G}_{\mathcal{F}_k} = \begin{pmatrix} \mathbf{0} & \dots & \mathbf{G}_{R_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{R_j} & \dots & \mathbf{0} \end{pmatrix} \tag{A.24}$$

Appendix B

Derivation of the Measurement-Differencing EKF-based Equations

Let $\mathbf{r}_k$ be expressed as,

$$\mathbf{r}_k \triangleq \mathbf{z}_k - \mathbf{\Lambda}_k \mathbf{z}_{k-1}$$

Substituting the observations $\mathbf{z}_k$ and $\mathbf{z}_{k-1}$ by their linearized expressions given by eq. (4.14),

$$\begin{aligned}\mathbf{r}_k \simeq\ & \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_k}, \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{H}_k(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k}) + \mathbf{G}_{\mathcal{F}_k}(\mathbf{y}_{\mathcal{F}_k} - \hat{\mathbf{y}}_{\mathcal{F}_k}) + \mathbf{w}_k \\ & - \mathbf{\Lambda}_k(\mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}}) + \mathbf{H}_{k-1}(\mathbf{x}_{\mathcal{R}_{k-1}} - \hat{\mathbf{x}}_{\mathcal{R}_{k-1}}) + \mathbf{G}_{\mathcal{F}_{k-1}}(\mathbf{y}_{\mathcal{F}_{k-1}} - \hat{\mathbf{y}}_{\mathcal{F}_{k-1}}) + \mathbf{w}_{k-1})\end{aligned}$$

Rearranging the error propagation equation given by eq. (4.9) following the approach described in [71] to avoid time-latency leads to,

$$\mathbf{x}_{\mathcal{R}_{k-1}} - \hat{\mathbf{x}}_{\mathcal{R}_{k-1}} \simeq \mathbf{F}_{k-1}^{-1}(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k}) - \mathbf{F}_{k-1}^{-1}\mathbf{v}_{k-1}$$

therefore,

$$\begin{aligned}\mathbf{r}_k \simeq\ & \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_k}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}}) \\ & + (\mathbf{H}_k - \mathbf{\Lambda}_k \mathbf{H}_{k-1}\mathbf{F}_{k-1}^{-1})(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k}) \\ & + \mathbf{\Lambda}_k \mathbf{H}_{k-1}\mathbf{F}_{k-1}^{-1}\mathbf{v}_{k-1} + \mathbf{w}_k - \mathbf{\Lambda}_k \mathbf{w}_{k-1} \\ & + \mathbf{G}_{\mathcal{F}_k}(\mathbf{y}_{\mathcal{F}_k} - \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{G}_{\mathcal{F}_{k-1}}(\mathbf{y}_{\mathcal{F}_{k-1}} - \hat{\mathbf{y}}_{\mathcal{F}_{k-1}})\end{aligned}$$

Substituting the linear relation between $\mathbf{y}_{\mathcal{F}_k}$ and $\mathbf{y}_{\mathcal{F}_{k-1}}$ given by eq. (4.19) with $\hat{\mathbf{y}}_{\mathcal{F}_k} = \mathbf{F}_{C_k}\hat{\mathbf{y}}_{\mathcal{F}_{k-1}}$ results in,

$$\begin{aligned}\mathbf{r}_k \simeq\ & \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_k}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}}) \\ & + (\mathbf{H}_k - \mathbf{\Lambda}_k \mathbf{H}_{k-1}\mathbf{F}_{k-1}^{-1})(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k}) \\ & + \mathbf{\Lambda}_k \mathbf{H}_{k-1}\mathbf{F}_{k-1}^{-1}\mathbf{v}_{k-1} + \mathbf{w}_k - \mathbf{\Lambda}_k \mathbf{w}_{k-1} + \mathbf{G}_{\mathcal{F}_k}\mathbf{n}_k \\ & + (\mathbf{G}_{\mathcal{F}_k}\mathbf{F}_{C_k} - \mathbf{\Lambda}_k \mathbf{G}_{\mathcal{F}_{k-1}})(\mathbf{y}_{\mathcal{F}_{k-1}} - \hat{\mathbf{y}}_{\mathcal{F}_{k-1}})\end{aligned}$$

which can finally be expressed as,

$$\mathbf{r}_k \simeq \mathbf{h}_k^* + \mathbf{H}_k^*(\mathbf{x}_{\mathcal{R}_k} - \hat{\mathbf{x}}_{\mathcal{R}_k}) + \mathbf{w}_{\mathbf{r}_k}$$

with,

$$\mathbf{h}_k^* = \mathbf{h}_k(\hat{\mathbf{x}}_{\mathcal{R}_k}, \hat{\mathbf{y}}_{\mathcal{F}_k}) - \mathbf{\Lambda}_k \mathbf{h}_{k-1}(\hat{\mathbf{x}}_{\mathcal{R}_{k-1}}, \hat{\mathbf{y}}_{\mathcal{F}_{k-1}})$$
$$\mathbf{H}_k^* = \mathbf{H}_k - \mathbf{\Lambda}_k \mathbf{H}_{k-1} \mathbf{F}_{k-1}^{-1}$$

and,

$$\mathbf{w}_{\mathbf{r}_k} = \mathbf{\Lambda}_k \mathbf{H}_{k-1} \mathbf{F}_{k-1}^{-1} \mathbf{v}_{k-1} + \mathbf{w}_k - \mathbf{\Lambda}_k \mathbf{w}_{k-1} + \mathbf{G}_{\mathcal{F}_k} \mathbf{n}_k$$

and, matrix $\mathbf{\Lambda}_k$ is computed such that the time-correlated components from the evolution of the measurement $\mathbf{r}_k$ are removed, thus is,

$$\mathbf{G}_{\mathcal{F}_k} \mathbf{F}_{C_k} - \mathbf{\Lambda}_k \mathbf{G}_{\mathcal{F}_{k-1}} \simeq \mathbf{0}$$

thus,

$$\mathbf{\Lambda}_k \simeq \mathbf{G}_{\mathcal{F}_k} \mathbf{F}_{C_k} \mathbf{G}_{\mathcal{F}_{k-1}}^T (\mathbf{G}_{\mathcal{F}_{k-1}} \mathbf{G}_{\mathcal{F}_{k-1}}^T)^{-1}$$

Appendix C

Derivation of the Measurement-Differencing EIF-based Equations

In this appendix we provide the algebraic derivation of the measurement-differencing EIF-based update equations from its EKF counterpart.

First, we briefly summarize the MD-EKF update equations (taken from Eqs. 4.29, 4.30 and 4.31):

- Filter gain:

$$\mathbf{K}_k = (\mathbf{P}_{k|k-1}\mathbf{H}_k^{*T} + \mathbf{C}_k)(\mathbf{H}_k^*\mathbf{P}_{k|k-1}\mathbf{H}_k^{*T} + \mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} + \mathbf{H}_k^*\mathbf{C}_k + \mathbf{C}_k^T\mathbf{H}_k^{*T})^{-1} \tag{C.1}$$

- State update:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{r}_k - \mathbf{h}_k^*) \tag{C.2}$$

- Covariance update:

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k(\mathbf{H}_k^*\mathbf{P}_{k|k-1}\mathbf{H}_k^{*T} + \mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} + \mathbf{H}_k^*\mathbf{C}_k + \mathbf{C}_k^T\mathbf{H}_k^{*T})\mathbf{K}_k^T \tag{C.3}$$

We are interested on obtaining the EIF-based update equations in terms of the information vector $\mathbf{i}_{k|k}$ and matrix $\mathbf{I}_{k|k}$ defined as,

$$\mathbf{I}_{k|k} = \mathbf{P}_{k|k}^{-1} \tag{C.4}$$

$$\mathbf{i}_{k|k} = \mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} \tag{C.5}$$

Hereinafter, we will simplify the notation to make reading easier and will omit subscripts. We make $\mathbf{P} = \mathbf{P}_{k|k-1}$ and $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{k|k-1}$ but maintain subscripts in $\mathbf{P}_{k|k}$ and $\hat{\mathbf{x}}_{k|k}$ to distinguish between the *a priori* and the *a posteriori* estimates.

We will first derive the update equation for $\mathbf{I}_{k|k}$. Inserting Eq. C.1 in C.3 we obtain these alternative expressions for $\mathbf{P}_{k|k}$:

$$\mathbf{P}_{k|k} = \mathbf{P} - (\mathbf{P}\mathbf{H}^{*T} + \mathbf{C})[(\mathbf{P}\mathbf{H}^{*T} + \mathbf{C})(\mathbf{H}^{*}\mathbf{P}\mathbf{H}^{*T} + \mathbf{P}_{\mathbf{w_r}} + \mathbf{H}^{*}\mathbf{C} + \mathbf{C}^{T}\mathbf{H}^{*T})^{-1}]^{T} \quad \text{(C.6)}$$

$$= \mathbf{P} - (\mathbf{P}\mathbf{H}^{*T} + \mathbf{C})(\mathbf{H}^{*}\mathbf{P}\mathbf{H}^{*T} + \mathbf{P}_{\mathbf{w_r}} + \mathbf{H}^{*}\mathbf{C} + \mathbf{C}^{T}\mathbf{H}^{*T})^{-1}(\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T}) \quad \text{(C.7)}$$

$$= \mathbf{P} - \mathbf{K}(\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T}) \quad \text{(C.8)}$$

Applying the following matrix inversion lemma on Eq. C.7:

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}$$

$$\mathbf{I}_{k|k} = \mathbf{P}_{k|k}^{-1} = \mathbf{P}^{-1} + \mathbf{P}^{-1}(\mathbf{P}\mathbf{H}^{*T} + \mathbf{C})\Big[(\mathbf{H}^{*}\mathbf{P}\mathbf{H}^{*T} + \mathbf{P}_{\mathbf{w_r}} + \mathbf{H}^{*}\mathbf{C} + \mathbf{C}^{T}\mathbf{H}^{*T}) - (\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T})\mathbf{P}^{-1}(\mathbf{P}\mathbf{H}^{*T} + \mathbf{C})\Big]^{-1}(\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T})\mathbf{P}^{-1} \quad \text{(C.9)}$$

$$= \mathbf{P}^{-1} + (\mathbf{H}^{*T} + \mathbf{P}^{-1}\mathbf{C})\Big[(\mathbf{H}^{*}\mathbf{P}\mathbf{H}^{*T} + \mathbf{P}_{\mathbf{w_r}} + \mathbf{H}^{*}\mathbf{C} + \mathbf{C}^{T}\mathbf{H}^{*T}) - (\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T})(\mathbf{H}^{*T} + \mathbf{P}^{-1}\mathbf{C})\Big]^{-1}(\mathbf{H}^{*} + \mathbf{C}^{T}\mathbf{P}^{-1}) \quad \text{(C.10)}$$

$$= \mathbf{P}^{-1} + (\mathbf{H}^{*T} + \mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}} - \mathbf{C}^{T}\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{H}^{*} + \mathbf{C}^{T}\mathbf{P}^{-1}) \quad \text{(C.11)}$$

Then, the measurement-differencing based information matrix update is:

$$\mathbf{I}_{k|k} = \mathbf{I}_{k|k-1} + (\mathbf{H}_{k}^{*T} + \mathbf{I}_{k|k-1}\mathbf{C}_{k})(\mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}} - \mathbf{C}_{k}^{T}\mathbf{I}_{k|k-1}\mathbf{C}_{k})^{-1}(\mathbf{H}_{k}^{*} + \mathbf{C}_{k}^{T}\mathbf{I}_{k|k-1}) \quad \text{(C.12)}$$

Next, we proceed to derive the information vector update equation. To this end, we will make use of the following expression for the Kalman gain obtained from C.8:

$$\mathbf{P}_{k|k} = \mathbf{P} - \mathbf{K}(\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T}) => \quad \text{(C.13)}$$

$$\mathbf{K} = (\mathbf{P} - \mathbf{P}_{k|k})(\mathbf{H}^{*}\mathbf{P} + \mathbf{C}^{T})^{+} \quad \text{(C.14)}$$

where $A^{+} = A^{T}(AA^{T})^{-1}$ is the Moore-Penrose generalized inverse or *pseudoinverse*.

We replace Eq. C.2 in Eq. C.5 and then Eq. C.11 on the first term.

$$\mathbf{i}_{k|k} = \mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} = \mathbf{P}_{k|k}^{-1}(\hat{\mathbf{x}} + \mathbf{K}(\mathbf{r} - \mathbf{h}^{*})) \quad \text{(C.15)}$$

$$= \mathbf{P}^{-1}\hat{\mathbf{x}} + (\mathbf{H}^{*T} + \mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}} - \mathbf{C}^{T}\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{H}^{*} + \mathbf{C}^{T}\mathbf{P}^{-1})\hat{\mathbf{x}} + \mathbf{P}_{k|k}^{-1}\mathbf{K}(\mathbf{r} - \mathbf{h}^{*}) \quad \text{(C.16)}$$

We develop the second term using C.11 and C.14

$$\mathbf{P}_{k|k}^{-1}\mathbf{K} = \mathbf{P}_{k|k}^{-1}(\mathbf{P}-\mathbf{P}_{k|k})(\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ \tag{C.17}$$

$$= \mathbf{P}_{k|k}^{-1}\mathbf{P}(\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ - (\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ \tag{C.18}$$

$$= \left(\mathbf{P}^{-1}+(\mathbf{H}^{*T}+\mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{H}^*+\mathbf{C}^T\mathbf{P}^{-1})\right)\mathbf{P}(\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ - (\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ \tag{C.19}$$

$$= (\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ + (\mathbf{H}^{*T}+\mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)(\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ - (\mathbf{H}^*\mathbf{P}+\mathbf{C}^T)^+ \tag{C.20}$$

$$= (\mathbf{H}^{*T}+\mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C})^{-1} \tag{C.21}$$

Inserting this result in Eq. C.16:

$$\mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} = \mathbf{P}^{-1}\hat{\mathbf{x}} + (\mathbf{H}^{*T}+\mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{H}^*+\mathbf{C}^T\mathbf{P}^{-1})\hat{\mathbf{x}} + (\mathbf{H}^{*T}+\mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{r}-\mathbf{h}^*) \tag{C.22}$$

$$= \mathbf{P}^{-1}\hat{\mathbf{x}} + (\mathbf{H}^{*T}+\mathbf{P}^{-1}\mathbf{C})(\mathbf{P}_{\mathbf{w_r}}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C})^{-1}(\mathbf{r}-\mathbf{h}^*+\mathbf{H}^*\hat{\mathbf{x}}+\mathbf{C}^T\mathbf{P}^{-1}\hat{\mathbf{x}}) \tag{C.23}$$

Finally, we express this result in terms of the information matrix and vector:

$$\mathbf{i}_{k|k} = \mathbf{i}_{k|k-1} + (\mathbf{H}_k^{*T}+\mathbf{I}_{k|k-1}\mathbf{C}_k)(\mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}}-\mathbf{C}_k^T\mathbf{I}_{k|k-1}\mathbf{C}_k)^{-1}(\mathbf{r}_k-\mathbf{h}_k^*+\mathbf{H}_k^*\hat{\mathbf{x}}_{k|k-1}+\mathbf{C}_k^T\mathbf{i}_{k|k-1}) \tag{C.24}$$

In summary, the measurement-differencing EIF-based update equations are:

$$\mathbf{I}_{k|k} = \mathbf{I}_{k|k-1} + (\mathbf{H}_k^{*T}+\mathbf{I}_{k|k-1}\mathbf{C}_k)(\mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}}-\mathbf{C}_k^T\mathbf{I}_{k|k-1}\mathbf{C}_k)^{-1}(\mathbf{H}_k^*+\mathbf{C}_k^T\mathbf{I}_{k|k-1}) \tag{C.25}$$

$$\mathbf{i}_{k|k} = \mathbf{i}_{k|k-1} + (\mathbf{H}_k^{*T}+\mathbf{I}_{k|k-1}\mathbf{C}_k)(\mathbf{P}_{\mathbf{w}_{\mathbf{r}_k}}-\mathbf{C}_k^T\mathbf{I}_{k|k-1}\mathbf{C}_k)^{-1}(\mathbf{r}_k-\mathbf{h}_k^*+\mathbf{H}_k^*\hat{\mathbf{x}}_{k|k-1}+\mathbf{C}_k^T\mathbf{i}_{k|k-1}) \tag{C.26}$$

Notice how, when there are no time-correlated measurements, $\mathbf{C}_k = \mathbf{0}$ and previous equations reduce to the standard EIF update equations shown in section 2.4.3.

Bibliography

[1] L.A.A. Andersson and J. Nygards. C-SAM: Multi-robot SLAM using square root information smoothing. In *IEEE International Conference on Robotics and Automation*, pages 2798–2805, Pasadena, CA, USA, May 19-23 2008. (Cited on page 100.)

[2] J.R. Asensio and L. Montano. A kinematic and dynamic model-based motion controller for mobile robots. In *The 15th IFAC Triennial World Congress*, Barcelona, Spain, July 21-26 2002. (Cited on page 35.)

[3] A. Bahr, M.R. Walter, and J.J. Leonard. Consistent cooperative localization. In *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17 2009. (Cited on page 45.)

[4] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte. Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2859–2865, May 9-13 2011. (Cited on page 80.)

[5] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, Oct. 2006. (Cited on pages 45 and 54.)

[6] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley and sons, May 2001. (Cited on pages 20, 45, 46, 49, 51, 52, and 73.)

[7] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, second edition, 2006. (Cited on pages 10, 12, 80, and 83.)

[8] L. Blackmore, H. Li, and B. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *American Control Conference*, 2006. (Cited on page 60.)

[9] J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25 –30, 1965. (Cited on page 61.)

[10] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley and sons, New York, 1992. (Cited on page 52.)

[11] A.E. Bryson and D.E. Johansen. Linear filtering for time-varying systems using measurements containing colored noise. *IEEE Transactions on Automatic Control*, Jan. 1968. (Cited on pages 45 and 51.)

[12] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *National Conference on Artificial Intelligence (AAAI)*, 1996. (Cited on page 44.)

[13] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J.D. Tardós. A comparison of SLAM algorithms based on a graph of relations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 2009. (Cited on page 112.)

[14] L. Carlone, M.K. Ng, J. Du, B. Bona, and M. Indri. Rao-blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication. In *IEEE International Conference on Robotics and Automation*, pages 243–249, Anchorage, Alaska, May 3-8 2010. (Cited on page 81.)

[15] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardós, and J. Neira. Robocentric map joining: Improving the consistency of ekf-slam. *Robotics and Autonomous Systems*, 55(1):21 – 29, 2007. (Cited on page 46.)

[16] J.A. Castellanos, J. Neira, and J.D. Tardós. Limits to the consistency of EKF-based SLAM. In *5th IFAC Symp. on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004. (Cited on page 45.)

[17] J.A. Castellanos, J.D. Tardós, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *IEEE International Conference on Robotics and Automation*, pages 1053–1059, Albuquerque, New Mexico, USA, Apr. 20-25 1997. (Cited on page 15.)

[18] A. Censi, D. Calisi, A. De Luca, and G. Oriolo. A bayesian framework for optimal motion planning with uncertainty. In *IEEE International Conference on Robotics and Automation*, pages 1798–1805, May 2008. (Cited on page 60.)

[19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, Wiley, New York, 1991. (Cited on page 73.)

[20] A. Cunningham, M. Paluri, and F. Dellaert. DDF-SAM: Fully distributed SLAM using constrained factor graphs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3025–3030, Taipei, Taiwan, Oct. 18-22 2010. (Cited on page 100.)

[21] A. Cunningham, K.M. Wurm, W. Burgard, and F. Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *IEEE International Conference on Robotics and Automation*, pages 1093–1100, St. Paul, Minnesota, USA, May 14-18 2012. (Cited on page 100.)

[22] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, and C.J. Taylor. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825, Oct. 2002. (Cited on page 34.)

[23] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, Dec. 2006. (Cited on page 10.)

[24] F. Dellaert, A. Kipp, and P. Krauthausen. A multifrontal QR factorization approach to distributed inference applied to multi-robot localization and mapping. In *AAAI National Conference on Artificial Intelligence*, 2005. (Cited on page 99.)

[25] C. Estrada, J. Neira, and J.D. Tardós. Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, Aug. 2005. (Cited on page 15.)

[26] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, Dec. 2006. (Cited on page 86.)

[27] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 2000. (Cited on page 45.)

[28] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998. (Cited on page 44.)

[29] B. Gerkey, R.T. Vaughan, and A. Howard. The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proc. of the 11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, June 2003. (Cited on page 54.)

[30] S. Grime and H. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849–863, 1994. (Cited on page 80.)

[31] G. Grisetti, R. Kümmerle, and K. Ni. Robust optimization of factor graphs by using condensed measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 581–588, Vilamoura, Portugal, Oct. 7-12 2012. (Cited on pages 27, 98, 100, and 102.)

[32] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, pages 31–43, 2010. (Cited on page 24.)

[33] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2d and 3d mapping. In *IEEE International Conference on Robotics and Automation*, pages 273–278, May 2010. (Cited on page 26.)

[34] N. Gulec and M. Unel. A novel coordination scheme applied to nonholonomic mobile robots. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 5089–5094, Dec 2005. (Cited on page 34.)

[35] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968. (Cited on page 60.)

[36] C. Hertzberg. A framework for sparse, non-linear least squares problems on manifolds. Master's thesis, Universität Bremen, 2008. (Cited on page 26.)

[37] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013. (Cited on page 26.)

[38] G.P. Huang, N. Trawny, A.I. Mourikis, and S.I. Roumeliotis. On the consistency of multi-robot cooperative localization. In *Robotics Science and Systems*, Seattle, Washington, 2009. (Cited on page 45.)

[39] S. Huang and G. Dissanayake. Convergence and consistency analysis for extended kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5):1036–1049, Oct. 2007. (Cited on page 45.)

[40] S. Huang, Z. Wang, and G. Dissanayake. Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24(5):1121–1130, Oct 2008. (Cited on page 15.)

[41] S.J. Julier. The scaled unscented transformation. In *American Control Conference*, volume 6, pages 4555 – 4559, 2002. (Cited on page 29.)

[42] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, Dec. 2008. (Cited on pages 24 and 101.)

[43] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller. Multiple relative pose graphs for robust cooperative mapping. In *IEEE International Conference on Robotics and Automation*, pages 3185–3192, Anchorage, Alaska, May 3-8 2010. (Cited on page 100.)

[44] S. Koenig and M. Likhachev. d^* lite. In *AAAI National Conference on Artificial Intelligence*, pages 476–483, 2002. (Cited on page 60.)

[45] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. (Cited on page 10.)

[46] F.R. Kschischang, B.J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, Feb. 2001. (Cited on page 13.)

[47] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*, pages 3607–3613, Shangai, China, May 9-13 2011. (Cited on pages 24 and 101.)

[48] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation*, pages 1250–1257, Los Alamitos, CA, USA, 1994. (Cited on page 44.)

[49] M.T. Lázaro and J.A. Castellanos. Localization of probabilistic robot formations in SLAM. In *IEEE International Conference on Robotics and Automation*, pages 3179–3184, Anchorage, Alaska, May 3-8 2010. (Cited on page 5.)

[50] M.T. Lázaro, L.M. Paz, P. Piniés, and J.A. Castellanos. Distributed localization and submapping for robot formations using a prior map. In *The 2013 IFAC Intelligent Autonomous Vehicles Symposium*, Gold Coast, Australia, June 26-28 2013. (Cited on page 6.)

[51] M.T. Lázaro, L.M. Paz, P. Piniés, J.A. Castellanos, and G. Grisetti. Multi-robot SLAM using condensed measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo Big Sight, Japan, Nov. 3-8 2013. (Cited on page 6.)

[52] M.T. Lázaro, P. Urcola, L. Montano, and J.A. Castellanos. Position tracking and path planning in uncertain maps for robot formations. In *The 2nd IFAC Workshop on Multivehicle Systems*, pages 7–12, Espoo, Finland, Oct. 3-4 2012. (Cited on page 5.)

[53] J.M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. 2003. (Cited on page 26.)

[54] K.Y.K. Leung, T.D. Barfoot, and H.H.T. Liu. Distributed and decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks. In *IEEE International Conference on Robotics and Automation*, pages 3841–3847, Shangai, China, May 9-13 2011. (Cited on page 81.)

[55] M.A. Lewis and K.-H. Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4(4):387–403, Oct. 1997. (Cited on page 34.)

[56] E.Z. MacArthur and C.D. Crane. Compliant formation control of a multi-vehicle system. In *International Symposium on Computational Intelligence in Robotics and Automation*, pages 479–484, June 2007. (Cited on page 34.)

[57] J. Minguez and L. Montano. Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, Feb. 2004. (Cited on page 39.)

[58] P.E. Missiuro and N. Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *IEEE International Conference on Robotics and Automation*, pages 1261–1267, May 2006. (Cited on page 59.)

[59] H. Moravec and A. E. Elfes. High resolution maps from wide angle sonar. In *ICRA*, pages 116 – 121, March 1985. (Cited on page 16.)

[60] A.I. Mourikis, S.I. Roumeliotis, and J.W. Burdick. SC-KF mobile robot localization: A stochastic cloning-kalman filter for processing relative-state measurements. *IEEE Transactions on Robotics*, 23(4):717–730, Aug. 2007. (Cited on page 45.)

[61] A.G.O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, Inc., Boca Raton, FL, USA, 1998. (Cited on page 21.)

[62] J. Neira, J.D. Tardós, and J.A. Castellanos. Linear time vehicle relocation in SLAM. In *IEEE International Conference on Robotics and Automation*, pages 427–433, Taipei, Taiwan, 2003. (Cited on pages 44 and 62.)

[63] E.D. Nerurkar, S.I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE International Conference on Robotics and Automation*, pages 1402–1409, Kobe, Japan, May 12-17 2009. (Cited on page 80.)

[64] E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh. Decentralised SLAM with low-bandwidth communication for teams of vehicles. In *International Conference on Field and Service Robotics*, pages 179–188, Lake Yamanaka, Japan, July 14-16 2003. (Cited on page 81.)

[65] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013. (Cited on page 112.)

[66] E. Olson, M. Walter, J. Leonard, and S. Teller. Single cluster graph partitioning for robotics applications. In *RSS*, pages 265–272, 2005. (Cited on page 108.)

[67] E.B. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, 2008. (Cited on page 111.)

[68] L.M. Paz and J. Neira. Optimal local map size for EKF-based SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 2006. (Cited on page 89.)

[69] L.M. Paz, P. Piniés, J. Neira, and J.D. Tardós. Global localization in SLAM in bilinear time. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 655–661, Edmonton, Canada, 2005. (Cited on pages 44, 56, and 62.)

[70] L.M. Paz, P. Piniés, J.D. Tardós, and J. Neira. Large scale 6DOF SLAM with a stereo camera in hand. *IEEE Transactions on Robotics*, 24(5), Oct. 2008. (Cited on page 15.)

[71] M.G. Petovello, K. O'Keefe, G. Lachapelle, and M.E. Cannon. Consideration of time-correlated errors in a kalman filter applicable to GNSS. *Journal of Geodesy*, 83:51–56, 2009. (Cited on pages 45, 51, 52, and 131.)

[72] P. Piniés and J.D. Tardós. Large scale SLAM building conditionally independent local maps: Application to monocular vision. *IEEE Transactions on Robotics*, 24(5):1094–1106, Oct. 2008. (Cited on pages 80 and 90.)

[73] I.M. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *The 15th International Joint Conference on Artificial Intelligence*, pages 1340–1345, Nagoya, Japan, Aug. 23-29 1997. (Cited on page 44.)

[74] C. Rizzo, D. Tardioli, D. Sicignano, L. Riazuelo, J.L. Villarroel, and L. Montano. Signal based deployment planning for robot teams in tunnel-like fading environments. *The International Journal of Robotics Research*, 32(12):1381–1397, Oct. 2013. (Cited on page 33.)

[75] S.I. Roumeliotis and G.A. Bekey. Collective localization: A distributed kalman filter approach to localization of groups of mobile robots. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, Apr. 2000. (Cited on page 44.)

[76] S.I. Roumeliotis and G.A. Bekey. Distributed multi-robot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, Oct. 2002. (Cited on page 80.)

[77] A.C. Sanderson. A distributed algorithm for cooperative navigation among multiple mobile robots. *Advanced Robotics*, 12(4):335–349, 1998. (Cited on page 44.)

[78] A.C. Schultz and W. Adams. Continuous localization using evidence grids. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998. (Cited on page 44.)

[79] D. Tardioli and J.L. Villarroel. Real time communications over 802.11: RT-WMP. In *IEEE International Conference on Mobile Adhoc and Sensor Systems MASS*, pages 1–11, Pisa, Italy, 2007. (Cited on page 73.)

[80] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998. (Cited on page 44.)

[81] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. (Cited on page 10.)

[82] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 2001. (Cited on page 44.)

[83] P. Urcola and L. Montano. Cooperative robot team navigation strategies based on an environment model. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4577–4583, Oct. 2009. (Cited on pages 39, 67, and 68.)

[84] P. Urcola and L. Montano. Adapting robot team behavior from interaction with a group of people. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2887–2894, San Francisco, CA, USA, Sep. 25-30 2011. (Cited on page 33.)

[85] P. Urcola, L. Riazuelo, M.T. Lázaro, and L. Montano. Cooperative navigation using environment compliant robot formations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2789–2794, Nice, France, Sep. 22-26 2008. (Cited on pages 5, 34, 39, and 56.)

[86] J. van den Berg, P. Abbeel, and K. Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, 2011. (Cited on page 60.)

[87] R. Vidal, O. Shakernia, and S. Sastry. Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 584–589, Sep. 2003. (Cited on page 34.)

[88] M.R. Walter, R.M. Eustice, and J.J. Leonard. Exactly sparse extended information filters for feature-based SLAM. *The International Journal of Robotics Research*, 26(4):335–359, Apr. 2007. (Cited on page 83.)

[89] S.B. Williams, G. Dissanayake, and H. Durrant-whyte. An efficient approach to the simultaneous localisation and mapping problem. In *IEEE International Conference on Robotics and Automation*, pages 406–411, 2002. (Cited on page 15.)