SAPIENZA
UNIVERSITÀ DI ROMA

PhD Thesis

Mutual localization from anonymous measurements in multi-robot systems

Dipartimento di Ingegneria Informatica,
Automatica e Gestionale

PhD in System Engineering
ING/INF 04
XXIV Ciclo

Paolo Stegagno

Advisor
Prof. Giuseppe Oriolo

April 2012

PhD in System Engineering
ING/INF 04
XXIV Ciclo (2008-2012)

Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Via Ariosto 25, Rome (RM), 00185, Italy

Università di Roma La Sapienza
Piazzale Aldo Moro, 5, Rome (RM), 00185, Italy

Author: Paolo Stegagno
Advisor: Prof. Giuseppe Oriolo
PhD Coordinator: Prof. Carlo Bruni, Prof. Salvatore Monaco

Mutual localization from anonymous measurements in multi-robot systems
PhD Thesis
Printed in Italy, April 2012

Contact information

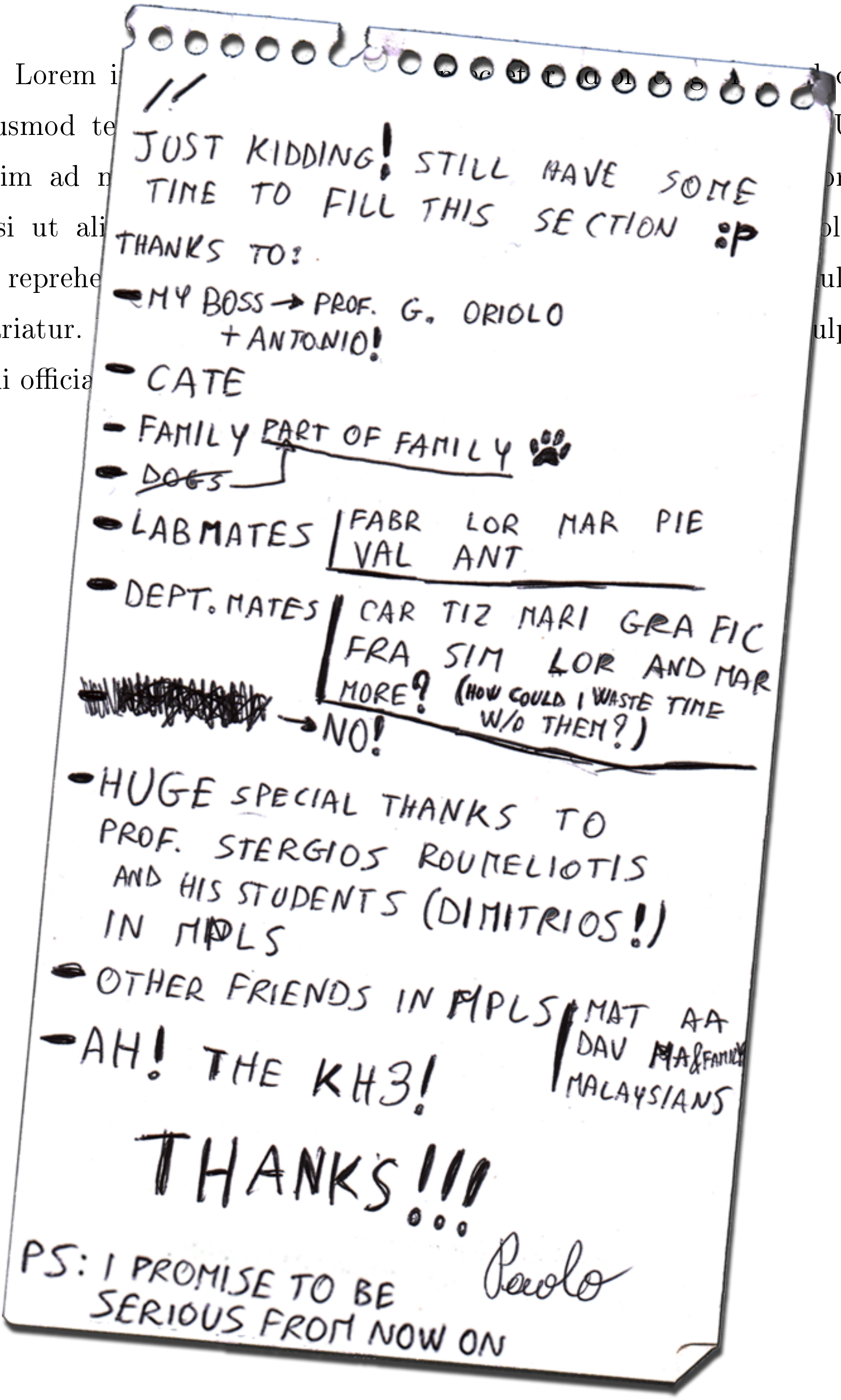
www: http://www.dis.uniroma1.it/∼stegagno/
e-mail: paolostegagno@gmail.com
stegagno@dis.uniroma1.it

Acknowledgments

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgments

Lorem i do eiusmod te Ut enim ad n oris nisi ut ali olor in reprehe ulla pariatur. ulpa qui officia



//

JUST KIDDING! STILL HAVE SOME TIME TO FILL THIS SECTION :P

THANKS TO:

- MY BOSS → PROF. G. ORIOLO
  + ANTONIO!
- CATE
- FAMILY PART OF FAMILY 🐾
- ~~DOGS~~
- LABMATES | FABR LOR MAR PIE VAL ANT
- DEPT. MATES | CAR TIZ MARI GRA FIC FRA SIM LOR AND MAR MORE? (HOW COULD I WASTE TIME W/O THEM?)
- [redacted] → NO!
- HUGE SPECIAL THANKS TO PROF. STERGIOS ROUMELIOTIS AND HIS STUDENTS (DIMITRIOS!) IN MPLS
- OTHER FRIENDS IN MPLS | MAT AA DAV MA&FAMILY MALAYSIANS
- AH! THE KH3!

THANKS!!!

PS: I PROMISE TO BE SERIOUS FROM NOW ON

Paolo

Citation!? What...?

P. Stegagno, *Breaking promises*, 29 Nov 2011

Contents

Introduction

Recently, the problem of controlling multi-agent (in particular, multi-robot) systems has attracted increasing attention in view of their pervasive application potential, increased performances and robustness with respect to a single-robot solution. However, their application usually requires a good knowledge of the mutual position and orientation of the components of the system.

A great number of techniques have been developed to achieve this result, mainly based on recursive filters, and most of them assume the knowledge of the identity of the measured robots. A still open problem is the data association between measurements and current estimates, i.e., assuming that at a given time a robot has an estimate on the pose of each of its mates and some measurements, the problem of associating each measurement to the originating robot.

This problem arises when the robots are equipped with sensors unable to discriminate among different robots, such as range finders, or in adverse environmental conditions. Its impact is reflected in all aspects concerning multi-robot localization, from the formulation of the problem to its solution. For example, the presence of false positives measurements (measurements of obstacles mistaken for robots) is allowed only by this assumptions. Moreover, we will see how the static problem of reconstructing the state of a multi-robot system from the measurements gathered from all its components admits in some particular configurations more than one solution due to the anonymity of the measurements. Last, the filtering itself using the odometry measurements of each robot cannot be performed without reconstructing the identity of the robots. In fact, the knowledge of the identity of the robot sending a given odometry would be useless without the knowledge of the identities of the current estimates.

This work will focus mainly on a 2D scenario and a team of differential drive robots modeled as unicycles with the ability of sensing each other's positions. Since we want to design methods suitable for real world implementation, we will assume the presence of false positive and negatives measurements and limited field of view of the sensors. Despite the system being decentralized, we will not discuss synchronization issues, assuming that the robots move slow enough to avoid the introduction of significant error during the delay times.

We will explore a number of different possibilities to solve the data association problem, from the more classical ones such as maximum likelihood criterion, to more sophisticated systems based on geometrical considerations or multi-tracking tools. An extensive experimentation will highlight the pros and cons of each method, as well as some extensions of the proposed methods dealing with different types of measurements or models will give rise to interesting considerations.

The rest of this work is organized as follows.

Part I: we provide a literature review on multi-robot localization.

Part II: in chapter 2 we formally state the problem addressed in this work; in chapter 3 we show the motivation that led us to face it; in chapter 4 we explore the consequences of anonymity; in chapter 5 we expose a solution coming from literature based on FastSLAM. Some of the ideas contained in this Part have been published in [1].

Part III: we propose a localization system based on a geometric multiple registration algorithm that we have developed. This part has been published in [2, 3, 4], and the developed system has been used in [5].

Part IV: in chapter 8 we provide a description of our experimental platform; in chapter 9 we compare the results of the developed method with the one based on FastSLAM. The ideas contained in Part II, Part III and Part IV have been collected in [6].

Part V: in chapter 10 we show an extension of the method presented in Part III using bearing-only measurements, published in [7]; in chapter 11 we show a 3-D extension of the bearing-only method, considering both bearing-only and bearing+distance measurements, published in [8, 9]; in chapter 12 we present a control law developed in [1] to drive the robots out from symmetric (hence ambiguous, as we will see later) configurations; in chapter 13 we present a different approach coming from multi-target tracking theory, that we will develop in the future; in chapter 14 we draw some conclusions.

Some videoclips of the experiments presented throughout this work are available online on the webpage

http://www.dis.uniroma1.it/∼labrob/

and youtube channel

http://www.youtube.com/RoboticsLabSapienza

of the Robotics Laboratory of the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Università di Roma La Sapienza. In particular, the reader may refer to

http://www.dis.uniroma1.it/∼labrob/research/mutLoc.html

for the videclips concerning the mutual localization algorithms, and to

http://www.dis.uniroma1.it/∼labrob/research/encirclement.html

for one of its applications.

part I

Introduction to multi-robot localization

chapter 1

State of the art

In this chapter we briefly recall the scientific research produced on multi-robot localization until now. Many authors have addressed this problem because it represents a prerequisite for all the tasks that implies *data fusion* of the sensory perceptions of the components of the system. Among the most studied topics, one may mention exploration [10], coverage [11], cooperative transportation [12], consensus [13], flocking [14, 15], formation control [16], connectivity maintenance [17, 18], pursuit-evasion [19], distributed estimation and sensing [20, 21]. The quality of the execution of the task depends on the accuracy of the estimates of the change of coordinates among the robots available to each robot.

In literature, we can identify three main problems related with multi-robot localization, depending on the assumptions taken by the authors. The localization of the components of a multi-robot team in a fixed frame common to all the robots is usually referred as *cooperative localization*. However, agreeing on a common fixed frame already implies a form of centralization. A more decentralized approach to this problem can consider either a moving frame attached to each agent, or a different fixed frame for each agent. We define relative mutual localization (RML) as the problem of estimating the relative poses among the moving frames attached to the agents, and absolute mutual localization (AML) as the problem of estimating the relative poses among the various fixed frames. RML and AML are equivalent if each agent is localized with respect to its own fixed frame. We will refer as *mutual localization* (ML) the class of problems including RML and AML.

All these problems have been studied deeply during the years, and a great number of authors have proposed a plethora of different approach under a huge number of different conditions. We will focus this work on ML, and more specifically on RML. However, the importance of the ideas originated from the study of CL and its close relationship to ML induce us to include it in this literature review. In the following, we will use the term position to indicate the pair of coordinates that identifies a representative point of a robot, while pose will indicate the position plus the orientation.

In a centralized system, robot localization can be addressed using external systems, such as GPS or fixed cameras. An example of this can be found in [22], in which the authors propose a method for the escorting of an autonomous agent. An external camera system provides the absolute position of the robots that is used to compute the control law. This approach is applicable only in known and structured environments and requires additional hardware. Using a GPS, the main limitation is the necessity of keeping the line-of-sight with the satellites, confining its application to open air environments. To overcome those limitations, the scientific community started to deal with the problem of estimating the pose of the robots using only their own sensory perceptions.

The first works in this field considered only single robots systems. Knowing that the exclusive use of proprioceptive sensors (such as encoders and IMUs) brings the localization error to grow with the traveled distance, some authors started to consider the idea to improve the localization by the use of exteroceptive sensors providing a representation of the environment. One of the first proposed technique suggest to identify some landmarks in the environments, such as walls or pillars, to allow the robot to compute its own pose relative to them. The knowledge of the pose of the landmarks in an absolute frame allows the computation of the absolute pose of the robot.

1.1 Cooperative positioning

The first work explicitly in the field of multi-robot localization can be considered [23]. The authors point out that the landmark method requires the knowledge of the disposition of the landmarks in the environment. To avoid this limitation, they propose the use of the same robot team members as landmarks. They divide the team in two subgroups A e B. In the beginning, the subgroup A stands still in a known configuration, while the subgroup B moves. When the members of B stop moving, they can estimate with a good precision their localization relative to the components of A. Then it is time for A to move and the robots repeat the above steps exchanging the roles of A and B. This approach is known in literature as *cooperative positioning*, and can be considered as the ancestor of CL.

The authors implement and extend this method in [24] and [25], in which they present a working system with three cooperating robots, with only one of them able to gather relative measurements, and propose different moving strategies to improve the accuracy of the localization.

A similar independent work is presented in [26] in which the localization of an heterogeneous team of robot for the exploration of unknown environment is addressed by allowing each robot to localize using distance measurements of other still robots. The authors develop also an ad hoc protocol to compute the starting pose of each robot, that is required to initialize the cooperative positioning algorithm. However, despite the high accuracy of the estimates of the poses of the robots, the method has some drawbacks. First, the robots must always keep line-of-sight among them. Second, the time spent standing still slows down the system. This is enough to nullify almost all the advantages inherent the use of a multi-robot team instead of a single robot.

1.2 A geometric algorithm

All the aforementioned papers take the assumption of known data association among measurements and robots. In [27], the authors propose an algorithm based on triangle recognition for the solution of the static RML problem with bearing measurements with unknown data association. The measurements are obtained processing the output of the omnidirectional cameras mounted atop each robot. The algorithm performs the following steps:

i) compute the differences among the measurements to obtain the angles between two following robots

ii) look for triplets of robots containing triplets of angles whose sum is 180°; each of these triplets is a possible triangle

iii) find out all the common edges between the triangles

iv) for each pair of triangles with a common edge, check if other triangles with common edges exist

v) if so, match the triangles and use them to compute the directions of the other robots.

This paper represents one of the first examples of algorithm based on geometrical considerations for the solution of RML. We will present here other methods, however this will be the only one assuming unknown data association. The use of omnidirectional sensors, however, limits the class of the feasible sensors to very few possibilities. Moreover, the algorithm is not designed to include outliers, that is, false measurements of object recognized as robots.

1.3 Filter based approaches

In [28] and [29] a MonteCarlo localization method (estimate of the pose of a robot in a known environment with landmarks using a particle filter) is applied to multi-robot. Two robots sharing a portion of the map of the environment are then able to perform CL. The use of relative pose measurements can further improve the estimates. Those measurements are obtained through a camera and the recognition of codified markers specific for each robot, so that the measurements include relative bearing, relative orientation and identity of the measured robot. The distance is retrieved through the use of a range finder such as a laser scanner.

The same measurement model is assumed in [30], in which the authors derive the equations of an extended Kalman filter for the estimation of the configuration vector, that is, the pose of all the robots in a common fixed frame. The system update is performed using odometry measurements, while the pose measurements are used for the measurement update. The authors are allowed to assume pairwise encounters of the robots, since a multiple encounter is decomposed in a certain number of pairwise encounters. The covariance matrix P of the filter is $(N \times M) \times (N \times M)$, with N number of variables describing the state of a robot and M number of robots. The matrix P is decomposed in $M \times M$ square blocks P_{mn} of dimension $N \times N$, and the measurement update derived from the encounter of robots i and j

involves only the blocks P_{ii}, P_{ij}, P_{ji} and P_{jj}. Eventually, the authors propose a distribution of the derived filter based on a factorization of the blocks of P. Each block P_{ij} can be decomposed in two matrices P_i and P_j for which the time update can be performed using only the local odometry measurements of robots i and j respectively. When an encounter happens, the two robots exchange their covariance matrices and are able to compute the blocks needed during the measurement update. The overall result is a drastic reduction of the computation time required during the time update.

The extended Kalman filter presented in [30] is generalized in [31] to use more simple measurements, as relative distance, bearing and orientation only. However, despite the simplification introduced in the sensor model, the authors still assume the identification of the measured robots. The same authors considered in [32] the problem of simultaneously localizing two mobile robots able to perform relative observations among them and equipped with proprioceptive sensors like encoders, producing an observability analysis based on the observability rank condition introduced in [33] for nonlinear systems. This analysis considers four different relative observations, showing the relative bearing as the best observation between the robots. Indeed, the part of the system which is observable is in general larger than for the other relative observations (relative distance and relative orientation). In [34], the authors examine more thoroughly the case of distance measurements.

Another example of filter based approach is [35]. The authors propose the estimate of the relative pose through the relative pose measurements gathered by the other robots. Each robot of a team is able to measure the relative pose and the identity of its neighbors and the displacement in its own pose in two consecutive time instants. The algorithm is developed from the point of view of R_o, while R_*, $* \in \{i, j\}$ generically denotes any other robot of the team. R_o estimates the probability density $p(x_i)$ of the pose of the other robots expressed in its own attached frame through a particle filter. The measurement update of the filter consider 5 different types of measurements:

i) m_i, measurement of the displacement of R_i;

ii) m_o, measurement of the displacement of R_o;

iii) r_{io}, measurement gathered by R_o of the pose of R_i;

iv) r_{oi}, measurement gathered by R_i of the pose of R_o;

v) r_{ji}, measurement gathered by R_i of the pose of R_j.

A measurement of type i produces a translation and a 'blur' (because of the growth of the uncertainty) of $p(x_i)$, while a measurement of type ii) produces a translation and a 'blur' in all the $p(x_i)$. If a measurement of type iii or iv occurs, the Bayesian update of the filter reduces the uncertainty. Since R_o is supposed to be always in $(0, 0, 0)$, the measurement of type iv is inverted to obtain an indirect measurement of the pose of R_i.

The last type of measurement v is the more tricky, since its straight use can cause dependencies in the measurements. The authors identify circular reasonings that cause the implicit reuse of already used measurements. This would jeopardize the estimate causing an excessive convergence rate whose result is a very precise but not accurate estimate. To avoid this situation, they propose a *dependency tree* for the $n - 1$ probability distributions. Each distribution has exactly one *parent* distribution and a certain number (possibly zero) son distribution. A distribution must not be used to update one of its ancestors.

This solution reduces the number of circular reasonings, but does not eliminate them completely. Moreover, a lot of measurements of type v must be discarded. The alternative to the dependency tree is to estimate the joint probability densities $p(x_i, x_j)$, but the result would be too computationally demanding requiring $(n-1)^2$ particle filters.

The technological development of the latest years pushes robotics to an always wider application of aerial vehicles. Thus, the most recent research has started to focus its attention to the 3-dimensional case. In this contest, some authors have studied the solvability of the 3-D RML problem, investigating the minimal sets of data needed to determine the robot-to-robot 3-D relative pose [36] and proposing observability analysis and estimators [37, 38].

1.4 Theory on rigidity of the formation

In this work we will develop a theory for the inversion of the measurement function in case of anonymous measurements that, for the best of our knowledge, represents a completely new research topic. Many authors have addressed the measurement function inversion under a great number of different assumptions, but not considering unknown data association. In particular, a whole collection of papers [39, 40, 41, 42, 43] have theoretically investigated the study of the rigidity of a formation of autonomous vehicles in which some relative distances (links) must be constant.

In [39] the authors illustrate two well known rigidity conditions based on the rank of the rigidity matrix. Their main achievement is the development of an algorithm to build generically rigid formations. In the following papers, the authors expand and apply this work. In [40] they solve the problem of the rank closure, intended as the problem of regaining rigidity through the addition of new links to a rigid formation that suddenly loses one of its component. In [41] the authors introduce the use of angular information between the robots and in [42] e [43] they propose beacon localization as an application of their rigidity theory. In particular, the authors are interested in the required minimum set of measurements to obtain a unique solution to the localization problem.

The last papers point out the importance of this theory in our work. The interpretation of the links as measurements and not as constraints establishes a common ground with our work. We will see how the loss of information due to the anonymity of the measurements will result in a 'loss of rigidity', intended in a wider sense as the loss of unique solvability of the problem.

part II

The problem of anonymous measurements

chapter 2

Definition

In this chapter we formally state the problem of mutual localization with anonymous position measurements, making clear all the assumptions that will accompany us throughout this work. Consider a system of $n \geq 2$ agents (henceforth called *robots*) $\mathcal{A}_1, \ldots, \mathcal{A}_n$, where n is unknown and may change during the operation. Denote by $\mathcal{N} = \{1, \ldots, n\}$ the robot index set, and let $\mathcal{N}_i = \mathcal{N}/\{i\}$. The robots move in the plane and a moving frame $\mathcal{F}_i$ is rigidly attached to each $\mathcal{A}_i$ (see Fig. 1a). The superscripts t and $1:t$ denote the value of a variable at the discrete time instant t and the set of all its values at time instants $1, 2, \ldots, t$, respectively. For ease of notation, we use n instead of n^t even if the number of robots may vary during operations. We will denote with u_i^t the spatial displacement of robot i between time $t-1$ and t.

We will define the localization problem from the point of view of a generic $\mathcal{A}_i$, as in [44] and [45]. The 3-vector describing the position $z_j = (z_{j1}, z_{j2}) \in \mathbb{R}^2$ and orientation $\theta_j \in \mathbb{S}^1$ of $\mathcal{F}_j$ w.r.t. $\mathcal{F}_i$ is the *relative pose* $x_j \in SE(2)$ of $\mathcal{A}_j$, $j \in \mathcal{N}_i$. Let $R(\phi) \in SO(2)$ denote the rotation matrix associated to an angle ϕ. Considering two poses $x_a = (z_a, \theta_a), x_b = (z_b, \theta_b)$, we denote by $x_a \oplus x_b$ and $x_a \ominus x_b$, respectively, the composition and the inverse composition of two poses, defined by the following formulas [46]:

$$
\begin{aligned}
x_a \oplus x_b &= (z_a + R(\theta_a) z_b, \theta_a + \theta_b) \\
x_a \ominus x_b &= (R(-\theta_b)(z_a - z_b), \theta_a - \theta_b).
\end{aligned}
$$

Discarding the last angular component (the rotation), operators $\oplus$ and $\ominus$ are also used to compose two-dimensional position vectors with three-dimensional poses, and the result is again a position. In particular, given the coordinates z of a point expressed in $\mathcal{F}_i$, whose pose w.r.t. $\mathcal{F}$ is x_i, the operation $x_i \oplus z$ gives the coordinates of the same point expressed in $\mathcal{F}$. Conversely, given x_i and the coordinates z of a point expressed in $\mathcal{F}$, the operation $z \ominus x_i$ gives the coordinates of the same point expressed in $\mathcal{F}_i$, whose pose w.r.t. $\mathcal{F}$ is x_i.

$$
\begin{aligned}
x_a \oplus z_b &= (z_a + R(\theta_a) z_b) \\
z_a \ominus x_b &= (R(-\theta_b)(z_a - z_b)).
\end{aligned}
$$

These operators may also be used with a set Z of points, by letting $x_i \oplus Z := \{x_i \oplus z \mid z \in Z\}$, and $Z \ominus x_i := \{z \ominus x_i \mid z \in Z\}$.

Each robot $\mathcal{A}_k$, $k \in \mathcal{N}$, is equipped with a *motion detector* that provides $\bar{u}_k^t$, a noisy measurement of its displacement between $t-1$ and t. The motion detector is characterized by a probabilistic motion model $p(u|\bar{u})$, where u and $\bar{u}$ are, respectively, the 'true' and the measured displacement.

In addition, each $\mathcal{A}_k$ is equipped with a *robot detector*, a sensor device that measures the relative position (typically, as bearing and distance) of other robots in $\mathcal{F}_k$, without the associated identity (see Figure 1b–c). Robot $\mathcal{A}_h$, $h \in \mathcal{N}_k$, is detected if it is placed in a *perception set* D_p that is rigidly attached to $\mathcal{F}_k$. No assumption is taken

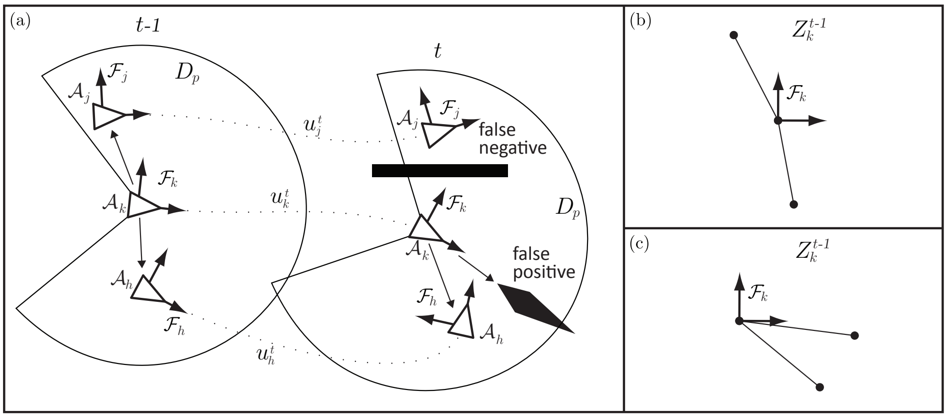


Figure 1 Settings of the mutual localization problem. Triangles are robots, black polygons are obstacles. The white regions surrounding the robots are the perception sets D_p at time instant $t-1$ and t.

on the shape of D_p. As shown in Fig. 1a, the robot detector is prone to false positives (it can be deceived by objects that look like robots) and false negatives (robots belonging to D_p which are not detected, e.g., due to line-of-sight occlusions). The measurements coming from the robot detector will be generically referred to as *features*, as a reminder of the fact that they are anonymous and, in addition, may or may not represent actual robots. We denote by $\bar{Z}_k^t$ the set of features detected by $\mathcal{A}_k$ at time t. For ease of notation, we define also the *observation* of the k-th robot at time t as the set of features in $\mathbb{R}^2$ detected by robot k plus the origin, and we denote it with $Z_k^t = \bar{Z}_k^t \cup \{(0,0)\}$. It represents the positions of the robots as measured by the robot detector of the k-th robot, i.e., relative to $\mathcal{F}_k$. Apart from the origin, which stands for $\mathcal{A}_k$ itself, Z_k^t does not convey any information about the identity of the robot located at a certain point (*anonymity*), nor about its orientation. Furthermore, the observation does not convey any information to distinguish a correct measurement from a false positive.

Finally, each robot $\mathcal{A}_k$, $k \in \mathcal{N}$, comes with a *communication module* that can send/receive data to/from any other robot $\mathcal{A}_h$, $h \in \mathcal{N}_k$, contained in a communication set D_c rigidly attached to $\mathcal{F}_k$. We assume that $D_p \subset D_c$, so that if $\mathcal{A}_k$ can detect $\mathcal{A}_h$ it can also communicate with it. Each message sent by $\mathcal{A}_k$ contains: (1) the robot signature (the index k), (2) the current composition of the motion displacements $\bar{v}_k^t = \bar{u}_k^1 \oplus \ldots \oplus \bar{u}_k^t$ incrementally obtained from the elementary measurements provided by the motion detector, (3) the feature set Z_k^t. Note that the receiver of two messages from $\mathcal{A}_k$ at time t and $t' < t$ can easily compute the motion displacement of $\mathcal{A}_k$ between the two instants as $\bar{v}^t \ominus \bar{v}^{t'}$. False negatives may also affect the communication (robot belonging to D_c that do not receive messages), whereas false positives in the communication may be easily avoided by appropriate message coding. We denote by C_k^t the set of robots which communicate with $\mathcal{A}_k$ at time t (*communication neighbors* of $\mathcal{A}_k$) and we let $C_k^{1:t} = \cup_{\tau=1}^t C_k^\tau$.

The Relative Mutual Localization (RML) problem with anonymous position measurements requires the generic robot $\mathcal{A}_i$ to compute, at each time instant t, its belief about the relative poses of those robots with which $\mathcal{A}_i$ has communicated, on the basis of the anonymous relative position measurements gathered directly by itself and obtained via communication with other robots.

Problem 1 (RML with anonymous position measurements)**.** *For each $t = 1, 2, \ldots$ and $j \in C_i^{1:t}$, compute the* belief

$$\text{bel}(x_j^t) := p(x_j^t | \bar{u}_i^{1:t}, Z_i^{1:t}, \{\bar{v}_j^\tau, Z_j^\tau\}_{\tau=1,\ldots,t,\, j \in C_i^\tau}),$$

given $\bar{u}_i^{1:t}$, $Z_i^{1:t}$, $\bar{v}_j^\tau$ and Z_j^τ, $\forall(\tau, j)$ s.t. $\tau = 1, \ldots, t$ and $j \in C_i^\tau$.

The study of the solution of this problem will be the main object of this thesis.

chapter 3

Motivation

All authors dealing with multi-robot localization have in one way or another addressed the data association problem. Some authors have found also unusual ways to solve it, such has collecting all the data and then manually associating the measurements to the robots. However, the most popular solution is tagging. The word *tagging* comes from the transitive verb *to tag*, whose meaning is [47]

- to supply with an identifying marker or price
- to provide with a name or epithet: label, brand

In robotics, tagging means to provide each robot with a distinctive feature that allows its unique recognition. It is usually implemented through the addition of different colors or patterns to each robot, so that their recognition can be performed using feature tracking algorithms on the images provided by cameras.

However, the application of tagging has its drawbacks and is not always reliable. First, tagging is in some way centralized, since it implies an *a priori* common knowledge among the robots in the form of a common transcoding table. Moreover, depending on the sensory equipment of the robots, tagging could be not applicable. For example, this is the case of distance sensors as range finders. The non applicability of tagging is evident in Figure 2, which shows an example of lidar scan and the features extracted from it. Not only the reconstruction of the identity of the sensed robots is impossible, but also some obstacles are mistakenly recognized as possible robots.

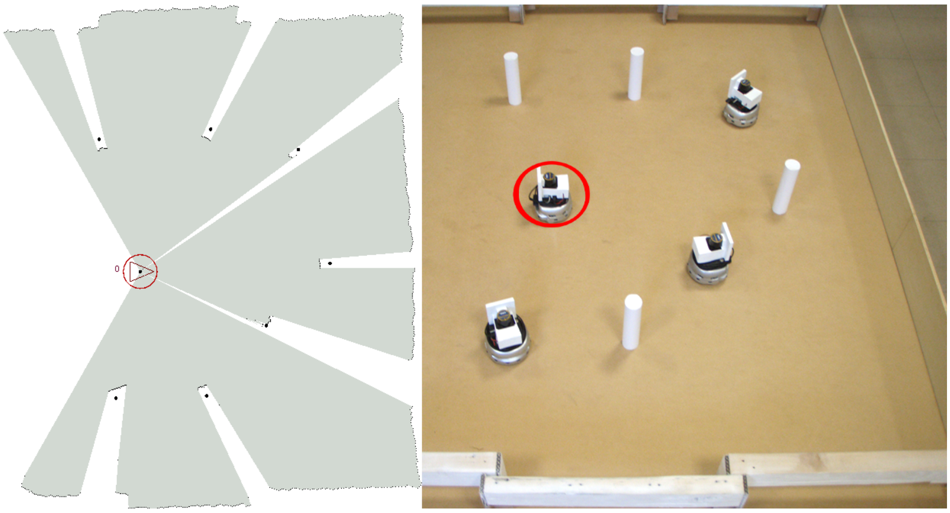


Figure 2 Example of lidar scan, with the extracted features.

We can draw the same conclusion on the non-applicability of tagging by considering swarm robotics, that usually involves a great number of small robots with limited sensing capabilities. In such situation, tagging could result unreliable. Assuming that each robot computes an estimate of the state of its neighbors, one may object that a great number of swarm behaviors can be obtained without the knowledge of their identities. Paradoxically, the estimation of the identities is needed to improve the estimate of the state. In fact, assuming that the robots communicate their odometries (or inertial measurements) to the neighbors, this information would be useless if the estimates of their state would not be accompanied with the identities. By improving the estimates used to compute the control law, the control itself would indirectly benefits from the estimation of the identities.

Other variables affecting visual tagging (i.e.: performed with cameras) can be environmental conditions. Low visibility can severely affect the capacity of recognition making tagging unavailable or at least unreliable. This is the case of smoky environments in emergency situations as well as conditions of low illumination typical of underwater and nocturnal environments.

Last, but not least, tagging could affect the task. An example of this happens when the task is disguising enemy agents. Imagine a scenario in which a robot needs to complete a path to deliver a packet, and some enemy agents are in charge to stop it. A feasible strategy could be to send a large number of support robots pretending to be the actual deliverer. All the robots could then cooperate and look for the best strategy to deceive the enemies. In such case, showing up the identity of the robots, revealing also the identity of the actual deliverer, is not a viable solution.

We are aware that the actual trend in robotics pushes for an always growing use of cameras, and tagging is a good solution in a great number of applications. However, there are some niches, of which we have given some examples, that denies this technique. In these situations the problem of anonymous measurements indeed materializes and its solution can further expand the always growing field of application of multi-robot systems.

chapter 4

Consequences of anonymity

In the previous chapter we have shown the motivation for assuming an unknown data association between measurements and robots, that is what we call anonymous measurements. However, our choice has some drawbacks. When the data association is known, given all the exteroceptive measurements (i.e.: measurements from the robot detector) of all the robots in a given time instant and assuming no noise on the measurements, the reconstruction of the formation is always possible when it is rigid, and can be computed through simple geometric arguments. In particular, given position measurements, any formation is rigid when the measurement graph is complete, that happens in general in case of unlimited field of view and no false negatives. False positives measurements are intrinsically removed by the assumption of known data association. In case of noisy measurements, the formation can be reconstructed through some optimization method.

If the data association is unknown, there exist formations in which, even in the simple case of complete measuring graph, the solution is non-unique. For example, trying to arrange the measurement sets (Figure 3b) obtained from the formation in Figure 3a, leads to two equivalent feasible formations (Figure 3c).

In this context, before tackling Problem 1, we want here to formalize its static equivalent and present some interesting results over its solution. In particular, we want to identify conditions under which

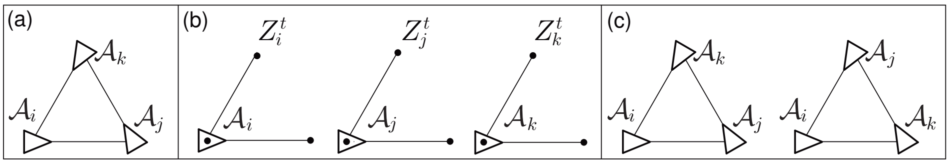


Figure 3 Example of ambiguity: a) actual formation; b) measurements sets; c) feasible formations from the measurements.

the spatial arrangement of the robots can be uniquely reconstructed (up to roto-translations) from the knowledge of all exteroceptive sensory data.

To provide a more formal definition of the problem, we introduce here a reference frame $\mathcal{F}$, in which we can express as z_i and θ_i respectively the origin and orientation of $\mathcal{F}_i$ with respect to $\mathcal{F}$. For simplicity, we use the same symbol (e.g., z) to indicate a point and its Cartesian coordinates; the actual meaning will be clear from the context. Since $\mathbb{R}^2 \times S^1$ is homeomorphic to $SE(2)$, any pose may also be interpreted as a roto-translation. A *formation* is a set of n poses $\{x_1, \ldots, x_n\}$ in $\mathcal{F}$, with x_i assigned to $\mathcal{A}_i$. Since we are interested in computing the group formation up to roto-translations, we can set without loss of generality $\mathcal{F} = \mathcal{F}_1$, so that $x_1 = ((0\ 0)^T, 0)$. This means that all formations will be expressed in the frame attached to $\mathcal{A}_1$. Clearly, all results can be expressed in another frame $\mathcal{F}'$ provided that the pose of $\mathcal{A}_1$ with respect to $\mathcal{F}'$ is known.

To simplify the problem, we assume for now complete measuring graph. Under these conditions, all the observations of a given group are the same up to roto-translations. We can now state the following problem:

Problem 2 (Static Mutual Localization with Anonymous Position Measurements and Complete Measuring Graph)**.** *Given n observations*

$Z_1, \dots, Z_n$, find all the possible pairs of functions

$$\hat{z} : \{2, \dots, n\} \to Z_1 \backslash (0\,0)^T$$
$$\hat{\theta} : \{2, \dots, n\} \to [0, 2\pi),$$

with $\hat{z}$ bijective, such that

$$Z_1 \ominus \hat{x}_i = Z_i \qquad i = 2, \dots, n, \tag{1}$$

where $\hat{x}_i := (\hat{z}(i), \hat{\theta}(i))$.

Function $\hat{z}$ assigns each point of Z_1 (with the exception of the origin) to one and only one robot in $\{\mathcal{A}_2, \dots, \mathcal{A}_n\}$, whose orientation is then defined by $\hat{\theta}$. Note that $\mathcal{A}_1$ is directly associated to the origin, with orientation equal to zero, in all solutions to the problem. Stated differently, Problem 2 consists in finding all the formations $\{\hat{x}_1 = ((0 \quad 0)^T, 0), \hat{x}_2 \dots, \hat{x}_n\}$ that are compatible with the given observations, i.e., satisfy (1).

In general, a solution to Problem 2 may exist or not. In the following, we assume that each observation Z_i, $i = 1, \dots, n$, has been gathered by robot $\mathcal{A}_i$ with reference to the same spatial arrangement of the group. This is sufficient to claim that Problem 2 admits at least one solution.

4.1 Unique Solvability, Structure and Number of Solutions

In this section we give a necessary and sufficient condition for the unique solvability of Problem 2 (Proposition 1), an associated test (Proposition 2), and a quantitative and qualitative characterization of the solutions (Propositions 3 and 4). In particular, we show that the

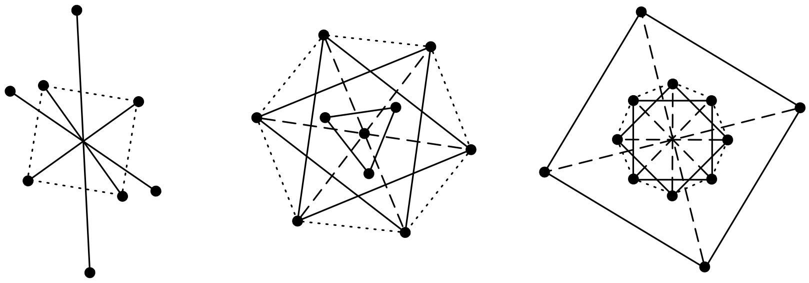


Figure 4 Three rotational symmetric sets of points. From left to right, the associated proper symmetric groups are respectively G_2, G_3 and G_4. Note that only the second set contains its centroid. Solid line segments join points that belong to the same set of the rotational symmetric partition. Dotted line segments show the presence of partial higher-degree symmetries which are not relevant for the analysis: from left to right, they identify respectively a square, an hexagon and an octagon. Dashed line segments meet at the centroid of each set.

problem is uniquely solvable if and only if the set of points represented by observation Z_1 does not have a rotational symmetry (remember that all observations are the same up to roto-translations). Furthermore, we show that in the case of non-unique solvability the number of solutions increases factorially with n, the number of robots. To establish these results, we first recall a few basic concepts on rotational symmetry.

4.1.1 A brush-up on rotational symmetry

Consider a set of n points $Z \subset \mathbb{R}^2$. Let $\mathcal{S}_Z$ denote the *proper symmetry* group of Z, i.e., the subgroup of its orientation-preserving isometries (roto-translations) under which it is invariant. It is known from symmetry group theory [48] that, since Z is a bounded set, $\mathcal{S}_Z$ can be represented as a subgroup of $SO(2)$ (the group of planar rotations), by choosing the origin to be its fixed point, i.e., the centroid of Z (since after any rotation in $\mathcal{S}_Z$ the set of points Z remains the same, also the centroid remains the same, hence the centroid is the fixed point). In particular, there exists a positive integer l such that $\mathcal{S}_Z = G_l$, where G_l is the *cyclic group of order* l, whose generator is the rotation of

$2\pi/l$. Z is said to be *rotational symmetric* if $\mathcal{S}_Z \neq G_1$, where G_1 is the trivial group containing only the identity operation.

Assume that $\mathcal{S}_Z = G_l$ and let c be the centroid of Z. Denote by $q_\phi = (c - R(\phi)c, \phi)$ the rotation by an angle ϕ around c, and in particular by

$$q_k := (c - R(2k\pi/l)c, 2k\pi/l), \tag{2}$$

the rotation by $2k\pi/l$, for $k = 0, 1, \dots, l-1$. We have then $Z = Z \ominus q_k = q_k \oplus Z$, for $k = 0, 1, \dots, l-1$. Note that rotational symmetry is invariant under isometries: if Z is rotational symmetric, also $Z \ominus x$ is rotational symmetric, for any $x \in SE(2)$. Examples of rotational symmetric sets of points are shown in Fig. 4.

The following Lemma establishes a property which is valid for any finite set of points and has an important role in the study of the unique solvability of Problem 2.

Lemma 1 (Rotational Symmetric Partition). *For each set Z of n points for which $\mathcal{S}_Z = G_l$, there exists a partition $\mathcal{E}_Z = \{E_1, \dots, E_m\}$ of Z such that E_j, $j = 1, \dots, m$, is invariant under any rotation in G_l around the centroid c, i.e.,*

$$E_j = E_j \ominus q_k\,, \quad k = 0, 1 \dots, l-1.$$

If $c \notin Z$, then l divides n, $m = n/l$, and the cardinality of each subset of the partition $\mathcal{E}_Z$ is l. If $c \in Z$, then l divides $n-1$, $m = 1 + (n-1)/l$ and the cardinality of each subset in $\mathcal{E}_Z \backslash \{c\}$ is l.

Proof. Suppose without loss of generality that c is the origin. Chosen a point $z \in Z\backslash\{c\}$, the set $E(z)$ of all points obtained applying an element of G_l to z is a subset of Z by definition. Clearly, $E(z)$ has cardinality l and is invariant under G_l. Now choose a point z' in

n	possible values of l										
1	1										
2	1	2									
3	1	2	3								
4	1	2	3	4							
5	1	2		4	5						
6	1	2	3		5	6					
7	1	2	3			6	7				
8	1	2		4			7	8			
9	1	2	3	4				8	9		
10	1	2	3		5				9	10	
$\vdots$	$\vdots$										$\ddots$

Figure 5 The possible values of the integer l for the cyclic groups G_l that can be the proper symmetry groups of a set Z of n points. Note that, since Z can always be non-rotational symmetric, $l = 1$ is ubiquitous. Also, $l = 2$ is always possible since for any odd value of n one point can be always placed in the centroid.

$Z \backslash E(z)$, repeat the above construction to obtain $E(z')$, and proceed as before. If $c \notin Z$, the collection of all the distinct sets $E(z)$ for all $z \in Z$ gives the subsets $E_1, \dots, E_m$ of the partition $\mathcal{E}_Z$, with $m = n/l$. On the other hand, if $c \in Z$ then set $E(c)$ is a singleton and must be added to the previous collection, which consists in this case of $(n-1)/l$ subsets. ∎

Figure 4 shows the partitions for three different rotational symmetric set of points, while in Figure 5 the possible values of l are tabulated for sets of $n = 1, \dots, 10$ points. Limit cases are $l = 1$ (the set of points is not rotational symmetric, and the partition consists of n singletons) and $l = n$ (the set of points may be a regular n-gon, and the partition consists if a single set containing all the points in Z).

4.2 Unique solvability of Problem 2

In the rest of this section, we assume that $\mathcal{S}_{Z_1} = G_l$ and denote by c the centroid of Z_1. The role of rotational symmetry in the unique solvability of Problem 2 is clarified by the following result.

Proposition 1 (Unique Solvability)**.** *Assume that Problem 2 admits a solution. The solution is unique if and only if Z_1 is not rotational symmetric.*

Proof. Assume that Problem 2 admits multiple solutions. Then there exists i and two poses $\hat{x}'_i$ and $\hat{x}''_i \neq \hat{x}'_i$ such that $Z_1 \ominus \hat{x}'_i = Z_i$ and $Z_1 \ominus \hat{x}''_i = Z_i$. Then $Z_1 = \hat{x}''_i \oplus Z_1 \ominus \hat{x}'_i$, i.e., there exists a non-zero roto-translation which transforms Z_1 in itself; this means that Z_1 is rotational symmetric. On the other hand, assume that Z_1 is rotational symmetric. Since a solution $\{x_1, \dots, x_n\}$ exists, i.e., $Z_1 \ominus \hat{x}_i = Z_i$, $i = 1, \dots, n$, there exists a non-zero roto-translation x which transforms Z_1 in itself, i.e., $Z_1 = Z_1 \ominus x$. This means that $\{x \ominus x_1, \dots, x \ominus x_n\}$ is also a solution. ■

Proposition 1 implies that the number of solutions to Problem 2 is invariant with respect to changes in the orientations of the robots in the formation (in spite of the fact that the observations change).

Unique solvability may be tested with the aid of the following result.

Proposition 2 (Unique Solvability Test)**.** *Denote with $Z_1(\phi)$ the set of points obtained by rotating the observation Z_1 by an angle ϕ around its centroid c, i.e.:*

$$Z_1(\phi) := \{R(\phi)(z - c) + c \mid z \in Z_1\}. \tag{3}$$

If $c \notin Z$, Problem 2 has a unique solution if and only if

$$Z_1 \neq Z_1(2\pi/m) \quad \forall m \text{ prime factor of } n. \tag{4}$$

If $c \in Z$, n must be replaced by $n - 1$ in (4).

Proof. Since Z_1 has n points, its proper-symmetry group $\mathcal{S}_{Z_1}$ can only be one of the cyclic groups $G_1, \dots, G_n$. In addition, since G_l, $2 \leq l \leq n$, also belongs to any G_m with m prime factor of l, and l can only be a divisor of n (if $c \notin Z_1$) or $n-1$ (if $c \in Z_1$), it is sufficient to check the rotations that are generators of the cyclic groups G_m, with m prime factor of n or $n-1$. ■

Assume that $c \notin Z_1$. Since condition (4) requires n checks for any value of m, the overall complexity of the test is $O(n \cdot \pi(n))$, where the *prime-counting function* $\pi(n)$ can be approximated by $n/\log(n)$. If $c \in Z_1$, the complexity is $O((n-1) \cdot \pi(n-1))$.

4.3 Structure and number of multiple solutions

We now turn our attention to the case when there are multiple solutions to Problem 2.

Proposition 3 (Structure of the Solutions)**.** *Let $i = 2, \ldots, n$. If $\hat{x}_i$ is a feasible pose for $\mathcal{A}_i$, in the sense that $\hat{x}_i = (\hat{z}_i, \hat{\theta}_i)$ satisfies (1), then all the non-zero poses obtained as $q_k \oplus \hat{x}_i$, with $k = 0, 1, \ldots, l-1$ and q_k defined by (2), are feasible for $\mathcal{A}_i$, and vice versa.*

Proof. Being $Z_1 \ominus \hat{x}_i = Z_i$ and $Z_1 = Z_1 \ominus q_k$, we have $(q_k \oplus Z_1) \ominus \hat{x}_i = Z_i$. Developing the pose compositions for an element z of Z_1 we have that

$$\begin{aligned}
(q_k \oplus z) \ominus \hat{x}_i &= (c_k + R(\phi_k)z) \ominus \hat{x}_i \\
&= R(-\hat{\theta}_i)(c_k + R(\phi_k)z - \hat{z}_i) \\
&= R(-\hat{\theta}_i)R(\phi_k)(z - R(-\phi_k)(\hat{z}_i - c_k)) \\
&= z \ominus (\hat{x}_i \ominus q_k).
\end{aligned}$$

Hence $(q_k \oplus Z_1) \ominus \hat{x}_i = Z_1 \ominus (\hat{x}_i \ominus q_k)$ and $\hat{x}_i \ominus q_k$ is a feasible solution, for $k = 0, 1, \ldots, l-1$, which is equivalent to say that $q_k \oplus \hat{x}_i$ is a feasible solution, for $k = 0, 1, \ldots, l-1$. Similarly, it is simple to show that for any other feasible pose $x' \oplus \hat{x}_i$, x' must belong to $\{q_k\}_{k=0,1,\ldots,l-1}$. ■

Proposition 3 essentially states that if the observations of Problem 2 are generated by a formation $\{x_1, \ldots, x_n\}$, then $\mathcal{A}_i$ can be assigned to position z_i as well as to all the other positions of the subset of $\mathcal{E}_{Z_1}$ which contains z_i. This leads to the following results.

Proposition 4 (Number of Solutions)**.** *The number of solutions to Problem 2 is*

$$(l-1)! \cdot (l!)^{\frac{n}{l}-1} \qquad \textit{if}\, c \notin Z_1 \qquad (5)$$

$$(l!)^{\frac{n-1}{l}} \qquad \textit{if}\, c \in Z_1. \qquad (6)$$

Proof. Remember that in all solutions $\mathcal{A}_1$ is at $(0\ \ 0)^T$. If $c \notin Z_1$, $\mathcal{E}_{Z_1}$ has n/l sets, each consisting of l positions. Each set of $\mathcal{E}_{Z_1}$ has l robots associated, and, in each solution, each of these robots (except for $\mathcal{A}_1$) can be placed in any position of the set, provided that this position is not occupied by another robot. Hence, $(l-1)!$ possible permutations correspond to the set of $\mathcal{E}_{Z_1}$ associated to $\mathcal{A}_1$, and $l!$ possible permutations correspond to the remaining $n/l - 1$ sets of $\mathcal{E}_{Z_1}$. Multiplying these possibilities we obtain (5). A similar analysis leads to (6) if $c \in Z_1$, noting that $\mathcal{A}_i$ associated to the set $\{c\}$ of $\mathcal{E}_{Z_1}$ has l possible poses if $i \neq 1$. ∎

Corollary 1. *For a given n, the maximum number of possible solutions to Problem 2 is $(n-1)!$. This number is actually reached when Z_1 is a regular n-gon if $c \notin Z_1$, and when $Z_1 \backslash c$ is a regular $(n-1)$-gon if $c \in Z_1$.*

Proof. If $c \in Z_1$ and $l = n-1$ then $(l!)^{\frac{n-1}{l}} = (n-1)!$ and $Z_1 \backslash c$ is a regular $(n-1)$-gon. If $l < n-1$, then l is a factor of $n-1$ and $m = (n-1)/l \in \mathbb{N}$. Both the numerator and denominator of

$$r = \frac{(l!)^m}{(n-1)!} = \frac{(l!)^{m-1}}{(n-1)(n-2)\ldots(l+1)}$$

are products of $l(m-1)$ factors and the smallest factor of the denominator is larger than the largest factor of the numerator. Then $r < 1$, and we can write $(l!)^{\frac{n-1}{l}} < (n-1)!$. For $c \notin Z_1$, a similar reasoning leads to $(l-1!)(l!)^{\frac{n}{l}-1} < (n-1)!$ if $l < n$, while if $l = n$ the number of solutions is $(n-1)!$ and Z_1 is a regular n-gon. ■

Summarizing, each point of Z_1 can be assigned to one and only one subset of partition $\mathcal{E}_{Z_1}$. If $c \notin Z_1$, Lemma 1 implies that each subset of $\mathcal{E}_{Z_1}$ has l positions and l robots assigned to it. Conversely, each robot can assume l different poses which correspond to all the l positions in its subset, with l different orientations that differ by a multiple of $2\pi/l$. The robots associated to the set to which $\mathcal{A}_1$ is associated have only $l-1$ possible poses instead of l. Note that all the robots associated to the same set have the observations equal up to a pure rotation. If $c \in Z_1$ then $\mathcal{A}_i$ $(i \neq 1)$ associated to $\{c\}$ has l different possible poses with the same position.

All the solutions are generated by independently permuting the possible poses of each robot, with the constraint that two robots cannot be at the same position. Hence, the set of solutions of Problem 2 is implicitly represented by (1) the set Z_1 (2) the partition $\mathcal{E}_{Z_1}$ of Z_1 (3) the association between each robot $\mathcal{A}_i$, $i = 1, \ldots, n$, and the corresponding set of $\mathcal{E}_{Z_1}$.

4.4 Non-complete measuring graph

We now sketch an extension of our problem to the case of non-complete measuring graph, i.e., observations that include only subsets of the group of the robots. This situation arises when limited-range or anisotropic sensors are used, or when line-of-sight occlusions occur due to obstacles or robots.

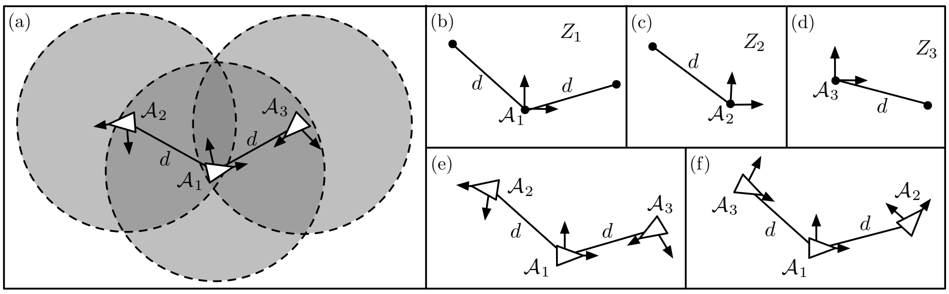


Figure 6 (a) An example of non-complete measuring graph due to sensor range limitations (b-d) observations of $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ (e-f) the two possible solutions, none of them rotational symmetric.

Problem 3 (Mutual Localization with Anonymous Position Measurements and Generic Measuring Graph)**.** *Given n observations $Z_1, \ldots, Z_n$, with $|Z_i| = m_i \leq n$, find all the sets $\hat{Z} \supset Z_1$, with $|\hat{Z}| = n$, together with the associated pairs of functions $\hat{z} : \{2, \ldots, n\} \to \hat{Z} \backslash (0\ 0)^T$, $\hat{\theta} : \{2, \ldots, n\} \to [0, 2\pi)$, with $\hat{z}$ bijective, such that*

$$\hat{Z} \ominus \hat{x}_i \supset Z_i \qquad i = 2, \ldots, n, \tag{7}$$

where $\hat{x}_i := (\hat{z}(i), \hat{\theta}(i))$.

Proposition 5 (Necessary Condition for Unique Solvability)**.** *If Problem 3 admits a solution in which $\hat{Z}$ is rotational symmetric, this solution is not unique.*

This result is obvious since a solution for which $\hat{Z}$ is rotational symmetric generates (by permutation of the robots in the same subset of $\mathcal{E}_{\hat{Z}}$) additional feasible solutions of Problem 3 that differ for the associated functions $\hat{z}$ and $\hat{\theta}$.

The condition of Proposition 5 is not sufficient. As shown by Fig. 6, there are cases in which multiple solutions exist, but none of them is associated to a rotational symmetric $\hat{Z}$.

chapter 5

Classical approach

For the best of our knowledge, no author has ever investigated Problem 1 as we have formulated here. However, some techniques have been developed through time to solve problems in multi-filtering when the data association of the measurements is unknown. Solution in closed form of Problem 1 can be performed by considering all possible data associations and keeping a bank of filters as the one described in [31], each representing one possible association. The estimates would be given by the filter that maximizes the likelihood.

Obviously, this solution is not applicable in practice due to the factorial number of possible associations. By considering only the measurements gathered by the owner of the algorithm the factorial number of associations would lead to an unsustainable number of filters. Assuming a team of n robots, the owner of the filter should perform one Kalman filter (whose state is $3n$-dimensional) for each possible data association, with $n!$ possible associations for each measurement step. After p measurement steps, the number of filter would be of $n!^p$. Taking into account also the measurements gathered from the other robots, the number of filters would further increase.

On the other hand, an algorithm could keep track only of the best data association for each time step, where *best* must be intended as the data association that maximizes the likelihood of the measurements. This way, although computationally tractable, is hazardous, since even only one mismatched data association could jeopardize the quality of the estimate.

The only feasible way to perform the data association is then to adopt a strategy midway between the two. An example of this strategy can be found in a tightly related field of mobile robotics, SLAM. Simultaneous Localization And Mapping is the problem for the robot to “acquire a map of its environment while simultaneously localizing itself relative to this map” [49]. Historically, the earliest SLAM algorithm is based on the EKF. In particular, EKF SLAM algorithm is based on the construction of feature-based maps composed of point landmarks. The data association between measurements and landmarks is performed using a maximum likelihood criterion. For this reason, EKF SLAM requires significant engineering of feature detectors, sometimes using artificial beacons. Moreover, the EKF SLAM is computationally demanding as the number of feature increases, limiting the applicability to some thousands of landmarks.

5.1 FastSLAM

Another possibility in filtering is the application of particle filters instead of EKFs. The straight application of particle filters to SLAM would bring to a computationally non-tractable algorithm. In fact, the number of required particles for the particle filter to work properly increases exponentially with the dimension of the state. In the case of 2D SLAM, the state would be of dimension 3 for the pose of the robot plus 2 for the position of each feature. To overcome this issue, the authors of [50] proposed FastSLAM, an algorithm that uses particles to represent the uncertainty on the pose of the robot while using EKFs for the estimate of the features. The algorithm relies on the fact that dependencies in the estimates of the feature position arise only through robot pose uncertainty. So, for each particle (in which the pose of the robot is assumed known) the individual map errors are conditionally independent. Hence, the estimation of the positions of the features can be performed through the use of separate 2-dimensional EKFs, one for each individual feature.

The conditional independence assumption results in the factorization of the posterior as:

$$p(y_{1:t}|z_{1:t}, u_{1:t}, c_{1:t}) = p(x_{1:t}|z_{1:t}, u_{1:t}, c_{1:t}) \prod_{n=1}^{N} p(m_n|x_{1:t}, z_{1:t}, c_{1:t}) \qquad (8)$$

where x is the state of the owner of the filter (the robot), m_n is the state of the n-th feature, z are the measurements, u are the control inputs and c is a variable that expresses the correspondences between measurements and features. Note that c is considered known in the computation of the posterior, so this algorithm considers known data association. However, in [51] the author extends FastSLAM to deal with the unknown data association assumption. Since each particle represents an hypothesis, the authors suggest to perform the data association on a per-particle basis. In particular, the simplest approach is to adopt the maximum likelihood assignment procedure used by EKFs independently in each particle. Particles that pick the correct data association will receive high probabilities, while particles that assign observations incorrectly will receive lower probabilities and will be removed during resampling. The authors underline that in this way is possible to remove wrong associations made in the past, while the effect of wrong associations can never be removed while using a simple EKF approach with maximum likelihood data association.

The authors bring this reasoning a step further proposing a Monte-Carlo data association. Each particle, instead of the application of the maximum likelihood criterion, draws a random association weighted by the probabilities of each landmark having generated the observation.

For the best of our knowledge, FastSLAM with Monte-Carlo data association is one of the most robust method to solve online a problem of data association. For this reason, we will compare the results obtained by FastSLAM in the solution of Problem 1 to the method that we will develop in the rest of this work. However, since Problem 1 is significantly different from the problem addressed by SLAM we have implemented and tested a modification of FastSLAM that applies to our scenario. In particular, $\mathcal{A}_i$ performs a *robo-centric* estimation of other robots as well as fixed obstacles that looks like robots (false positives) considering them as the features of FastSLAM. By the term robo-centric, we mean that the estimation is conducted in the frame $\mathcal{F}_i$ attached to $\mathcal{A}_i$. This has two main implications. First, in each particle the state of $\mathcal{A}_i$ is always null, so that the particle distribution will be used to estimate only the data association and not the state of $\mathcal{A}_i$. Second, all the EKFs of each particle, also the ones relative to fixed features, must be updated with the odometry of $\mathcal{A}_i$. Moreover, since the other robots move and communicate their odometry, $\mathcal{A}_i$ can use this information to update the corresponding EKFs filter of each particle.

In this framework we have also introduced measurements gathered and communicated by other robots. Unluckily, we can not use them to update the EKFs, since this would create dependences among the estimates, thus affecting the validity of equation (8). Assume that at a certain time $\mathcal{A}_i$ receive the measurements gathered by $\mathcal{A}_j$. To use them to update the filter, $\mathcal{A}_i$ needs to express those measurements in its frame of reference. However, its only knowledge on the pose of $\mathcal{A}_j$ is the estimate that it has computed, but if it uses that estimate to report $\mathcal{A}_j$'s measurements in its frame and then uses those quantities, then the updated estimates will be dependent from the estimate on the pose of $\mathcal{A}_j$, violating the assumption of conditional independence.

For this reason, we use the measurements of the other robots to update only the weights of the particles. Since each particle represents an hypothesis on the formation of the team, it is possible to compute for each particle the expected set of features of each robot (i.e.: a prediction of the measurements based on the state of the system in that particle). The particles receiving higher weights will be the ones whose expected measurements of all the robots are more similar to the actual measurements. We are aware that this use of the measurements of the other robots might still introduce some dependencies in the estimates. However, we have tested also the FastSLAM algorithm without using the measurements of the other robots and its performances are definitely worse.

The resulting algorithm has the same computational complexity of FastSLAM. However, as admitted also by the authors of [51], letting the algorithm estimate the data association requires a number of particles potentially factorial. We say here *potentially* because it depends also on the number of times in which two robots/features are close enough to be exchanged in the data association process. If all features and robots are well separated, this algorithm is likely to work properly. However, this situation is significantly easier with respect to our case of study in which the robots are moving in densely populated environments. If the occurrence of situations in which the robots/features can be exchanged is frequent, FastSLAM needs to keep track of all possible combinatorial data associations, leading to a factorial-growing number of required particles.

This problem is dramatically present above all in the first steps of the algorithm, and is magnified by the linearization introduced by the

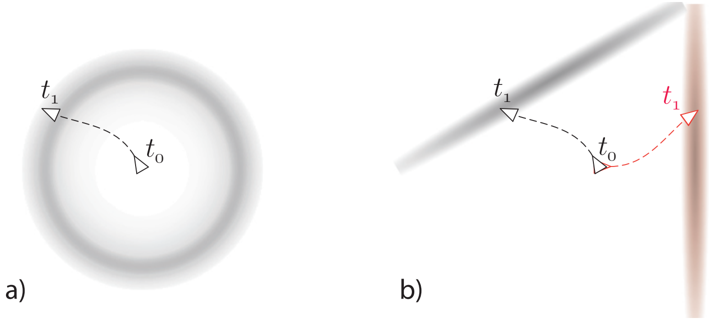


Figure 7 Real (a) and estimated by an EKF (b) pdf on the position of a moving unicycle; red cloud in (b) is the pdf estimated by an EKF initialized with a large orientation error.

EKF on the system model. This last issue can be explained through an example. Assume to know with no uncertainty the position of the unicycle at time t_0 in Figure 7a, but no knowledge is given on its orientation. If the unicycle moves and sends its odometry, the probability density function (pdf) of its position at time t_1 becomes a circle, as depicted in Figure 7a. An EKF with a correct initial orientation (despite the uncertainty on it) would behave as depicted in Figure 7b, where the gray cloud is the computed pdf. However, an EKF with a completely wrong initial orientation would generate a pdf depicted as the red cloud. Since the real position is not even close to the red cloud, a badly initialized EKF would probably fail to converge. This is even more probable under our assumption of anonymity of the measurements, since a maximum likelihood algorithm would probably never assign again a measurement to the badly initialized EKF.

The solution of this problem is suggested by Figure 7b itself. In fact, the estimation of the probability density function can be demanded to a bigger number of EKFs, each of them initialized with a different orientation. The pdfs estimated by banks of 2, 4, 8 and 16 EKFs whose initial orientations are regularly chosen are depicted in Figure 8. By Figure 8, we can say that each robot estimated in FastSLAM must be

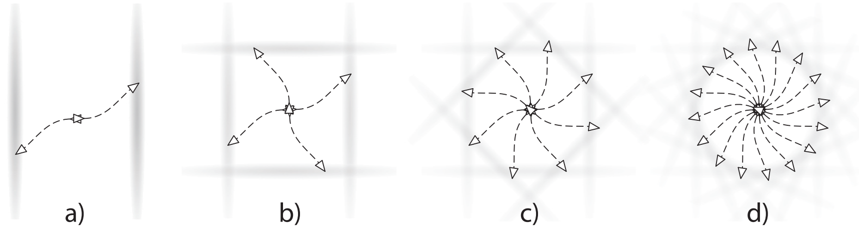


Figure 8 Pdfs estimated by banks of respectively 2, 4, 8 and 16 EKFs, whose initial orientation is regularly chosen.

initialized at least with 2 different orientations. This means that to keep track of all possible formations in the beginning (when there is no knowledge on the initial orientation), given a formation of all other estimated robots, each robot must be in at least 2 different particles with different orientations. This consideration lead to count at least for 2 each robot in the required number of particles, that is given by

$$(2 * n + m)! \tag{9}$$

where n is the number of robots and m is the number of other features.

The use of *negative information*, as described in [51], can help to overcome some of these computational issues. Negative information is when the robot expects to see a particular landmark (robot or feature) and does not. If this happens repeatedly, probably the estimation relative to that landmark is not correct. In this way, the robot could decide to reinitialize (in case of a robot, that is known to exist because it communicates) or eliminate (in case of a feature) a given landmark in a particle. This technique can be useful to 'save' particles in which the initial guess of the association was correct but the starting orientation of one robot was wrong.

To conclude, we want to underline that the above presented initial orientation problem arises only when other robots are part of the estimated landmarks, and this is why the authors of [51] were able to claim a simple factorial computational complexity with the number of features m. In the following section we show some experimental results.

5.2 Experimental Results

The following experiments are performed on a team of four Khepera III robots using MIP, an open source project devoted to multi-robot applications. Each robot is equipped with an Hokuyo URG-04LX (240° field of view) and encoders. The feature detector is a simple feature extraction algorithm that inspects the laser scan, looking for the indentations given by the small 'hat' mounted atop each robot. The ground truth is given by a system of a 3 cameras mounted over the arena in which the robots move. The simulations are performed with the same software. The details on the system, will be given in Chapter 8.

In the first experiment the robots start in a square (hence rotational symmetric) configuration. One of them moves along a closed path, while the others, including the owner of the algorithm $\mathcal{A}_1$, stand still. Figure 9 shows three snapshots of the experiment and the estimates computed at the same time instants. The triangles in the estimates represent the ground truth, while $\mathcal{A}_1$ is the robot circled in red. The estimates are printed as dots in the mean value of each EKF. The first snapshot is taken only few seconds after the start, and shows as the estimates of the moving robot (cyan) are sparse near all three true poses of the robots. In the second snapshot the moving robot has traveled about 270 deg of a circle, and most of its estimates are close to its real pose. In the last snapshot the moving robot has returned to the starting position, and all the estimates are so close to the real value to

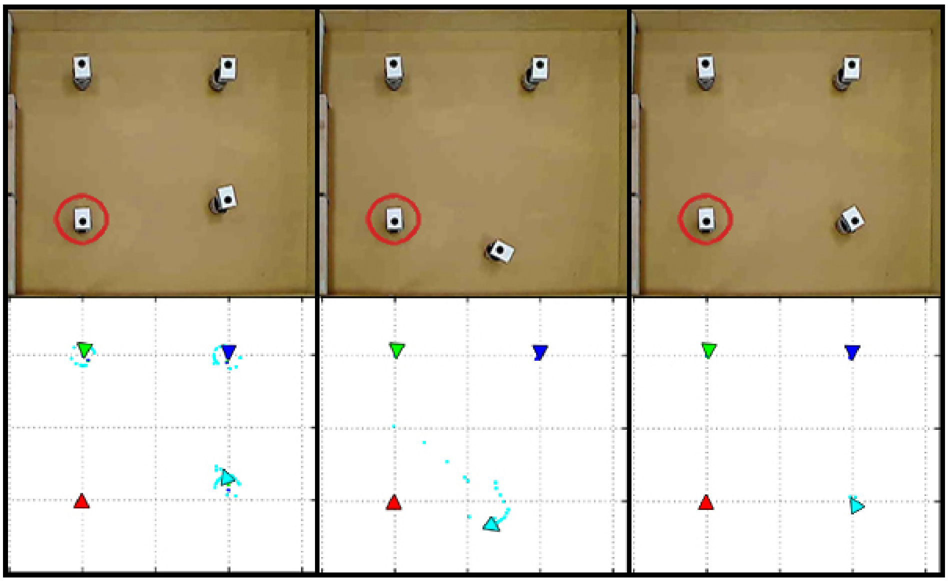


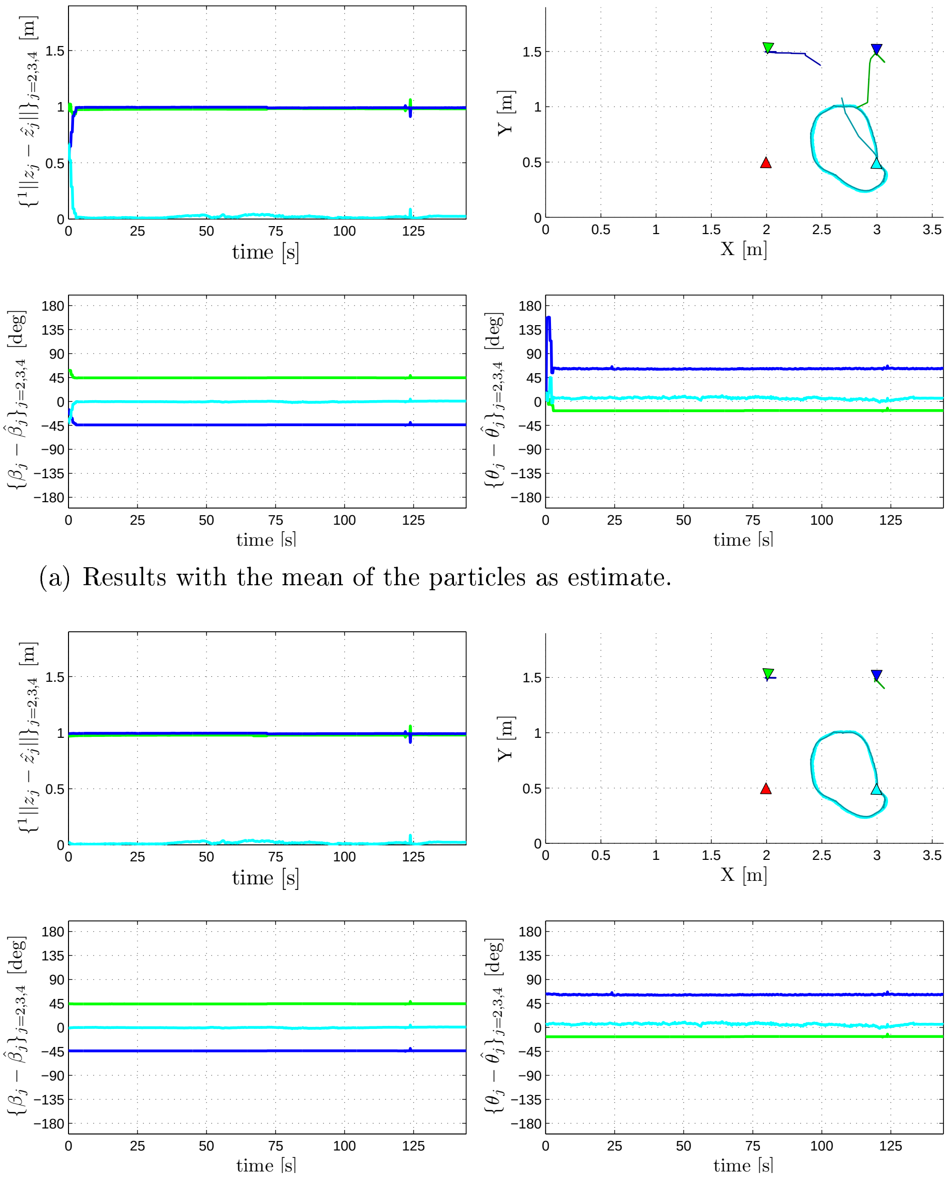
Figure 9 Top: snapshots of three instants of the first experiment; bottom: estimates computed by $\mathcal{A}_1$ (in red) in the same time instants. The triangles are the ground truth, while the mean of each EKF is printed as a dot. The estimation is conducted with 1000 particles.

be shadowed by the ground truth. The same applies to the estimates of the fixed (blue and green) robots, that in all three snapshots are shadowed by the ground truth.

Those snapshots are taken from an estimation performed with 1000 particles, however we have performed the estimation other 2 times with 10 and 100 particles, to highlight the complexity problem of the algorithm and the numerical results are presented in Figure 10, 11 and 12 respectively for 10, 100 and 1000 particles estimation. Each Figure shows the distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors for each robot, plus the trajectories of the robots and the estimates of the position on the XY plane (top-right). The errors are computed considering as estimate the mean of the particles (Figures 10(a), 11(a) and 12(a)) as well as the particle receiving the highest rating (Figures 10(b), 11(b) and 12(b)) However, since the mean criterion shows better results with respect to the best particle criterion, we will adopt this as the standard criterion for all following experiments, not showing again the results for the best particle. In particular the mean value, although a slower convergence, produces more stable estimates.

This experiment is useful to validate Equation (9), whose prevision is of $(3 * 2)! = 720$ required particles. If we try to run the localization system with only 10 particles (Figure 10) the result is not good, since the identities of R_2 and R_3 (green and blue in the Figure) are exchanged. In fact, during the experiment the distance error converges to 1 m for both the robots, that is also the distance between them. Running the filter with 100 particles (Figure 11) produces a better estimate and the distance error relative to all the robots converges to zero. However, the orientation of R_2 and R_3 is still affected by significant error. The same experiment run with 1000 particles (Figure 12) produces an estimate in which all errors converge to zero.

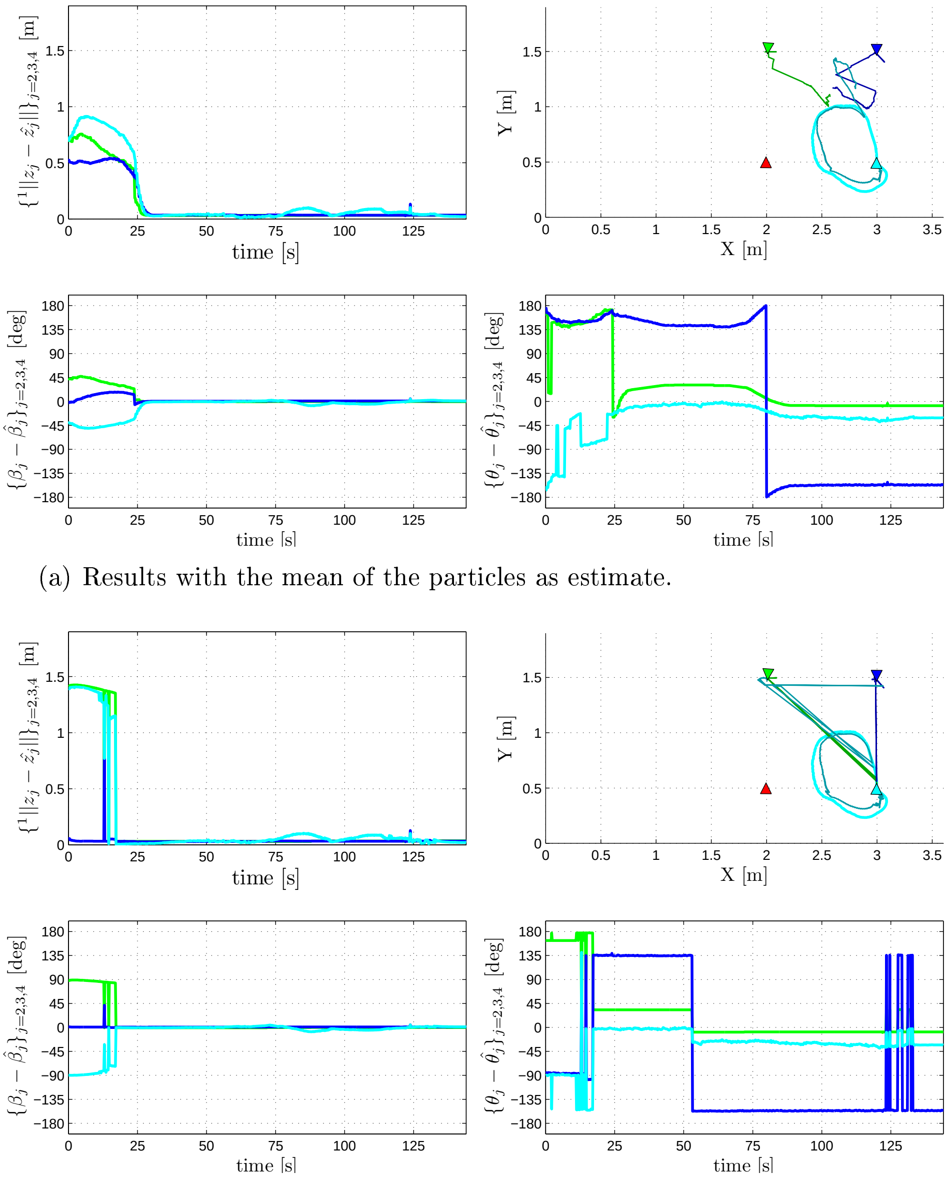
In the second and more general experiment depicted in Figure 13, all the four robots move from a starting lattice rotational symmetric configuration completed by two fixed obstacles detected as features. The increased difficulty of the experiments completely disrupts the estimate, as shown by both the snapshots and the distance, bearing and orientation errors plotted in Figure 14. In this situation, in fact, the number of required particles given by Equation (9) is 40320, way over the capabilities of any commercial computer. The first snapshot shows the initial configuration. In the beginning of the algorithm, the estimates of two robots are performing well. In particular, most of the green and cyan particles are around the respective ground truth in the first snapshot. However, very few blue particles are close to the ground truth of the correct robot, while much more are visible near the the black circles representing the obstacles. As the experiment



(a) Results with the mean of the particles as estimate.

(b) Results with the best of the particles as estimate.

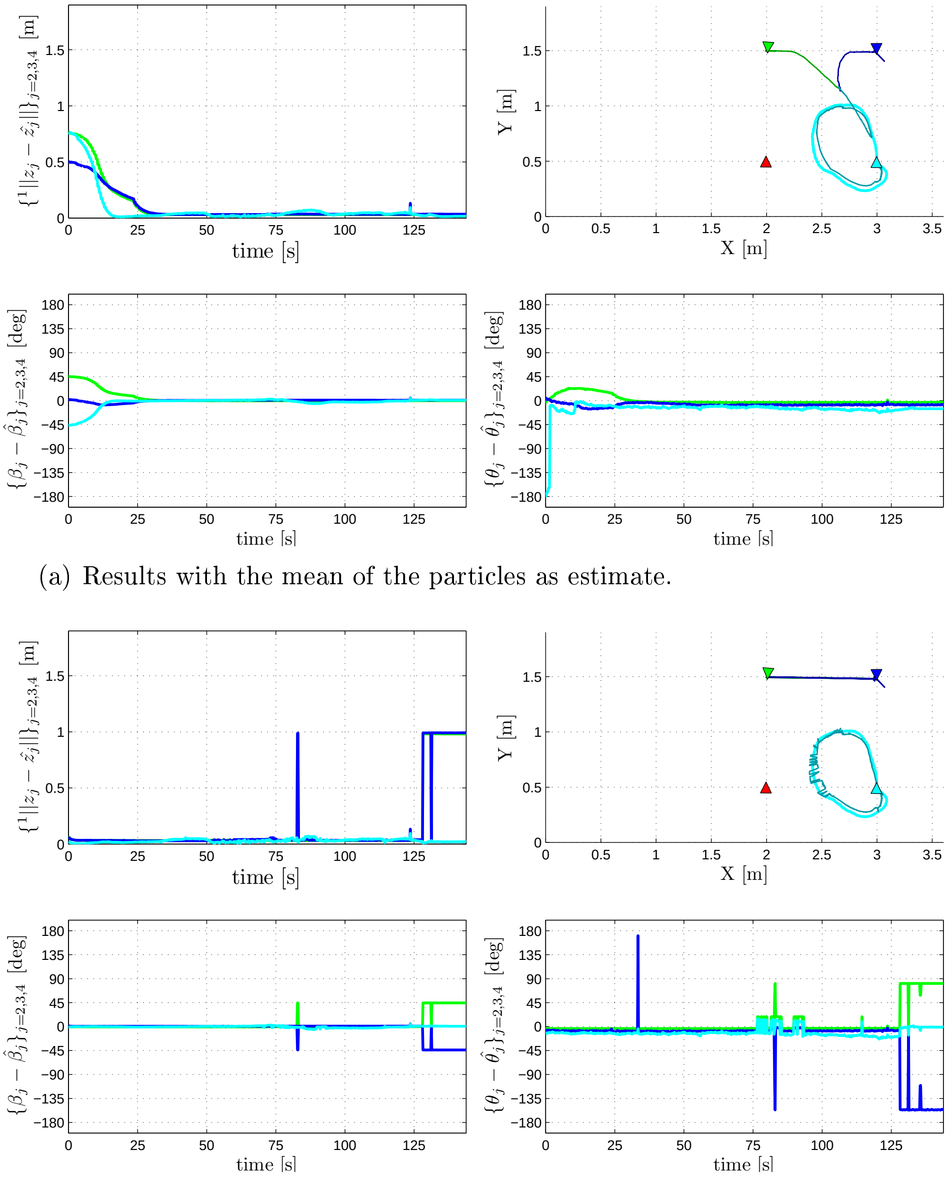
Figure 10 Distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors on the estimates produced by a filter with 10 particles as mean of the particles (a) and particle with the highest value (b) in the first experiment. In top-right the real trajectories and the estimates are plotted on the XY plane; triangles are the starting configurations of the robots; the red robot is the robot performing the filter.



(a) Results with the mean of the particles as estimate.

(b) Results with the best of the particles as estimate.

Figure 11 Distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors on the estimates produced by a filter with 100 particles as mean of the particles (a) and particle with the highest value (b) in the first experiment. In top-right the real trajectories and the estimates are plotted on the XY plane; triangles are the starting configurations of the robots; the red robot is the robot performing the filter.



(a) Results with the mean of the particles as estimate.

(b) Results with the best of the particles as estimate.

Figure 12 Distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors on the estimates produced by a filter with 1000 particles as mean of the particles (a) and particle with the highest value (b) in the first experiment. In top-right the real trajectories and the estimates are plotted on the XY plane; triangles are the starting configurations of the robots; the red robot is the robot performing the filter.

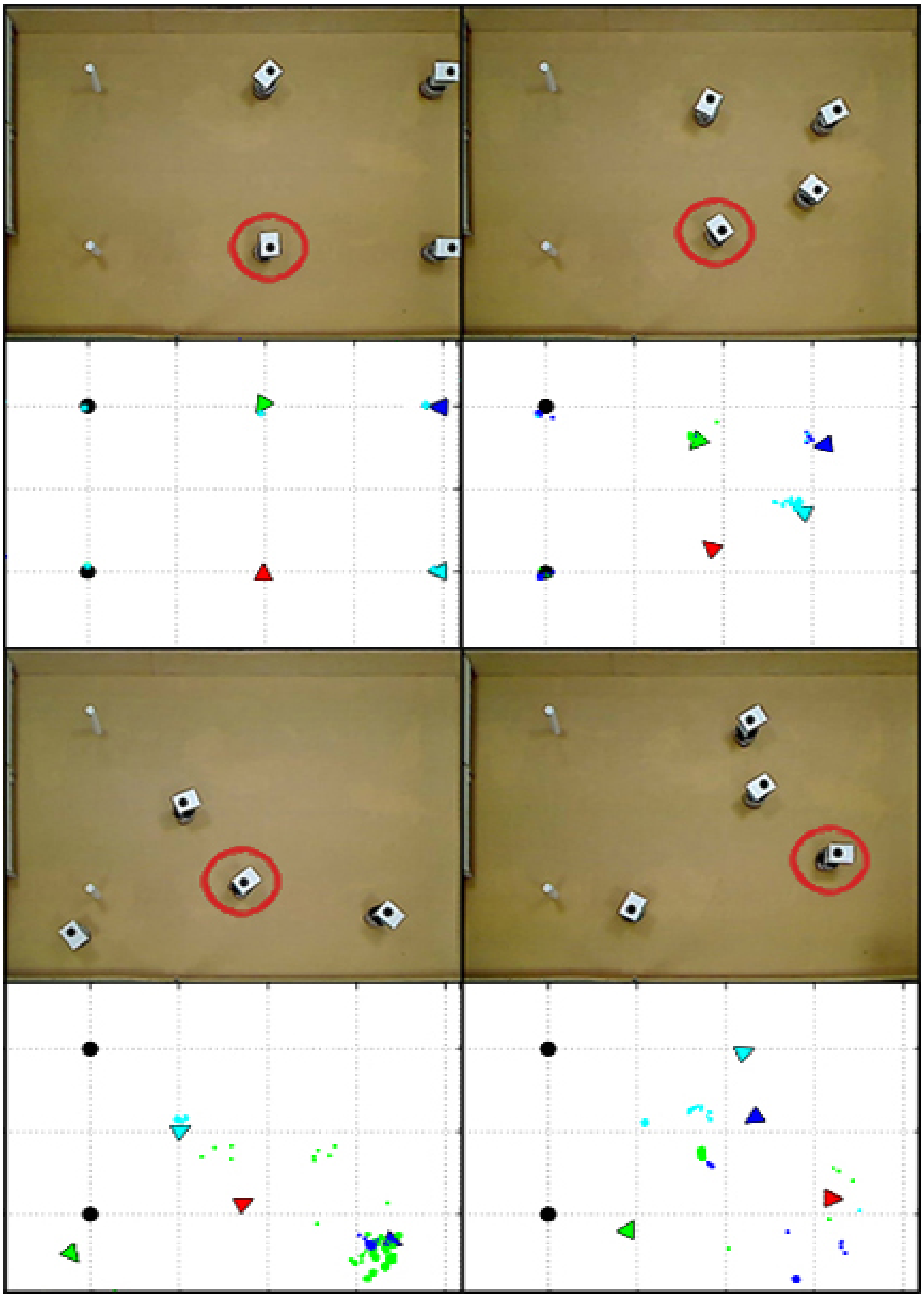


Figure 13 Top: snapshots of four instants of the second experiment; bottom: estimates computed by $\mathcal{A}_1$ (in red) in the same time instants. The estimation is conducted with 1000 particles.

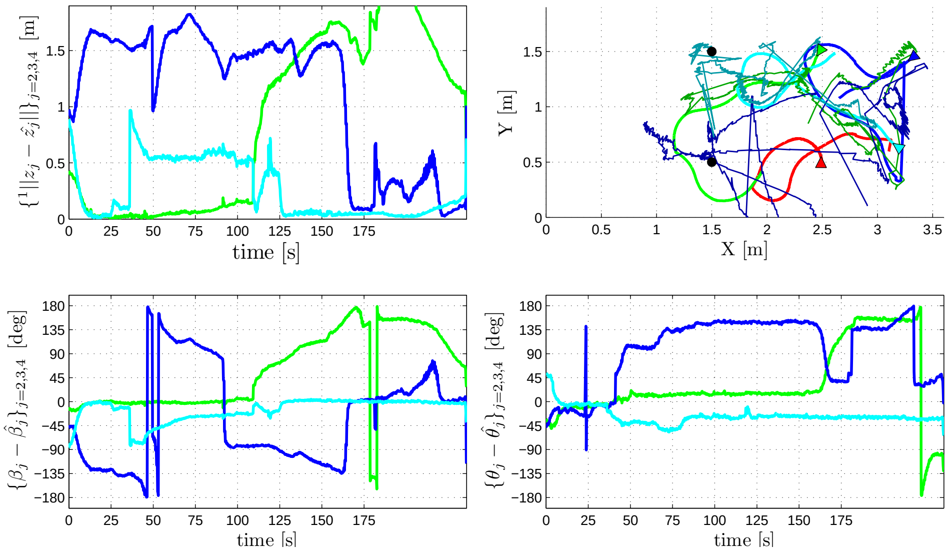


Figure 14 Distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors on the estimates produced by a filter with 1000 particles as mean of the particles in the second experiment. In top-right the real trajectories and the estimates are plotted on the XY plane; triangles are the starting configurations of the robots; the red robot is the robot performing the filter.

continues, the estimation errors continue to increase and the final result is a complete nonsense estimate of the formation as depicted in the fourth snapshot. The same conclusions can be drawn from the plots of the errors.

As prove of the fact that the main problem is the estimation of the identities of the robots, we have tested this algorithm in a pure SLAM scenario, in a simulation with 6 well separated fixed features.

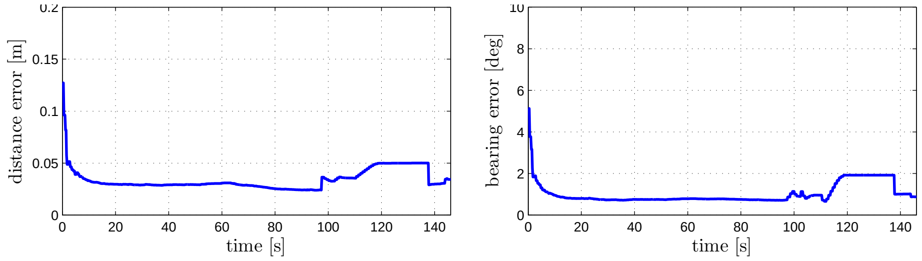


Figure 15 Mean square distance and bearing errors in a SLAM simulation; the estimation is conducted with 100 particles.

Figure 15 shows the mean square distance and bearing errors of all the features detected during the simulation.

The experiments presented in this section prove that this approach, although valid for the SLAM problem, offers a good solution for the mutual localization problem with anonymous measurements only in very limited situations, with small number of robots and features.

part III

Multiple registration-based method

chapter 6

Mutual localization system

The distinctive aspects of Problem 1 with respect to standard versions of the mutual localization problem (e.g., those in [29, 44, 52]) are the anonymity of the measurements and the presence of false positives/negatives. The classical approach explained in the previous chapter is reasonably successful provided that robots and features in general are well spaced. Only in this way the ambiguities can be avoided and are occasional. Moreover, the initial ambiguity cannot be avoided even if the starting configuration is not rotational symmetric, leading to an exponential complexity dependent on the number of robots plus the number of features.

Here, we would like to design a mutual localization method that is effective for tasks requiring densely populated environments, with frequent non pairwise encounters among robots (e.g., in the course of formation control, cooperative exploration, multiple-view environment monitoring, sensor data fusion). With this objective, we adopt the approach outlined in Figure 16. The generic robot $\mathcal{A}_i$ applies a multiple registration algorithm to compute the most likely relative poses of the robots belonging to C_i^t, on the basis of the sets of features Z_i^t, $\{Z_j^t\}_{j \in C_i^t}$, and the current beliefs $\overline{\mathrm{bel}}(x_j^t) = p(x_j^t | \bar{u}_i^{1:t}, Z_i^{1:t-1}, \{\bar{v}_h^{1:t}, Z_h^{1:t-1}\}_{h \in C_i^{1:t}})_{j \in C_i^t}$ about $\{x_j^t\}_{j \in C_i^t}$. The relative poses thus obtained, together with the measurements from the motion detector $\bar{u}_i^t$ and $\{\bar{v}_j^t\}_{j \in C_i^t}$, are used by $|C_i^{1:t}|$ particle filters to update the belief about the pose of each robot in $C_i^{1:t}$ (where $|\cdot|$ denotes

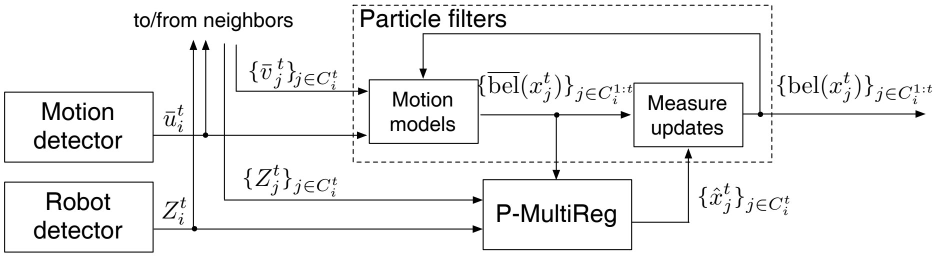

to/from neighbors
Particle filters
Motion detector
Robot detector
$\bar{u}_i^t$
Z_i^t
$\{\bar{v}_j^t\}_{j\in C_i^t}$
$\{Z_j^t\}_{j\in C_i^t}$
Motion models
$\{\overline{\text{bel}}(x_j^t)\}_{j\in C_i^{1:t}}$
Measure updates
$\{\text{bel}(x_j^t)\}_{j\in C_i^{1:t}}$
P-MultiReg
$\{\hat{x}_j^t\}_{j\in C_i^t}$

Figure 16 Scheme of the mutual localization system that runs on $\mathcal{A}_i$.

the cardinality of a set). The multiple registration algorithm and the particle filters are respectively described in Sections 6.1 and 6.2.

The advantages of our two-stage approach with respect to the more classical approach discussed in Chapter 5 are:

1. the mutual exclusive structure of the set of features is exploited in the registration phase;

2. the increased dimension of the measurements (the relative orientation is also provided by P-MultiReg) results in a faster convergence of the estimation process;

3. the multi-robot system achieves, in a distributed way, a result comparable with the outcome of a single centralized sensor taking simultaneous snapshots of the same scene from different viewpoints. This expands the perception capabilities of the system members beyond those of the individual robots, without asking for a central data processor. For example, $\mathcal{A}_i$ can now estimate the pose of robots which are occluded by an obstacle or in a blind region of the robot detector, provided that they are seen by other robots;

4. the robot detector probabilistic model in the particle filter is simpler since it does not have to take into account the identity of the robots.

These advantages come at an acceptable price. In fact, the increased complexity experienced when the initial arrangement of the system is close to being ambiguous, i.e, rotational symmetric (as shown in Section 4.1) would still be unavoidable with the classical approach. The situation is even better if the system comes from a steadily non-ambiguous situation so that the localization method has been able to reduce uncertainty to a minimum. In this case, the arising of rotational symmetries does not affect the complexity of the algorithm. To overcome the complexity issue of a possible ambiguous starting configuration, we will present in Chapter 12 a control law that breaks symmetry. In any case, as will be shown in the experiments, the particle filter copes flawlessly with the multiple hypotheses computed in this situation by the registration algorithm for the relative pose of the robots.

6.1 Multiple registration algorithm

At each time instant t, the generic robot $\mathcal{A}_i$ runs *Probabilistic MultiReg* (abbreviated as P-MultiReg), a multiple registration algorithm that represents the part of the localization system which directly interfaces with the particle filters (see Figure 16). In general, *registration* is the process of computing the relative pose between two or more different viewpoints of the same scene. In our case, P-MultiReg derives a set of feasible estimates $\{\hat{x}_j^t\}$ of each relative pose x_j^t, $j \in C_i^t$, given the observations Z_i^t, $\{Z_j^t\}_{j \in C_i^t}$, and the beliefs $\{\overline{\mathrm{bel}}(x_j^t)\}_{j \in C_i^t}$, which are obtained from the motion model blocks of the particle filters (see Figure 16).

In the following we will be giving the formal definition of the multiple registration problem

Problem 4 (Multiple registration with false positives)**.** *Given n observations $Z_1, \ldots, Z_n$, find all the sets $\hat{Z}_i \supset Z_i$ such that $|\hat{Z}_i| \geq n$ and all the possible pairs of functions*

$$\hat{z} : \mathcal{N}_i \to \hat{Z}_i \backslash (0\,0)^T$$
$$\hat{\theta} : \mathcal{N}_i \to [0, 2\pi)$$

with $\hat{z}$ injective, such that

$$\hat{Z}_i \ominus \hat{x}_j \supset Z_j \qquad j \in \mathcal{N}_i \tag{10}$$

where $\hat{x}_i := (\hat{z}(j), \hat{\theta}(j))$.

Problem 4 could be solved in closed form through the enumeration of all possible correspondences between features of different observations. However, this method has obviously a combinatorial complexity that refrains by its application. No other techniques are known for the solution in closed form of Problem 4. Here we propose an algorithm, P-MultiReg, that uses RANSAC [53] to identify the most likely correspondences between tuples of features belonging to Z_i^t, $\{Z_j^t\}_{j \in C_i^t}$ and a least squares estimation to compute the poses $\hat{x}_j^t$ that best fit those correspondences, i.e., that maximize the likelihood of the measurements. A pseudo-code description of P-MultiReg is given in Algorithm 1. Below, we describe in detail its operation.

We remark that each observation in input of P-MultiReg is partially labeled, since the feature in $(0, 0)$ is labeled as $\mathcal{A}_k$, and all other features are unlabeled. In the first step, P-MultiReg performs $|C_i^t|$ binary registrations (see Figure 17) between Z_i^t and every set in $\{Z_j^t\}_{j \in C_i^t}$. Our implementation of binary registration is inspired to the algorithm presented in [54]. The intermediate results obtained from

Algorithm 1: P-MultiReg

```
input  : observations $Z_i^t$, $\{Z_j^t\}_{j \in C_i^t}$, beliefs $\overline{\mathrm{bel}}\{x_j\}_{j \in C_i^t}$
output: estimate relative poses $\{\hat{x}_j^t\}_{j \in C_i^t}$
for $i \leftarrow 1$ to $|C_i^t|$ do
    perform binary registrations among $Z_i^t$ and all the $Z_j^t, j \in C_i^t$, not registered yet;
    select a maximal subset of irreconcilable solutions;
    select the partial solutions whose metric (11) is above a certain threshold;
    foreach selected solution $S$ do
        expand the aggregated set of feature with $S$;
        tune all the already estimated relative poses taking into account new correspondences;
        create a new branch of the algorithm if more than one solution;
```

these registrations are: (1) a group of aggregate sets of features (now with two labeled features) obtained from each binary registration (2) a group of (temporary) relative poses, one for each aggregate set. Then, the algorithm prunes the results, discarding those which are either redundant or do not fit adequately the corresponding current belief in $\{\overline{\mathrm{bel}}(x_j^t)\}_{j \in C_i^t}$ according to the metric function

$$\int p(\hat{x}_j^t | x_j^t) \overline{\mathrm{bel}}(x_j^t) dx_j^t \tag{11}$$

and a suitable threshold. In the above function, $p(\hat{x}_j^t | x_j^t)$ is the probability to measurement $\hat{x}_j^t$ given that the robot is in x_j^t, as computed from the perception model of the robot detector.

Non-redundant solutions arise only when either two different robots are associated to the same feature in two different solutions, or two different features are associated to the same robot. We will call irreconcilable this type of solutions. See Figure 18 for an example of irreconcilable solutions.

In a normal situation, when the uncertainty is small, at the end of this first iteration (binary registrations plus pruning), a single aggregate set of features Z^t_{ik} ($k \in C^t_i$) survives, with the associated temporary estimate of $\hat{x}^t_k$. The algorithm then performs a second iteration consisting of $|C^t_i| - 1$ binary registrations between Z^t_{ik} and every set in $\{Z^t_j\}_{j \in C^t_i \setminus \{k\}}$, followed by a pruning of their outputs. The intermediate result is an aggregate set of features Z^t_{ikl} ($l \in C^t_i \setminus \{k\}$), now with three labeled features, and two temporary pose estimates $\hat{x}^t_k$ and $\hat{x}^t_l$. Note that $\hat{x}^t_k$ is updated to account for new correspondences between features of Z^t_k and Z^t_l in the aggregate set Z^t_{ikl}. The algorithm then repeats the basic iteration until no feature sets from $\{Z^t_j\}_{j \in C^t_i}$ are left, producing in the end a pose estimate $\hat{x}^t_j$ for each $j \in C^t_i$ with which registration was successful (the registration had at least a solution) and sufficiently consistent with the current belief.

Now assume that at the end of the h-th iteration the algorithm selects two partial solutions. Then it branches, continuing with the following $|C^t_i| - h$ iterations in two parallel executions, each of them assuming a different partial solution for the h-th iteration. The two branches will lead to two different solutions, and a potentially different estimate $\hat{x}^t_j$ for each $j \in C^t_i$ for each robot j whose observation is registered after the h-th step. In a general execution, the algorithm creates a number of different branches, each of them generating a different configuration for the whole system. Among these configurations, another step of pruning is performed following the above criteria. The selected solutions are the solution of the algorithm. Thus, it will produce a set of estimates of the possible poses for each robot j such that $j \in C^t_i$.

Now, assume that all the binary registrations at the h-th iteration in a branch of the algorithm fail (i.e.: they find no solution). In this case,

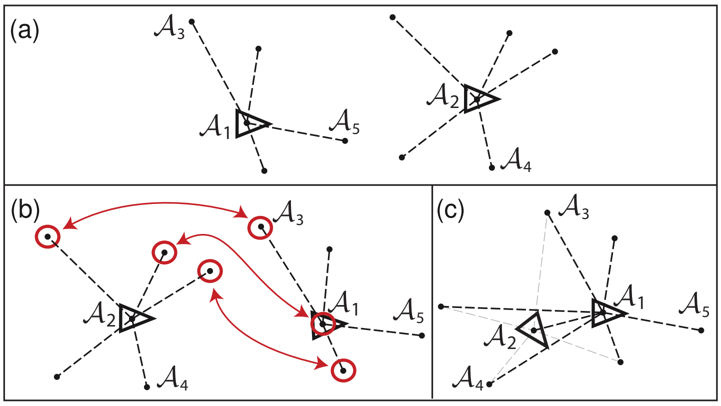


Figure 17 An example of a binary registration between partially labeled sets of features: (a) initial feature sets Z_1 and Z_2 (b) proposed association (involved features are circled in red) (c) aggregate set Z_{12} and estimated relative pose between $\mathcal{A}_1$ and $\mathcal{A}_2$ (the displacement between the two triangles).

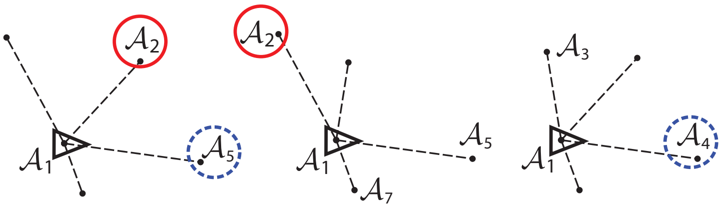


Figure 18 Three non-redundant (irreconcilable) intermediate results (note that the feature sets come from different binary registrations and therefore do not coincide). The first and the second are irreconcilable because $\mathcal{A}_2$ is assigned to different features, while the first and the third are irreconcilable because the same feature is assigned to different robots ($\mathcal{A}_4$ and $\mathcal{A}_5$).

the partial solution at the previous step can be considered as general solution found by that branch, and will be considered among the others in the final pruning with all the other solutions of the other branches. This is particularly important when two robots do not observe the same scene but communicate, that is allowed by the assumption that $D_p \subset D_c$.

To gain a deeper understanding of how P-MultiReg works, consider now the more complicated situation of Figure 19. Here, the robots are arranged in a formation close to be rotational symmetric (Figure 19a),

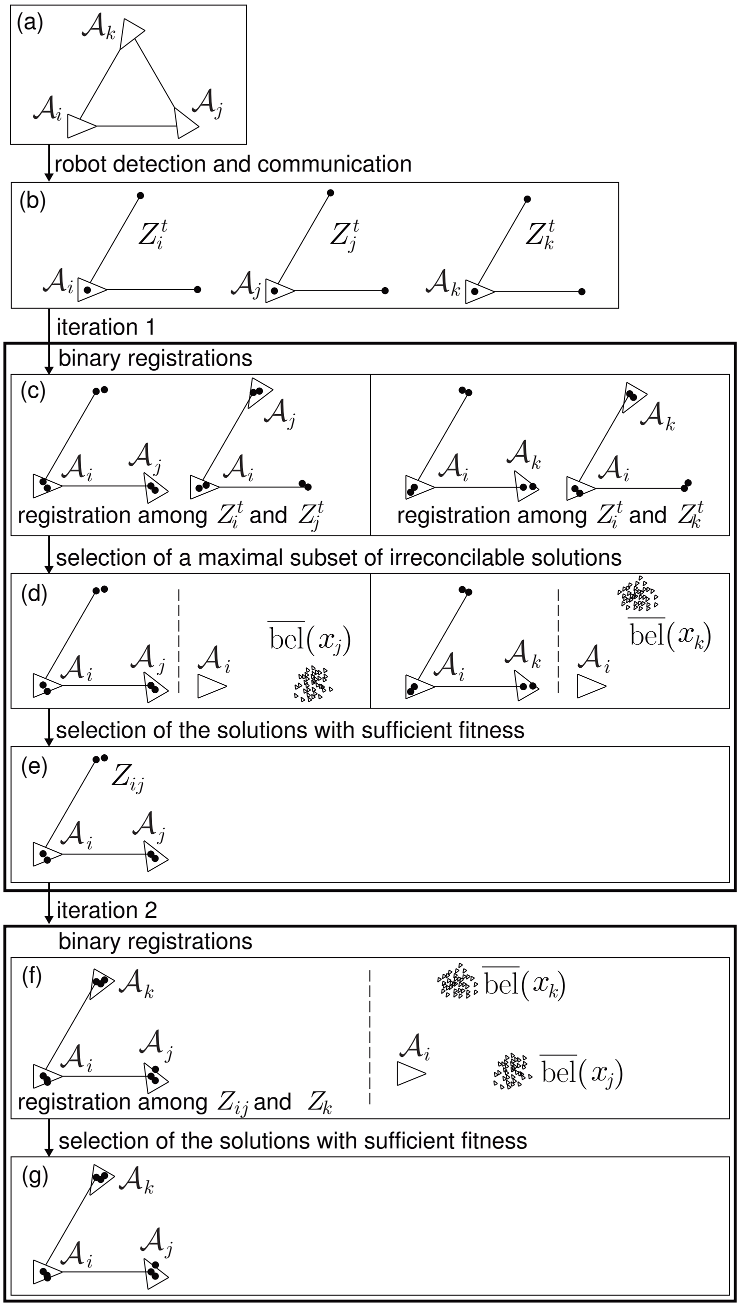

(a)
$\mathcal{A}_k$
$\mathcal{A}_i$
$\mathcal{A}_j$
robot detection and communication
(b)
Z_i^t
Z_j^t
Z_k^t
$\mathcal{A}_i$
$\mathcal{A}_j$
$\mathcal{A}_k$
iteration 1
binary registrations
(c)
registration among Z_i^t and Z_j^t
registration among Z_i^t and Z_k^t
selection of a maximal subset of irreconcilable solutions
(d)
$\overline{\mathrm{bel}}(x_j)$
$\overline{\mathrm{bel}}(x_k)$
selection of the solutions with sufficient fitness
(e)
Z_{ij}
iteration 2
binary registrations
(f)
$\overline{\mathrm{bel}}(x_k)$
$\overline{\mathrm{bel}}(x_j)$
registration among Z_{ij} and Z_k
selection of the solutions with sufficient fitness
(g)

Figure 19 Execution of P-MultiReg in a simple ambiguous situation: (a) actual configuration (b) initial feature sets with the addition of the labeled features at the origin (c) results of the binary registrations between Z_i and Z_j, Z_k, respectively (d) selection of a maximal subset of irreconcilable solutions (e) selection of the solutions with sufficient fitness w.r.t. the belief (f) result of the binary registration between Z_{ij} and Z_k (g) final result.

so that the system configuration is ambiguous from the registration viewpoint, as explained in Section 4.1. The objective of the algorithm is to register the directly perceived feature set Z_i^t with the communicated feature sets $\{Z_j^t, Z_k^t\}$ (Figure 19b). At the start, P-MultiReg performs $|C_i^t| = 2$ binary registrations between Z_i^t and Z_j^t, Z_k^t, respectively (Figure 19c), obtaining a maximum subset with two irreconcilable solutions (Figure 19d). These are rated on the basis of their fitness with respect to the current belief according to the metric function (11); for example, assume that in this case the current belief indicates that only the solution placing $\mathcal{A}_j$ at the rightmost feature can be accepted. If both solutions pass the fitness test (for example, because the current belief on the poses of $\mathcal{A}_j$ and $\mathcal{A}_j$ is uniform), the algorithm would expand two branches, leading to two different solutions. The result of the first iteration is therefore the aggregate set of features Z_{ij}^t (Figure 19e) together with a temporary estimate for the relative pose of $\mathcal{A}_j$. In the second iteration, the algorithm performs a single binary registration between Z_{ij}^t and the remaining feature set, i.e., Z_k^t (Figure 19f), and checks the fitness of the result w.r.t. the current belief; if it is above the threshold, the final solution and the associated estimates $\hat{x}_j^t$ and $\hat{x}_k^t$ are produced (Figure 19g).

The running time of P-MultiReg, which accounts for most of the cycle time of our mutual localization system, depends on the number $|C_i^t| + 1 \leq n$ of feature sets it receives as input. In normal operation (no ambiguity, or ambiguities that can be resolved based on the belief function), P-MultiReg expands only one branch, which executes $(|C_i^t|-1)(|C_i^t|-2)/2$ binary registrations to produce a solution: moreover, each binary registration requires constant time. This leads to a worst-case complexity $O(n^2)$, while the average-case complexity can be significantly lower if the number of communicating robots is smaller than n. The complexity of P-MultiReg is exponential in n in the presence of ambiguities that cannot be resolved. In fact, for Corollary 1, the maximum number of possible solutions is $(n-1)!$, that would lead the algorithm to expand $(n-1)!$ branches. Since each branch has quadratic computational complexity, in this case the algorithm is factorial. However, it is important to remark that this worst case happens only in case of ambiguous starting configurations.

6.2 Filtering

The generic robot $\mathcal{A}_i$ maintains one particle filter for each $\mathcal{A}_j$. The use of separate beliefs $p(x_j^t)$, with $j \in C_i^{1:t}$, instead of a single joint belief $\{x_j^t\}_{j \in C_i^{1:t}}$ relies on the independence assumption, i.e., $p(\{x_j^t\}_{j \in C_i^{1:t}}) = \prod_{j \in C_i^{1:t}} p(x_j^t)$. The assumption is true in a pure localization scenario, while in certain situations, e.g., distributed control, it is an acceptable approximation. However, maintaining $p(\{x_j^t\}_{j \in C_i^{1:t}})$ is not feasible from a computational point of view, since it would require a number of particles growing exponentially with the number of robots.

At time t, the j-th filter ($j \in C_i^{1:t}$) receives as inputs the motion displacement $\bar{u}_i$ of $\mathcal{A}_i$ plus, for each $j \in C_i^t$:

1. the motion displacements $\bar{u}_j^{1:t}$ (sent by $\mathcal{A}_j$);
2. the set of relative pose estimates $\{\hat{x}_j^t\}$ (computed by P-MultiReg).

In particular, each $\hat{x}_j^t$ is approximated as a gaussian measurement with a covariance which reflects the uncertainty in the registration steps of the algorithm.

The update rule that accounts for the motion of $\mathcal{A}_i$ is

$$p(x_j|\bar{u}_i) = N_i \int p(u'|\bar{u}_i) p(x_j \oplus u') du', \tag{12}$$

where N_i is a normalization factor and $p(u'|\bar{u})$ is the motion detector model. Equation (12) leads to the following update for the single particle:

$$x_j = x_j \ominus (\bar{u}_i \oplus n_u), \tag{13}$$

where n_u is a sample taken by $p(u'|\bar{u})$. Similarly, the update rule that accounts for the motion of $\mathcal{A}_j$ is

$$p(x_j|\bar{u}_j) = N_j \int p(u'|\bar{u}_j)p(x_j \ominus u')du', \tag{14}$$

where N_j is a normalization factor, and the update equation for the single particle is

$$x_j = x_j \oplus (\bar{u}_j \oplus n_u). \tag{15}$$

Updates due to the motion of $\mathcal{A}_i$ and $\mathcal{A}_j$ cause a translation of $p(x_j)$, while the additive noise introduces a blur. Assume now that the set of relative pose estimates $\{\hat{x}_j^t\}$ is composed by a single element. Then, using Bayes law, the measurement update is given by

$$p(x_j|\hat{x}_j^t) = Np(\hat{x}_j^t|x_j)p(x_j), \tag{16}$$

where N is another normalization factor. If the $\{\hat{x}_j^t\}$ is composed by more than one element, we assume that each of them is equally probable, so that using the theorem of total probability we can use the update equation:

$$p(x_j|\hat{x}_j^t) = N \sum_{\hat{x}_j^t \in \{\hat{x}_j^t\}} p(\hat{x}_j^t|x_j)p(x_j), \tag{17}$$

in which N in (17) is different from the one in (16), and the sensor model $p(\hat{x}_j^t|x_j)$ is a Gaussian with zero-mean and covariance computed experimentally.

Normally, $\mathcal{A}_i$ uses $\bar{v}_j^t \ominus \bar{v}_j^{t-1} = \bar{u}_j^t$ as motion measurement for the motion update of the robot $\mathcal{A}_j$. However, when $\mathcal{A}_i$ and $\mathcal{A}_j$ do not communicate for a time interval $[t_a + 1, t_b - 1]$ (e.g., due to the fact that the robots are far from each other) the motion update of $\mathcal{A}_j$ cannot be performed. When, at t_b, the communication is resumed, $\mathcal{A}_i$ uses $\bar{v}_j^{t_b} \ominus \bar{v}_j^{t_a} = \bar{u}_j^{t_a+1} \oplus \ldots \oplus \bar{u}_j^{t_b}$ as motion measurement for the motion update. This explains why the robots send $\bar{v}_j^t$ instead of $\bar{u}_j^t$.

A number of standard practical techniques have been used to improve the performance of the filter. For example, the initial prior distribution is generated using the first measurements. Moreover, at each step a small percentage of particles are re-initialized using the new measurements; this enables the localization system to deal with the *kidnapped robot* situation (see the next section). We have also reduced the frequency of the measurement update with respect to the motion update to guarantee the independence of subsequent measurements.

chapter 7

Experimental results

We present here the results produced by this method in the same experiments introduced in Section 5.2. The first, with four robots starting in a square (hence ambiguous) configuration, with three of them standing still and the fourth traveling a close path to stop in the starting point, is depicted in Figure 20 with the relative computed estimates. In the beginning the algorithm is not able to discriminate among six different formations in which the positions of $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$ are permuted, so that each particle filter receives three hypotheses for the measurement update. Then, the particles representing each robot are equally distributed among three possible configurations. When $\mathcal{A}_4$ starts moving, the symmetry is broken and P-MultiReg is able to find a unique solution. Hence, it produces a single hypothesis on the pose of each robot, and the clouds of particle separate while the estimates suddenly converge to the correct values. Then the new measurements are used to improve and update the estimates. When $\mathcal{A}_4$ reaches the starting position, (hence bringing back the configuration to be ambiguous) P-MultiReg is still able to find a single solution thanks to the use of the current prior.

Figure 21 shows the distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors in the estimates of the poses of $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$. Top-right shows the trajectories of the robots (bold lighter lines) and the estimates (thin darker lines) of the position on the XY plane. The triangles represent the starting configurations of the robots.

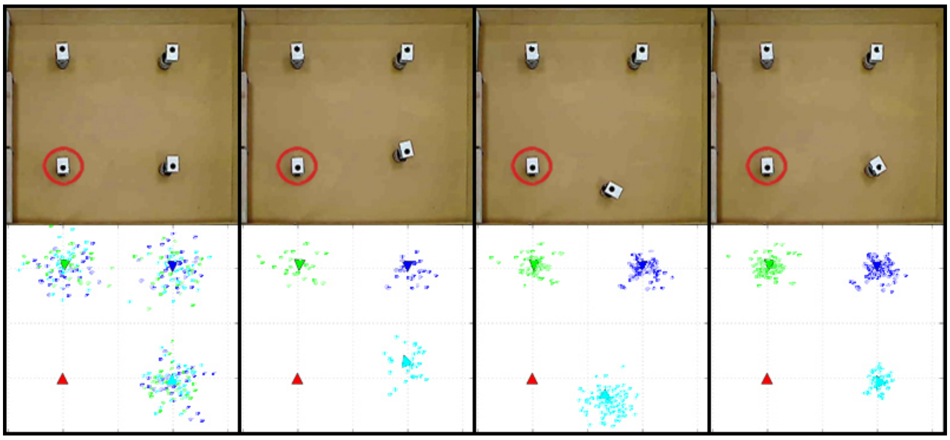


Figure 20 Top: snapshots of four instants of the first experiment; bottom: estimates computed by $\mathcal{A}_1$ (in red) in the same time instants. The estimation is conducted with 400 particles.

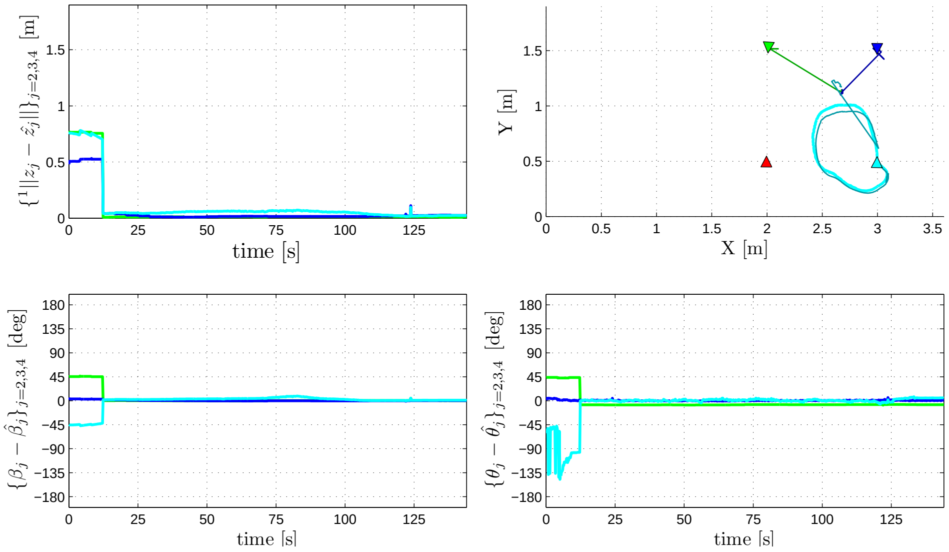


Figure 21 Distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors on the estimates produced by a filter with 400 particles as mean of the particles in the first experiment. In top-right the trajectories and the estimates are plotted on the XY plane; triangles are the starting configurations of the robots; the red robot is the robot performing the filter.

The peculiarity of the second experiment (Figures 22 and 23) is the ambiguous starting configuration of the robots complicated by the presence of two obstacles detected as features (*deceiving obstacles*) disposed as to complete a lattice. While the initial estimates suffer

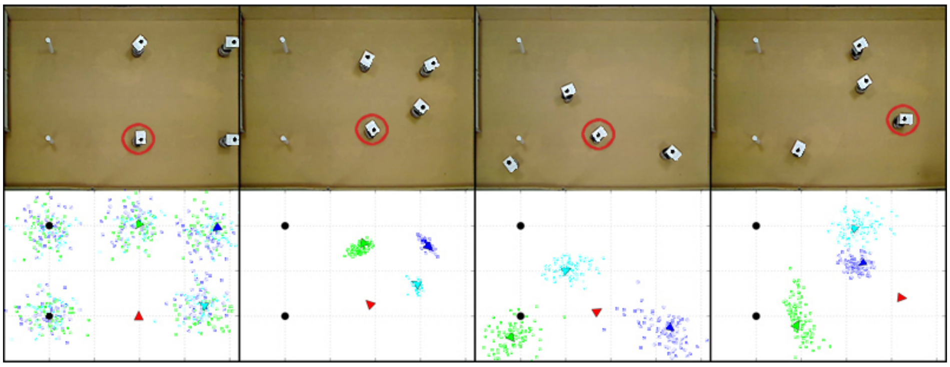


Figure 22 Top: snapshots of four instants of the second experiment; bottom: estimates computed by $\mathcal{A}_1$ (in red) in the same time instants. The estimation is conducted with 400 particles.

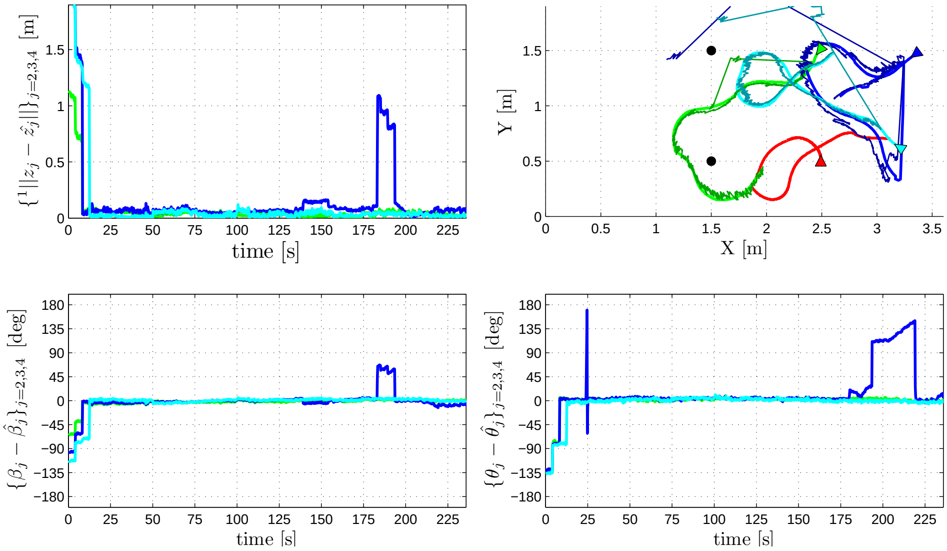


Figure 23 Distance (top-left), bearing (bottom-left) and orientation (bottom-right) errors on the estimates produced by a filter with 400 particles as mean of the particles in the second experiment. In top-right the trajectories and the estimates are plotted on the XY plane; triangles are the starting configurations of the robots; the red robot is the robot performing the filter.

from major errors, also in this case when the robots start moving the symmetry is broken and the errors converges to zero. At the end of the experiment, near $t = 180$ s, $\mathcal{A}_3$ (in blue) is kidnapped, i.e.: its pose undergoes a sudden displacement not registered by the odometry. However, thanks to the resampling on the measurements in the filter, $\mathcal{A}_1$ takes only about 10-15 s to recover a good estimate.

part IV

A comparative case study

chapter 8

Implementation

We have produced an extensive experimentation of the methods presented in this work on a team of 4 Khepera III robots moving in a 3.7 m×1.9 m flat arena. The ground truth to validate the experiments is provided by a system of three cameras providing at each time instant the pose of the robots in the arena with a linear precision of 1 cm and an angular precision of 5 deg. The simulation environment (for the development phase) is provided by Player/Stage[1], an open source project providing general purpose interfaces for robotics as well as cinematic simulations of multiple robots moving in 2D environments. The robots are either manually driven or following a pseudo-random navigation algorithm, depending on the experiment. In this Chapter we will give the details of both the hardware and software components of the multi-robot team.

8.1 Hardware

The Khepera III is a wheeled minirobot produced by K-TEAM corporation whose modular architecture allows its customization to the needs of the user. In its basic configuration, the Khepera III is a cylindric-shaped robot with 13 cm diameter and 7 cm height, equipped with motors, encoders and a microprocessor. The two 4 cm wheels are independently driven and a castor in the front of the vehicles ensures

[1] http://playerstage.sourceforge.net/

its stability, so that the cinematic is well described by the unicycle model. The encoders resolution is 0.52 deg/impulse, corresponding to about 55 impulse for a 10 mm displacement. The motion detector as defined in Chapter 2 is provided by the numeric integration of the encoders reading over a time interval of 0.1 s.

The KoreBot LE expansion offers an Intel XScale PXA-255 400MHz, 64 MBytes RAM, 32 MBytes FLASH and two USB interfaces. Running onboard an embedded systems-oriented Linux OS (kernel 2.4.19), the robot has the capabilities of a small PC and is programmable in C++. The use of the Korebot libraries allows access to the hardware (motors, sensor), while connectivity abilities are offered by a WiFi Compact Flash Ethernet card. However, a big lack of this platform is the absence in the CPU of a dedicated floating point unit. Floating point operations are in fact emulated by the CPU with 30-50 integer operations each, slowing down the execution of any algorithm. To overcome this inconvenience, we have decided to perform most of the computation on a desktop computer connected to the Khepera III through wireless connection. For this reason, we implemented a client-server architecture so that each robot manage sensors and low level control executing the instructions coming from a client performing high level decisions. The inter-robot communication is performed through UDP/IP protocol between the clients connecting to different robots.

To implement the robot detector, we have added atop each robot an Hokuyo URG-04LX range sensor, that is small, light and low-consuming with respect to other sensors of the same type. The Hokuyo is connected to the robot through USB connection and its sensory perceptions are managed by the custom driver that we have developed and sent to the client at 10 Hz. The 240 deg field of view with

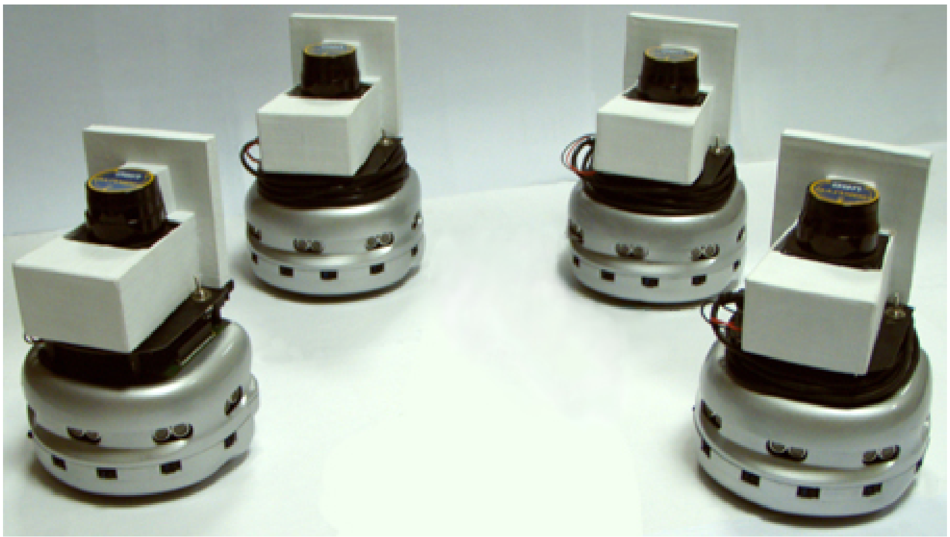


Figure 24 Khepera II robots with the Hokuyos and the 'hats' for the recognition in the scan

0.36 deg angular resolution (683 measurements for each scan) and 20-4095 mm distance range with 1 mm linear resolution provides a wide perception set. Moreover, given the angular resolution the sensor is able to detect 6 mm obstacles at a distance of 1 m, 12 mm at 2 m and 25 mm at 4 m. The single measurement have a precision of ±10 mm, but the accuracy is affected by the color of the sensed object and is calibrated to have maximum accuracy for white non-reflective surfaces. In correspondence of glossy black surfaces the sensor is likely to give null measurements, since the laser ray is completely absorbed by the obstacle.

The robot detector is implemented as a simple feature extraction algorithm on the scans looking for the small indentations produced by the small 'hat' mounted atop each robot. The underlying idea is to recognize in 10-120 mm wide objects the profile of other possible robots. However, this implies each laser to measure the distance of the lasers mounted atop the other robots. This is not directly possible, due to the aforementioned problem in sensing black glossy objects as the Hokuyo are. For this reason we covered the Hokuyos with white ‘hats’ so that the robots can more easily sense each other. Clearly, the ‘hats’ are designed so that they do not obstruct field of view of the sensors. A picture of the team of Khepera III with the laser scanners and the ‘hats’ is shown in Figure 24.

8.2 Software

The algorithm developed in this work have been implemented in MIP (Multi-robot Integration Platform), a general purpose open source project for decision-making, planning, control, and estimation in multi-robot systems. MIP is developed in C++ on Unix systems from the DIS Robotics Laboratory of the Sapienza Università di Roma. The guidelines of the project are software re-usability and easiness of embedding, that are achieved through a good level of modularity, virtualization of the classes, the use of abstracted low-level robot interfaces and the fact that a different instance of the same executable controls each robot. The last characteristic allows also a good scalability with the number of the robots and robustness to faults of single agents. Inter-robot communication, that is mandatory for multi-robot applications and for split instantiations of the robot control processes, is guaranteed by an IP-based communication module.

Despite some basic classes being implemented in MIP (such as classes representing poses and positions), more advanced functionalities are implemented wrapping the best open source libraries available on-line, such as Armadillo for linear algebra, openCV for image processing and orocos-BFL for standard bayesian filtering. The compilation files are automatically generated through the use of CMake, a cross-platform, open-source build system. The code is thought and tested on Ubuntu GNU-Linux systems, but is in principle compatible with other Unix-based operative systems, and in particular Mac OS X.

The code is fully documented by the use of Doxygen, an automatic tool that generates an on-line browser documentation using formatted comments in the code. The resulting project, MIP, consists of over 400000 lines of code and the implemented tasks ranges from exploration of unknown environments through Khepera III robots to visual hovering with simulated UAV. The web page of the project is available at http://www.dis.uniroma1.it/∼labrob/software/MIP/.

MIP is composed by the following components :

- Baselib: basic library for general purpose functionalities, e.g., pose, laser scan, IP communication, class serialization, multi-threading, file management, user option management, maps, probability density functions, etc...

- Algorithms: class collection of algorithms for robotics, e.g., geometric and sensor data processing (Voronoi diagrams, feature extraction,...), estimate (Kalman filtering, particle filtering,...), control (trajectory control, obstacle avoidance,...), etc...

- Resources: classes derived from the Resource class, providing interface modules respect to the hardware or the MIP platform facilities (motors, sensors, communication modules, keyboard, logging/tracing, 2-D/3-D display,...)

- Tasks: classes derived from the Task class, which actually perform the robot activities that must be executed in parallel, acting as glue between algorithms and resources. Example of activities are: tracking, deployment, target navigation, mutual localization, entrapment, exploration,... Each task is a finite

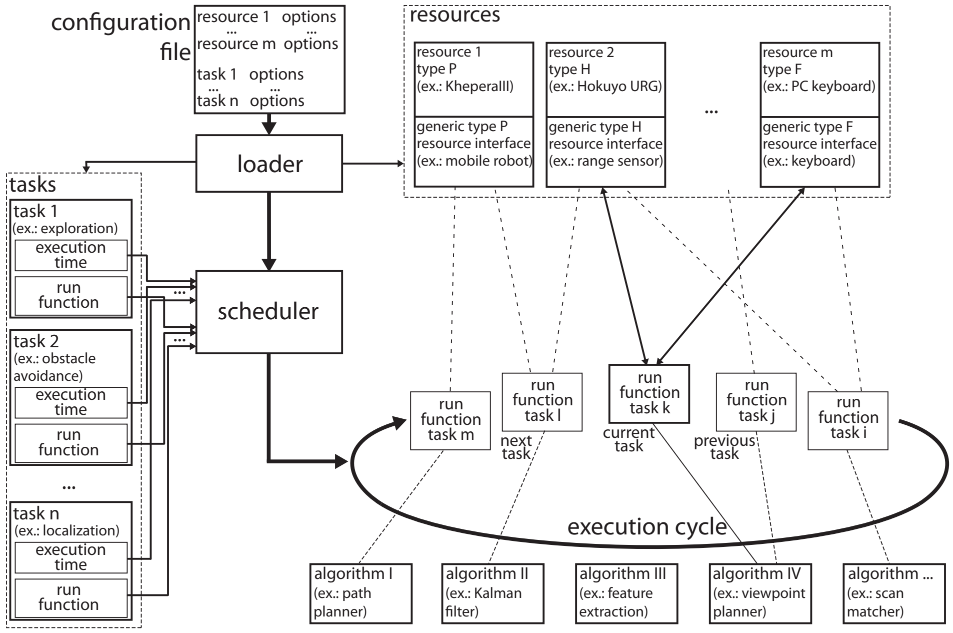


Figure 25 A block scheme of MIP.

state machine that uses the algorithms and the resources to gain its objective. A task can benefit of outputs or provide inputs from/to other concurrent scheduled tasks. The inter-task data exchange pass through the resources, for this reason the resources include also a shared memory for tasks.

- Main: main of the program. Here is created and launched the Scheduler. The scheduler cyclically executes a list of task, checking the timing correctness and managing the frequency of execution, as requested from every task. The Scheduler is not preemptive.

A block scheme of MIP is presented in Figure 25. The MIP execution begins with the Loader reading a configuration file, that is a simple text file listing all the Resources and the Tasks with the respective options that must be instantiated for the correct execution of the requested behavior. An example of configuration file is reported in Table 1. The robot instantiated in the example is a remote Khepera III, that is a Khepera III robot available on a given IP address and waiting for commands. The resource that is instantiated is derived from a generic mobile robot class. Its interface is standardized and can be replaced with any other standardized mobile robot derived class, such as a class connecting to a simulated robot. The other resources include a Keyboard for the input, a shared memory between different task (SharedMemory) to allow communication among the tasks, an IP-based communication module to allow communication among the robots, and finally a visualizer implemented in openGL to allow a visualization of the results of the estimation process.

The requested tasks are one handler of the keyboard (KeyboardHit), a task to manually drive the robot and interpret the input of the keyboard (JoystickTask), a task to automatically drive the robot using the instantaneous reading of a range finder scan mounted atop the Khepera III (LaserNavigationTask), a task to extract features on lidar scans acting as robot detector (FeatureExtraction), the task that performs the mutual localization algorithm (ParticleMutualLocalization), and a task that performs an obstacle avoidance on the trajectories defined from the two navigation tasks (Driver). Note that the JoystickTask, ParticleMutualLocalization and Driver comes with some option set.

Once the loading phase is completed, the Scheduler begins its work by cyclically executing the run function of each loaded task. The Scheduler tries as much as possible to respect the constraints in the execution times specified by each task. The execution of MIP ends when a task decide that the experiment or simulation is over, that may happen for a direct command of the user as well as for the satisfaction of certain conditions specific for each task.

```
RESOURCES
  SharedMemory
  IPCommModule
  MobileRobotRemoteKhepera
  Keyboard
  DrawingWindowGL
TASKS
  KeyboardHit
  JoystickTask
    -joystickCmdType 1
  LaserNavigationTask
  FeatureExtraction
  ParticleMutualLocalization
    -mutLocMyFrameType attached
  Driver
    -driverInput 1
```

Table 1 Example of configuration file in MIP for the execution of the mutual localization algorithm on a team of Khepera III robots.

The peculiar architecture of MIP allows its easy extension to add new features such as new algorithms and tasks, as well as new resources and types of robots. The virtualization of the interfaces between the robots and the tasks allows the porting of algorithms and methods from simulation to experiments with no effort. To conclude, we want to specify that MIP offers also the possibility of collecting data during experiments and then run the estimation tasks offline through the OfflinerTask. This last characteristic is particularly important in this work, since it allows the comparison of the different methods for the mutual localization on the same datasets.

chapter 9

Experimental results

In this section we will compare the estimates computed by the proposed methods in four different experiments. Although the plot of the distance, bearing and orientation errors are good indicators for the quality of the estimates, here we will provide the percentage of time in which the errors are below some given thresholds to quantify and summarize the differences between the two method.

The comparison will be conducted over the two experiments already introduced in the previous Chapters, and two more experiments each one of them emphasizing different aspects of the possible behaviors of the robots.

9.1 Experiment 1

Figure 26 shows the plot of the errors of distance (top), bearing (middle) and orientation (bottom) on the estimates of the pose of $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$ in the the first experiment by the FastSLAM (left column) and P-MultiReg-based (right column) algorithm. Table 2 reports the percentage of time in which the errors on the estimates of the pose of each robot are less than 0.1 m for the distance error and 7 deg for the bearing and orientation errors.

From the plots of the errors the two algorithm show a similar behavior. However, the values in Table 2 reveal that the P-MultiReg based approach produces a better estimate with respect to the FastSLAM based approach for about 5–10% of the time.

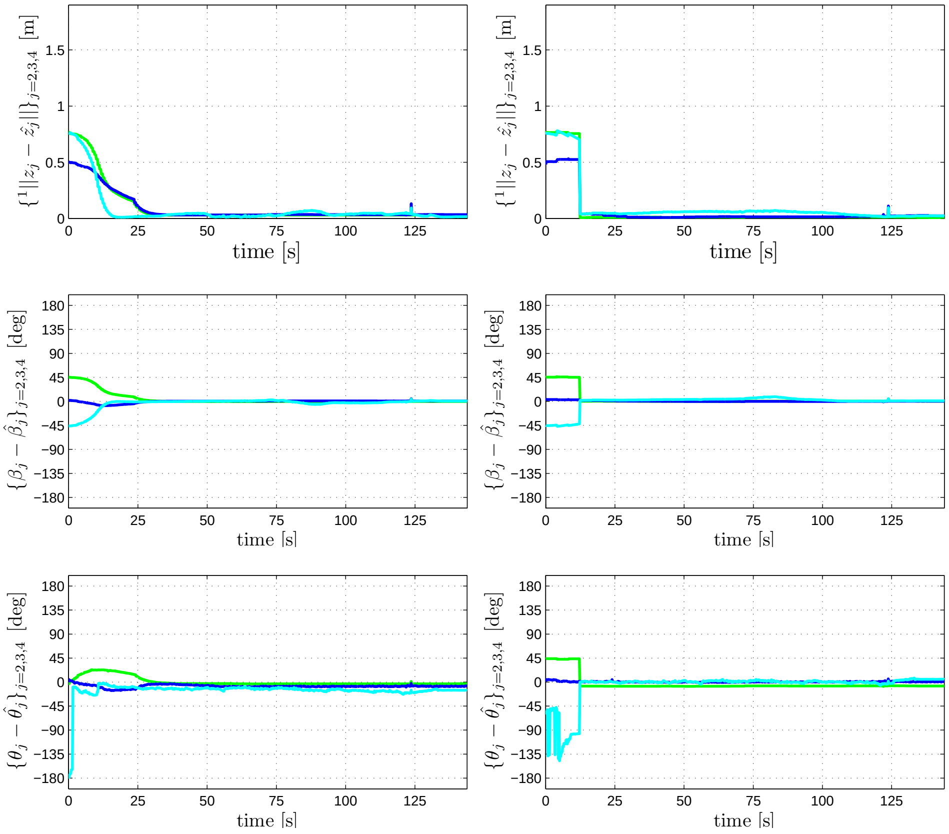


Figure 26 Distance (top), bearing (middle) and orientation (bottom) errors on the estimates produced by FastSLAM (left column) and P-MultiReg (right column) based methods in the first experiment.

	FastSLAM based			P-MultiReg based		
	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
distance	82.6%	82.1%	90.8%	91.4 %	91.3%	91.4%
bearing	83.2%	95.8%	91.0%	91.4%	100%	84.6%
orientation	83.3%	26.5%	3.1%	2.8%	100%	91.1%

Table 2 Percentage of time of the first experiment in which distance, bearing and orientation errors are less than 0.1 m, 7 deg and 7 deg respectively.

9.2 Experiment 2

Figure 27 shows the plot of the errors of distance (top), bearing (middle) and orientation (bottom) on the estimates of the pose of $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$ in the the first experiment by the FastSLAM (left column) and P-MultiReg-based (right column) algorithm. Table 3 reports the percentage of time in which the errors on the estimates of the pose of

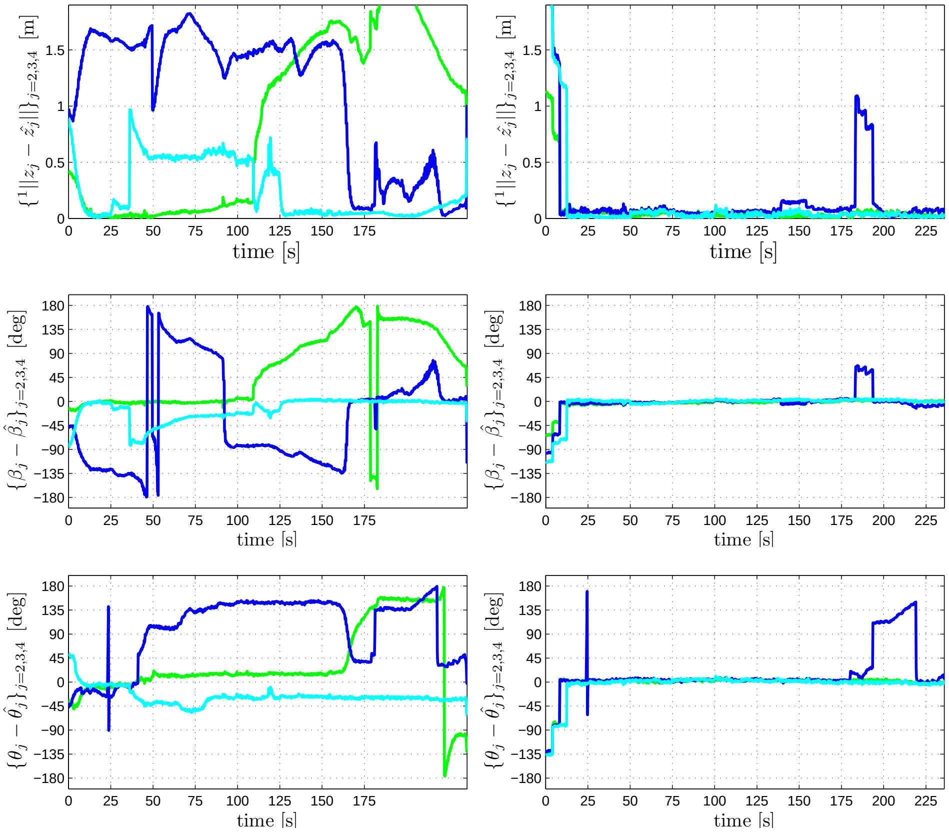


Figure 27 Distance (top), bearing (middle) and orientation (bottom) errors on the estimates produced by FastSLAM (left column) and P-MultiReg (right column) based methods in the second experiment.

	FastSLAM based			P-MultiReg based		
	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
distance	33.6%	9.3%	48.1%	96.2%	81.5%	94.6%
bearing	42.4%	13.5%	54.0%	95.7%	85.2%	94.7%
orientation	8.7%	3.0%	10.8%	95.9%	75.9%	93.6%

Table 3 Percentage of time of the second experiment in which distance, bearing and orientation errors are less than 0.1 m, 7 deg and 7 deg respectively.

each robot are less than 0.1 m for the distance error and 7 deg for the bearing and orientation errors. In this second experiment the P-MultiReg based method outperforms FastSLAM, showing a good reliability also in the numerical data of the Table 3. On the contrary, the FastSLAM approach shows all its limits.

9.3 Experiment 3

Both previous experiments begin with an ambiguous starting configuration, that is a disturbing element for both the algorithms and a possible source of error. As we have explained in Chapter 5, the initialization and the first steps are fundamental for the success of the estimation through FastSLAM. For this reason, we propose here the results of an experiment in which the starting configuration is non ambiguous, and is depicted in the first snapshot of Figure 28. The robots start moving (second snapshot) and after about 60 s one $\mathcal{A}_4$ is kidnapped (third snapshot). The final configuration of the system is depicted in the fourth snapshot.

The distance, bearing and orientation errors of the estimates computed by the two method are plotted in Figure 29. From the very first step of the experiment the P-MultiReg method is able to compute acceptable estimates. On the contrary the FastSLAM based method struggles to keep the errors bounded until the kidnapping. When the kidnapping occurs, the P-MultiReg method takes few seconds to restore a good estimate, while the already low-quality estimate provided by FastSLAM is disrupted. This experiment highlights another issue of the FastSLAM method. Since each particle of FastSLAM represents one formation/map of the environment, the error on the pose of one robot can affect the weight of the particles containing also good estimates for other robots, so that a consistent error on the pose of one robot can result in the disruption also of the other

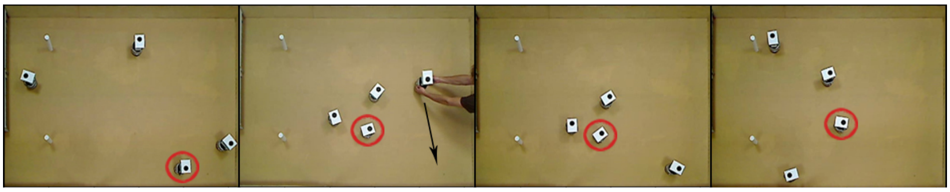


Figure 28 Snapshots of four instants of the third experiment.

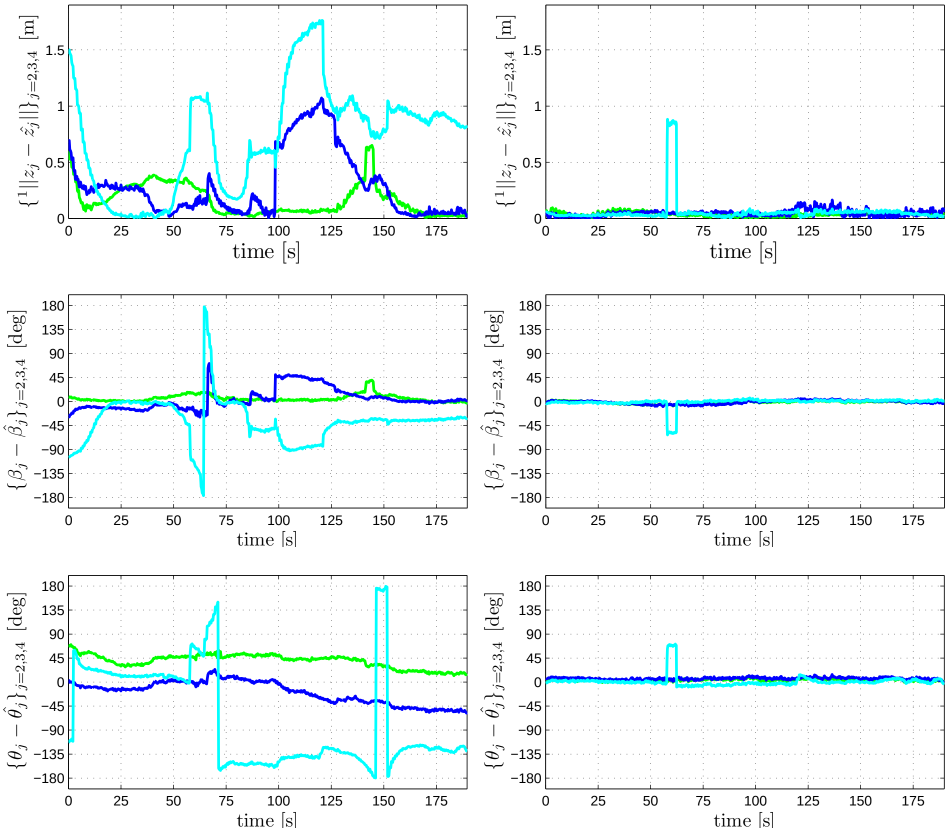


Figure 29 Distance (top), bearing (middle) and orientation (bottom) errors on the estimates produced by FastSLAM (left column) and P-MultiReg (right column) based methods in the third experiment.

estimates. On the contrary, the policy of the P-MultiReg method of maintaining one filter for each robot is more robust with respect to errors in the estimate of one component of the team. In fact, the big error in the estimate of $\mathcal{A}_4$ when it is kidnapped does not affect the estimate of the other robots.

	FastSLAM based			P-MultiReg based		
	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
distance	48.0%	34.6%	16.1%	100%	94.9%	97.6%
bearing	70.3%	39.8%	22.6%	96.8%	93.8%	97.6%
orientation	0%	29.1%	3.2%	92.9%	53.4%	90.4%

Table 4 Percentage of time of the third experiment in which distance, bearing and orientation errors are less than 0.1 m, 7 deg and 7 deg respectively.

9.4 Experiment 4

The peculiarity of the last experiment (Figures 30 and 31) is the clustered start of the team, that gradually moves from the left to the right of the arena. This could be a typical scenario for an application, and could present some traps for the localization since in a similar scenario the robots moving in the front does not gather measurements of the other robots for a long time period. This is exactly what happens between $t = 120$ s and $t = 220$ s, in which $\mathcal{A}_1$ is already in the right half of the arena while $\mathcal{A}_3$ and $\mathcal{A}_4$ (blue and cyan in the plots respectively) are going back in the left half of the arena.

In the P-MultiReg method, this results in a long time with no results from P-MultiReg including those robots (i.e.: results from P-MultiReg includes only $\mathcal{A}_2$, green in the plots), so that the particles filters in charge of the estimates of the poses of $\mathcal{A}_3$ and $\mathcal{A}_4$ work using only the odometry for the time update. Consequently, the errors increase with time. When $\mathcal{A}_3$ and $\mathcal{A}_4$ go back closer to $\mathcal{A}_1$, they start to see more features in common with it, and P-MultiReg is able to find some hypotheses on the poses of the two robot to feed the filter. This happens after $t = 220$ s, when the errors start to converge back to zero.

The same issue is instead disruptive for the estimates produced by the FastSLAM method. In fact, having only measurements of $\mathcal{A}_2$, the algorithm starts to select particles with the correct pose for $\mathcal{A}_2$, modifying also the poses of the non-sensed robots without any criterion. This effect can be seen also by the percentage provided in Table 5 in comparison with the percentage provided by the Tables

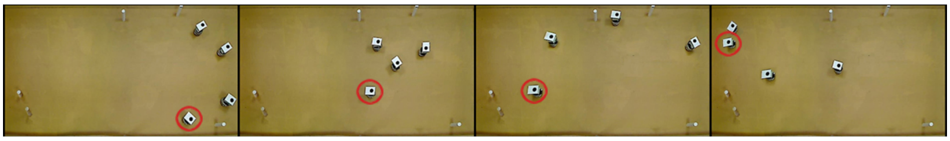


Figure 30 Snapshots of four instants of the fourth experiment.

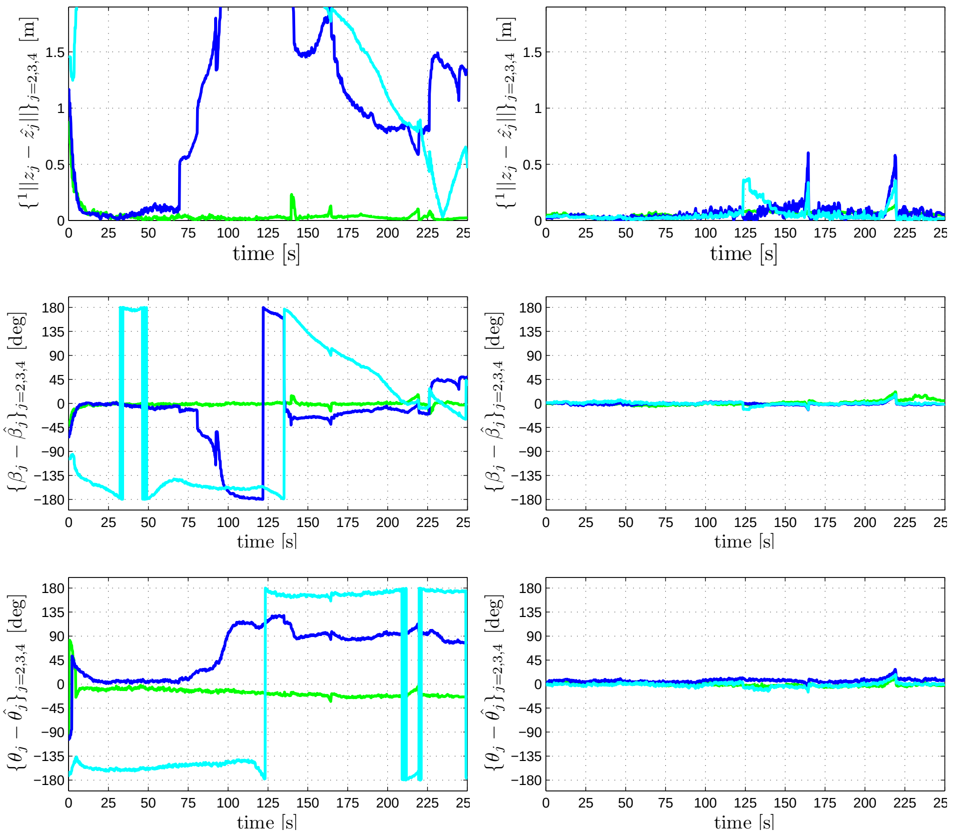


Figure 31 Distance (top), bearing (middle) and orientation (bottom) errors on the estimates produced by FastSLAM (left column) and P-MultiReg (right column) based methods in the fourth experiment.

of the other experiments. In the FastSLAM columns, an increase of quality of the best estimate ($\mathcal{A}_2$ in this experiment and in the previous, $\mathcal{A}_4$ in the second) correspond to a decrease in the quality of the second best estimate ($\mathcal{A}_3$ in this experiment and in the previous, $\mathcal{A}_2$ in the second).

	FastSLAM based			P-MultiReg based		
	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
distance	94.5%	19.5%	1.3%	98.4%	84.3%	88.4%
bearing	92.2%	21.3%	8.1%	89.8%	97.8%	95.3%
orientation	4.6%	18.9%	0%	97.9%	62.5%	91.3%

Table 5 Percentage of time of the fourth experiment in which distance, bearing and orientation errors are less than 0.1 m, 7 deg and 7 deg respectively.

part V

Extensions and future work

chapter 10

Bearing-only extension

As already stated in this thesis, the recent trend in robotics pushes for a growing use of cameras as exteroceptive sensors because they are cheap and high-information-gathering. However, being non-depth sensors they are not compatible with the relative position measurement model that we have assumed until now (range-measuring capabilities are typical of more complex sensors as Kinect or stereo cameras). This consideration suggests us an extension of our multiple registration method to the case of bearing only measurements to allow its wider application. The first step is to reformulate the problem with the new sensor model. We remark that the use of cameras for the detection of other robots does not automatically solve the problem of the identification. In Chapter 3 we have given some example of situation in which the anonymity of the measurements is mandatory even if the exteroceptive sensors are cameras.

10.1 Problem Formulation

We refer to Chapter 2 for all the assumptions taken on the robots frames, movement and communication. The only difference concerns the robot detector. In this chapter we assume that each $\mathcal{A}_i$ carries as robot detector a sensor device that detects other robots within the perception set D_p and returns a measurement of their *bearing angles* β_j with respect to $\mathcal{F}_i$, without the associated identity. As for the case of position measurements, this detector is prone to false positives

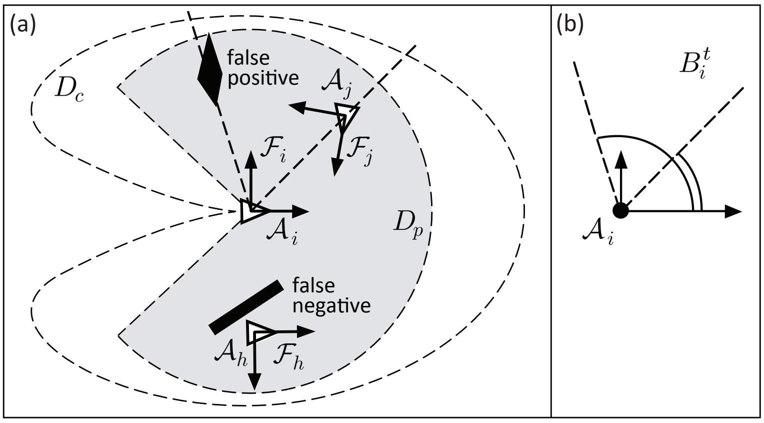


Figure 32 Bearing only robot detection. (a) A group of robots (white triangles) with the associated moving frames. Also shown are the perception and communications sets for robot $\mathcal{A}_i$; note how in this case the obstacles (black polygons) create one false positive and one false negative. (b) As a consequence, the feature set B_i^t includes two anonymous bearing measurements.

(detected objects that look like robots) and false negatives (undetected robots in D_p, e.g., due to occlusions). We will denote by B_i^t the set of bearing features detected by $\mathcal{A}_i$ at time t. An example of bearing only robot detection is given in Figure 32.

In the following, we will use the symbol ξ_j to identify the *relative bearing-orientation* of $\mathcal{A}_j$ with respect to $\mathcal{A}_i$, i.e., the 2-vector containing the bearing angle and orientation of $\mathcal{F}_j$ expressed in $\mathcal{F}_i$. Note that ξ_j can be considered as a partial representation of x_j without the *scale* information.

In a probabilistic framework, solving the RML problem with anonymous bearing measurements requires $\mathcal{A}_i$ to compute its current belief about the relative poses of all the robots with which $\mathcal{A}_i$ has communicated, on the basis of the odometry and bearing measurements gathered directly by $\mathcal{A}_i$ or obtained via communication with other robots:

Problem 5 (Probabilistic RML with anonymous bearing measure-

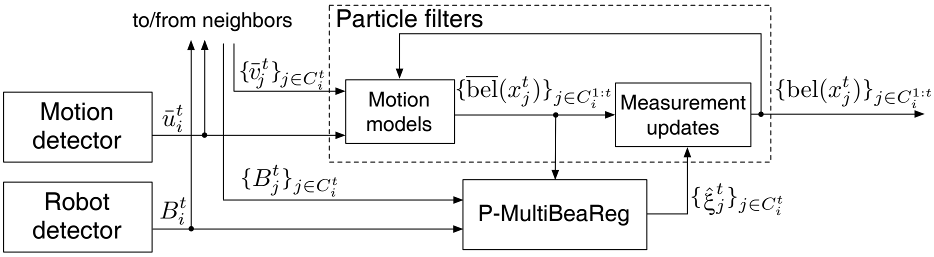

to/from neighbors
Particle filters
$\{\bar{v}_j^t\}_{j\in C_i^t}$
Motion detector
$\bar{u}_i^t$
Motion models
$\{\overline{\text{bel}}(x_j^t)\}_{j\in C_i^{1:t}}$
Measurement updates
$\{\text{bel}(x_j^t)\}_{j\in C_i^{1:t}}$
Robot detector
B_i^t
$\{B_j^t\}_{j\in C_i^t}$
P-MultiBeaReg
$\{\hat{\xi}_j^t\}_{j\in C_i^t}$

Figure 33 Scheme of the mutual localization system that runs on $\mathcal{A}_i$.

ments). *For* $t = 1, 2, \ldots$ *and* $j \in C_i^{1:t}$*, compute the following* belief

$$\text{bel}(x_j^t) = p(x_j^t | \bar{u}_i^{1:t}, B_i^{1:t}, \{\bar{v}_j^\tau, B_j^\tau\}_{j\in C_i^\tau, \tau=1,\ldots,t})$$

10.2 Proposed Approach

The mutual localization system that we propose for the solution of Problem 5 is similar to that in Chapter 6, with the appropriate modifications to account for the availability of bearing (rather than position) measurements. As shown in Figure 33, $\mathcal{A}_i$ first applies the multiple registration algorithm P-MultiBeaReg to compute the most 'likely' relative bearing-orientation of each robot belonging to C_i^t, based on the sets of features B_i^t and $\{B_j^t\}_{j\in C_i^t}$, as well as on the current beliefs about $\{x_j^t\}_{j\in C_i^t}$. The relative bearing-orientations thus obtained, together with the measurements from the motion detector $\bar{u}_i^t$ and $\{\bar{v}_j^t\}_{j\in C_i^t}$, are used by particle filters to update the belief about the pose of each robot in $C_i^{1:t}$.

The extension of our multiple registration based method is then non-trivial, since it requires a completely new multiple registration algorithm and minor adaptations also in the design of the particle filters. The latter because P-MultiBeaReg will provide only relative bearing-orientation of robots belonging to C_i^t, as opposed to the full relative poses provided by P-MultiReg.

Algorithm 2: P-MultiBeaReg

```
   input  : feature sets $B_i^t$, $\{B_j^t\}_{j \in C_i^t}$, beliefs $\overline{\mathrm{bel}}\{x_j\}_{j \in C_i^t}$
   output: relative bearing-orientation estimates $\{\hat{\xi}_j^t\}_{j \in C_i^t}$
1  Identify triangles from the feature sets;
2  Rate triangles according to their number of 3-intersections and
   collect those above a certain threshold in a set $\mathcal{T}$;
3  Extract from $\mathcal{T}$ a maximal subset $\mathcal{T}_{\mathrm{irr}}$ of irreconcilable triangles
   containing $\mathcal{A}_i$;
4  Define *partial solution* each triangle in $\mathcal{T}_{\mathrm{irr}}$ whose metric (18) is above
   a certain threshold;
5  **foreach** *partial solution* $S$ **do**
6      Expand $S$ with each triangle $T_m \in \mathcal{T}$ such that such that $S$ and
       $T_m$ have a common side, and $T_m \notin S$;
7      For each new partial solution, compute the number of
       3-intersections and select the solutions whose rating is above a
       certain threshold;
8      Extract a maximal subset of irreconcilable solutions;
9      Prune solutions whose metric (18) is below a certain threshold;
10     **if** *no new partial solution* **then** end branch;
11     **else goto** 5
```

10.3 Probabilistic Multiple Bearing Registration

P-MultiBeaReg is a probabilistic multiple registration algorithm run by $\mathcal{A}_i$ at each time instant t to feed the measurement update of the particle filters (see Figure 33). In general, *registration* is the process of computing the relative pose between two or more different viewpoints of the same scene. In our case, since the 'scene' consists only of bearing angles, the scale of the relative pose cannot be recovered. In particular, given the sets of features B_i^t, $\{B_j^t\}_{j \in C_i^t}$ and the current beliefs $\{\overline{\mathrm{bel}}(x_j^t)\}_{j \in C_i^t}$ computed in the particle filters through the motion model of the robots, P-MultiBeaReg derives an estimate $\hat{\xi}_j^t$ of the relative bearing-orientation of $\mathcal{A}_j^t$, $j \in C_i^t$, with respect to $\mathcal{A}_i$. A pseudo-code description of P-MultiBeaReg is given in Algorithm 2. The basic steps are illustrated in Figure 34.

Consider the configuration of the multi-robot system shown in Figure 34a, with the corresponding feature sets in Figure 34b. Note that each pair of features (bearing angles) in the same feature set can be equivalently represented by their difference angle. Now take a triplet of robots that 'see' each other, e.g., $\mathcal{A}_i$, $\mathcal{A}_j$, $\mathcal{A}_k$, and make $\mathcal{A}_h$ 'disappear' for a moment, so that each robot in the triplet sees only two features, or equivalently one difference angle; since the triplet defines a triangle, the sum of the three difference angles must be π. The algorithm exploits this basic observation by scanning all the possible triplets of feature sets and looking for triplets of difference angles (one from each feature set) whose sum is π, with a certain tolerance. Each of these triplets defines a triangle; more precisely, it defines a class of equivalence, because the triangle is defined only up to a scaling factor. Note that a triangle includes the identity of the robots at its vertices.

When three robots forming a triangle see a fourth robot (e.g., $\mathcal{A}_h$ in Figure 34a), their sets of features include three rays (one from each feature set) that intersect in a single point (we call this a *3-intersection*). Based on this idea, the algorithm rates all the triangles by counting their 3-intersections, and discards those below a certain threshold. A simple 2-intersection does not provide any useful information, since two non-parallel rays will always intersect at a point. From the remaining set $\mathcal{T}$, one extracts a maximal subset $\mathcal{T}_{\mathrm{irr}}$ of irreconcilable triangles containing $\mathcal{A}_i$ (two triangles are *irreconcilable* if they associate the same robot to different features of the same set, or two different robots to the same feature).

The results of this process of triangle finding and rating are illustrated in Figure 34c–f. In particular, Figure 34c shows all the triangles having one 3-intersection and containing $\mathcal{A}_i$; Figure 34e shows all the triangles having one 3-intersection but not containing $\mathcal{A}_i$;

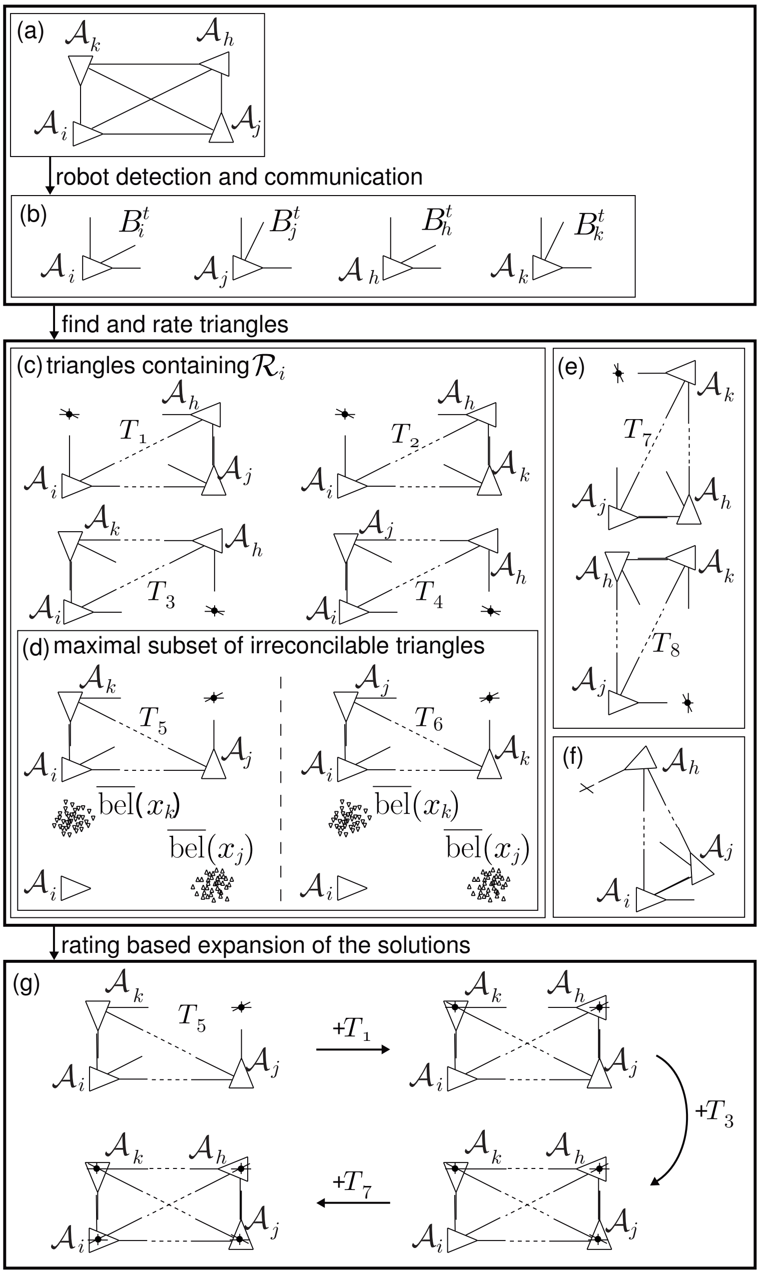

(a)
$\mathcal{A}_k$
$\mathcal{A}_h$
$\mathcal{A}_i$
$\mathcal{A}_j$
robot detection and communication
(b)
B_i^t
B_j^t
B_h^t
B_k^t
find and rate triangles
(c) triangles containing $\mathcal{R}_i$
T_1
T_2
T_3
T_4
(d) maximal subset of irreconcilable triangles
T_5
T_6
$\overline{\mathrm{bel}}(x_k)$
$\overline{\mathrm{bel}}(x_j)$
(e)
T_7
T_8
(f)
rating based expansion of the solutions
(g)
$+T_1$
$+T_3$
$+T_7$

Figure 34 Illustration of the basic steps of P-MultiBeaReg in a simple situation: (a) actual configuration (b) feature sets (c) all the triangles having one 3-intersection and containing $\mathcal{A}_i$ (d) one choice for the maximal subset $\mathcal{T}_{\mathrm{irr}}$ and comparison with the current belief (e) all the triangles having one 3-intersection but not containing $\mathcal{A}_i$ (f) one of the triangles without 3-intersections (g) expansion of the partial solution using the remaining triangles in $\mathcal{T}$.

whereas Figure 34f depicts one of the triangles without 3-intersections. One choice for the maximal subset $\mathcal{T}_{\text{irr}}$ is shown in Figure 34d.

The next step is aimed at validating the triangles in $\mathcal{T}_{\text{irr}}$ on the basis of the current belief about the pose of the robots. To this end, we use the metric function

$$P(\hat{x}_j^t) = \int p(\hat{x}_j^t | x_j^t) \overline{\text{bel}}(x_j^t) dx_j^t. \tag{18}$$

First, the scale of each triangle is computed so as to maximize the function, and then an adaptive thresholding of these maximum values is used to keep only the triangles that better fit the belief.

Each triangle surviving the previous step is used as initialization of a *partial solution*, and originates a branch of the algorithm aimed at iteratively expanding the partial solution with the addition of other triangles (see Figure 34g).

In particular, let S be the partial solution (a collection of triangles) associated to a branch at a given step. Denote by $\mathcal{T}_S = \{T_{S1}, \ldots, T_{SM}\}$ the triangles in $\mathcal{T}$ that are not contained in S and have one common side with S. The algorithm builds M partial solutions by expanding S with T_{Si}, $i = 1, \ldots, M$. Each solution is then rated by counting its total number of 3-intersections. Note that two 3-intersections that match generate a 4-intersection, that counts as four 3-intersections. In general, an n-intersection counts as $n!/[(n-3)!3!]$ 3-intersections. A triangle vertex matching a 3-intersection counts as an additional 4-intersection; adding a triangle that is not in the solution but whose vertexes are already in the solution also counts as an additional 4-intersection. Then, the algorithm selects a subset of partial solutions whose rating is above a certain threshold, extracts from this set a maximal subset of irreconcilable elements, and rates them using the metric function (18). Only the solutions which adequately fit the current belief according to an adaptive threshold are passed as partial solutions to following step, in which each solution originates a new branch of the algorithm. Each branch is expanded in an iterative process until the associated $\mathcal{T}_S$ becomes empty. The final solutions produced by all branches are then rated to identify the most likely relative bearing-orientation estimates.

10.4 Particle Filters

The particle filters designed in Chapter 6 needs only a few adaptations to work with angular measurements. We report here the whole design for completeness. Robot $\mathcal{A}_i$ maintains one particle filter for each $\mathcal{A}_j$, $j \in C_i^{1:t}$. The inputs of the j-th filter at time t are the displacement $\bar{u}_i^t$ of $\mathcal{A}_i$, the total displacement $\bar{v}_j^t = \bar{u}_j^1 \oplus \cdots \oplus \bar{u}_j^t$ of $\mathcal{A}_j$ (sent by $\mathcal{A}_j$) and the relative bearing-orientation estimate $\hat{\xi}_j^t$ (computed by P-MultiBeaReg). The latter is used to generate a gaussian measurement model with mean value $\hat{\xi}_j^t$ and appropriate covariance. If P-MultiBeaReg generates $m > 1$ estimates (e.g., due to ambiguity), the model is given by the normalized sum of m gaussians centered at the estimates.

The update rules accounting for the motion of $\mathcal{A}_i$ and $\mathcal{A}_j$ are respectively

$$p(x_j|\bar{u}_i) = N_i \int p(u'|\bar{u}_i)p(x_j \oplus u')du'$$
$$p(x_j|\bar{u}_j) = N_j \int p(u'|\bar{u}_j)p(x_j \ominus u')du',$$

with N_i and N_j normalization factors. These lead to the following update for the single particle:

$$x_j = (x_j \ominus (\bar{u}_i \oplus n_{u_i})) \oplus (\bar{u}_j \oplus n_{u_j}),$$

where n_{u_i} and n_{u_j} are samples taken by $p(u'|u)$.

Note that if $\mathcal{A}_i$ and $\mathcal{A}_j$ do not communicate over a time interval (t_a, t_b) (e.g., due to the fact that they are far from each other) the motion update of $\mathcal{A}_j$ is not performed. At t_b, when communication is resumed, $\mathcal{A}_i$ uses $\bar{v}_j^{t_b} \ominus \bar{v}_j^{t_a}$ as displacement for the motion update. This explains why the robots send out the total displacement $\bar{v}_j^t$ rather than the last incremental displacement $\bar{u}_j^t$. The outcome of the update step are the beliefs $\{\overline{\mathrm{bel}}(x_j^t)\}$.

The main difference with respect to the filter in Chapter 6 is in the measurement update. Since P-MultiBeaReg only produces relative bearing-orientation, the measurements have a lower dimension than the state. However, the generalization of the update rule is straightforward and given by

$$p(x_j|\hat{\xi}_j) = Np(\hat{\xi}_j|x_j)p(x_j), \tag{19}$$

where N is another normalization factor. Equation (19) allows the computation of the posteriors $\{\mathrm{bel(x_j^t)}\}$ depicted in Figure 33 by using the beliefs $\{\overline{\mathrm{bel}}(x_j^t)\}$ and the relative bearing-orientation estimate given by P-MultiBeaReg.

10.5 Experimental Results

We have implemented and tested the proposed bearing only mutual localization system on our team of Khepera III robots. We have simulated the bearing only robot detector using only the bearing (not the distance) information coming from the position detector used to test the other methods.

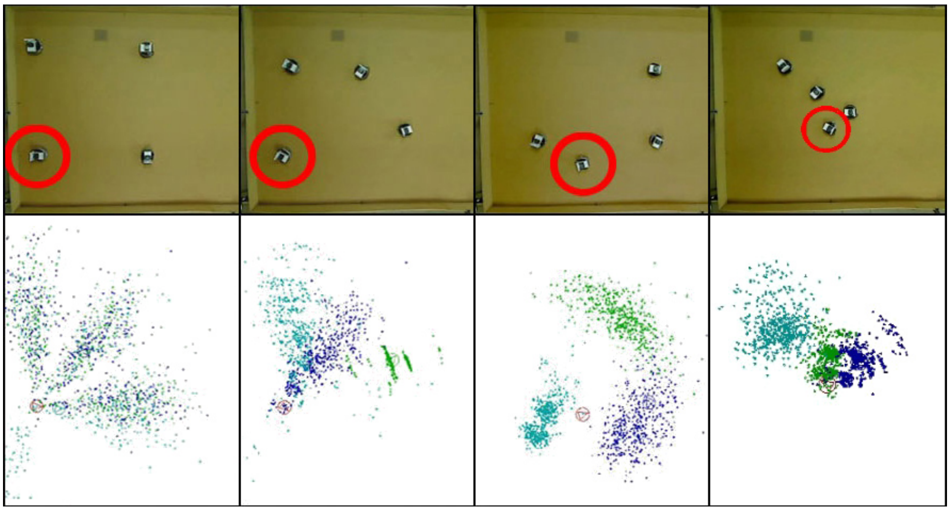


Figure 35 Experiment. Top: Snapshots of the scene. Bottom: Sample distributions computed by $\mathcal{A}_4$ (circled).

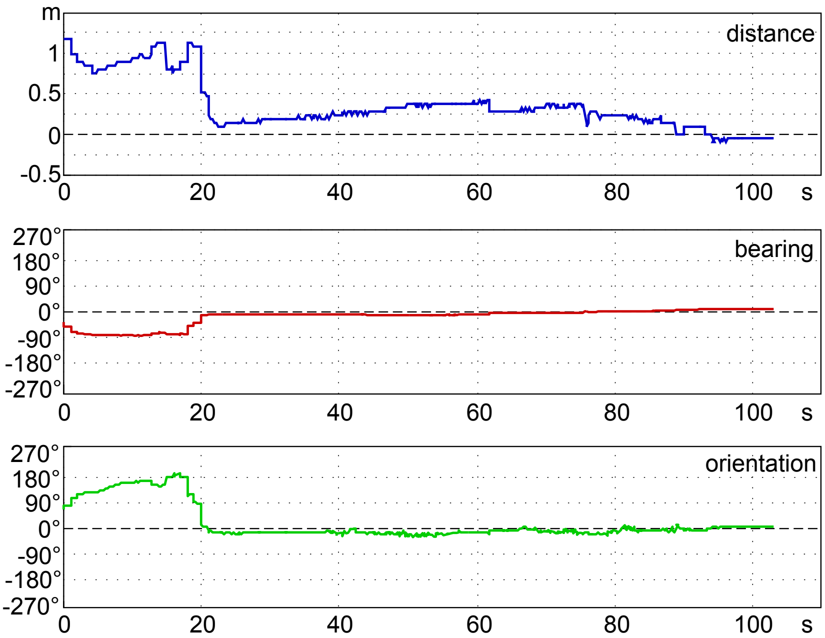


Figure 36 Experiment: Differences between the pose estimate of $\mathcal{A}_1$ computed by $\mathcal{A}_4$ using the proposed method and the method in Chapter 6.

Results from an experiment using 500 particles for each filter are shown in Figure 35. The robots start in a square, ambiguous configuration in which the registration problem has multiple solutions (first snapshot). As a consequence, the particles are very sparse at the beginning, and there is no separation between the clouds associated to the different robots. When the robots start moving, the symmetry of the formation is broken, and P-MultiBeaReg is able to compute a single solution. Hence, the particle clouds start to separate. Still, a small displacement does not allow the filters to recover an acceptable estimate of the scale, so that the particles are distributed over circular sectors (second snapshot). When the robots have moved enough for the filters to recover the scale (third and fourth snapshots), the results of P-MultiBeaReg are essentially used to update and improve the estimates.

For comparison, we have also run the localization method described in Chapter 6, whose results are obtained using the full position (distance plus bearing angle) measurements, and represent thus a sort of 'ground truth' for our new partially-informed localization system. Figure 36 plots the differences between the pose estimate of $\mathcal{A}_1$ computed by $\mathcal{A}_4$ using the two methods. There is a clear mismatch at the beginning of the experiment; however, it should be considered that the ambiguity also affects the fully informed method. As soon as the ambiguity is broken, the mismatch between the two methods becomes negligible.

Due to the small number of robots and to the limited field of view of the Hokuyo sensor, it happened frequently in the experiments that there was no triplet of robots 'seeing' each other. This is obviously a problem for P-MultiBeaReg. For example, the growth of the distance error in Figure 36 between $t = 20$ and $t = 40$ is due to this phenomenon.

In order to avoid the above difficulty, we have tested the algorithm also in simulation with a $360°$ field of view. The results are shown in Figure 37, and the errors on the pose estimate of $\mathcal{A}_1$ computed by $\mathcal{A}_4$ are plotted in Figure 38. In this case, a ground truth was obviously provided by the simulator. As expected, the convergence is faster and

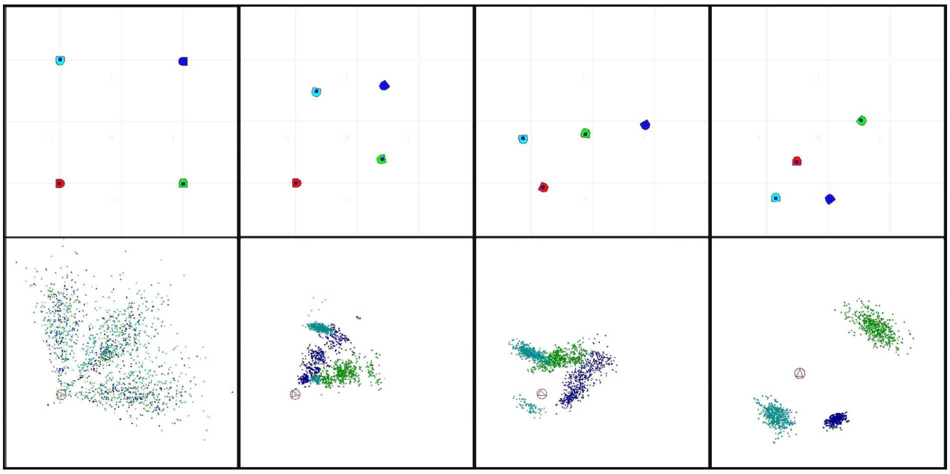


Figure 37 Simulation. Top: Snapshots of the scene. Bottom: Sample distributions computed by $\mathcal{A}_4$ (red).

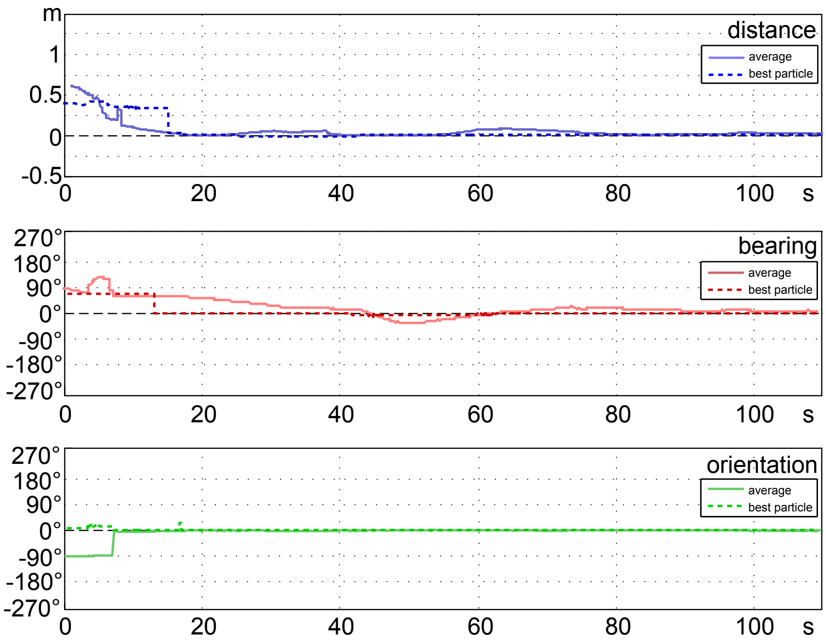


Figure 38 Simulation: Errors on the pose estimate of $\mathcal{A}_1$ computed by $\mathcal{A}_4$. Dashed lines refer to the estimate of the best particle, solid lines refer to the estimate obtained by averaging the particles.

the estimates are more precise than in the previous case. Figure 38 shows also the errors for the best particle (i.e., the particle with the largest weight), which converges quickly as expected. In particular, when the symmetry is broken, this error reduces to zero immediately.

chapter 11

3-D extension

In the last years, research in robotics has begun to focus its interest on systems working in 3-dimensional space. This is mainly due to the increasing growth of the computational speed and the miniaturization of the devices that allow the construction of vehicles with always better performances and their control in real time. This pushes robotic researchers to extend to the 'newly rediscovered' dimension all the algorithms and methods that where firstly designed for 2-dimensional scenarios.

For this reason, we spent our effort to design a 3-D extension of our multiple registration based method. The resulting algorithm will be tested on a team of quadrotors, but its potential application ranges from unmanned underwater vehicles (UUV) to wheeled vehicles on generic surfaces. However, the 3-D extension is once again non-trivial. We start reformulating the problem of mutual localization from anonymous measurements. Due to the complete change of the system model, as well as the forced introduction of IMUs instead of encoders, we restate from the beginning all the assumptions taken in this work, that will be valid only in this chapter.

A common way to obtain relative measurements is through the use of a feature tracking algorithm on the images of a stream video of a camera. This usually produces bearing only measurements. However, from a camera system is also possible to extract a rough measurement of the distance of the measured robots (i.e.: through the knowledge

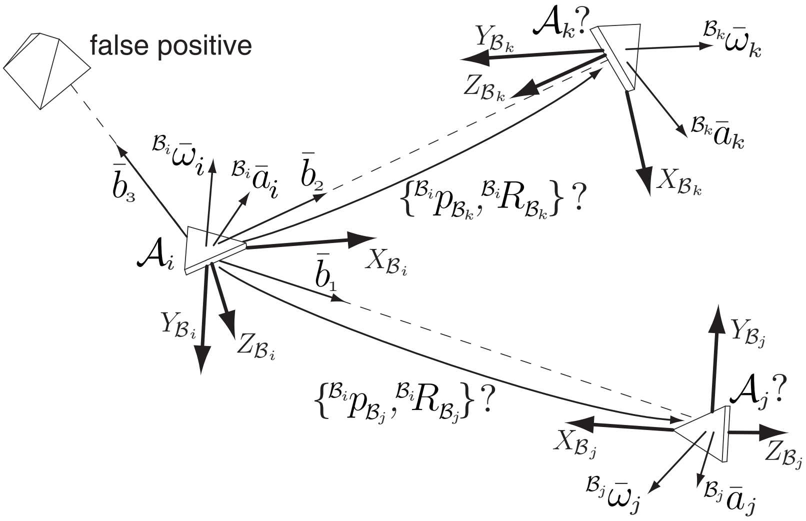

false positive
$\mathcal{A}_k$?
$Y_{\mathcal{B}_k}$
$^{\mathcal{B}_k}\bar{\omega}_k$
$Z_{\mathcal{B}_k}$
$^{\mathcal{B}_k}\bar{a}_k$
$X_{\mathcal{B}_k}$
$^{\mathcal{B}_i}\bar{\omega}_i$
$^{\mathcal{B}_i}\bar{a}_i$
$\bar{b}_3$
$\bar{b}_2$
$\{^{\mathcal{B}_i}p_{\mathcal{B}_k}, {^{\mathcal{B}_i}R_{\mathcal{B}_k}}\}$?
$\mathcal{A}_i$
$X_{\mathcal{B}_i}$
$\bar{b}_1$
$Y_{\mathcal{B}_i}$
$Z_{\mathcal{B}_i}$
$Y_{\mathcal{B}_j}$
$\mathcal{A}_j$?
$\{^{\mathcal{B}_i}p_{\mathcal{B}_j}, {^{\mathcal{B}_i}R_{\mathcal{B}_j}}\}$?
$X_{\mathcal{B}_j}$
$Z_{\mathcal{B}_j}$
$^{\mathcal{B}_j}\bar{\omega}_j$
$^{\mathcal{B}_j}\bar{a}_j$

Figure 39 Mutual localization with anonymous bearing measurements. Triangles are robots with their attached frame, acceleration, angular velocity measurements and robot detection with false positives.

of the size of the other robots). Obviously, this measurement will be affected by large noise, and could be unfit for the direct use in a geometric algorithm or in an filter. For this reason, we will consider in this section two different scenarios. The first considers the 3-D RML with anonymous bearing measurements, while the second includes also the use of distance measurements affected by large noise.

11.1 Problem formulation

We consider a system of n robots $\mathcal{A}_1, \ldots, \mathcal{A}_n$, with n unknown (hence, it may vary during the operation). Denote by $\mathcal{N} = \{1, \ldots, n\}$ the set of robot indices, and let $\mathcal{N}_i = \mathcal{N}/\{i\}$. Each robot is a rigid body in $\mathbb{R}^3$. Denote by $\mathcal{W} : \{O_\mathcal{W}, X_\mathcal{W}, Y_\mathcal{W}, Z_\mathcal{W}\}$ and $\mathcal{B}_i : \{O_{\mathcal{B}_i}, X_{\mathcal{B}_i}, Y_{\mathcal{B}_i}, Z_{\mathcal{B}_i}\}$, respectively, the inertial (world) frame and the body frame attached to the center of mass of $\mathcal{A}_i$. Body frames conform to the North-East-Down (NED) convention, as common in the aerospace field (see Figure 39). The configuration of $\mathcal{A}_i$ is represented by the position $^{\mathcal{W}}p_{\mathcal{B}_i} \in \mathbb{R}^3$ of the origin of $\mathcal{B}_i$ in $\mathcal{W}$ and the rotation matrix ${}^{\mathcal{W}}R_{\mathcal{B}_i} \in SO(3)$ between $\mathcal{W}$ and $\mathcal{B}_i$. Denote with $R_X(\cdot), R_Y(\cdot), R_Z(\cdot)$ the canonical rotation matrices about the axes x, y, z respectively. Then ${}^{\mathcal{W}}R_{\mathcal{B}_i}$ can be written as ${}^{\mathcal{W}}R_{\mathcal{B}_i} = R_Z(\psi_{\mathcal{B}_i})R_Y(\theta_{\mathcal{B}_i})R_X(\phi_{\mathcal{B}_i})$, where $\psi_{\mathcal{B}_i}, \theta_{\mathcal{B}_i}, \phi_{\mathcal{B}_i} \in \mathbb{S}^1$ are the *yaw*, *pitch*, and *roll* angles of $\mathcal{A}_i$, respectively, and $\mathbb{S}^1$ denotes the unit circle. The derivative of ${}^{\mathcal{W}}R_{\mathcal{B}_i}$ is ${}^{\mathcal{W}}\dot{R}_{\mathcal{B}_i} = [{}^{\mathcal{W}}\omega_{\mathcal{B}_i}]_\times {}^{\mathcal{W}}R_{\mathcal{B}_i}$, where

$$
{}^{\mathcal{W}}\omega_{\mathcal{B}_i} = \begin{pmatrix} {}^{\mathcal{W}}p_i \\ {}^{\mathcal{W}}q_i \\ {}^{\mathcal{W}}r_i \end{pmatrix}, \quad [{}^{\mathcal{W}}\omega_{\mathcal{B}_i}]_\times = \begin{pmatrix} 0 & -{}^{\mathcal{W}}r_i & {}^{\mathcal{W}}q_i \\ {}^{\mathcal{W}}r_i & 0 & {}^{\mathcal{W}}p_i \\ {}^{\mathcal{W}}q_i & -{}^{\mathcal{W}}p_i & 0 \end{pmatrix},
$$

and ${}^{\mathcal{W}}\omega_{\mathcal{B}_i}$ is the angular velocity in world frame.

Since we are interested in mutual localization among robots, we define the following relative quantities

$$
{}^{\mathcal{B}_i}p_{\mathcal{B}_j} = {}^{\mathcal{W}}R^T_{\mathcal{B}_i}({}^{\mathcal{W}}p_{\mathcal{B}_j} - {}^{\mathcal{W}}p_{\mathcal{B}_i}) \tag{20}
$$

$$
{}^{\mathcal{B}_i}R_{\mathcal{B}_j} = {}^{\mathcal{W}}R^T_{\mathcal{B}_i}\,{}^{\mathcal{W}}R_{\mathcal{B}_j} \tag{21}
$$

and denote by ${}^{\mathcal{B}_i}x_{\mathcal{B}_j} = \{{}^{\mathcal{B}_i}p_{\mathcal{B}_j}, {}^{\mathcal{B}_i}R_{\mathcal{B}_j}\}$ the *full relative pose* between $\mathcal{A}_i$ and $\mathcal{A}_j$.

Each robot $\mathcal{A}_i$ is equipped with a *motion detector*, such as an Inertial Measurement Unit (IMU), that provides measurements ${}^{\mathcal{B}_i}\bar{a}_i, {}^{\mathcal{B}_i}\bar{\omega}_i$ of its proper acceleration ${}^{\mathcal{B}_i}a_i$ and angular velocity ${}^{\mathcal{B}_i}\omega_i$ in body frame, given by

$$
{}^{\mathcal{B}_i}a_i = {}^{\mathcal{W}}R^T_{\mathcal{B}_i}({}^{\mathcal{W}}\ddot{p}_{\mathcal{B}_i} - g\,e_3) \tag{22}
$$

$$
{}^{\mathcal{B}_i}\omega_i = {}^{\mathcal{W}}R^T_{\mathcal{B}_i}\,{}^{\mathcal{W}}\omega_{\mathcal{B}_i} \tag{23}
$$

where g is the gravity acceleration and $e_3 = (0\ 0\ 1)^T$.

In addition, $\mathcal{A}_i$ comes with a *robot detector*, a sensor device which detects other robots and returns an anonymous measurement ${}^{\mathcal{B}_i}\bar{b}_{\mathcal{B}_j}$ of their *relative bearing*

$$^{\mathcal{B}_i}b_{\mathcal{B}_j} = {}^{\mathcal{W}}R^T_{\mathcal{B}_i} \frac{^{\mathcal{W}}p_{\mathcal{B}_j} - {}^{\mathcal{W}}p_{\mathcal{B}_i}}{\|^{\mathcal{W}}p_{\mathcal{B}_j} - {}^{\mathcal{W}}p_{\mathcal{B}_i}\|} \in \mathbb{S}^2 \tag{24}$$

that is, the unit-norm vector in $\mathbb{R}^3$ pointing toward the center of mass of $\mathcal{A}_j$, expressed in $\mathcal{B}_i$. The measurement $^{\mathcal{B}_i}\bar{b}_{\mathcal{B}_j}$ is available whenever $^{\mathcal{B}_i}p_{\mathcal{B}_j} \in D_p$, the *perception set* attached to the robot.

In addition to being subject to false positives (due to objects that look like robots) and false negatives (due to occlusions), relative bearing measurements do not contain the identity of the measured robot (see Fig. 39). Therefore, the output of the robot detector is a set $B_{\mathcal{B}_i}$ of measurements whose ordering has no relation to the robot indexing; in addition, each measurement may or not refer to an actual robot. For this reason, in the following, relative bearing measurements will be generically referred to as *features*, to emphasize that they are anonymous and, in any case, may or may not represent actual robots.

A measurement of (24) can be obtained, for example, by using a feature tracking algorithm on the images provided by a calibrated camera mounted on the robot. The choice and description of the tracking algorithm belongs to the computer vision field and is outside the scope of this paper. However, the knowledge of the size of the robots allows also the extraction from the images of rough measurements $^{\mathcal{B}_i}\bar{d}_{\mathcal{B}_j}$ of the relative distance

$$^{\mathcal{B}_i}d_{\mathcal{B}_j} = ||^{\mathcal{W}}p_{\mathcal{B}_i} - {}^{\mathcal{W}}p_{\mathcal{B}_j}||_2 \tag{25}$$

The measurements extracted with those methods, however, are usually affected by consistent noise, and could result unreliable. For this reason, we consider here two different Scenarios:

Scenario I or *bearing-only*: the output of the robot detector is a set $B_{\mathcal{B}_i}$ of bearing measurements;

Scenario II or *bearing+distance*: the output of the robot detector is a set $C_{\mathcal{B}_i}$ of bearing+distance measurements, with the uncertainty on the distance much larger than the uncertainty on the bearing.

The equipment of each robot is completed by a *communication module* that can send/receive data to/from any other robot contained in a *communication set* D_c around itself. We assume that $D_p \subset D_c$, so that if $\mathcal{A}_i$ can detect $\mathcal{A}_j$ it can also communicate with it. Each message by $\mathcal{A}_i$ is composed by: (1) the robot signature (the index i), (2) the transformed acceleration measurement $\hat{a}_i$, (3) the transformed feature set $\hat{B}_i/\hat{C}_i$, and (4) the partial estimates $\hat{\phi}_{\mathcal{B}_i}, \hat{\theta}_{\mathcal{B}_i}, \hat{\dot{\psi}}_i$. The definition of $\hat{a}_i$, $\hat{B}_i$, $\hat{C}_i$, $\hat{\phi}_{\mathcal{B}_i}$, $\hat{\theta}_{\mathcal{B}_i}$, $\hat{\dot{\psi}}_i$ is given in Sect. 11.2.

From now on, we consider the relative localization problem from the point of view of the generic robot $\mathcal{A}_i$. Denote with N_i the *neighbors* of $\mathcal{A}_i$, i.e., the set of robots from which it is receiving communication. In a probabilistic framework, the RML problem with anonymous bearing measurements requires the generic robot $\mathcal{A}_i$ to compute its belief about the relative poses of robots that are or have been its neighbors, using inertial and bearing measurements coming from its own sensory equipment or obtained via communication. In particular, using the superscripts t and $1:t$ to denote the value of a variable at time t and the history of its values at times $1, 2, \ldots, t$, we can formulate the following problem.

Problem 6. (3-D Probabilistic RML with anonymous measurements) *For* $t = 1, 2, \ldots$ *and* $j \in N_i^{1:t}$, *compute the* belief $\text{bel}({}^{\mathcal{B}_i}x_{\mathcal{B}_j}) = P({}^{\mathcal{B}_i}x^t_{\mathcal{B}_j} | {}^{\mathcal{B}_i}\bar{a}_i^{1:t}, {}^{\mathcal{B}_i}\bar{\omega}_i^{1:t}, I^{1:t}_{\mathcal{B}_i}, \{{}^{\mathcal{B}_j}\bar{a}_j^{\tau}, {}^{\mathcal{B}_j}\bar{\omega}_j^{\tau},\ I^{\tau}_{\mathcal{B}_j}\}_{j \in N_i^{\tau}, \tau=1,\ldots,t})$

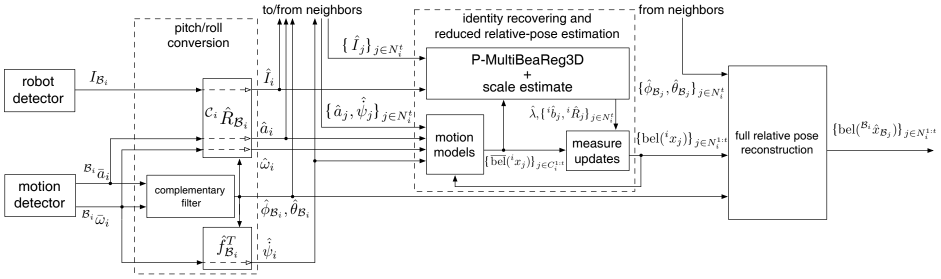

to/from neighbors
from neighbors
pitch/roll conversion
identity recovering and reduced relative-pose estimation
robot detector
$I_{\mathcal{B}_i}$
$\hat{I}_i$
$\{\hat{I}_j\}_{j\in N_i^t}$
P-MultiBeaReg3D + scale estimate
$\{\hat{\phi}_{\mathcal{B}_j}, \hat{\theta}_{\mathcal{B}_j}\}_{j\in N_i^t}$
${}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}$
$\{\hat{a}_j, \hat{\dot{\psi}}_j\}_{j\in N_i^t}$
$\hat{\lambda}, \{{}^i\hat{b}_j, {}^i\hat{R}_j\}_{j\in N_i^t}$
$\hat{a}_i$
motion models
measure updates
$\{\mathrm{bel}({}^i x_j)\}_{j\in N_i^{1:t}}$
full relative pose reconstruction
$\{\mathrm{bel}({}^{\mathcal{B}_i}\hat{x}_{\mathcal{B}_j})\}_{j\in N_i^{1:t}}$
$\{\overline{\mathrm{bel}}({}^i x_j)\}_{j\in C_i^{1:t}}$
$\hat{\omega}_i$
${}^{\mathcal{B}_i}\bar{a}_i$
motion detector
complementary filter
${}^{\mathcal{B}_i}\bar{\omega}_i$
$\hat{\phi}_{\mathcal{B}_i}, \hat{\theta}_{\mathcal{B}_i}$
$\hat{f}^T_{\mathcal{B}_i}$
$\hat{\dot{\psi}}_i$

Figure 40 Scheme of the mutual localization system that runs on $\mathcal{A}_i$.

$$\textit{with } I^{\mathcal{T}}_{\mathcal{B}_j} = \begin{cases} B^{\mathcal{T}}_{\mathcal{B}_j} & \text{in Scenario I} \\ C^{\mathcal{T}}_{\mathcal{B}_j} & \text{in Scenario II} \end{cases}$$

11.2 3-D pose estimation

For $k \in \mathcal{N}$, denote with $\mathcal{C}_i = \{O_{\mathcal{C}_i}, X_{\mathcal{C}_i}, Y_{\mathcal{C}_i}, Z_{\mathcal{C}_i}\}$ the frame having the same origin as $\mathcal{B}_i$ and such that ${}^{\mathcal{W}}R_{\mathcal{C}_i} = R_Z(\psi_{\mathcal{B}_i})$. Being ${}^{\mathcal{C}_i}R_{\mathcal{B}_i} = R_Y(\theta_{\mathcal{B}_i})R_X(\phi_{\mathcal{B}_i})$, we have

$$ {}^{\mathcal{C}_i}R_{\mathcal{B}_i} = \begin{pmatrix} c\theta_{\mathcal{B}_i} & s\phi_{\mathcal{B}_i}s\theta_{\mathcal{B}_i} & c\phi_{\mathcal{B}_i}s\theta_{\mathcal{B}_i} \\ 0 & c\phi_{\mathcal{B}_i} & -s\phi_{\mathcal{B}_i} \\ -s\theta_{\mathcal{B}_i} & s\phi_{\mathcal{B}_i}c\theta_{\mathcal{B}_i} & c\phi_{\mathcal{B}_i}c\theta_{\mathcal{B}_i} \end{pmatrix}. \tag{26}$$

The scheme of our estimation algorithm is shown in Fig. 40. We split Problem 6, i.e., the problem of estimating ${}^{\mathcal{B}_i}x_{\mathcal{B}_j}$, $j \in N_i^{1:t}$, in two subproblems.

Estimation of pitch and roll

First we let any $\mathcal{A}_i$ to independently estimate its roll $\phi_{\mathcal{B}_i}$ and pitch $\theta_{\mathcal{B}_i}$ by only using its own motion detector measurements ${}^{\mathcal{B}_i}a_i^t, {}^{\mathcal{B}_i}\omega_i^t$. This goal is achieved with a complementary filter (see [55, 56]) and we denote with $\hat{\phi}_{\mathcal{B}_i}$ and $\hat{\theta}_{\mathcal{B}_i}$ the estimates. As a consequence $\mathcal{A}_i$ can compute the estimate ${}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}$ of ${}^{\mathcal{C}_i}R_{\mathcal{B}_i}$ using (26).

Estimation of the reduced relative pose

We solve a problem which is simpler than Problem 6, and consists in retrieving the identities of the relative bearing measurements and estimating a *reduced relative pose* ${}^{i}x_j = \{{}^{i}p_j, {}^{i}R_j\}$, $j \in N_i^{1:t}$, where

$$ {}^{i}p_j = {}^{\mathcal{W}}R_{\mathcal{C}_i}^{T}({}^{\mathcal{W}}p_{\mathcal{C}_j} - {}^{\mathcal{W}}p_{\mathcal{C}_i}) \tag{27}$$

$$ {}^{i}R_j = R_Z(\psi_{\mathcal{B}_i})^T R_Z(\psi_{\mathcal{B}_j}). \tag{28}$$

Denote by ${}^{i}\hat{x}_j = \{{}^{i}\hat{p}_j, {}^{i}\hat{R}_j\}$ the corresponding estimates.

Once both subproblems are solved, it is immediate to compute an estimate ${}^{\mathcal{B}_i}\hat{x}_{\mathcal{B}_j} = \{{}^{\mathcal{B}_i}\hat{p}_{\mathcal{B}_j}, {}^{\mathcal{B}_i}\hat{R}_{\mathcal{B}_j}\}$ of the relative pose required by Problem 6 by setting ${}^{\mathcal{B}_i}\hat{p}_{\mathcal{B}_j} = {}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}{}^{T}\,{}^{i}\hat{p}_j$ and ${}^{\mathcal{B}_i}\hat{R}_{\mathcal{B}_j} = {}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}{}^{T}\,{}^{i}\hat{R}_j$.

For the estimation of the reduced relative pose ${}^{i}x_j$, rather than the motion and robot detector measurements (22–24) we use the corresponding quantities in the $\mathcal{C}_i$ frame

$$ a_i = {}^{\mathcal{W}}R_{\mathcal{C}_i}^{T}({}^{\mathcal{W}}\ddot{p}_{\mathcal{C}_i} - ge_3) \tag{29}$$

$$ \omega_i = {}^{\mathcal{W}}R_{\mathcal{C}_i}^{T}\,{}^{\mathcal{W}}\omega_{\mathcal{C}_i} \tag{30}$$

$$ {}^{i}b_j = \frac{{}^{i}p_j}{\|{}^{i}p_j\|} = {}^{\mathcal{W}}R_{\mathcal{C}}^{T}\frac{{}^{\mathcal{W}}p_{\mathcal{C}_j} - {}^{\mathcal{W}}p_{\mathcal{C}_i}}{\|{}^{\mathcal{W}}p_{\mathcal{C}_j} - {}^{\mathcal{W}}p_{\mathcal{C}_i}\|}. \tag{31}$$

Using the roll and pitch estimates from the complementary filter we have

$$ \hat{a}_i = {}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}(0 \;\; 0 \;\; {}^{\mathcal{B}_i}\bar{a}_{i_Z})^T \tag{32}$$

$$ \hat{\omega}_i = {}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}(0 \;\; 0 \;\; {}^{\mathcal{B}_i}\bar{\omega}_{i_Z})^T \tag{33}$$

$$ {}^{i}\hat{b}_j = {}^{\mathcal{C}_i}\hat{R}_{\mathcal{B}_i}{}^{\mathcal{B}_i}\bar{b}_j. \tag{34}$$

while the distance measurements i ${}^{i}\bar{d}_j$ are invariant w.r.t. rotations.

Note that $\hat{a}_i$ and $\hat{\omega}_i$ are computed using only the z component of the respective vectors, implicitly neglecting the first two components. We need to take this approximation in order to preserve the independence of the measurements and to avoid to use twice the x and y components of ${}^{\mathcal{B}_i}\bar{a}_i$ and ${}^{\mathcal{B}_i}\bar{\omega}_i$, since they have already been used to compute the estimates of roll and pitch. For example, in a typical quadrotor [56] this approximation can be safely taken assuming that the linear velocities are less than 5 m/s and the roll and pitch angles are less than 25 deg. We emphasize that the estimates of the transformed relative measurements ${}^i\hat{b}_j$ and ${}^i\bar{d}_j$ are still anonymous.

In addition, the system uses an estimate estimate $\hat{\dot{\psi}}_{\mathcal{B}_k}$ of the yaw rate, which is computed plugging the roll and pitch estimates into the formula

$$\dot{\psi}_{\mathcal{B}_i} = \begin{pmatrix} 0 & \dfrac{\sin\phi_{\mathcal{B}_i}}{\cos\theta_{\mathcal{B}_i}} & \dfrac{\cos\phi_{\mathcal{B}_i}}{\cos\theta_{\mathcal{B}_i}} \end{pmatrix} {}^{\mathcal{B}_i}\bar{\omega}_i = f_{\mathcal{B}_i}^T {}^{\mathcal{B}_i}\bar{\omega}_i, \tag{35}$$

where $f_{\mathcal{B}_i}^T =$ is the co-vector which transforms the angular velocity in body frame into the yaw rate.

This leads to the following reformulation of Problem 6.

Problem 7. *For* $t = 1, 2, \ldots$ *and* $j \in N_i^{1:t}$*, compute the* belief*:* $\mathrm{bel}({}^ix_j) = P({}^ix_j^t|\, \hat{a}_i^{1:t}, \hat{\omega}_i^{1:t}, \hat{I}_i^{1:t}, \hat{\dot{\psi}}_i^{1:t}, \{\hat{a}_j^\tau, \hat{I}_j^\tau, \hat{\dot{\psi}}_j^\tau, \}_{j\in N_i^\tau, \tau=1,\ldots,t})$ *with* $\hat{I}_j^\tau = \begin{cases} \hat{B}_j^\tau & \text{in Scenario I} \\ \hat{C}_j^\tau & \text{in Scenario II} \end{cases}$

In order to solve Problem 7 we need to recover: (1) the identities of the measurements in $\hat{I}_i$ and $\hat{I}_j$, (2) the relative orientations ${}^iR_j$, and (3) the relative distances ${}^id_j$. To this aim we use a two-step approach in analogy with the 2-D case.

First, a multiple registration algorithm (P-MultiBeaReg3D, described in Sect. 11.3) is used to retrieve the identities in $\hat{I}_i$ and the ${}^i\hat{R}_j$ matrices. Then, its output is used to feed a bank of Particle

Filters (PF) (described in Sect. 11.5), one for each $\mathcal{A}_j$, $j \in N_i^{1:t}$ to filter out the noise. We will see that in Scenario I the PFs also retrieve the scale of the formation, while in Scenario II this is done at the end of the multiple registration algorithm through the use of the distance measurements.

11.3 Multiple registration algorithm

P-MultiBeaReg3D is the probabilistic multiple registration algorithm run by $\mathcal{A}_i$ at each time instant t to feed the measurement update of the particle filters (see Fig. 40). In Scenario I, since the 'scene' consists only of sets of bearing measurements, the scale of the relative poses cannot be recovered. In particular, given the sets of features $\hat{B}_i$, $\{\hat{B}_j\}_{j \in N_i}$ and the current beliefs $\{\overline{\mathrm{bel}}({}^i x_j)\}_{j \in N_i}$ computed by the particle filters through the motion model of the robots (see Fig. 40), P-MultiBeaReg3D derives a set of guesses for the relative bearing-orientation (${}^i\hat{b}_j$ and ${}^i\hat{R}_j$) of $\mathcal{A}_j$, $j \in N_i$, w.r.t. $\mathcal{A}_i$.

In Scenario II, the algorithm is able to retrieve an estimate ${}^i\hat{d}_j$ of the distances through the ${}^i\bar{d}_j$ measurements. Thus its output for each $\mathcal{A}_j$, $j \in N_i$, is a set of guesses for the reduced poses ${}^i\hat{x}_j$. However, being the ${}^i\bar{d}_j$'s affected by consistent noise, we chose not to use them in the recover of the identities. So, the algorithm in this scenario is the same as the one developed for Scenario I, with only some differences that will be discussed in Sect. 11.4. A pseudo-code description of P-MultiBeaReg3D is given in Algorithm 3 and its basic steps are illustrated in Fig. 41.

1) *Azimuth/Zenith-distance representation:* consider the situation in Fig. 41a, where four robots are arranged in a 'square' formation with the opposite vertices at the same height and the corresponding feature sets in Fig. 41b. Each bearing measurement can be represented by an

```
Algorithm 3: P-MultiBeaReg3D
   input  : feature sets $\hat{B}_i$, $\{\hat{B}_j\}_{j \in N_i}$, beliefs $\overline{\text{bel}}\{{}^i x_j\}_{j \in N_i}$
   output: relative bearing-orientation estimates
1  Identify triangles from the feature sets;
2  Rate triangles according to their 2-intersections and collect those
   above a certain threshold in a set $\mathcal{T}$;
3  Extract from $\mathcal{T}$ a maximal subset $\mathcal{T}_{\text{irr}}$ of irreconcilable triangles
   containing $\mathcal{A}_i$;
4  Define partial solution each triangle in $\mathcal{T}_{\text{irr}}$ whose metric (37) is above
   a certain threshold;
5  foreach partial solution $S$ do
6      Expand $S$ with each triangle $T_m \in \mathcal{T}$ such that such that $S$ and
       $T_m$ have a common side, and $T_m \notin S$;
7      For each new partial solution, compute its 2-intersections and
       select solutions with rating above a threshold;
8      Select a maximal subset of irreconcilable solutions and set them
       as partial solutions for the next step;
9      Prune solutions whose metric (37) is under an adaptive threshold;
10     if no new partial solution then end branch;
11     else goto 5
```

azimuth ${}^i\alpha_j$ and zenith-distance ${}^i\zeta_j$ pair, i.e., a given bearing ${}^i b_j$ can be represented as $({}^i\alpha_j, {}^i\zeta_j) \in [0, 2\pi) \times [0, \pi)$, since

$$ {}^i b_j = (\sin {}^i\zeta_j \cos {}^i\alpha_j \;\; \sin {}^i\zeta_j \sin {}^i\alpha_j \;\; \cos {}^i\zeta_j)^T. \tag{36}$$

The projection of $\hat{B}_i$ on the XY plane of $\mathcal{C}_i$ preserves only the azimuth information. Furthermore, each pair of azimuth angles in the same feature set (i.e., belonging to the same robot) can be equivalently represented by their difference. Note that such differences representation of the azimuth angles of the feature set of a robot does not change if we choose the XY plane of a different robot, since all the XY planes of the $\mathcal{C}_i$ are parallel. Then, an azimuth angle difference represents a feasible internal angle of a planar triangle.

2) *Triangle finding:* consider now a triplet of robots that ‘see’ each

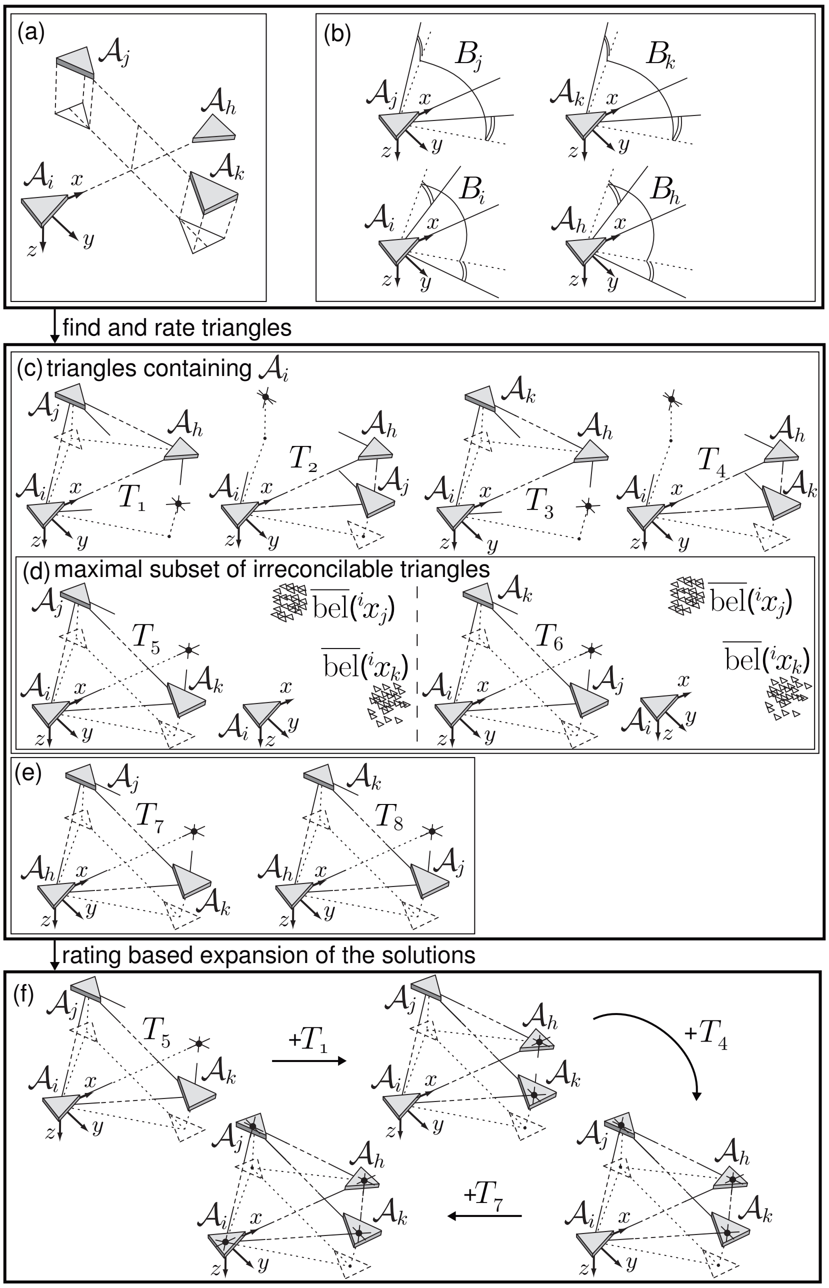


Figure 41 Execution of P-MultiBeaReg3D in an ambiguous situation: (a) actual configuration (b) initial feature sets (c) triangle found in the first step containing the owner of the algorithm and their triple intersections (d) maximal subset of irreconcilable triangles and their comparison with the current belief (e) other triangles found in the first step of the algorithm and their triple intersections (f) expansion of the solution using the remaining triangles.

other, e.g., $\mathcal{A}_i$, $\mathcal{A}_j$, $\mathcal{A}_k$, and make $\mathcal{A}_h$ 'disappear' for a moment, so that each robot in the triplet sees only two features, or equivalently one difference angle. Since the projection of a 3-D triangle on $\mathcal{C}_i$'s XY plane defines a planar triangle, the sum of the three difference angles must be π. The algorithm scans all the possible triplets coming from different feature sets and looks for triplets of difference angles (one from each feature set) whose sum is π, with a certain tolerance. Each of these triplets defines a planar triangle; more precisely, it defines a class of equivalence, because the triangle is defined only up to a scaling factor. Note that a triangle encodes also the identity of the robots at its vertices. Such triangles must satisfy also an additional condition. In fact, each azimuth angle comes with a zenith-distance angle associated. By building the triangle as explained, we are implying that a certain feature of a set is the equivalent of another feature of another set. Then, the sum of the zenith-distances of two associated bearings must be equal to π, with a certain tolerance.

3) *2-intersections rating:* when two robots in a triangle see another robot that is not the third vertex of the triangle, their feature sets will contain two intersecting rays, one for each set. We will call this a *2-intersection*. A triangle can also have *3-intersections*, when all three robots forming it see a fourth robot (e.g., $\mathcal{A}_h$ in Fig. 41a). In general, an n-intersection, that is, n intersecting rays from n different robots, accounts for $n!/2(n-2)! = n(n-1)/2$ 2-intersections. Hence, the algorithm rates all the triangles by counting their 2-intersections and collects those above a certain threshold in a set $\mathcal{T}$ (Fig. 41c–e).

4) *Irreconcilable triangles:* the algorithm extracts from $\mathcal{T}$ a maximal subset $\mathcal{T}_{\text{irr}}$ (Fig. 41d) of irreconcilable triangles containing $\mathcal{A}_i$; two triangles are said to be *irreconcilable* if they associate the same robot to different features of the same set (e.g., $\mathcal{A}_j$ in T_5 and T_6),

or different robots to the same feature (e.g., $\mathcal{A}_j$ and $\mathcal{A}_k$ in T_5 and T_6).

5) *Belief rating:* the triangles in $\mathcal{T}_{\text{irr}}$ are validated (Fig. 41d) on the basis of the current belief about the pose of the robots. To this end, we use the metric function

$$P(\{{}^i\hat{b}_j, {}^i\hat{R}_j\}) = \int p(\{{}^i\hat{b}_j, {}^i\hat{R}_j\}|{}^ix_j)\overline{\text{bel}}({}^ix_j)d^ix_j \tag{37}$$

where $\overline{\text{bel}}({}^ix_j)$ comes from the particle filters. First, the scale of each triangle is calculated so as to maximize this function; then, an adaptive thresholding of these maximum values is used to select the triangles that better fit the belief. Those triangles are collected in a set $\mathcal{T}_{best}$.

6) *Partial solutions:* each triangle of $\mathcal{T}_{best}$ is the base of a branch of the algorithm and constitutes the partial solution at the first step of its branch. Let S be the partial solution of a branch at a given step; S includes (1) a collection of triangles (2) the change of coordinates between them (3) the total number of 2-intersections. In Fig. 41f the only branch has T_5 as first partial solution.

7) *Iterative expansion:* the partial solution of each branch is iteratively expanded looking for triangles that have common edges with it (see Fig. 41f). Let $\mathcal{T}_S = \{T_m, m = 1, \dots, M\}$ be the set of triangles $T_m \in \mathcal{T}$ not yet in S having a common edge with one triangle in S. Then the algorithm builds a set of M possible partial solutions for the next step expanding S with $T_m, m = 1, \dots, M$. Each solution is then rated counting out its total number of 2-intersections. As in the case of the triangles, among the best rated partial solutions of each branch the algorithm selects a maximal subset of irreconcilable solutions. Among those, only the solutions that fit with the current belief according to equation (37) are used as partial solutions at following step, expanding a branch for each of them.

In the case of Fig. 41f, the algorithm expands a partial solution by joining to the triangle T_5 the triangles T_1, T_4, T_7 respectively at the first, second and third iteration.

The iterative process continues in each branch until $\mathcal{T}_{\mathcal{S}}$ becomes empty in that branch. In the end, each branch finds a solution, and the best of them are selected, again with the 2-intersection and belief criteria. Since each branch of the algorithm may in principle produce a different pair ${}^i\hat{b}_k, {}^i\hat{R}_k$ for each $\mathcal{A}_k$, each with its own weight, the result is a list of such pairs for the generic robot $\mathcal{A}_k$.

11.4 Scale estimate using the distances

As stated before, each branch of the algorithm finds a formation up to an unknown scaling factor λ. The knowledge of just one distance in the formation would be enough to produce an estimate of λ, hence of the whole formation. However, the high level of noise affecting the distance measurements associated to the bearing measurements in Scenario II discourages their usage in this way. In fact, using just one of the ${}^i\bar{d}_j$ would produce estimates of ${}^ix_j$ affected by the same level of noise affecting ${}^i\bar{d}_j$.

However, each ${}^i\bar{d}_j$ can be equivalently thought as a measurement of λ. In this perspective, each ${}^i\bar{b}_j$ comes with an associated measurement ${}^i\bar{\lambda}_j$. By taking the mean value of all those measurements, we can produce a more accurate estimate

$$\hat{\lambda} = \sum_{\{i,j\}} {}^i\bar{\lambda}_j / l \tag{38}$$

where l is the number of available distance measurements. Since each triangle includes at least 6 bearing measurements, even the scaling factor of a formation with few robots can be estimated quite accurately.

This estimate of λ is used twice in the algorithm: at the end of the algorithm to estimate the scale, and in the step of validation through the belief, where is possible to use the estimated scale of each partial solution instead of the scale that maximizes the equation (37).

11.5 Filtering

The generic $\mathcal{A}_i$ runs one particle filter (PF$_j$) for each $\mathcal{A}_j$ to fuse the estimates coming from P-MultiBeaReg3D with the metric informations provided by the IMUs of $\mathcal{A}_i$ and $\mathcal{A}_j$. While in Scenario II the PF needs only to filter the noise, in Scenario I it is in charge of retrieving the distances between the robots. In both cases, the use of separate beliefs $P({}^i\chi_j)$ instead of a single joint belief $P(\{{}^i\chi_j\}_{j\in N_i^{1:t}})$ is based on the independence assumption $P(\{{}^i\chi_j\}_{j\in N_i^{1:t}}) = \prod_{j\in N_i^{1:t}} P({}^i\chi_j)$. This assumption is true in a pure localization scenario, while in certain situations it is only an acceptable approximation. In any case, $P(\{{}^i\chi_j\}_{j\in N_i^{1:t}})$ cannot be maintained due to its computational cost, as the dimension of its distribution grows exponentially with the number of robots. The observability of the system is guaranteed by [38], whose analysis can be used to generate exciting trajectories.

The equations of motion of the system are

$$
\begin{aligned}
{}^i\dot{p}_j &= {}^iv_j && (39)\\
{}^i\dot{v}_j &= {}^iR_j a_j - a_i + [\omega_i]_\times {}^iv_j && (40)\\
{}^i\dot{R}_j &= ({}^iR_j[\omega_j]_\times - [\omega_i]_\times){}^iR_j && (41)
\end{aligned}
$$

where we denoted with ${}^iv_j$ the velocity of $O_{\mathcal{C}_j}$ in $\mathcal{C}_i$. Since

$$
{}^iR_j = R_Z(-\psi_{\mathcal{B}_i})R_Z(\psi_{\mathcal{B}_j}) = R_Z(\psi_{\mathcal{B}_j} - \psi_{\mathcal{B}_i}), \tag{42}
$$

we can replace (41) with

$$ {}^{i}\dot{\psi}_j = \dot{\psi}_{\mathcal{B}_j} - \dot{\psi}_{\mathcal{B}_i} = f_{\mathcal{B}_j}^{T}\,{}^{\mathcal{B}_j}\omega_j - f_{\mathcal{B}_i}^{T}\,{}^{\mathcal{B}_i}\omega_i, \tag{43} $$

being ${}^{i}\psi_j = \psi_{\mathcal{B}_j} - \psi_{\mathcal{B}_i}$ and $f_{\mathcal{B}_i}^{T}, f_{\mathcal{B}_j}^{T}$ defined by (35), and compute ${}^{i}R_j$ in (40) from (42). Therefore the state of each particle in PF$_j$ is the 7-dimensional tuple ${}^{i}\chi_j = ({}^{i}p_j, {}^{i}v_j, {}^{i}\psi_j) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{S}^1$. The observability of the system is guaranteed by [38]. In particular, one can use the analysis in that paper to generate exciting trajectories.

The motion update step of PF$_j$ is obtained by plugging $\hat{a}_i$, $\hat{a}_j$, $\hat{\omega}_i$, $\hat{\dot{\psi}}_{\mathcal{B}_i}$, $\hat{\dot{\psi}}_{\mathcal{B}_j}$ in (39–43). The new state probability is predicted by means of the integration of the motion measurements with the knowledge of the measurement noise.

Coming to the measurement update step, note first that, at each t, the algorithm P-MultiBeaReg3D may return more than one solution per robot, i.e., more than one pair ${}^{i}\hat{b}_j, {}^{i}\hat{R}_j$, each solution rated on the basis of its uncertainty during the registration steps of the algorithm. For this reason, each solutions is approximated in PF$_j$ as a gaussian measurement with a covariance proportional to its uncertainty. Therefore, the measurement model is given by the normalized sum of gaussians centered at the solutions of P-MultiBeaReg3D.

Denote with ${}^{i}\hat{\psi}_j$ the estimate of ${}^{i}\psi_j$ obtained from ${}^{i}\hat{R}_j$. The measurement update produces a rating of the predicted particles by using Bayes' law

$$ P({}^{i}\chi_j | {}^{i}\hat{b}_j, {}^{i}\hat{\psi}_j) = N P({}^{i}\hat{b}_j, {}^{i}\hat{\psi}_j | {}^{i}\chi_j) P({}^{i}\chi_j), \tag{44} $$

where N is a normalization factor.

In Scenario II, the only difference is in the measurement update,

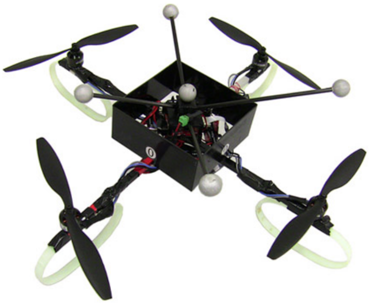


Figure 42 Quadrotor model used for the experiments.

since P-MultiBeaReg3D returns also an estimate $\hat{\lambda}$ of the scaling factor. The resulting update law is

$$P({}^{i}\chi_j | {}^{i}\hat{b}_j, {}^{i}\hat{\psi}_j, \hat{\lambda}) = N P({}^{i}\hat{b}_j, {}^{i}\hat{\psi}_j, \hat{\lambda} | {}^{i}\chi_j) P({}^{i}\chi_j). \tag{45}$$

A number of standard practical techniques have been used to improve the performance of the filter. For example, the initial prior distribution is generated using the first measurements, and we have used a Tustin integration to smooth the acceleration data coming from the motion detector.

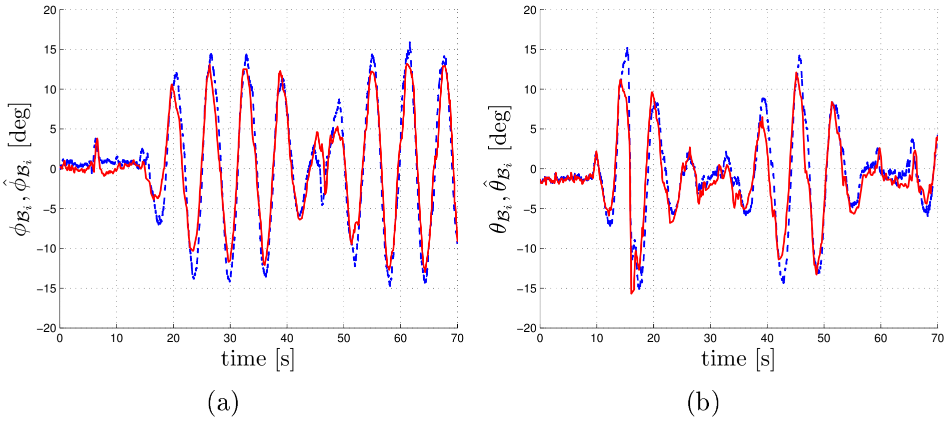
11.6 Experimental results

The proposed localization system has been experimentally tested by using quadrotors[2] as mobile robots, like the one depicted in Figure 42. We used an external Motion Capture System[3] (mocap) endowed with 16 infrared cameras as ground truth, which reaches a precision of about 1 mm for the translations and 1° for the rotations.

As a motion detector, we used the IMU available on the microcontroller board, composed by one three-axis linear MEMS

[2] http://www.mikrokopter.com

[3] http://www.vicon.com



(a)

(b)

Figure 43 Real values (dashed blue) and estimates (solid red) of roll 43(a), and pitch 43(b) during a typical experiment.

accelerometer plus three orthogonally mounted angular rate sensors. The microcontroller acquires these measurements at 400 Hz, and runs at the same frequency the complementary filter to recover the current roll and pitch. In particular, the estimate $\hat{\phi}_{\mathcal{B}_i}$ ($\hat{\theta}_{\mathcal{B}_i}$) of the roll (pitch) is computed by fusing the accelerometer measurement ${}^{\mathcal{B}_i}\bar{a}_{i_Y}$ (${}^{\mathcal{B}_i}\bar{a}_{i_X}$) with the gyroscope measurement ${}^{\mathcal{B}_i}\bar{\omega}_{i_X}$ (${}^{\mathcal{B}_i}\bar{\omega}_{i_Y}$). For our quadrotor, the dynamics of the filter is

$$\begin{aligned}\dot{\hat{\phi}}_{\mathcal{B}_i} &= {}^{\mathcal{B}_i}\bar{\omega}_{i_Y} + k_{\phi_i}({}^{\mathcal{B}_i}\bar{a}_{i_Y} - \hat{\phi}_{\mathcal{B}_i})\\ \dot{\hat{\theta}}_{\mathcal{B}_i} &= {}^{\mathcal{B}_i}\bar{\omega}_{i_X} + k_{\theta_i}({}^{\mathcal{B}_i}\bar{a}_{i_X} - \hat{\theta}_{\mathcal{B}_i}).\end{aligned}$$

The typical performance of the filter is shown in Fig. 43; here, the mean error is 1.92° for roll and 2.67° for pitch.

Due to the limited memory and processing power of the microcontroller, the localization algorithm runs on a GNU-Linux machine to which $({}^{\mathcal{B}_i}\bar{a}_i, {}^{\mathcal{B}_i}\bar{\omega}_i, \hat{\phi}_{\mathcal{B}_i}, \hat{\theta}_{\mathcal{B}_i})$ are transmitted through a serial connection. This connection is slow (average rate 20 Hz with standard deviation 4 ms) and represents a bottleneck in our testbed but also a challenge

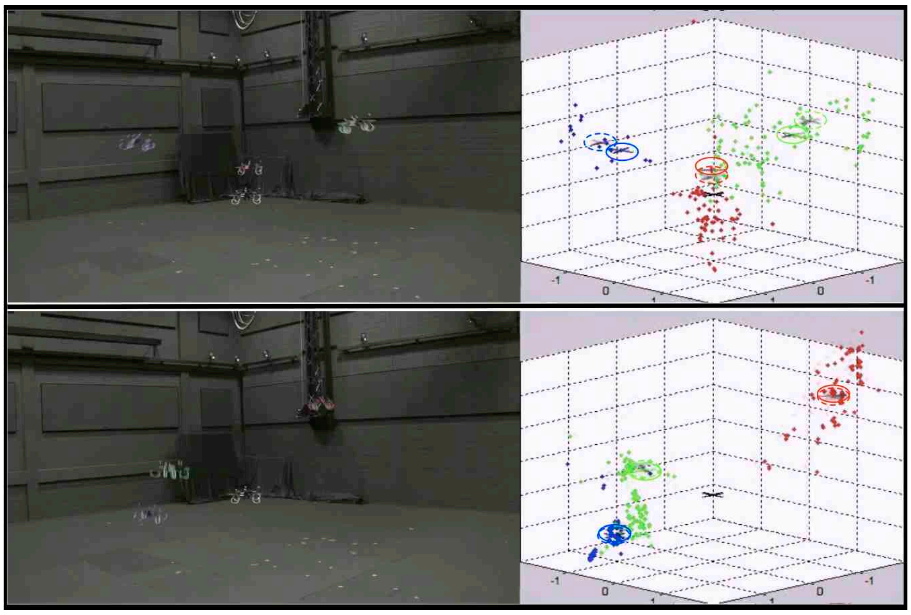


Figure 44 Two snapshots from the first experiment. On the left, the scene as seen by a fixed camera. On the right, the estimates computed by the filters at the same time instants. The dots represent the best 100 particles. For each quadrotor, the solid circled cross represents the best particle, while the dashed circled cross represents the ground truth. The first snapshot is taken at the very beginning of the experiment, while in the second snapshot the relative distances among the robots have already been retrieved.

for the localization algorithm.

The scaling factors for the IMU readings and the noise statistics of ${}^{\mathcal{B}_i}\bar{a}_i, {}^{\mathcal{B}_i}\bar{\omega}_i$ have been identified via a preliminary statistical analysis conducted over a set of data collected with the quadrotor in simple hovering. In particular, the resolutions of the accelerometer and of the gyroscope are respectively $(0.019, 0.019, 0.019)$ m/s^2 and $(0.29, 0.29, 0.29)$ deg/s, while their variances are $(0.1, 0.1, 0.6)$ m/s^2 and $(0.64, 0.64, 1.12)$ deg/s. The large variance for the accelerometer is also due to the vibrations induced by the motors/propellers.

We simulated the behavior of an on-board robot detector by analytically computing the relative bearing from the motion capture system via (24), adding to the azimuth and zenith-distance angles a

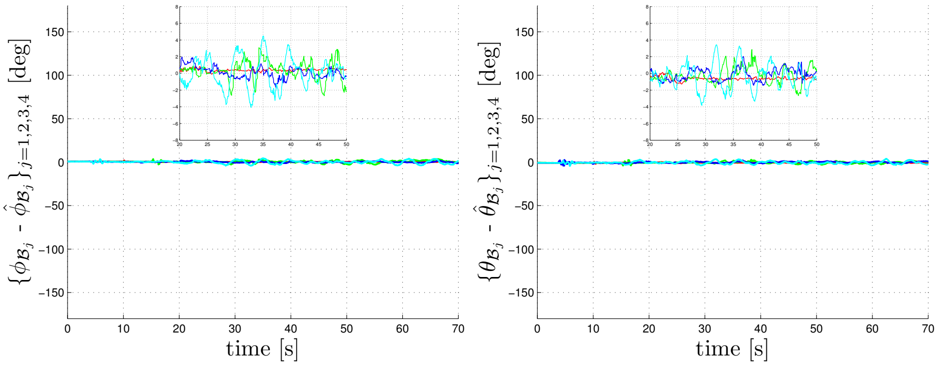


Figure 45 First experiment: errors on roll and pitch estimates for $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, computed by the complementary filters. Each plot contains a zoom in $t \in [20, 50]$ s

zero-mean gaussian noise with standard deviation σ_b of 5 deg (the noise typically observed in visual tracking experiments with the same system), and randomly introducing false positives and negatives. The distance measurements are obtained accordingly, adding a zero-mean gaussian noise with standard deviation σ_d varying from 0.3 to 1.5 m in different experiments.

Because of this strategy, we were able to gather the data for the experiment by running the robots in a sequential way. The data for one experiment are collected in multiple sessions, each session collecting the data from one robot. After, the data are subsequently synchronized and the estimation is conducted offline.

Some snapshots and the results of a bearing-only complete experiment with 4 robots starting in an square ambiguous configuration are shown in Figure 44–46. In the experiment the robot running the mutual localization process is the one with $i = 1$. The "best particle" has been used as estimate, since it showed a better behaviour with respect to the average of the particles. This is due to the multiple registration algorithm that could return more than one solution for each robot. Another possible solution is to make a clustering analysis

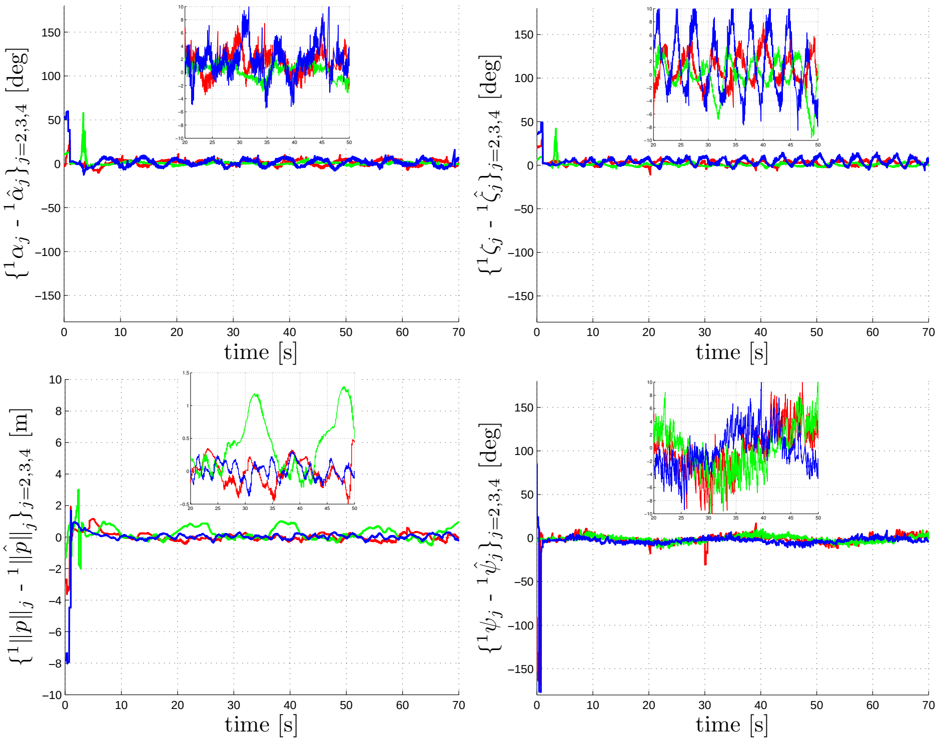


Figure 46 First experiment: errors on azimuth, zenith-distance, distance and yaw estimates for $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, computed by the particle filters of $\mathcal{A}_1$. Each plot contains a zoom in $t \in [20, 50]$ s

on the particles.

Plots in Figure 45 show the errors of the roll and pitch estimated by the complementary filters of $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$. Figure 46 shows the errors of azimuth, zenith-distance, distance and yaw estimates computed by the 3 particle filters running on $\mathcal{A}_1$. Since the starting configuration is ambiguous the initial errors are big. Moreover, the initial distance is completely unknown and the particles were on purpose initialized to be at a random relative distance with big mean error in order to show the ability of the algorithm to recover from big initial errors. When the symmetry is broken and the robots have moved enough the filters are able to recover the correct depths. Note that the convergence of the estimates is faster with respect to the the

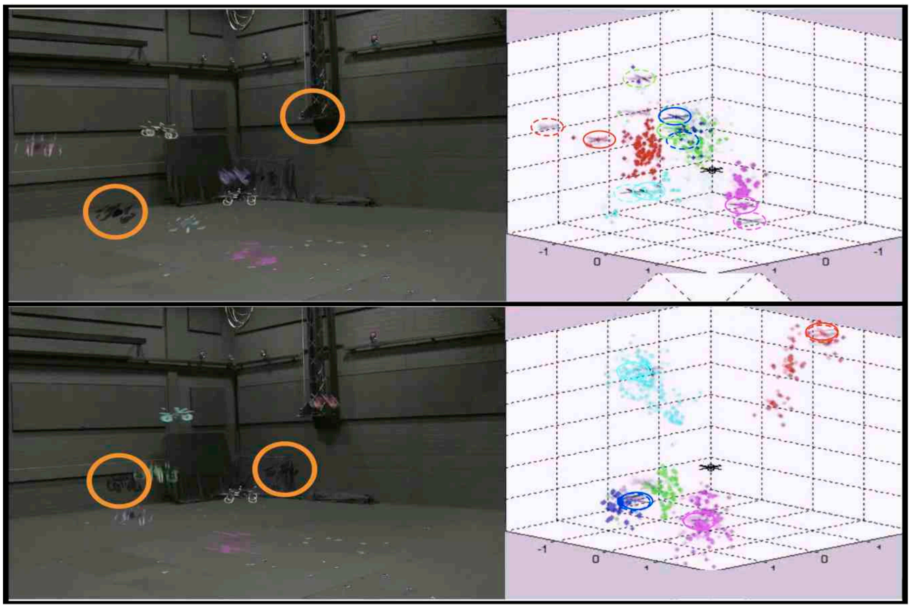


Figure 47 Two snapshots from the second experiment. On the left, the scene as seen by a fixed camera; the circled quadrotors act as false positives and do not communicate with the others. On the right, the estimates computed by the filters at the same time instants. The dots represent the best 100 particles. For each quadrotor, the solid circled cross represents the best particle, while the dashed circled cross represents the ground truth. The first snapshot is taken at the very beginning of the experiment, while in the second snapshot the relative distances among the robots have already been retrieved.

2D case since the multiple registration algorithm is more often able to find an unique solution.

In the second experiment (Figure 47), still bearing-only, we wanted to validate our framework in presence of false positives and false negatives with a group of six robots, whose errors on roll and pitch estimates are shown in Figure 48. False negatives are simulated randomly deleting a feature from the generic feature set $\hat{B}_k$ for a random time interval $t \in [1, 3]$ s; false positives are emulated by adding two additional robots that are not in the group communication set D_c. The azimuth, zenith-distance, distance and yaw error ranges in Figure 49 prove the robustness to false positives and/or false negatives because they do not affect the estimates.

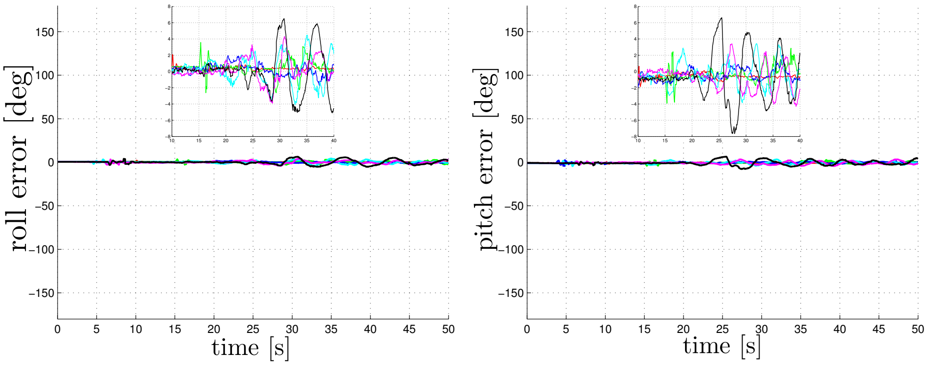


Figure 48 Second experiment: errors on roll and pitch estimates for $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, $\mathcal{A}_5$, $\mathcal{A}_6$, computed by the complementary filters. Each plot contains a zoom in $t \in [10, 40]$ s

We have used this second experiment also to test the effect of the distance measurements in the bearing+distance Scenario. Figure 50 shows the estimation errors of azimuth, zenith, distance and yaw for the estimates of $\{\mathcal{A}_2, \dots, \mathcal{A}_6\}$ computed by PFs of robot $\mathcal{A}_1$ using also the distance measurements, with 0.5 m standard deviation of the additive gaussian noise. Table 6 shows the mean (maximum) azimuth, zenith, distance and yaw errors w.r.t. the standard deviation of the noise on the distance measurements. The maximum distance error in the bearing only experiment (first row) is considered after the first 5 s, to allow the algorithm to retrieve the intial scale. The values show that the usage of the distance measurements significantly improves the quality of the estimates even when affected by large noise. The two methods obtain comparable results only when the standard deviation of the noise on the distances exceeds 100% of the measurements.

The same conclusions can be drawn from the plots (Figure 51) of the circular error probable, defined as the probability

$$p\left(e_d = \sqrt{e_x^2 + e_y^2 + e_z^2} < d\right)$$

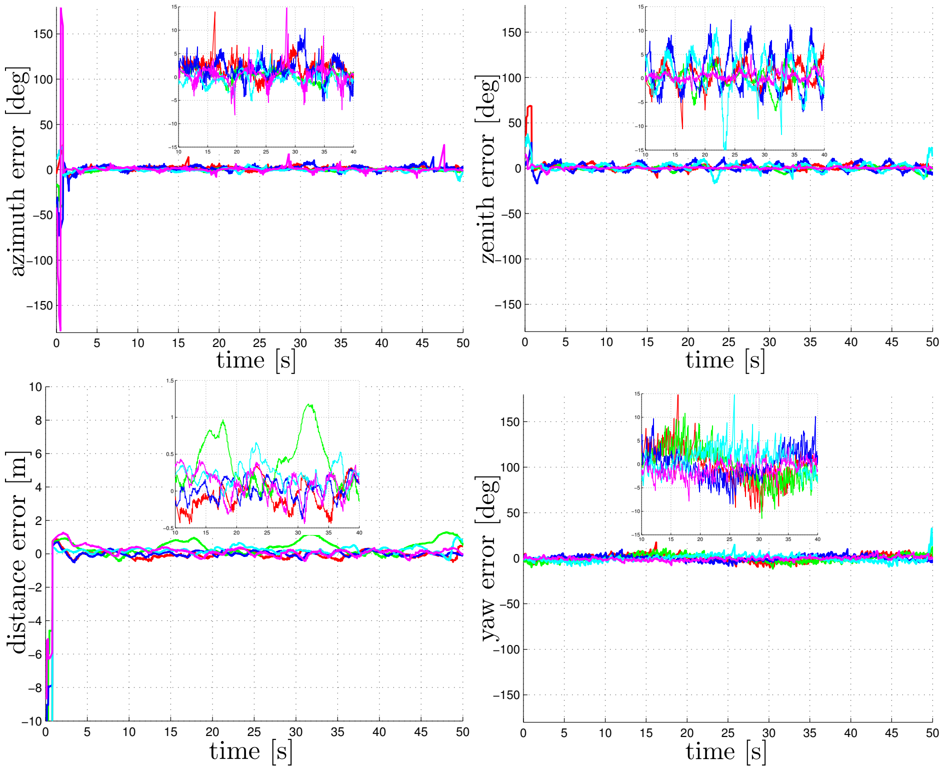


Figure 49 Second experiment - bearing-only: errors on azimuth, zenith-distance, distance and yaw estimates for $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, $\mathcal{A}_5$, $\mathcal{A}_6$, computed by the particle filters of $\mathcal{A}_1$. Each plot contains a zoom in $t \in [10, 40]$ s

	azimuth[deg]	zenith[deg]	distance[m]	yaw[deg]
bearing only	5.50 (21.87)	6.45 (21.96)	0.13 (1.04)	0.25 (19.61)
σ_d=1.5 m	2.64 (21.60)	4.75 (26.45)	0.08 (0.87)	0.27 (12.48)
σ_d=1.0 m	2.32 (21.59)	4.46 (25.73)	0.07 (0.82)	0.32 (20.65)
σ_d=0.3 m	0.74 (11.99)	2.93 (12.70)	0.04 (0.43)	0.24 (12.45)

Table 6 Mean (maximum) azimuth, zenith, distance and yaw errors w.r.t. the standard deviation of the noise on the distance measurements.

that the radial error e_d is less or equal to a parameter d, where e_x, e_y, e_z are the errors on the estimates. The plots show how the error needed to satisfy a given probability is in general lower including the distance measurements.

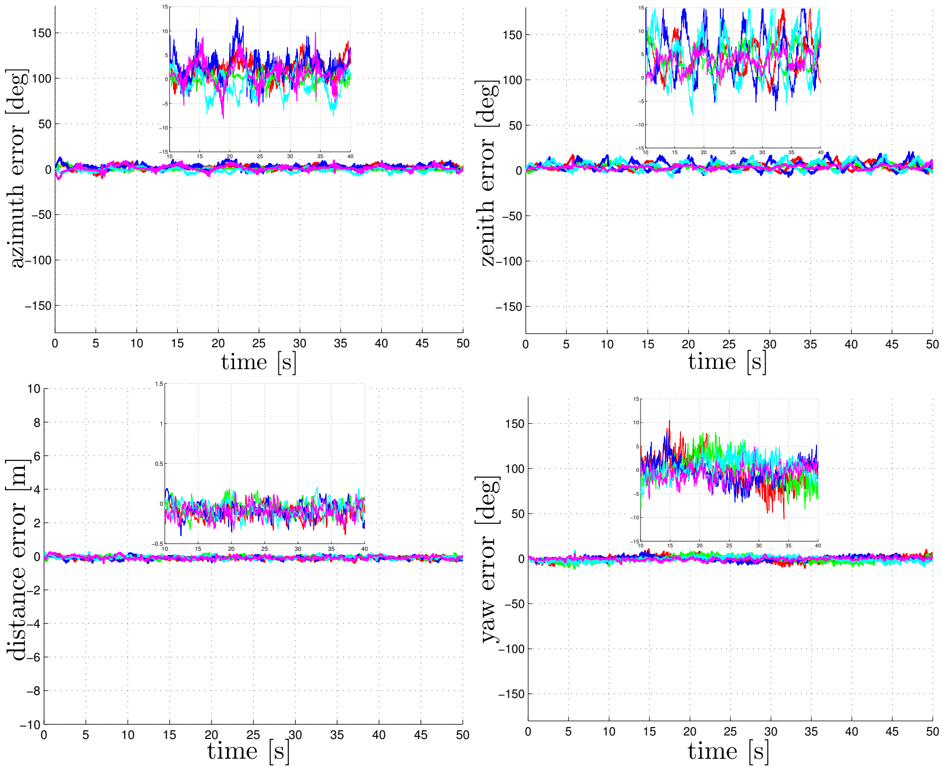


Figure 50 Second experiment - bearing+distance: errors on azimuth, zenith-distance, distance and yaw estimates for $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, $\mathcal{A}_5$, $\mathcal{A}_6$, computed by the particle filters of $\mathcal{A}_1$. Each plot contains a zoom in $t \in [10, 40]$ s

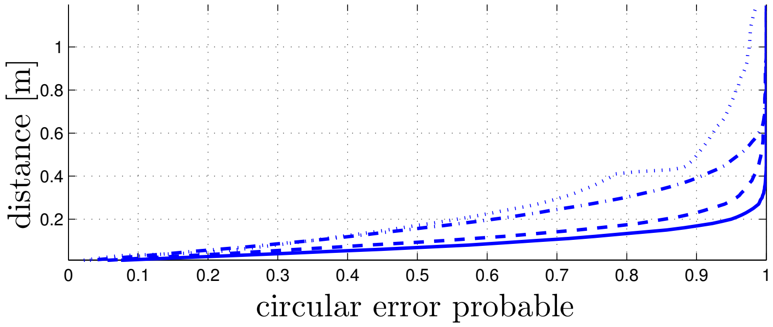


Figure 51 Circular error probable computed for the bearing-only (dotted) and distance with 0.3 m (solid), 1.0 m (dashed), 1.5 m (dash-dotted) standard deviation noise experiments.

chapter 12

Anti-symmetry control law

Assume that a group of robots must perform a collaborative task which requires mutual localization, and that only anonymous position measurements are available. If the robots are initially arranged in a formation resulting in observations that are rotational symmetric, mutual localization will be computationally heavier and will not provide a single solution. In the stochastic case, as mentioned in Chapter 4, problems will arise whenever the observations are *close* to being rotational symmetric. We have also pointed out that an initialization of the system in a rotational symmetric configuration is the only situation in which our multiple registration method would fail to find a unique solution. For this reason, we introduce in this section a continuous function that measures the distance of sets of points from rotational symmetry. This will be used to design a control law aimed at keeping the solution to Problem 2 unique. We mention that the symmetry distance function proposed in [57] is not practical for our purposes because its computation cannot be executed in real time.

Given the set of points Z_1 and an angle $\phi \in [0, 2\pi)$, define the *symmetry metric function*

$$\gamma_Z(\phi) := e(Z_1, Z_1(\phi)),$$

where $Z_1(\phi)$ is defined in (3) and

$$e(Z', Z'') := \sum_{z' \in Z'} \min_{z'' \in Z''} \|z' - z''\|^2$$

is the *closest point metric* between Z' and Z''.

Proposition 6 (Properties of γ_Z)**.** *The following statements are true:*

1. $\gamma_{Z_1}(0) = 0$.

2. γ_{Z_1} *is zero only at* $\{2k\pi/l, k = 0, \ldots, l-1\}$*, where* l *is the integer such that* $\mathcal{S}_Z = G_l$*.*

3. Z_1 *is rotational symmetric if and only if* γ_{Z_1} *is zero for some* ϕ *other than* 0*.*

4. *There exist* ϕ_1, ϕ_2*, with* $0 < \phi_1 < \phi_2 < 2\pi$*, such that* γ_{Z_1} *is strictly increasing in* $[0, \phi_1)$ *and strictly decreasing in* $(\phi_1, 2\pi)$*.*

Proof. 1) is true by definition. Moreover, $\gamma(\phi) = 0$ if and only if for any $p' \in Z_1$ exists $p'' \in Z_1(\phi)$ s.t. $z' = z''$. Hence, $Z_1 = Z_1(\phi)$, i.e., the rotation $R(\phi)$ belongs to G_l. This implies 2). Also, 2) implies 3). Finally, consider the function $\hat{\gamma}_Z(\phi) = \sum_{z \in Z_1} \|(z - c) - R(\phi)(z - c)\|^2$, which is equal to $\sum_{z \in Z_1} (2(z-c)\sin(\phi/2))^2$, that is monotonically increasing in $[0, \pi]$ and monotonically decreasing in $[\pi, 2\pi]$. For each $z \in Z_1$ there is a neighborhood of $\phi = 0$ in which $\min_{z' \in Z_1(\phi)} \|z - z'\|^2 = \|(z - c) - R(\phi)(z - c)\|^2$, i.e., in which $\gamma_Z(\phi) = \hat{\gamma}_Z(\phi)$. Denote by $\Phi \subset [0, 2\pi)$ the set in which $\gamma_Z(\phi) = \hat{\gamma}_Z(\phi)$. Then, 4) is proven by taking $\phi_1 = \max_{\Phi \cap [0,\pi]} \phi$ and $\phi_2 = \min_{\Phi \cap [\pi, 2\pi]} \phi$. ∎

As in the proof, define $\phi_1 = \max_{\Phi \cap [0,\pi]} \phi$ and $\phi_2 = \min_{\Phi \cap [\pi,2\pi]} \phi$. According to Proposition 6, the minimum value of function γ_{Z_1} in the interval $[\phi_1, \phi_2]$ (called *internal* minimum value in the following) is

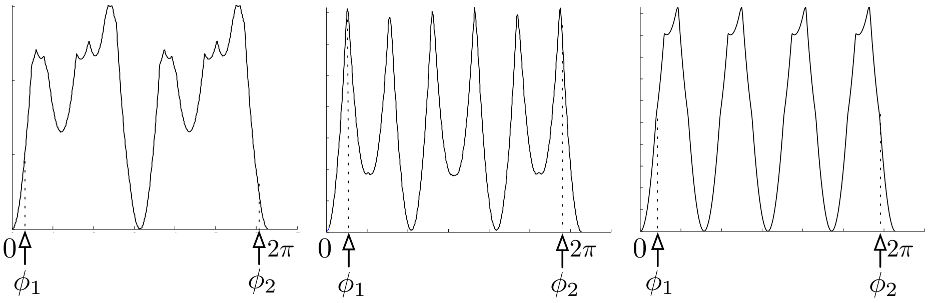


Figure 52 The symmetry metric function γ for the three set of points of Fig. 4, in the same order from left to right.

a continuous measurement of the distance of Z_1 from being rotational symmetric. If the minimum is zero, Z_1 is actually symmetric. A control action aimed at keeping Problem 2 uniquely solvable can then be based on the strategy of increasing such minimum value.

In particular, assume for simplicity that the position of each robot obeys an omnidirectional kinematic model:

$$\dot{z}_i = u_i, \qquad i = 1, \ldots, n,$$

where u_i is the two-dimensional vector of velocity inputs for $\mathcal{A}_i$. Consider the following *anti-symmetry* control law

$$u_i = \alpha \frac{\bar{z}_i - z_i}{\|\bar{z}_i - z_i\|} \qquad i = 1, \ldots, n, \tag{46}$$

where α is a positive gain and

$$\bar{z}_i := \arg\min_{z \in Z_1(\bar{\phi})} \|z_i - z\| \qquad \bar{\phi} := \arg\min_{\phi \in [\phi_1, \phi_2]} \gamma_{Z_1}(\phi).$$

This control law has a simple interpretation. Once the rotation angle $\bar{\phi}$ that minimizes γ_{Z_1} in $[\phi_1, \phi_2]$ has been identified (e.g., numerically), $Z_1(\bar{\phi})$ is built by rotating Z_1 by $\bar{\phi}$. The closest point $\bar{z}_i \in Z_1(\bar{\phi})$ is found for any $z_i \in Z_1$, and the velocity input is chosen so as push $\mathcal{A}_i$

away from $\bar{z}_i$ along the segment $z_i\bar{z}_i$, leading to an increase of $\gamma_{Z_1}(\bar{\phi})$.

Note that (46) is undefined if Z_1 is rotational symmetric. In this case a simple randomized control can be used for the small time sufficient to break the symmetry.

12.1 Simulations

We have validated the results of Sections 4.1 and 12 through extensive simulations of the anti-symmetry control law.

The results of the first simulation are shown in Fig. 53 (top). The 9-robot system starts in a lattice formation whose proper symmetry group is G_4, and moves under the action of the anti-symmetry control. Symmetry is readily broken, as shown by change in symmetry metric function γ_{Z_1}, which has 3 internal zeros at start. As the simulation proceeds, the internal minimum value of γ_{Z_1} increases.

Figure 53 (top) also shows the consequence of measurement noise on the accuracy of the estimated solution in the neighborhood of the initial rotational symmetric formation. To compute the solutions of Problem 2, we have used P-MultiReg with uniform prior belief. At each step, we have obtained multiple sets of noisy observations by adding a gaussian noise to the observations of the current arrangement. The figure shows all the possible poses of the circled robot as estimated by P-MultiReg on the basis of these data. At the start, when the formation is rotational symmetric, the estimated solutions are evenly distributed in 4 clusters of poses. The clusters are centered on all the feasible positions of a single subset of the partition $\mathcal{E}_{Z_1}$, as predicted by Proposition 3. The number of solutions (576) found by P-MultiReg matches with the one theoretically derived in Proposition 4. When the symmetry is completely broken, at $t = 4.0$ s, the surviving estimates have a gaussian distribution centered on the real pose and a covariance

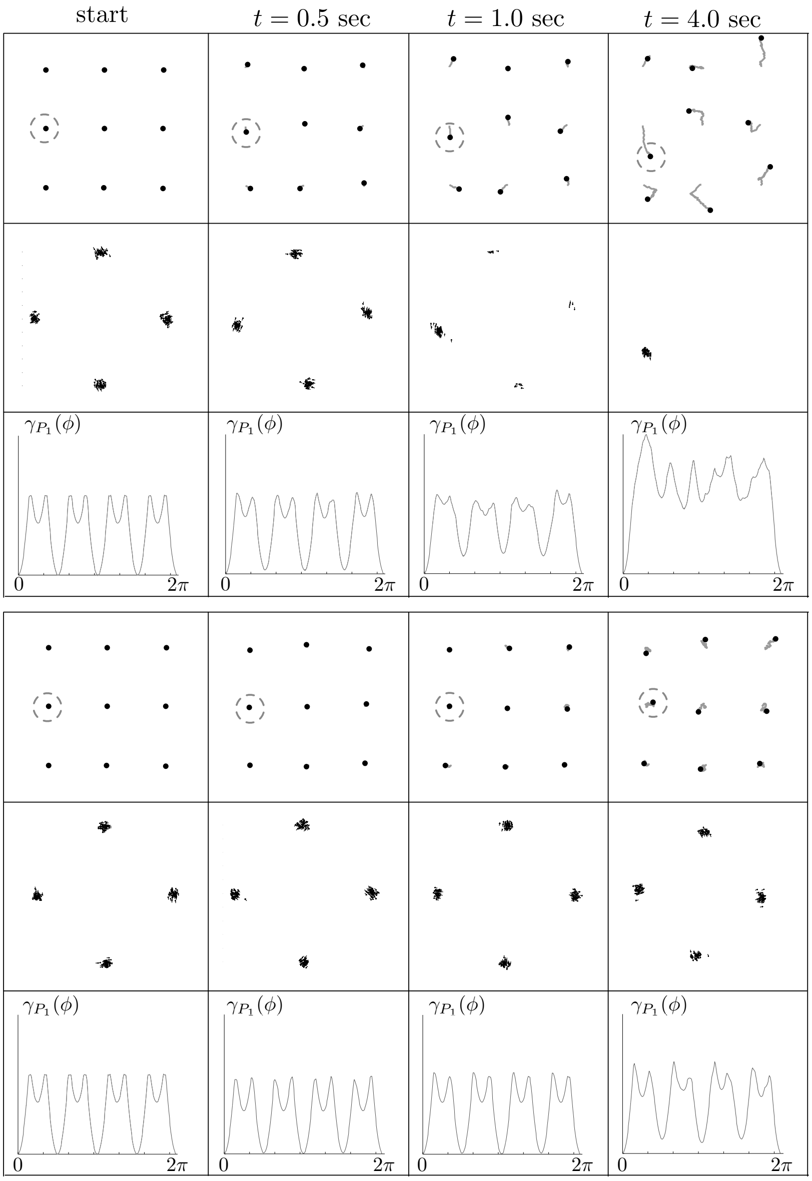


Figure 53 Above: the use of the anti-symmetry control law to break up a 9-robot lattice formation whose proper symmetry group is G_4; 4 snapshots of the formation (top), the estimated positions for the circled robot (center), the symmetry metric function γ_{Z_1} (bottom). Below: the same results with a random control law.

comparable to that of the additive noise.

In the intermediate frames, in which the formation is close to being rotational symmetric, the solutions of MultiReg are distributed in more than one cluster, but not evenly. The largest cluster is centered on

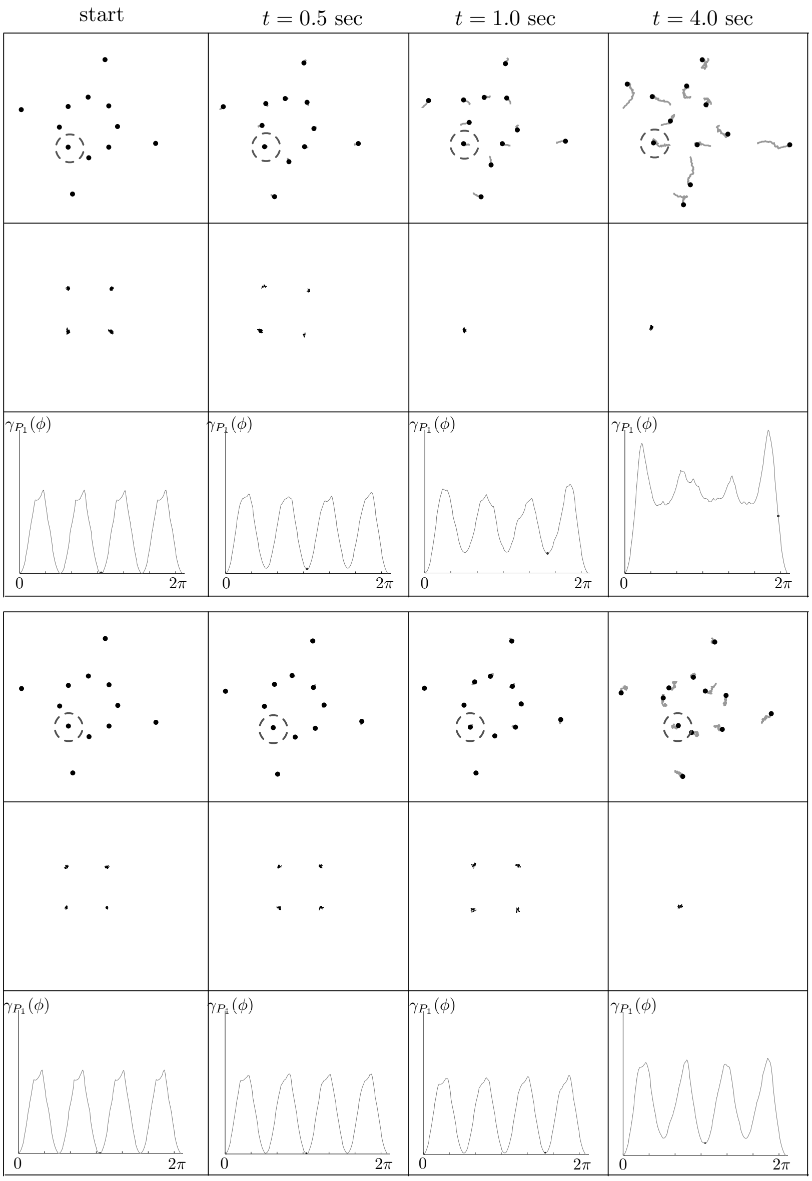


Figure 54 As in Fig. 53 for a starting formation with proper symmetry group G_4.

the real pose of the estimated robot. The other clusters, with less solutions, become feasible configurations only when the additive noise on the observations restores the rotational symmetry.

For comparison, we have also simulated the same 9-robot system under the action of a random control law (Figure 53, bottom). In fact,

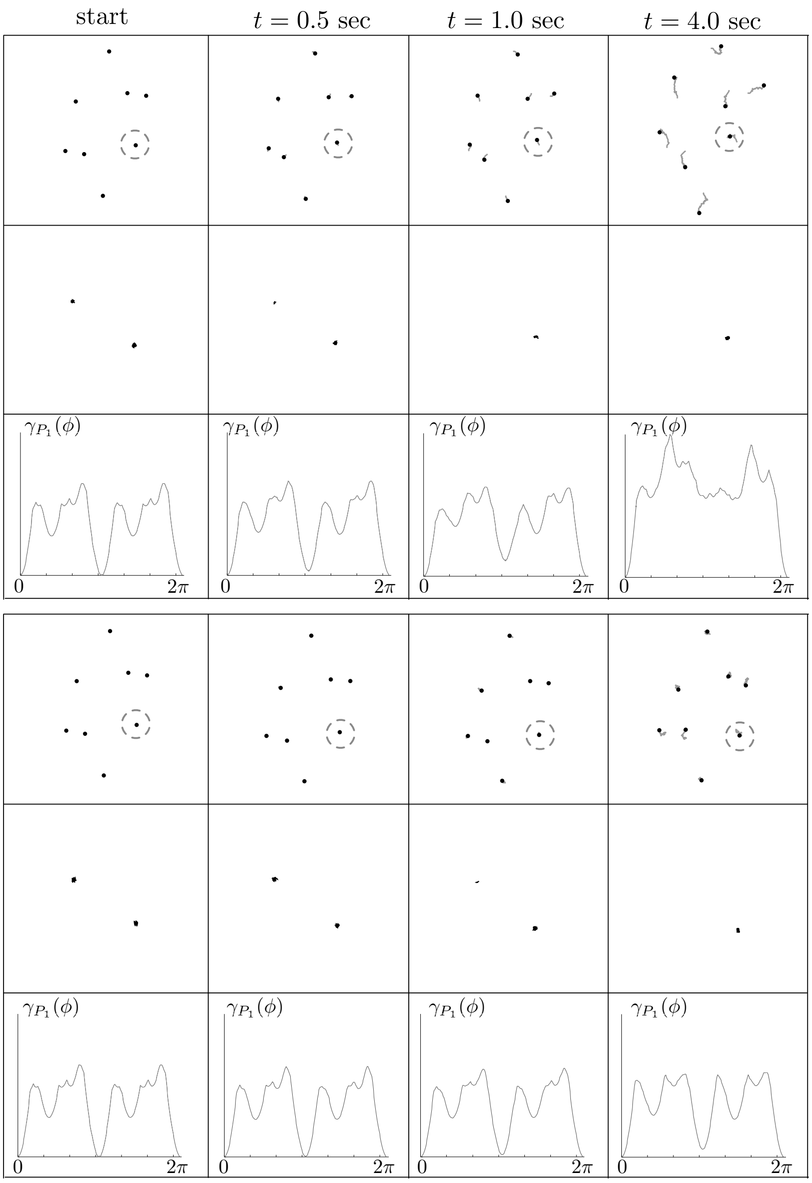


Figure 55 As in Fig. 53 for a starting formation with proper symmetry group G_2.

since the subset of symmetric configurations has zero measure in the configuration space, a random control law can be expected to break the symmetry. However, the results show that the anti-symmetry control is much more effective in achieving this than the random control. In fact, the increase of the internal minima of γ_{Z_1} with the random control

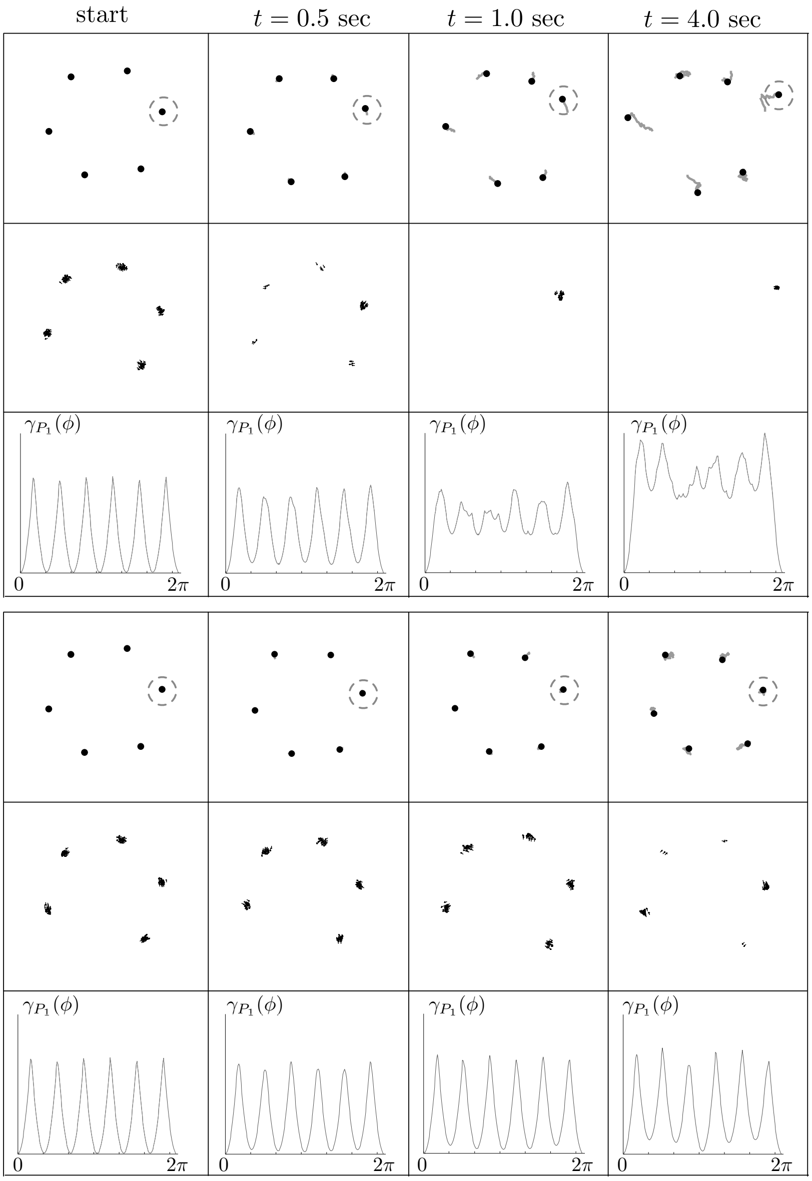


Figure 56 As in Fig. 53 for a starting formation with proper symmetry group G_6.

is slower and non-monotonous. Correspondingly, the multiple clusters of the estimation do not disappear.

The results of the same simulation for different symmetric starting configurations are shown in Figs. 54–56.

chapter 13

Future work: PHD filter based approach

One of the main issues affecting our P-MultiReg based approach for the solution of the mutual localization problem shows up when two robots, not looking each other and not looking at the same portion of environment, gather similar features sets because, for example, of an environment which is densely populated by deceiving obstacles disposed along repetitive patterns. In this case, if they communicate, P-MultiReg could match feature sets taken in completely different places, and find wrong hypotheses to feed the filters. Figure 57 depicts an example of this situation.

To find a partial solution to this issue, a possible workaround could be to keep memory of the features of the environment detected in the past. If in the above example $\mathcal{A}_i$ had already measured the features detected by $\mathcal{A}_j$, it could use this information to recognize that the

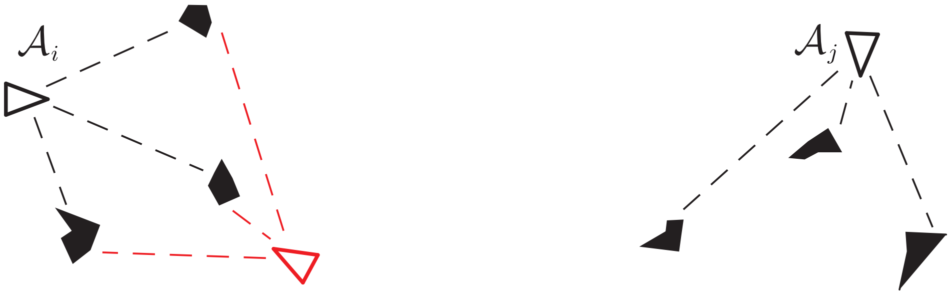


Figure 57 The robots $\mathcal{A}_i$ and $\mathcal{A}_j$ (triangles) observe different objects (black polygons) but gather similar observations (black dashed lines), so that $\mathcal{A}_i$ could localize $\mathcal{A}_j$ in the pose showed by the red triangle.

possible configuration of $\mathcal{A}_j$ on a simple registration of feature sets base are both the true and the red hypotheses. Moreover, by the use of the prior knowledge on the pose of $\mathcal{A}_j$, $\mathcal{A}_i$, gathered by a previous encounter, could disambiguate the solution in favor of the true hypothesis.

A possible way to enforce this idea comes from multi-target tracking theory. Assuming to have n_k targets $\{1, 2, \ldots, n_k\}$, with n_k unknown and variable over time, the probability hypothesis density (PHD) Filter [58] basic idea is to estimate the probability hypothesis density $D(\mathbf{x}')$ of the generic targets, where $D(\mathbf{x}')$ is the function whose integral over any subset $S \subseteq \mathcal{X}$ of the state space of the targets $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the expected number of targets $N(S)$ in those subset. The mathematical definition of $D(\mathbf{x})$ is given by the following relationship

$$N(S) = \int_S D(\mathbf{x}')d\mathbf{x}' \tag{47}$$

The derivation of the filter is based on random finite set (RFS) theory, and is quite tough and long. However, the resulting filter is composed of two steps, time update and measurements update, whose interpretation is straightforward.

The generic equation for the time update is

$$\begin{aligned} & D_{k+1|k}(\mathbf{x}) = \\ & = b_{k+1|k}(\mathbf{x}) + \int [p_S(\mathbf{x}')f_{k+1|k}(\mathbf{x}|\mathbf{x}') + b_{k+1|k}(\mathbf{x}|\mathbf{x}')]D_{k|k}(\mathbf{x}')d\mathbf{x}' \end{aligned} \tag{48}$$

where $b_{k+1|k}(\mathbf{x})$ is the probability that a new target appears in $\mathbf{x}$, $p_S(\mathbf{w})$ is the probability that a target with state $\mathbf{w}$ at time-step k will survive into time-step $k+1$, $b_{k+1|k}(\mathbf{x}|\mathbf{w})$ is the probability that a new target spawns in $\mathbf{x}$ at time-step $k+1$ given that there is a target in $\mathbf{w}$ at time-step k, and $f_{k+1|k}(\mathbf{x}|\mathbf{w})$ is the single target Markov transition density. Assuming no target appearances or disappearances, the previous equation reduces to

$$D_{k+1|k}(\mathbf{x}) = \int f_{k+1|k}(\mathbf{x}|\mathbf{x}')D_{k|k}(\mathbf{x}')d\mathbf{x}' \tag{49}$$

showing that in this case the PHD is governed by the same law of motion as that which governs the time evolution of the posterior probability density function of any single target in the multi-target system.

The generic equation for the measurement update is

$$\begin{aligned} D_{k+1|k+1}(\mathbf{x}) = & \\ = & D_{k+1|k}(\mathbf{x}) \left[\sum_{\mathbf{z} \in Z_{k+1}} \frac{p_D(\mathbf{x})g(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z}) + \int p_D(\mathbf{x}')g(\mathbf{z}|\mathbf{x}')D_{k+1|k}(\mathbf{x}')d\mathbf{x}'} + 1 - p_D(\mathbf{x}) \right] \end{aligned} \tag{50}$$

where $g(\mathbf{z}|\mathbf{x})$ is the sensor likelihood function, $p_D(\mathbf{x})$ is the probability that an observation will be collected from a target with state $\mathbf{x}$, and $\lambda c(\mathbf{z})$ expresses the probability that a given measurement $\mathbf{z}$ is a false alarm. The simplified case with no missed detections $p_D(\mathbf{x}) = 1$ and no false alarms $\lambda = 1$ is

$$D_{k+1|k+1}(\mathbf{x}) = D_{k+1|k}(\mathbf{x}) \left[\sum_{\mathbf{z} \in Z_{k+1}} \frac{g(\mathbf{z}|\mathbf{x})}{\int p_D(\mathbf{x}')g(\mathbf{z}|\mathbf{x}')D_{k+1|k}(\mathbf{x}')d\mathbf{x}'} \right] \tag{51}$$

The measurements update equation expresses a Bayes-like rule with all the measures and all the targets, each measurement associated to each target, and each association with a weight that is computed by the probability. In this way the data association in the PHD filter is fully probability driven and promises to keep track of all possible associations without computational complexity explosion.

The bad news is that, the filter as first designed in [58], lacked of a simple form for the implementation. For this reason, the authors of [59] proposed a gaussian mixture implementation of the PHD filter, assuming the gaussianity of all uncertainty and noises in the model of targets and measurements. The resulting filter is applicable in practice.

In [60] the authors discretize the space of the targets in a given number of subsets and estimate the probability of each bin to be occupied by a target by the use of measurements and the motion model of the targets. The resulting bin-occupancy filter is proved to be the discretized version of the PHD filter, and offers another possibility for its practical implementation.

The robotics community has recently started to apply the PHD filter to solve problems as SLAM in [61], whose authors propose a feature based SLAM in which the features are tracked through the use of a PHD filter.

13.1 PHD filter based mutual localization system

Here we want to design a system for the mutual localization of the components of a team of robots in which the data association is driven by a PHD-filter like policy. The straight application of the PHD filter is not feasible since our problem implies the reconstruction of the identities of the measured robots and the use of their odometries. In particular, the second needs the first to be applied. For this reason we have to modify equation (48) so that it can include the estimation of the identities and the use of the odometries, while using the same equation (50) for the measurement update. Its gaussian mixture implementation will provide a sort of robocentric feature based map of the environment in the form of a gaussian mixture, including in it

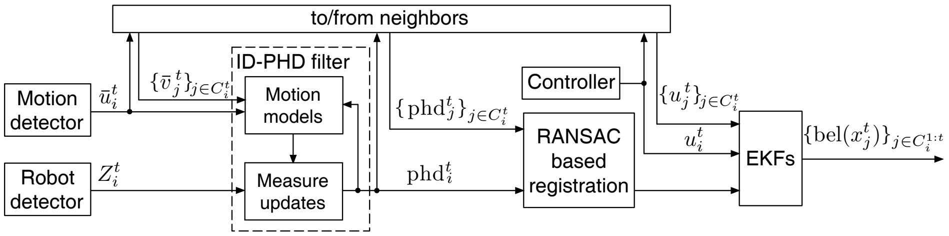

to/from neighbors
ID-PHD filter
Motion detector
$\bar{u}_i^t$
$\{\bar{v}_j^t\}_{j \in C_i^t}$
Motion models
Robot detector
Z_i^t
Measure updates
$\{\mathrm{phd}_j^t\}_{j \in C_i^t}$
phd_i^t
Controller
RANSAC based registration
$\{u_j^t\}_{j \in C_i^t}$
u_i^t
EKFs
$\{\mathrm{bel}(x_j^t)\}_{j \in C_i^{1:t}}$

Figure 58 System architecture of the PHD filter based system for the mutual localization running on $\mathcal{A}_i$.

also some gaussians with an ID representing the Probability hypothesis density of the other robots. We will call this filter the ID-PHD filter.

However, it will not be directly possible to include the use of the measurements gathered from the other robots in the filter. In fact their use would require the joint estimation of the targets to deal with the dependence that it would introduce, but such estimation is outside of the PHD filter framework.

To include measurements from the other robots we propose the architecture depicted in Figure 58. The ID-PHD filter is fed with the odometries of the robots for the time update and with the measurements gathered by the robot detector for the measurement update. The result of the filter phd_i^t is the estimate of the probability hypothesis density computed by $\mathcal{A}_i$ at time t is communicated to the neighbor and used to feed a RANSAC based registration algorithm with the phd computed by the other robots $\{\mathrm{phd}_j^t\}_{j \in C_i^t}$. The registration phase is easier with respect to the case of P-MultiReg, since the information provided by phd_h^t is more rich w.r.t the one gathered by Z_i^t, including also the identity of the robots and their orientation. However, the use of a RANSAC paradigm is required since each robot can be in principle represented by more than one gaussian.

The result of the registration is finally used in the measurement updates of a bank of EKF, one EKF for each $\mathcal{A}_j$. The time update of the EKFs can not be carried on by the use of the odometries, since this information has already been used to gather the measurements. For this reason, the time update is fed with the integration over time of the real control input as provided by the controller.

13.2 ID-PHD filter

Here we explain how is possible to include the estimate of the identities in the PHD filter by deriving the time update from scratch, using a different point of view over the probability hypothesis density. To do this, we start by introducing the notation for this section.

We assume to have n_k targets $\{1, 2, \dots, n_k\}$, with n_k unknown and variable over time. The state $\mathbf{x}^h$ of the h-th target, $h = 1, \dots, n_k$ evolves in the state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ following a Markov process described by the transition density $f(\cdot|\cdot)$. That is, given a state $\mathbf{x}'$ at time $k-1$ and the control input $\mathbf{u}_k$ applied between time $k-1$ and k, $f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k)$ is the probability density of transition to the state $\mathbf{x}$ at time k. We assume that at each time k we receive a set $U_k = \{\mathbf{u}_k^j,\ j = 1, \dots, r_k\}$ of control inputs through the communication network. This means that each target is communicating his control input to the system in charge of the estimate. We will denote with $U_{1:k} = \{U_1, \dots, U_k\}$ the set of the sets of control inputs received up to step k.

The Markov process is partially observed in the observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. These observations are modeled by the likelihood function $g_k(\cdot|\cdot)$, that is, given a state $\mathbf{x}$ at time k, the probability density of receiving the observation $\mathbf{z}_k \in \mathcal{Z}$ at the same time is given by $g_k(\mathbf{z}_k|\mathbf{x})$. Without loss of generality, we can assume that the function $g_k(\cdot|\cdot)$ is invariant with time, so that $g_k(\cdot|\cdot) = g(\cdot|\cdot)$. We assume that each sensor provides at each time step k a set of observations $Z_k = \{\mathbf{z}_k^i,\ i =$

$1, \ldots, m_k\}$. Each observation $\mathbf{z}_k^i$ is the result of a measurement of one target, without the knowledge of the identity of the measured target. We will denote with $Z_{1:k} = \{Z_1, \ldots, Z_k\}$ the set of the sets of observations obtained until time k. For brevity of notation, we introduce the symbol $Y_a^b = \{Z_{1:a}, U_{1:b}\}$ to indicate the set including all the control inputs up to step b and all the observations up to step a. We assume that, given a target in a certain location $\mathbf{x}$ at step k, the probability that it originates a measurement is modeled by a function $P_d(\mathbf{x}; k)$. This means that $P_d(\mathbf{x}; k)$ is bounded between 0 and 1, and variable over k. To simplify the notation, in the following we will drop the subscript k in n_k, m_k, r_k, assuming however that all this quantities are dependent from the time step k, and we will omit the dependency of $P_d(\mathbf{x}; k)$ from k.

In general, we will denote with $p(\cdot)$ a probability density function, and with $P(\cdot)$ a probability mass function. We will also use the superscript $(\cdot)^h$ to refer a particular function, probability or quantity to the target h, the superscript $(\cdot)^i$ to refer a particular function, probability or quantity to the i-th measurement and the superscript $(\cdot)^j$ to refer a particular function, probability or quantity to the j-th control input. We can define $p^h(\mathbf{x}|Y_k^k)$ as the probability density of the state of target h at time k given all the observations $Z_{1:k}$ and all the control inputs $U_{1:k}$ up to time k. Similarly, $p^h(\mathbf{x}|Y_{k-1}^k)$ is the probability density of the state of target h at time k given all the observations $Z_{1:k-1}$ and all the control inputs $U_{1:k}$ up to time $k-1$ and k respectively. Let be also $p^h(\mathbf{z}_k^i|Y_{k-1}^k)$ the probability density that the h-th target originates the observation $\mathbf{z}_k^i$ given the observations $Z_{1:k-1}$ up to step $k-1$ and the control input $U_{1:k}$ up to step k.

Here we address the problem of estimating the number and state of the targets by the estimation of the probability hypothesis density

$D(\mathbf{x}')$, defined by the relationship:

$$N(S) = \int_S D(\mathbf{x}')d\mathbf{x}' \tag{52}$$

where $N(S)$ is the expected number of targets in any subset $S \subseteq \mathcal{X}$. In [58], Mahler shows the uniqueness of this function. Note that we can always factorize the probability hypothesis density of generic targets in a sum of elements

$$D(\mathbf{x}|\cdot) = \sum_{h=1}^{n} d^h(\mathbf{x}|\cdot) = \sum_{h=1}^{n} P_E^h(\cdot)p^h(\mathbf{x}|\cdot) \tag{53}$$

each one of them representing a target with probability of existence

$$P(E^h|\cdot) = P_E^h(\cdot) = \int_{\mathcal{X}} d^h(\mathbf{x}|\cdot)d\mathbf{x}, \quad h = 1, \dots, n \tag{54}$$

where E^h is the event: **the target** h **exists**. By interpreting the probability density function $p^h_{(\cdot)}(\mathbf{x})$ as the probability hypothesis density of a target whose probability of existence is $P^h_{E_{(\cdot)}} = 1$ we can write

$$p^h(\mathbf{x}|\cdot) = d^h(\mathbf{x}|E^h, \cdot) \tag{55}$$

Then, $d^h(\mathbf{x}|\cdot)$ the probability hypothesis densities of the single target h, while $D(\mathbf{x}|\cdot)$ is the probability hypothesis density of generic targets.

13.2.1 Time Update

The time update for a single target in a standard Bayesian filter is given by:

$$\begin{aligned} p^h(\mathbf{x}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) = \\ = \int f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k) p^h(\mathbf{x}'|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}' \end{aligned} \tag{56}$$

Let be $P^h(\mathbf{u}_k^j)$ the probability that between time $k-1$ and k the j-th control input is referred to the h-th target. Then, since only one control input is the actual control of each target, these probabilities must satisfy the conditions

$$\sum_{j=1}^{r} P^h(\mathbf{u}_k^j) = 1, \quad h = 1, \ldots, n \tag{57}$$

Then, for the theorem of total probability the time update of the Bayesian filter can be written as

$$p^h(\mathbf{x}|Y_{k-1}^k) = \sum_{j=1}^{r} P^h(\mathbf{u}_k^j) \int f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k^j) p^h(\mathbf{x}'|Y_{k-1}^{k-1}) d\mathbf{x}' \tag{58}$$

Introducing the probability of existence of each target and computing the sum of the probability hypothesis densities for each target, the prior probability hypothesis density of generic targets is then given by

$$\begin{aligned} D(\mathbf{x}|Y_{k-1}^k) &= \sum_{h=1}^{n} d^h(\mathbf{x}|Y_{k-1}^k) = \sum_{h=1}^{n} P_E^h(Y_{k-1}^{k-1})\, p^h(\mathbf{x}|Y_{k-1}^k) = \\ &= \sum_{h=1}^{n} P_E^h(Y_{k-1}^{k-1}) \sum_{j=1}^{r} \int p^h(\mathbf{u}_k^j) f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k^j) p^h(\mathbf{x}'|Y_{k-1}^{k-1}) d\mathbf{x}' = \\ &= \sum_{j=1}^{r} \int f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k^j) \sum_{h=1}^{n} \left[p^h(\mathbf{u}_k^j) d^h(\mathbf{x}'|Y_{k-1}^{k-1}) \right] d\mathbf{x}' \end{aligned} \tag{59}$$

Assuming $p^h(\mathbf{u}_k^j) = p(\mathbf{u}_k^j),\ h = 1, \ldots, n$, then

$$
\begin{aligned}
D(\mathbf{x}|Y_{k-1}^k) &= \sum_{j=1}^{r} \int f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k^j) p(\mathbf{u}_k^j) \sum_{h=1}^{n} \left[d^h(\mathbf{x}'|Y_{k-1}^{k-1}) \right] d\mathbf{x}' = \\
&= \sum_{j=1}^{r} p(\mathbf{u}_k^j) \int f(\mathbf{x}|\mathbf{x}', \mathbf{u}_k^j) D(\mathbf{x}'|Y_{k-1}^{k-1}) d\mathbf{x} \qquad (60)
\end{aligned}
$$

In equation (59) we are implicitly assuming that the probability of existence does not change in the time update. It is possible to extend equation (59) replacing $P_E^h(Y_{k-1}^{k-1})$ with $P_E^h(Y_{k-1}^k) = P_S^h(k) P_E^h(Y_{k-1}^{k-1})$, where $P_S^h(k)$ is the probability that target h survived in the time elapsed between step $k-1$ and step k.

The ID-PHD filter is initialized on the first measurements gathered by the robot, creating one gaussian for each measurement with probability of existence equal to 1. The identity associated to these first gaussians are unknown, so the probability that it is a given communicating robot is equal for all robots and all gaussians. We are in the situation described in equation (60). Each gaussian is then atomized in $n + 1$ gaussians, each one of them associated with the relative identity, plus one that is the identity of the generic static feature. The weight of each of this gaussian is then $1/(n + 1)$. All the gaussians can be now propagated by the use of the corresponding odometry. In the following steps, the gaussians with the correct identities will be magnified in the measurement update, conducted following equation (50), while the others will vanish. The target appearance terms will again generate not associated gaussians, that will be atomized in single identities gaussians and so on. The identities in this case will be chosen among the identities of communicating robots whose pose has not been recovered yet by the filter.

13.3 Future work

We have already implemented and tested the ID-PHD filter with success. At this step of the development, we still need to implement the RANSAC based registration algorithm and connect the blocks of the system. Once the system is complete, we plan to extend the comparison presented in Chapter 9 to this method. After this experimental phase, we can consider the insertion of the developed ID-PHD filter in a more complex mutual localization system, adding also the *multiple registration* of the probability hypothesis densities computed by the robots and a feedback from the EKFs to this algorithm to chose among possible ambiguous configurations, thus obtaining the complete fusion of this system with the P-MultiReg based method.

chapter 14

Conclusions

The object of the study of this work is the mutual localization problem with anonymous position measurements in multi-robot teams. It arises in all those situations in which the component of a team of robots do not have the ability to recognize the identities of other robots, either for sensory limitation or hostile environmental condition.

We have theoretically analyzed the static situation in which one robot tries to reconstruct the pose of its teammates by the use of its and their measurements, proving that the introduction of unknown data association causes the loss of unique solvability of the inversion of the measurement map if the formation is rotational symmetric. A classical approach to deal with this problem could be to estimate during time the data association using a particle filter. This method, although working in simple problems, crashes on the factorial nature of the problem as the number of robots and the ambiguities increases.

The winning idea to solve the problem is to demand the solution of the data association to a probabilistic algorithm that deals with the mutual exclusion constraint, so that the computational complexity is limited. A subsequent filtering phase can avoid to explicitly consider the mutual exclusion constraint, since its information is already incorporated in the result of the multiple registration algorithm. We have carried on an extensive experimentation to validate these considerations, and, as expected, the multiple registration based method outperforms the classical approach.

However, its extension to different types of measurements and/or system is non-trivial, requiring a strong effort to design an equivalent of the multiple registration algorithm. Moreover, the algorithm suffers from a factorial computational complexity whenever the system starts in a rotational symmetric configuration. This issue can be solved through the use of a control law that leads the robot in non rotational symmetric configurations.

In the future, we want to try to extend the field of view of the sensors by the use of PHD filters, an algorithm coming from multi-target tracking theory. This would allow a registration phase on more reliable data, producing an even more robust localization system.

References

[1] A. Franchi, G. Oriolo, and P. Stegagno. On the solvability of the mutual localization problem with anonymous position measures. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 3193–3199, Anchorage, AK, May 2010. 10, 11

[2] A. Franchi, P. Stegagno, and G. Oriolo. Probabilistic mutual localization in multi-agent systems from anonymous position measures. In *49th IEEE Conf. on Decision and Control*, pages 6534–6540, Atlanta, GA, Dec. 2010. 11

[3] A. Franchi, G. Oriolo, and P. Stegagno. Mutual localization in a multi-robot system with anonymous relative position measures. In *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3974–3980, St. Louis, MO, Oct. 2009. 11

[4] A. Franchi, G. Oriolo, and P. Stegagno. Mutual localization in a multi-robot system with anonymous relative position measures. Technical report, Department of Computer and System Sciences Antonio Ruberti, Jan. 2009. 11

[5] A. Franchi, P. Stegagno, M. Di Rocco, and G. Oriolo. Distributed target localization and encirclement with a multi-robot system. In *7th IFAC Symp. on Intelligent Autonomous Vehicles*, pages 67–72, Lecce, Italy, Sep. 2010. 11

[6] A. Franchi, G. Oriolo, and P. Stegagno. Mutual localization in multi-robot systems using anonymous relative measurements. *In preparation*, 2012. 11

[7] P. Stegagno, M. Cognetti, A. Franchi, and G. Oriolo. Mutual localization using anonymous bearing-only measures. In *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 469–474, San Francisco, CA, Sep. 2011. 11

[8] M. Cognetti, P. Stegagno, A. Franchi, G. Oriolo, and H. H. Bülthoff. 3-D mutual localization with anonymous bearing measurements. In *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 2012. 11

[9] P. Stegagno, M. Cognetti, A. Franchi, G. Oriolo, and H. H. Bülthoff. Bearing-only vs. bearing+distance anonymous mutual

localization in multi-UAV systems. In *Submitted to the 2nd IFAC Workshop on Multivehicle Systems*, Espoo, Finland, Oct. 2012. 11

[10] A. Howard, L. E. Parker, and G. S. Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25(5-6):431–447, 2006. 13

[11] M. Schwager, B. J. Julian, and D. Rus. Optimal coverage for multiple hovering robots with downward facing cameras. In *2009 IEEE Int. Conf. on Robotics and Automation*, pages 3515–3522, Kobe, Japan, May 2009. 13

[12] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *International Journal of Robotics Research*, 30(3), 2010. 13

[13] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215 –233, 2007. 13

[14] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *IEEE Trans. on Automatic Control*, 52(5):863 –868, 2007. 13

[15] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. on Automatic Control*, 51(3):401–420, 2006. 13

[16] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009. 13

[17] R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity in mobile sensor networks. In *2008 American Control Conference*, pages 2678–2683, Seattle, WA, Jun. 2008. 13

[18] T. Gustavi, D. V. Dimarogonas, M. Egerstedt, and X. Hu. Sufficient conditions for connectivity maintenance and rendezvous in leader-follower networks. *Automatica*, 46(1):133–139, 2010. 13

[19] J. W. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion without mapping or global localization via local frontiers. *Autonomous Robots*, pages 1–15, 2011. 13

[20] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. On synchronous robotic networks - Part II: Time complexity of rendezvous and deployment algorithms. *IEEE Trans. on Automatic Control*, 52(12):2214–2226, 2007. 13

[21] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira. Sensing and coverage for a network of heterogeneous robots. In *47th IEEE Conf. on Decision and Control*, pages 3947–3952, Cancun, Mexico, Dec. 2008. 13

[22] G. Antonelli, F. Arrichiello, and S. Chiaverini. The entrapment/escorting mission for a multi-robot system: Theory and experiments. In *Proc. ieee/asme international conference on Advanced intelligent mechatronics*, pages 1–6, 2007. 14

[23] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1250–1257 vol.2, 1994. 15

[24] R. Kurazume, S. Hirose, S. Nagata, and N. Sashida. Study on cooperative positioning system (basic principle and measurement experiment). In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1421–1426 vol.2, 1996. 15

[25] R. Kurazume and S. Hirose. Study on cooperative positioning system: optimum moving strategies for cps-iii. In *Proc. IEEE International Conference on Robotics and Automation*, volume 4, pages 2896–2903 vol.4, 1998. 15

[26] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):43–52, 2000. 15

[27] H. Kato, K. Ishiguro and M. Barth. Identifying and localizing robots in a multi-robot system environment. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 966–971, 1999. 16

[28] D. Fox, W. Burgard, H. Kruppa, and Thrun S. Collaborative multi-robot localization. In *Proceedings of the 23rd Annual*

German Conference on Artificial Intelligence, pages 255–266, London, UK, 1999. Springer-Verlag. 17

[29] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000. 17, 58

[30] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Trans. on Robotics*, 18(5):781–795, 2002. 17, 18

[31] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *2005 IEEE Int. Conf. on Robotics and Automation*, pages 2797–2802, Barcelona, Spain, Apr. 2005. 18, 41

[32] A. Martinelli and R. Siegwart. Observability analysis for mobile robot localization. In *2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1471–1476, Edmonton, Canada, Aug. 2005. 18

[33] R. Hermann and A. Krener. Nonlinear controllability and observability. *Automatic Control, IEEE Transactions on*, 22(5):728 – 740, Oct 1977. 18

[34] X.S. Zhou and S.I. Roumeliotis. Determining the robot-to-robot relative pose using range-only measurements. In *2007 IEEE International Conference on Robotics and Automation*, pages 4025–4031, 2007. 18

[35] A. Howard, M. J. Matarić, and G. S. Sukhatme. An incremental deployment algorithm for mobile robot teams. In *2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2849–2854, Las Vegas, NV, Oct. 2003. 18

[36] X. S. Zhou and S. Roumeliotis. Determining the robot-to-robot 3d relative pose using combinations of range and bearing measurements (part II). In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 4736–4743, Shanghai, China, May. 2011. 20

[37] N. Trawny, X. S. Zhou, K. Zhou, and S. I. Roumeliotis. Inter-robot transformations in 3D. *IEEE Trans. on Robotics*, 26(2):225–243, 2010. 20

[38] A. Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *to appear in IEEE Trans. on Robotics*, 2011. 20, 116, 117

[39] T. Eren, P.N. Belhumeur, B.D.O Anderson, and A.S. Morse. A framework for maintaining formations based on rigidity. In *15th International Federation of Automatic Control (IFAC) World Congress*, 2002. 20

[40] T. Eren, P.N. Belhumeur, and A.S. Morse. Closing ranks in vehicle formations based on rigidity. In *41st Conference on Decision and Control*, volume 3, pages 2959–2964, 2002. 20

[41] T. Eren, W. Whiteley, A.S. Morse, P.N. Belhumeur, and B.D.O. Anderson. Sensor and network topologies of formations with direction, bearing, and angle information between agents. In *42nd Conference on Decision and Control*, volume 3, pages 3064–3069 Vol.3, 2003. 20, 21

[42] T. Eren, O.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In O.K. Goldenberg, editor, *Proc. INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673–2684 vol.4, 2004. 20, 21

[43] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Trans. on Mobile Computing*, 5(12):1663–1678, 2006. 20, 21

[44] A. Howard, M. J. Matarić, and G. S. Sukhatme. Putting the 'I' in 'team': An ego-centric approach to cooperative localization. In *2003 IEEE Int. Conf. on Robotics and Automation*, pages 868–892, Taipei, Taiwan, Sep. 2003. 23, 58

[45] A. Martinelli. Improving the precision on multi robot localization by using a series of filters hierarchically distributed. In *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1053–1058, San Diego, CA, Nov. 2007. 23

[46] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainly. *International Journal of Robotics Research*, 5(4):56–68, 1986. 24

[47] Merriam-webster online. http://www.merriam-webster.com/dictionary/tagging, last seen on Dec 2011. 27

[48] W. Miller. *Symmetry Groups and Their Applications.* Academic Press, 1972. 33

[49] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics.* The MIT Press, 2006. 42

[50] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002. 42

[51] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *2003 IEEE Int. Conf. on Robotics and Automation*, pages 1985–1991, Taipei, Taiwan, Sep. 2003. 43, 45, 47, 48

[52] X. S. Zhou and S. I. Roumeliotis. Robot-to-robot relative pose estimation from range measurements. *IEEE Trans. on Robotics*, 24(6):1379–1393, 2008. 58

[53] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 61

[54] D. Fontanelli, L. Ricciato, and S. Soatto. A fast ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements. In *IEEE 2007 International Conference on Automation Science and Engineering*, pages 597–602, 2007. 61

[55] R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. on Automatic Control*, 53(5):1203–1218, 2008. 107

[56] P. Martin and E. Salaün. The true role of accelerometer feedback in quadrotor control. In *2010 IEEE Int. Conf. on Robotics and*

Automation, pages 1623–1629, Anchorage, AK, May 2010. 107, 109

[57] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 17(12):1154–1166, 1995. 127

[58] R. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Trans. on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003. 136, 138, 142

[59] B. Vo and W. Ma. The gaussian mixture probability hypothesis density filter. *IEEE Trans. on Signal Processing*, 54(11):4091–4103, 2006. 138

[60] O. Erdinic, P. Willet, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the phd filters. *IEEE Trans. on Signal Processing*, 57(11):4232–4246, 2009. 138

[61] J. Mullane, B. Vo, and M. Adams. Rao-blackwellised phd slam. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 5410–5416, Anchorage, AK, May 2010. 138