Ground and Aerial Mutual Localization Using Anonymous Relative-Bearing Measurements

Paolo Stegagno, Marco Cognetti, Giuseppe Oriolo, Heinrich H Bülthoff, Antonio Franchi

Preprint version final version at http://ieeexplore.ieee.org/ Accepted for IEEE Transaction on Robotics 2016

Ground and Aerial Mutual Localization using Anonymous Relative-Bearing Measurements

Paolo Stegagno, Marco Cognetti, Giuseppe Oriolo, Heinrich H. Bülthoff, Antonio Franchi

***Abstract*—We present a decentralized algorithm for estimating mutual poses (i.e., relative positions and orientations) in a group of mobile robots. The algorithm uses only anonymous relative-bearing measurements obtainable, e.g., using onboard monocular cameras, and onboard motion measurements, such as inertial ones (acceleration and angular velocity). Onboard relative-bearing sensors supply anonymous measurements, i.e., they provide the directions along which other robots are located but each direction is not associated to any robot (identities are unknown). The issue of anonymity is often overlooked in theory but represents a real problem in practice, especially when employing onboard vision. The solution is first presented for ground robots, in SE(2), and then for aerial robots, in SE(3), in order to emphasize the difference between the two cases. The proposed method is based on a two-step approach, the first uses instantaneous geometrical arguments on the anonymous measurements in order to retrieve the most likely unscaled relative configurations together with the identities, the second uses a numeric Bayesian filtering in order to take advantage of the motion model over time and to retrieve the scale. The proposed method exhibits robustness w.r.t. false positives and negatives of the robot detector. An extensive experimental validation of the algorithm is performed using Khepera III ground mobile robots and quadrotor aerial robots.**

***Index Terms*—Bearing/vision-based mutual localization of ground and flying robots, mutual localization, bearing-only, bearing/visual-inertial sensor fusion, unknown data-association.**

I. Introduction

The designer of a decentralized multi-robot system must take into account the problem of the mutual localization among the agents, that is, each agent must be able to detect its teammates and compute an estimate of their configurations. This knowledge is mandatory for all robots in order to cooperate and exchange data, and is a prerequisite to perform in a decentralized way higher level tasks such as formation control [1], connectivity maintenance [2], cooperative exploration [3] and transportation [4], distributed estimation [5], coverage and sensing [6], [7].

The first efforts to solve this issue can be ascribed to [8], in which the authors propose a system for the estimation of the configuration of a team of robots that uses relative measurements collected by the robots themselves. The problem introduced by this work subsequently evolved in what is now known as *Cooperative Localization* (CL) which consists in estimating the configuration of the agents in a *common* fixed frame. Many authors have addressed CL through the use of particle filters, Kalman filters and MAP estimators [9], [10], [11], [12], [13], and the general result of those works is the improved accuracy of the localization of the entire system by the inclusion of relative measurements among the robots. However, we state that agreeing in advance on a common fixed frame already implies a form of centralized consensus between the agents, and should be avoided if possible in the development of a decentralized system.

A possible workaround is to assume that each agent expresses configurations and measurements in its own fixed frame. Then, it is necessary for each robot to solve the *Absolute Mutual Localization* (AML) i.e., to estimate the change of coordinates among its fixed frame and the fixed frames of the other robots. Otherwise, assuming that each robot is equipped with its own attached frame leads to the definition of the *Relative Mutual Localization* (RML) as the problem of estimating the change of coordinates among the agents' moving frames. Some authors have studied the solvability of the RML problem, investigating the minimal sets of data needed to determine the robot-to-robot 3D relative pose [14] and proposing observability analysis and estimators [15], [16].

Most of the previous works assume that robot-to-robot measurements come with the identity of the measured robot, or equivalently that the estimation is performed *pairwise*. The situation in which the identities of the measured robots is not retrieved by the sensor is known in robotics literature as unknown data-association or anonymous measurements. In [17], [18], we have addressed the problem of 2D RML with anonymous position measurements plus odometric data. The concept behind these papers is that the capability of achieving mutual localization with anonymous measurements increases the level of decentralization, widens the field of applicability and adds flexibility to the system. Hence, we proposed a two-phase localization system composed by (1) a multiple registration algorithm that compute all the possible changes of coordinates among the agents' relative frames using geometrical arguments to invert the measurement map (2) a bank of particle filters to account for the uncertainty and ambiguity of the process (see also [19]). The whole localization system relies on the exchange of measurements among the robots through communication.

In an effort of further generalization, in [20] and [21] we

Paolo Stegagno and Heinrich H. Bülthoff are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {paolo.stegagno, hhb}@tuebingen.mpg.de. Heinrich H. Bülthoff is also with the Department of Brain and Cognitive Engineering, Korea University, Seoul, 136-713 Korea.

Marco Cognetti and Giuseppe Oriolo are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy {cognetti, oriolo}@diag.uniroma1.it.

Antonio Franchi is with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, afranchi@laas.fr.

have considered 2D and 3D extensions to the case of bearing-only (rather than range-and-bearing) measurements, to allow the use of non-depth sensors such as simple cameras, whose employment widens the field of applicability of our system. The objective of this work is to offer a comprehensive point of view on bearing-only mutual localization with anonymous measurements. At this aim, we will tackle the problem both in 2D and 3D worlds considering three among the most significant settings for robotics applications, defined as follows:

Scenario A (2D, linear and angular velocity measurements): the robots lie on a plane, and are able to measure both their own linear and angular velocity in body frame;

Scenario B (3D, linear and angular velocity measurements): the robots live in a 3D world, and are able to measure both their own linear and angular velocity in body frame.

Scenario C (3D, linear acceleration and angular velocity measurements): the robots live in a 3D world, and are able to measure their own linear acceleration and angular velocity in body frame;

Note that Scenario A can be regarded (and will be formalized) as a particular case of Scenario B. However, as we will explain in the paper, not having the third dimension causes a loss of information in the measurements which calls for a different metric while evaluating the result of a geometric algorithm to solve the problem. In addition, with our system architecture we will be able to introduce an additional constraint in the 3D case. Moreover, the introduction of Scenario A is functional to the explanation of the general case, and provides more insights on some design choices.

Preliminary works regarding only Scenarios A and C appeared in our conference papers [20] and [21]. With respect to these works, in this paper we also address the newly formulated Scenario B, which is an interesting compromise among the other two, allowing us to produce a detailed discussion on the similarities and differences among the different Scenarios. Moreover, we provide a theoretical discussion on the consequences of anonymity, we present new experiments and compare the results of the different solutions.

The rest of the paper is organized as follows. In Section II we introduce the working principle and a qualitative description of the system aimed at the estimation of the relative configurations among the robots in the three Scenarios. In Section III we draw some conclusions on the consequences of the assumption of anonymous measurements. In Section IV we formally introduce the considered mutual localization problems, while in Section V we address Scenario A. The developed system is then modified in Section VI to tackle the increased dimensionality of Scenarios C and B. We discuss some important aspects of the system, as the computational complexity, in Section VII. Experimental results for the three Scenarios are presented in Section VIII, and some conclusions are given in Section IX.

II. System Overview

In general, the localization of a team of robots requires both proprioceptive and exteroceptive sensors. The use of GPS or motion capture system, and more in general off-board sensors,

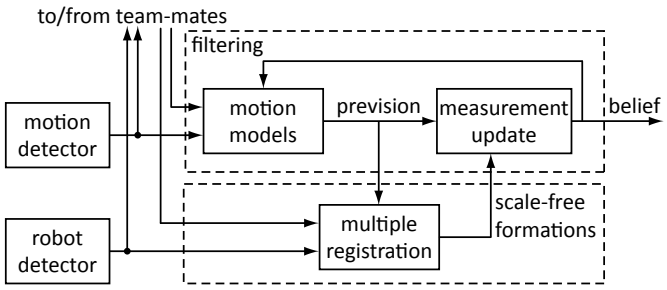


Fig. 1: The working principle of the developed mutual localization system.

bounds the use of the system to the presence of an external infrastructure. Hence, the choice of the sensor equipment of the robots will be crucial for the design of the system, and will vary depending on the considered Scenario. However, it is possible to draw some similarities on the functionalities of the sensors on different platforms as the ones moving in different spaces (2D and 3D).

In fact, in all Scenarios the generic robot will carry on:

- a *motion detector*, which provides some proprioceptive measurements of its own motion;
- a *robot detector*, which is able to detect the bearing of the adjacent robots and provides some exteroceptive measurements on their relative configurations, without detecting their identities; hence, the robot detector will provide at each time-step a set of untagged relative measurements.

In our system we want to ensure robustness against single-robot failures adopting a decentralized approach. For this reason we opted for a robo-centric estimation, in which each robot computes the relative configurations of the team-mates in its own frame of reference. In this way it is also addressed the problem that the robots do not have a common fixed frame of reference, and in general are not able to compute it. In fact, although some techniques for the computation of common frames of reference are known (see, e.g., [22]), to the best of our knowledge those techniques do not account for the situation of anonymous measurements.

Most previous works in mutual localization apply Bayesian filters and/or geometrical considerations to perform the estimate of the relative configurations of the robot. However, the standard use of those tools is favored by the assumption of known data association of the measurements. Instead, in this work we want to tackle the more challenging situation of anonymous measurements, which cannot be used directly to feed a filter or a geometrical algorithm. In fact, they require a foregoing step to recover the correspondences among the measurements and the measured robots.

Although this step can be implemented through methods such as Maximum Likelihood [23] and Joint Compatibility Test [24], those approaches are likely to be successful only in situations in which the robots are sparse, so that at each time-step the sets of measurements contain one or few elements and the beliefs on their configurations do not overlap. Instead, they are likely to make a high number of association errors if the robots operate close to each other, hence the beliefs are

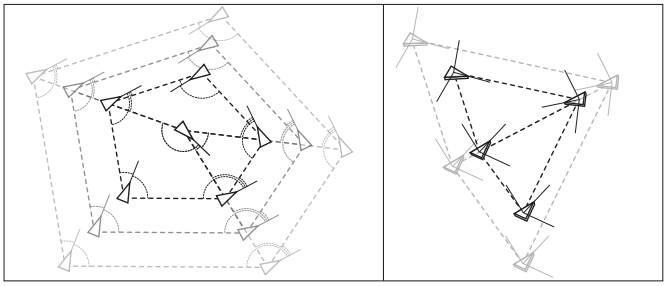


Fig. 2: Examples of 2D (left) and 3D (right) formations up to a scaling factor.

overlapping. Moreover, they do not solve the problem of the initialization, when there is no available belief.

In our proposed solution each robot of the team runs an independent mutual localization system as the one depicted in Fig. 1. In order to perform the mutual localization, each robot collects at each time-step the sensor perception communicated from the team-mates. Then, all the measurement sets provided by the robot detectors are registered by a custom algorithm (multiple registration) on the basis of geometrical computations to reconstruct the correspondences among measurements and measured robots. The multiple registration does not act as a standard data association technique, since it also reconstructs part of the unmeasured state (i.e., the relative orientation). In particular, since it processes only angular measurements, and since the multiple registration problem may admit multiple solutions (see Sec. III), its output is a set of most likely formations up to a scaling factor. In fact, no distance information is offered by the robot detector, hence it is not possible to reconstruct the scale of the registered formations.

The problem of discriminating among multiple solutions, as well as the reconstruction of the scale of the formation and the rejection of noise is demanded to the filtering stage. While the time update of the filtering is performed using appropriate models and the measurements from the motion detector of all robots, the output of the multiple registration is used to feed the measurement update. Note that the prediction computed by the filtering stage is used in the multiple registration to help in the reconstruction of the correspondences of measurements and robots, and prune the multiple solution before the filtering stage.

III. Consequences of Anonymity

One of the key point of our mutual localization system is the design of a multiple registration algorithm which reconstructs the correspondences among the measurements and the measured robots on the basis of geometric considerations.

It is very well known from literature that in general this problem admits multiple solutions [25] even in the simplified situation of non-anonymous position measurements and 2D world. In such case, frameworks (i.e.: spatial configuration and measurement graph) which admit a single solution are said to be rigid, and there are known conditions and algorithms to individuate rigid and non-rigid frameworks.

In our previous paper [19] we have shown that, in case of anonymous position measurements, the multiple registration problem applied in rotational symmetric configurations admits multiple solutions even if the underlying framework is rigid (i.e.: the framework is rigid considering the non-anonymous version of the measurements). Hence, the loss of the identity information results in the loss of unique solvability. Here, avoiding a formal description of the problem, we want to show some relevant properties that will be useful in the design of the algorithm. We will use the following abbreviations: MRB (MRAB) indicates multiple registration with (anonymous) bearing measurements, MRP (MRAP) indicates multiple registration with (anonymous) position (i.e.: bearing plus distance) measurements.

Property 1: *any configuration obtained through a change of scale of a solution of the MRAB problem is again a solution of the same problem*. Two simple examples in 2D and 3D (Fig. 2) show that bearing measurements are invariant w.r.t. the scale of the original formation. This is valid in general, and implies that a MRAB problem always admits infinite solutions. In the following analysis, we will not consider the infinite solutions derived by the unknown scale of the formation in case of bearing measurements. When referring to a particular solution to this problem we will implicitly indicate the class of equivalence of all the solutions up to a scale factor.

Property 2: *a solution of the MRAP problem is also a solution of the equivalent MRAB problem*, while the opposite is not necessarily true. In fact, the additional distance information contained in the position measurements constitutes additional constraints which are not present in the case of bearing measurements. Conversely, all constraints given by the bearing measurements are contained also in the position measurements. Consider the examples in Fig. 3. The first row shows an equilateral triangular formation which is rigid, but admits two different solutions both in case of anonymous position and bearing measurements. In the example, the robots A_j and A_k can be switched as vertices of the triangle, keeping the third one on A_i. Then, in this case, the loss of the metric information does not impact on the type and number of the solutions. The same does not apply when considering isosceles or scalene triangles, analyzed in the second and third rows. In both cases the solution to the problem with position measurements is unique, while it is possible to exchange the role of A_j and A_k in case of bearing measurements.

Property 3: *Consider the 2D MRAB problem. If the measurement sets of two robots coincide up to a 2D rotation of all the measurements then the 2D MRAB problem admits multiple solutions*. It is easy to show (e.g., Fig. 3) that if the measurement sets of two robots are equal up to a rotation, then the role of the two robots can be exchanged in a solution in order to create a different solution. This property can be generalized to the 3D MRAB problem: *if the measurement sets of two robots coincide up to a 3D rotation of all the measurements then the 3D MRAB problem admits multiple solutions.* Figure 4a shows an example of such situation.

Note that in the 3D case the rotation can be around any axis, hence the third dimension increases in general the difficulty of the problem. However, if the robots are able to agree on a common direction in space, they can restrict the problem on the plane perpendicular to that direction, bringing it back to

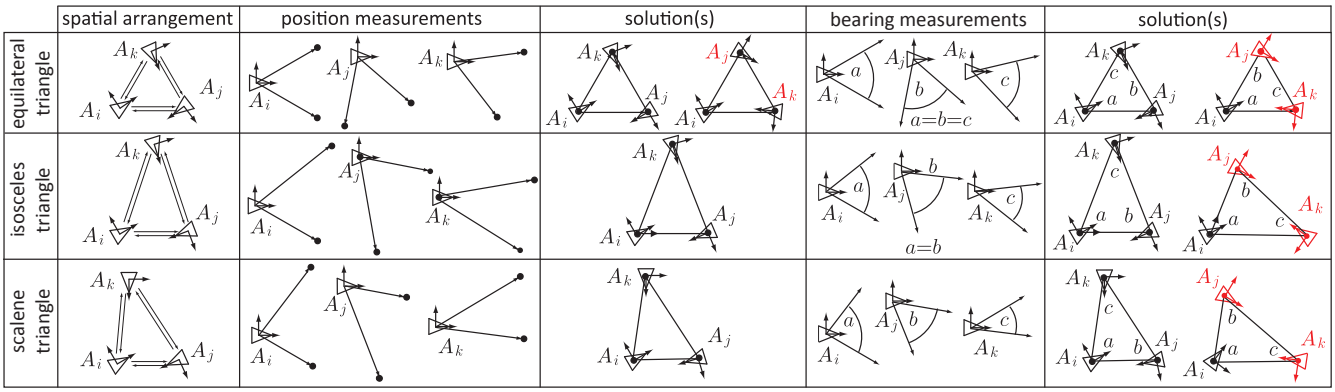


Fig. 3: Comparison of the solvabilty of the multiple registration problem in case of triangular formation using anonymous position or bearing measurements.

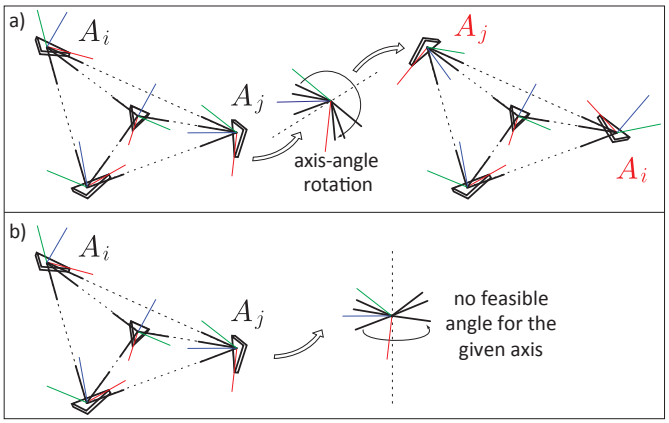


Fig. 4: a) Example of axis-angle rotation in the 3D MRAB problem; b) same situation with a common direction in space available to the robots.

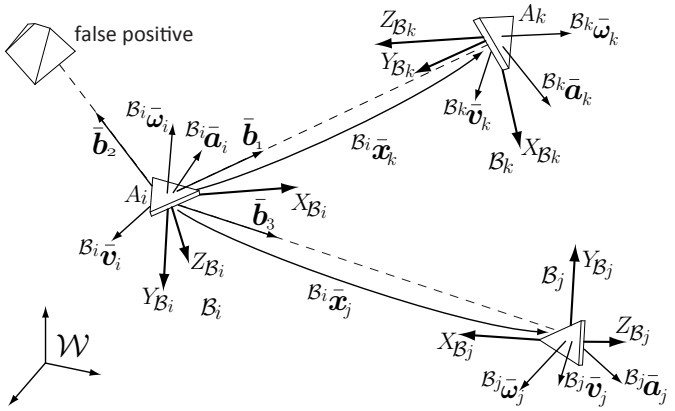


Fig. 5: Mutual localization with anonymous bearing measurements. Triangles are robots with their attached frame, velocity, acceleration, angular velocity measurements and robot detectors with false positives.

2D (Figure 4b). In particular, if the sensor equipment includes an IMU, each robot is able to estimate the gravity vector and obtain a direction known also by the other robots. In this way, the only feasible axis direction for the axis-angle pair becomes the direction of the gravity vector and the 3D MRAB problem can be restricted to the horizontal plane. The considerable simplification introduced by this approach is only one of two advantages. The second is that, as we will show in detail in Sec. VI, once the problem is reduced to 2D, the vertical information contained in the bearing measurements (the angle that the bearing forms with the direction of the gravity vector) provides additional constraints that reduces the number of solutions of the problem.

IV. Problem Formulation

A. Notation in the generic 3D Case

We refer the reader to Fig. 5 for illustration. Consider a system of n robots $A_1, \dots, A_n$, with n unknown (hence, it may vary during the operation). Let $K = \{1, \dots, n\}$ be the set of robot indices and $K_i = K \setminus \{i\}$. The robots are modeled as rigid bodies in $SE(3)$. 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 A_i. Body frames conform to the North-East-Down (NED) convention, as common in the aeronautic field.

The configuration of A_i is represented by the position ${}^\mathcal{W}\boldsymbol{p}_i \in \mathbb{R}^3$ of $O_{\mathcal{B}_i}$ in $\mathcal{W}$ and the rotation matrix ${}^\mathcal{W}\boldsymbol{R}_{\mathcal{B}_i} \in SO(3)$ between $\mathcal{W}$ and $\mathcal{B}_i$. Denote with $\boldsymbol{R}_X(\cdot), \boldsymbol{R}_Y(\cdot), \boldsymbol{R}_Z(\cdot)$ the canonical rotation matrices about the axes X, Y, Z respectively. Then ${}^\mathcal{W}\boldsymbol{R}_{\mathcal{B}_i}$ can be written as ${}^\mathcal{W}\boldsymbol{R}_{\mathcal{B}_i} = \boldsymbol{R}_Z({}^\mathcal{W}\psi_i)\boldsymbol{R}_Y({}^\mathcal{W}\theta_i)\boldsymbol{R}_X({}^\mathcal{W}\phi_i)\boldsymbol{R}_X(\pi)$, being ${}^\mathcal{W}\psi_i, {}^\mathcal{W}\theta_i, {}^\mathcal{W}\phi_i \in \mathbb{S}^1$ the *yaw*, *pitch*, and *roll* angles of A_i respectively, $\boldsymbol{R}_X(\pi)$ the rotation matrix between a NED frame to its corresponding NWU (North-West-Up) frame, and $\mathbb{S}^1$ denotes the unit circle.

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

$$ {}^{\mathcal{B}_i}\boldsymbol{p}_j = {}^\mathcal{W}\boldsymbol{R}_{\mathcal{B}_i}^T({}^\mathcal{W}\boldsymbol{p}_j - {}^\mathcal{W}\boldsymbol{p}_i) \tag{1} $$

$$ {}^{\mathcal{B}_i}\boldsymbol{R}_{\mathcal{B}_j} = {}^\mathcal{W}\boldsymbol{R}_{\mathcal{B}_i}^T\,{}^\mathcal{W}\boldsymbol{R}_{\mathcal{B}_j} \tag{2} $$

and we denote by ${}^{\mathcal{B}_i}\boldsymbol{x}_{\mathcal{B}_j} = \{{}^{\mathcal{B}_i}\boldsymbol{p}_j, {}^{\mathcal{B}_i}\boldsymbol{R}_{\mathcal{B}_j}\}$ the *full relative configuration* of A_j in $\mathcal{B}_i$. Computing an estimate of ${}^{\mathcal{B}_i}\boldsymbol{x}_{\mathcal{B}_j}$ is the final goal of the mutual localization.

It is convenient to introduce the frame $\mathcal{C}_i = \{O_{\mathcal{C}_i}, X_{\mathcal{C}_i}, Y_{\mathcal{C}_i}, Z_{\mathcal{C}_i}\}$, defined as the frame having the same

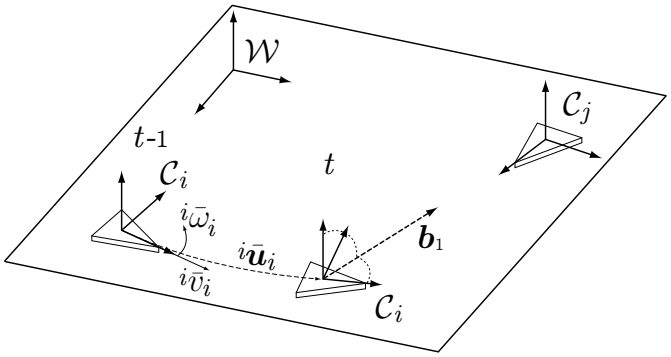


Fig. 6: Mutual localization with anonymous bearing measurements. The differences of the 2D case w.r.t. the other Scenarios.

origin as $\mathcal{B}_i$ ($O_{\mathcal{C}_i} \equiv O_{\mathcal{B}_i}$) and such that ${}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{C}_i} = \boldsymbol{R}_Z({}^{\mathcal{W}}\psi_i)$. Hence, the rotation matrix between $\mathcal{B}_i$ and $\mathcal{C}_i$ is

$$ {}^{\mathcal{C}_i}\boldsymbol{R}_{\mathcal{B}_i} = \boldsymbol{R}_Y({}^{\mathcal{W}}\theta_i)\boldsymbol{R}_X({}^{\mathcal{W}}\phi_i)\boldsymbol{R}_X(\pi). \tag{3} $$

In principle, the whole localization system could be expressed in NED frame, but the introduction of its corresponding NWU will be useful to develop a common notation with the 2D case. Note that the plane $X_{\mathcal{C}_i}Y_{\mathcal{C}_i}$ is parallel to the plane $X_{\mathcal{W}}Y_{\mathcal{W}}$, or equivalently $Z_{\mathcal{C}_i} \parallel Z_{\mathcal{W}}$. In general, we will use the left superscripts ${}^{\mathcal{B}_i}(\cdot)$ and ${}^{i}(\cdot)$ to denote quantities $(\cdot)$ expressed in $\mathcal{B}_i$ and $\mathcal{C}_i$ respectively.

B. Particularization to the 2D Case

If the robots lie on the $X_{\mathcal{W}}Y_{\mathcal{W}}$ plane the third component of the position ${}^{\mathcal{W}}\boldsymbol{p}_i$, as well as the roll ${}^{\mathcal{W}}\phi_i$ and pitch ${}^{\mathcal{W}}\theta_i$ angles are equal to zero. Hence, in Scenario A the configuration of A_i is completely determined as ${}^{\mathcal{W}}\tilde{\boldsymbol{x}}_i = ({}^{\mathcal{W}}\tilde{\boldsymbol{p}}_i^T, {}^{\mathcal{W}}\psi_i)^T$, where ${}^{\mathcal{W}}\tilde{\boldsymbol{p}}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} {}^{\mathcal{W}}\boldsymbol{p}_i \in \mathbb{R}^2$. Therefore in Scenario A equation (3) reduces to

$$ {}^{\mathcal{C}_i}\boldsymbol{R}_{\mathcal{B}_i} = \boldsymbol{R}_X(\pi). \tag{4} $$

For a visualization of the peculiarities of Scenario A refer to Fig 6.

C. Sensor and Communication Equipment

For what concerns the sensor equipment of the robots, we must distinguish among the different Scenarios. In Scenario A, which is compatible with a team of general purpose wheeled robots, A_i's motion detector provides the measurements: ${}^{i}\bar{\boldsymbol{v}}_i$, linear velocity of A_i in the $X_{\mathcal{W}}Y_{\mathcal{W}}$ plane in $\mathcal{C}_i$; and ${}^{i}\bar{\boldsymbol{\omega}}_i$, angular rate about the axis $Z_{\mathcal{W}}$. This can be easily done, e.g., through the use of encoders on the wheels. The integration of ${}^{i}\bar{\boldsymbol{v}}_i$, ${}^{i}\bar{\boldsymbol{\omega}}_i$ between two sampling instants $t-1$ and t, provides a measurement of the robot displacement $\bar{\boldsymbol{u}}_i^t$ on the plane between $t-1$ and t. Hence, the motion detector in this case will be particularized as an *odometer*.

Scenarios B and C are more complex to address, since they are compatible with a wider range of systems, from swarms of aerial vehicles to a multitude of underwater vehicles. A general purpose motion detector which is usually present on those platforms is the IMU (Inertial Measurement Unit), that provides to A_i noisy measurements ${}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i$, ${}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i$ of its proper acceleration ${}^{\mathcal{B}_i}\boldsymbol{a}_i$ and angular velocity ${}^{\mathcal{B}_i}\boldsymbol{\omega}_i$ in body frame, defined by

$$ {}^{\mathcal{B}_i}\boldsymbol{a}_i = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}_i}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_i - g\boldsymbol{e}_3) \tag{5} $$

$$ {}^{\mathcal{B}_i}\boldsymbol{\omega}_i = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}_i}^T {}^{\mathcal{W}}\boldsymbol{\omega}_i \tag{6} $$

where g is the gravity acceleration and $\boldsymbol{e}_3 = (0\ 0\ 1)^T$.

In Scenario B only, we assume that, in addition to the IMU, measurements ${}^{\mathcal{B}_i}\bar{\boldsymbol{v}}_i$ of the linear velocity ${}^{\mathcal{B}_i}\boldsymbol{v}_i$ are also available. Linear velocities can be recovered by the use of a camera and techniques based on the image optic flow or visual odometry (e.g: [26],[27], [28]).

For the robot detector, we are able to give a generic formulation independent from the Scenario by assuming the same type of sensor in all Scenarios. In particular, we assume that each A_i is able to obtain at each time t anonymous measurements of the *relative bearings*

$$ {}^{\mathcal{B}_i}\boldsymbol{b}_j = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}_i}^T \frac{{}^{\mathcal{W}}\boldsymbol{p}_j - {}^{\mathcal{W}}\boldsymbol{p}_i}{\|{}^{\mathcal{W}}\boldsymbol{p}_j - {}^{\mathcal{W}}\boldsymbol{p}_i\|} \tag{7} $$

of all robots A_j falling in the perception set D_p attached to $\mathcal{B}_i$.

The relative bearing ${}^{\mathcal{B}_i}\boldsymbol{b}_j \in \mathbb{S}^2$ is the unit-norm vector pointing toward the center of mass of A_j, expressed in $\mathcal{B}_i$. A measure of (7) can be obtained, for example, by using a feature tracking algorithm on the images provided by a calibrated camera mounted on the A_i.

Note that a relative bearing $\boldsymbol{b}$ can be equivalently represented by the use of an *azimuth* and *zenith-distance* pair $(\alpha, \zeta) \in [0, 2\pi) \times [0, \pi)$, since they are related by

$$ \boldsymbol{b} = (\sin\zeta\cos\alpha \quad \sin\zeta\sin\alpha \quad \cos\zeta)^T. \tag{8} $$

In addition to being subject to false positives (due to objects that look like robots) and false negatives (due to line of sight occlusions), relative bearing measurements do not contain the identities of the measured robots (see Fig. 5). Hence, the output of the robot detector is a set ${}^{\mathcal{B}_i}B^t = \{\bar{\boldsymbol{b}}_1, \bar{\boldsymbol{b}}_2, \ldots\}$ of measurements whose ordering has no relation to the robot indexing. In addition, each measurement may or may not refer to an actual robot due to the possible presence of false positives. For this reason, in the following we will generically refer to relative measurements as *features*. Eventually, each feature in ${}^{\mathcal{B}_i}B^t$ identifies a ray originating from A_i on which another robot may lay.

The only difference between the Scenarios is that in Scenario A the third component of a bearing measurement $\bar{\boldsymbol{b}}_k$ is always equal to zero, being its corresponding zenith-distance ζ always equal to $\pi/2$. Hence, in Scenario A each bearing can be represented by the use of a unit-norm vector in $\mathbb{S}^1$, or equivalently by mean of the only azimuth α angle. Note that, in this Scenario, it is immediate to express the features in $\mathcal{C}_i$ using the rotation matrix given by equation (4), while in Scenario B–C this operation requires the knowledge of the roll and pitch angles of A_i. We will denote with ${}^{i}B^t$ the set of features expressed in $\mathcal{C}_i$ at time t.

The equipment of A_i is completed by a *communication module* that can send/receive data to/from any other robot,

symbol	description	Scenario
A_i	i-th robot	ABC
K	set of robot indices	ABC
$\mathcal{W}$	world frame	ABC
$O_{\mathcal{F}}, X_{\mathcal{F}}, Y_{\mathcal{F}}, Z_{\mathcal{F}}$	origin and axes of the generic frame $\mathcal{F}$	ABC
${}^{\mathcal{F}}\boldsymbol{p}_i$	position of A_i in $\mathcal{F}$	ABC
${}^{\mathcal{F}}\boldsymbol{R}_{\mathcal{G}}$	rotation matrix from the frame $\mathcal{G}$ to the frame $\mathcal{F}$	ABC
${}^{\mathcal{W}}\psi_i$	yaw angle of A_i	ABC
${}^{\mathcal{W}}\theta_i, {}^{\mathcal{W}}\phi_i$	pitch and roll angles of A_i	BC
${}^{\mathcal{F}}\boldsymbol{x}_{\mathcal{G}}$	configuration (position and orientation) of $\mathcal{G}$ in $\mathcal{F}$	BC
${}^{\mathcal{F}}\tilde{\boldsymbol{p}}_{\mathcal{G}}$	reduced position of $\mathcal{G}$ in $\mathcal{F}$	BC
${}^{\mathcal{F}}\tilde{\boldsymbol{x}}_{\mathcal{G}}$	reduced configuration $({}^{\mathcal{F}}\tilde{\boldsymbol{p}}_{\mathcal{G}}^T, {}^{\mathcal{F}}\psi_{\mathcal{G}})$ of $\mathcal{G}$ in $\mathcal{F}$	BC
$\boldsymbol{v}_i$	linear velocity of A_i	ABC
$\boldsymbol{\omega}_i$	angular velocity of A_i	ABC
$\boldsymbol{u}_i^t$	displacement on the plane of A_i between $t-1$ and t	A
$\boldsymbol{a}_i$	linear acceleration of A_i	BC
${}^iB$	feature set relative bearing of A_j	ABC
$\boldsymbol{b}_j$	relative bearing of A_j	ABC
$\bar{\diamond}$	measurement of $\diamond$	ABC
$\hat{\diamond}$	estimate of $\diamond$	ABC
D_p	perception set	ABC
D_c	communication set	ABC
N_i	communication neighbors	ABC
α, ζ	azimuth and zenith-distance angles	ABC

TABLE I: Main symbols with the Scenarios where they are used.

provided that it lies in the *communication set* D_c rigidly attached to $\mathcal{B}_i$. In general, the robots use the communication module to exchange the sensor readings, and each robot signs its communication packet. However, since the content of the packets is dependent from the Scenario, we will give its detailed description while addressing each specific problem. A list of the main symbols used in each Scenario is given in Table I.

Eventually, we can formulate the relative mutual localization problem from the point of view of the generic robot A_i. Denote with N_i^t the *neighbors* of A_i at time t, i.e., the set of robots from which it is receiving communication. In a probabilistic framework, the RML problem with anonymous bearing measurements requires A_i to compute at each t a belief about the relative configuration ${}^{\mathcal{B}_i}\boldsymbol{x}_j$ of all A_j that have been in communication with A_i, (i.e.: such that $j \in N_i^{1:t} = N_i^1 \cup N_i^2 \cup \ldots \cup N_i^t$) given all the measurements from the robot and motion detectors of all communicating robots. We will formalize later the RML problem in each specific Scenario.

V. Scenario A

In Scenario A the robots lie on the $X_{\mathcal{W}}Y_{\mathcal{W}}$ plane. Using the definition of the motion and robot detector for this Scenario, the system architecture explained in Fig. 1 can be particularized as shown in Fig. 7. Hence, in this case the communication packet sent at time t by the generic robot A_i will contain, in addition to the robot signature,

1) the displacement $\bar{\boldsymbol{u}}_i^t$;
2) the feature set ${}^iB^t$;

and the RML problem can be formalized as

Problem 1: (2D RML with anonymous bearing measurements) For $t = 1, 2, \ldots$ and $j \in N_i^{1:t}$, compute the following *belief*:

$$\mathrm{bel}({}^i\tilde{\boldsymbol{x}}_j^t) = p({}^i\tilde{\boldsymbol{x}}_j^t | \bar{\boldsymbol{u}}_i^{1:t}, {}^iB^{1:t}, \{\bar{\boldsymbol{u}}_j^{1:t}, {}^jB^{1:t}\}_{j \in N_i^{1:t}}). \quad (9)$$

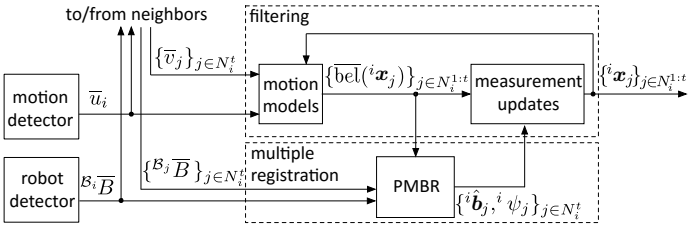


Fig. 7: Scheme of the mutual localization system that runs on A_i in Scenario A.

Now, we can particularize the multiple registration and the filtering stages for this Scenario.

A. Multiple Registration

Multiple registration in this scenario is performed by PMBR, a probabilistic multiple registration algorithm run by A_i at each time instant t to feed the measurement update of the particle filters (see Fig. 7). 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 ${}^iB^t$, $\{{}^jB^t\}_{j\in N_i^t}$ and the current beliefs $\{\overline{\mathrm{bel}}({}^i\tilde{\boldsymbol{x}}_j^t)\}_{j\in N_i^t}$ computed in the particle filters through the motion model of the robots, PMBR derives an estimate ${}^i\hat{\boldsymbol{\xi}}_j^t$ of the relative bearing-orientation of A_j, $j \in N_i^t$, w.r.t. A_i.

The basic steps of PMBR reported in the pseudo-code in Algorithm 1 are explained by means of an illustrative example in Fig. 8.

line 1) Difference angles representation: consider the configuration of the multi-robot system shown in Fig. 8a, with the corresponding feature sets in Fig. 8b. Note that each pair of features (bearing angles) in the same feature set can be equivalently represented by their difference angle. Let B a feature set, then the equivalent difference set D is defined as

$$D = \{\boldsymbol{d}_{rs} = \boldsymbol{b}_r - \boldsymbol{b}_s | \boldsymbol{d}_{rs} < \pi, \forall \{\boldsymbol{b}_r, \boldsymbol{b}_s\} \subseteq B\} \quad (10)$$

and in the pseudo-code is computed through the function $\mathrm{computeDiffAngleSet}(\diamond)$. Note that if B has m elements then D has at most $m(m-1)/2$ elements. In fact, there are $m(m-1)/2$ pairs $\boldsymbol{b}_r$, $\boldsymbol{b}_s$, and each pair originates two difference angles, one larger and one smaller than π, with only the second being part of D. If $\boldsymbol{b}_r - \boldsymbol{b}_s = \pi$, no difference angle is added to D.

line 2) Triangle finding: now take a triplet of robots that 'see' each other, e.g., A_i, A_j, A_k, and make A_h 'disappear' for a moment. Each robot in the triplet sees only two anonymous 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) $\boldsymbol{d}_1 \in D_i$, $\boldsymbol{d}_2 \in D_j$, $\boldsymbol{d}_3 \in D_k$ whose sum is π, with a certain tolerance τ_1 needed to account for measurement noise:

$$\boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 = \pi \pm \tau_1 \quad (11)$$

Algorithm 1: PMBR

input : feature sets ${}^iB^t$, $\{{}^jB^t\}_{j\in N_i^t}$, beliefs $\{\overline{\text{bel}}({}^i\tilde{\boldsymbol{x}}_j^t)\}_{j\in N_i^t}$

output: relative bearing-orientation estimates $\{{}^i\hat{\boldsymbol{\xi}}_j^t\}_{j\in N_i^t}$

init : $\mathcal{T}_{\text{raw}}=\varnothing$, $\mathcal{T}_{\text{rat}}=\varnothing$, $\mathcal{T}_{\text{irr}}=\varnothing$, $\mathcal{T}_{\text{best}}=\varnothing$, $\mathcal{S}=\varnothing$

1 **foreach** $j\in\{i\}\cup N_i^t$ **do** ${}^jD^t=\text{computeDiffAngleSet}({}^jB^t)$

2 **foreach** $\{D_1,D_2,D_3\}\subseteq\{{}^iD^t\}\cup\{{}^jD^t\}_{j\in N_i^t}$ **do**
  foreach $\{\boldsymbol{d}_1\in D_1,\boldsymbol{d}_2\in D_2,\boldsymbol{d}_3\in D_3\}$ **do**
    if $\boldsymbol{d}_1+\boldsymbol{d}_2+\boldsymbol{d}_3=\pi\pm\tau_1$ **then**
      $(T_a,T_b)=\text{computeTriangles}(\boldsymbol{d}_1,\boldsymbol{d}_2,\boldsymbol{d}_3)$
      $\mathcal{T}_{\text{raw}}=\mathcal{T}_{\text{raw}}\cup\{T_a,T_b\}$

3 **foreach** $T\in\mathcal{T}_{\text{raw}}$ **do**
  if $\text{count3Intersections}(T)\geq\text{th}_1$ **then** $\mathcal{T}_{\text{rat}}=\mathcal{T}_{\text{rat}}\cup\{T\}$

4 **foreach** $T\in\mathcal{T}_{\text{rat}}$ **do**
  if $\boldsymbol{d}_1\in{}^iD^t$ **and** $\text{areIrreconcilable}(T,T'),\forall T'\in\mathcal{T}_{\text{irr}}$ **then**
    $\mathcal{T}_{\text{irr}}=\mathcal{T}_{\text{irr}}\cup\{T\}$

5 **foreach** $T\in\mathcal{T}_{\text{irr}}$ **do**
  if $\text{beliefRating2D}(T)\geq\text{th}_2$ **then** $\mathcal{T}_{\text{best}}=\mathcal{T}_{\text{best}}\cup\{T\}$

6 **foreach** $T\in\mathcal{T}_{\text{best}}$ **do** $\mathcal{S}=\mathcal{S}\cup\text{branch}(T,\mathcal{T}_{\text{rat}})$

7 **foreach** $j\in N_i^t$ **do** ${}^i\hat{\boldsymbol{\xi}}_j^t=\text{extractAziOri}(\mathcal{S})$

function $\text{branch}(\text{partialSolution } S,\text{triangleSet } \mathcal{T}_{\text{rat}})$:

init : $\mathcal{S}_{\text{new}}=\varnothing$, $\mathcal{S}_{\text{rat}}=\varnothing$, $\mathcal{S}_{\text{irr}}=\varnothing$, $\mathcal{S}_{\text{best}}=\varnothing$

foreach $T\in\mathcal{T}_{\text{rat}}$ **do**
  if $\text{commonSide}(S,T)$ **and** $T\notin S$ **then**
    $\mathcal{S}_{\text{new}}=\mathcal{S}_{\text{new}}\cup\{\text{join}(S,T)\}$

foreach $S\in\mathcal{S}_{\text{new}}$ **do**
  if $\text{count3Intersections}(S)\geq\text{th}_3$ **then** $\mathcal{S}_{\text{rat}}=\mathcal{S}_{\text{rat}}\cup\{S\}$

foreach $S\in\mathcal{S}_{\text{rat}}$ **do**
  if $\text{areIrreconcilable}(S,S'),\forall S'\in\mathcal{S}_{\text{irr}}$ **then** $\mathcal{S}_{\text{irr}}=\mathcal{S}_{\text{irr}}\cup\{S\}$

foreach $S\in\mathcal{S}_{\text{irr}}$ **do**
  if $\text{beliefRating2D}(S)\geq\text{th}_4$ **then** $\mathcal{S}_{\text{best}}=\mathcal{S}_{\text{best}}\cup\{S\}$

if $\mathcal{S}_{\text{best}}=\varnothing$ **then return** S

else foreach $S\in\mathcal{S}_{\text{best}}$ **do** $\mathcal{S}=\mathcal{S}\cup\text{branch}(S,\mathcal{T}_{\text{rat}})$

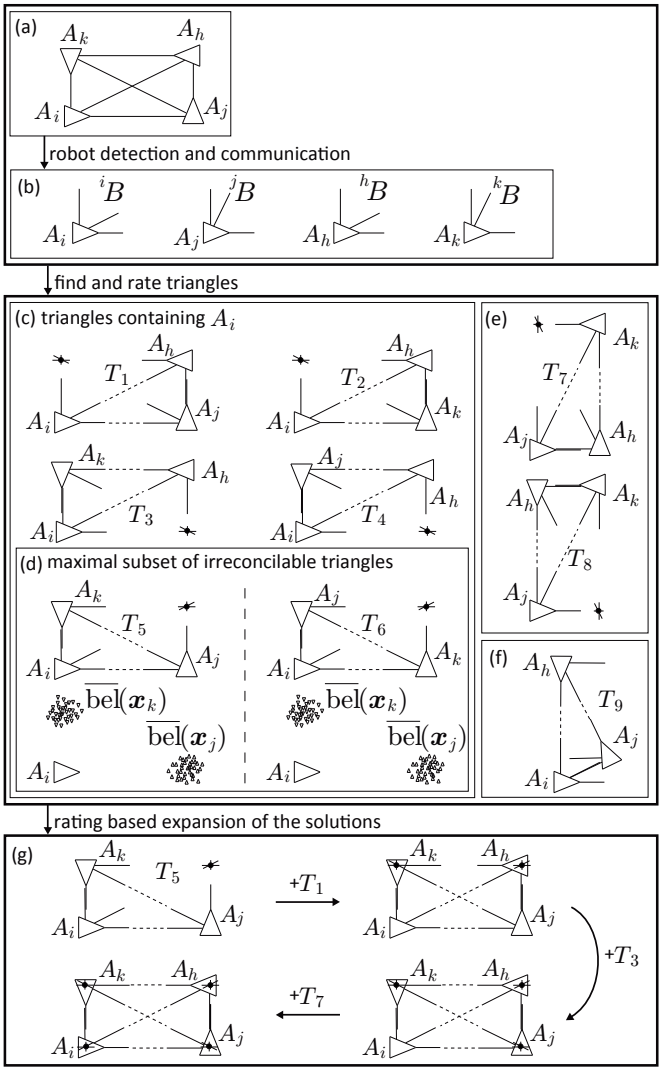


Fig. 8: Illustration of the basic steps of PMBR in a simple situation: (a) true configuration (b) feature sets (c) all the triangles having one 3-intersection and containing A_i (d) one choice for the maximal subset $\mathcal{T}_{\text{irr}}$ and comparison with the current belief (e) all the triangles having one 3-intersection but not containing A_i (f) one of the triangles without 3-intersections (g) expansion of the partial solution using the remaining triangles in $\mathcal{T}$.

As explained in Sect. III and Fig. 3, each of these triplets defines two triangles whose vertices are occupied by the three robots; more precisely, it defines a class of equivalence for each triangle, because the triangles are defined only up to a scaling factor. An example of a triangle pair originated by the same difference angles is given by T_1 in Fig. 8c and T_9 in Fig. 8f. Note that triangles include the identity of the robots at their vertices. Once appropriate triplets are identified, the triangle pairs are computed through the function $\text{computeTriangles}(\diamond,\square,\circ)$ and collected in a set $\mathcal{T}_{\text{raw}}$.

line 3) 3-intersection rating: when three robots of a triangle see a fourth robot (e.g., A_h in Fig. 8a), their sets of features include three rays (one from each feature set) which intersect in a single point (we call this a *3-intersection*). Based on this idea, the algorithm rates all triangles by counting their 3-intersections (function $\text{count3Intersections}(\diamond)$ in the pseudo-code), and discards those below a certain threshold th_1, with the remaining triangles collected in a set $\mathcal{T}_{\text{rat}}$. Note that a simple 2-intersection does not provide any useful information, since two non-parallel lines on the plane will always intersect.

line 4) Irreconcilable triangles: from the set $\mathcal{T}_{\text{rat}}$, one extracts a maximal subset $\mathcal{T}_{\text{irr}}$ of irreconcilable triangles containing A_i. Two triangles are *irreconcilable* if they associate the same robot to different features of the same set (e.g., A_j in T_5 and T_6), or two different robots to the same feature (e.g., A_k and A_j in T_2 and T_5). With this definition, it is easy to implement a function $\text{areIrreconcilable}(\diamond,\circ)$ to test whether two triangles are irreconcilable or not. The result of the triangle finding and rating process is illustrated in Fig. 8c–f.

line 5) Belief rating: with the function $\text{beliefRating2D}(\diamond)$, the algorithm rates each triangle $T\in\mathcal{T}_{\text{irr}}$ on the basis of the current belief about the configuration of the robots, using the metric function

$$P({}^i\hat{\boldsymbol{x}}_j^t)=\sum_{A_j\in T}\int p({}^i\hat{\boldsymbol{x}}_j^t|{}^i\tilde{\boldsymbol{x}}_j^t)\overline{\text{bel}}({}^i\tilde{\boldsymbol{x}}_j^t)d{}^i\tilde{\boldsymbol{x}}_j^t\,, \tag{12}$$

in which the scale of each triangle is computed so as to maximize (12). Then an adaptive thresholding of these maximum

values is used to keep only the triangles that better fit the belief, which are collected in a set $\mathcal{T}_{\text{best}}$.

line 6) Partial solutions: at this step a recursive algorithm is initialized through the function $\text{branch}(\diamond, \circ)$. Each triangle in $\mathcal{T}_{\text{best}}$ is the base of a branch of the algorithm, and constitutes the *partial solution* of that branch at the first step. 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 among them (3) the total number of the 3-intersections. In Fig. 8g, the only branch of the algorithm is originated by T_5. Each branch of the algorithm can originate multiple branches, and when the recursion terminates on a given branch, the partial solution of that branch is inserted in the set $\mathcal{S}$ of final solutions.

line 7) Relative azimuth-orientation extraction: the output of the algorithm, a set of relative azimuth-orientation for each $j \in N_i^t$, is easily extracted from $\mathcal{S}$ through the function $\text{extractAziOri}(\diamond)$. Since the solutions in $\mathcal{S}$ in principle come from many different branches, it is possible to once again rate them (with 3-intersections and beliefs) considering only the best to extract azimuth-orientation estimates.

lines 8-14) Recursive expansion: during one recursion in each branch, the current partial solution S is expanded looking for triangles in $\mathcal{T}_{\text{rat}}$ having common edges with it (through the function $\text{commonSide}(\diamond, \circ)$). In particular, using the function $\text{join}(\diamond, \circ)$ a new set $\mathcal{S}_{\text{new}}$ of partial solutions is created, each partial solution being the expansion of S plus a triangle which was not previously in S and which had (at least) one common side with S (line 9). As in the case of triangles, each solution in $\mathcal{S}_{\text{new}}$ is rated counting out its total number of 3-intersections (line 10). Having more triangles joined together, it is possible that more than three rays all intersect in the same point. In general, an n-intersection counts as $n!/3!(n-3)! = n(n-1)(n-2)/6$ 3-intersections. Among the best rated partial solutions of each branch, collected in $\mathcal{S}_{\text{rat}}$, the algorithm selects a maximal subset of irreconcilable solutions $\mathcal{S}_{\text{irr}}$ (line 11). Among those, only the solutions that fit with the current belief according to equation (12) are collected in $\mathcal{S}_{\text{best}}$ (line 12) and used as partial solutions at following step, expanding a branch for each of them (line 14).

In the example of Fig. 8g the algorithm expands T_5 by joining T_1, T_3 and T_7 respectively at the first, second and third iteration. The recursive process continues in each branch until $\mathcal{S}_{\text{best}}$ becomes empty (line 13), which may happen because either no expanded solution is good enough, or because all triangles have already been joined in S.

B. Filtering

The filtering stage is exploited by A_i by the use of a bank of particle filters, one for each A_j, $j \in N_i^{1:t}$. The use of a separate filters instead of one filter for the whole system implicates the computation of a separate belief for each robot rather than a single joint belief. While this is an approximation, the single joint belief computation is however not computationally feasible. In fact, the state of the filter would grow linearly with the number of robots, which would cause the number of required particles to grow exponentially.

Moreover, we preferred to use particle filters instead of extended Kalman filters because the firsts are inherently multi-modal, that is desirable in a situation in which multiple solutions may be returned from the multiple registration algorithm.

The inputs of the j-th filter at time t are the displacement $\bar{\boldsymbol{u}}_i^t$ of A_i, the displacement $\bar{\boldsymbol{u}}_j^t$ of A_j (sent by A_j) and the relative bearing-orientation estimate ${}^i\hat{\boldsymbol{\xi}}_j^t$ (computed by PMBR). The latter is used to generate a Gaussian measurement model with mean value ${}^i\hat{\boldsymbol{\xi}}_j^t$ and appropriate covariance. If PMBR 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 A_i and A_j are respectively

$$p({}^i\tilde{\boldsymbol{x}}_j|\boldsymbol{u}_i) = F_i \int p(\boldsymbol{u}'|\boldsymbol{u}_i)p({}^i\tilde{\boldsymbol{x}}_j \oplus \boldsymbol{u}')d\boldsymbol{u}'$$

$$p({}^i\tilde{\boldsymbol{x}}_j|\boldsymbol{u}_j) = F_j \int p(\boldsymbol{u}'|\boldsymbol{u}_j)p({}^i\tilde{\boldsymbol{x}}_j \ominus \boldsymbol{u}')d\boldsymbol{u}',$$

with F_i and F_j normalization factors, and $\oplus$, $\ominus$ are the composition operators introduced by [29]. These lead to the following update for the single particle:

$${}^i\tilde{\boldsymbol{x}}_j = \left({}^i\tilde{\boldsymbol{x}}_j \ominus (\boldsymbol{u}_i \oplus n_{\boldsymbol{u}_i})\right) \oplus (\boldsymbol{u}_j \oplus n_{\boldsymbol{u}_j}),$$

where $n_{\boldsymbol{u}_i}$ and $n_{\boldsymbol{u}_j}$ are samples taken by $p(\boldsymbol{u}'|\boldsymbol{u})$.

Note that if A_i and 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 or due to temporary or prolonged communication failures) the motion update of A_j is not performed. At t_b, when communication is resumed, A_j sends to A_i all odometry measurements $\bar{\boldsymbol{u}}_j^{t_a}, \ldots, \bar{\boldsymbol{u}}_j^{t_b}$, which are used in more consecutive time updates. Since no measurement update will happen for multiple motion updates, the resulting particles will be in general more scattered than during normal operation. The outcome of the update step are the beliefs $\{\overline{\text{bel}}({}^i\tilde{\boldsymbol{x}}_j^t)\}$ (see Fig. 7).

Since PMBR only produces relative azimuth-orientation, the measurements have a lower dimension than the state. However, the generalization of the update rule is straightforward and given by

$$p({}^i\tilde{\boldsymbol{x}}_j|\hat{\boldsymbol{\xi}}_j) = Fp({}^i\hat{\boldsymbol{\xi}}_j|{}^i\tilde{\boldsymbol{x}}_j)p({}^i\tilde{\boldsymbol{x}}_j), \tag{13}$$

where F is another normalization factor. Equation (13) allows the computation of the posteriors $\{\text{bel}({}^i\tilde{\boldsymbol{x}}_j^t)\}$ as depicted in Fig. 7 by using the the beliefs $\{\overline{\text{bel}}({}^i\tilde{\boldsymbol{x}}_j^t)\}$ and the relative azimuth-orientation estimate given by PMBR. Note that the filter is in charge of reconstructing the distance information among the robots, which is missing in the measurements (i.e.: in the estimates produced by PMBR). In fact, the authors of [30] proved that the system is observable, hence our combination of inputs and measurements is suitable to fulfill this task.

Standard instruments can be applied to improve the performance of the filter. For example, the initial prior distribution can be generated using the first measurements. Similarly, if new robots are encountered during the normal operation of the system, the relative particle filters will be added to the current bank of filters and initialized with the first available measurements. Also, it is advisable to reduce the frequency of the measurement update with respect to the motion update in

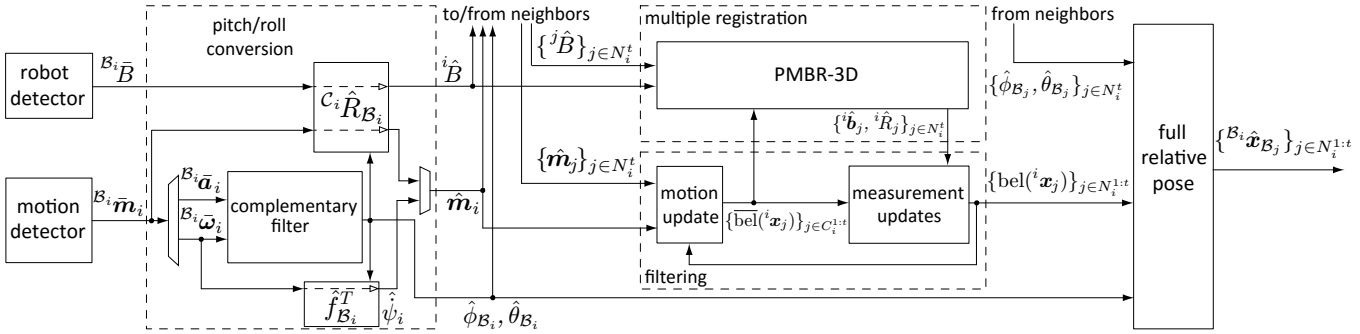


Fig. 9: Scheme of the mutual localization system that runs on A_i in Scenarios B and C.

order to guarantee the independence assumption for successive measurements.

VI. Scenarios B and C

The problem addressed in this section can be formalized as follows.

Problem 2: (3D RML with anonymous bearing measurements) For $t = 1, 2, \dots$ and $j \in N_i^{1:t}$, compute the following *belief*:

$$\text{bel}({}^{\mathcal{B}_i}\boldsymbol{x}_j^t) = p\left({}^{\mathcal{B}_i}\boldsymbol{x}_j^t | {}^{\mathcal{B}_i}\bar{\boldsymbol{m}}_i^{1:t}, {}^{\mathcal{B}_i}B^{1:t}, \{\boldsymbol{m}_j^{1:t}, {}^{\mathcal{B}_j}B^{1:t}\}_{j\in N_i^{1:t}}\right) \tag{14}$$

with

$${}^{\mathcal{B}_i}\bar{\boldsymbol{m}}_i^{1:t} = \begin{cases} \{{}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i^{1:t}, {}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i^{1:t}, {}^{\mathcal{B}_i}\bar{\boldsymbol{v}}_i^{1:t}\} & \text{in Scenario B} \\ \{{}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i^{1:t}, {}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i^{1:t}\} & \text{in Scenario C.} \end{cases} \tag{15}$$

The higher dimensionality of the state space w.r.t. Scenario A is the major challenge to be tackled in order to extend the method presented in Section V. In fact, it adds one and two degrees of freedom respectively to the position and orientation of the relative configuration. This additional problem will impact mainly on the multiple registration algorithm, in which the research of triangles switches from 2D to 3D. In particular, the additional non-zero roll and pitch angles enormously complicates the geometrical computations. Furthermore, the velocity of the robots becomes a state to be estimated in Scenario C, in addition to the pose.

Notice that the multiple registration algorithm would be easier to perform if roll and pitch angles in the team were all equal to zero. In fact, in this situation we could apply a multiple registration algorithm similar to the one developed in the 2D case. In addition, the third dimension would also be turned from an issue into a resource, allowing the use of the zenith-distance angles as additional check on the triangles.

Hence, in order to tackle the increased dimensionality we modified the conceptual scheme of Fig. 1, following a *divide et impera* strategy. The resulting system architecture is shown in Fig. 9. Note that the scheme, as well as the multiple registration algorithm, is valid for both Scenario B and C. In fact, those Scenarios differ only for the type of measurements provided by the motion detector which impacts only on the design of the filtering stage.

A. Estimation of Roll and Pitch

The basic idea of the new architecture is to let each A_i independently compute an estimate ${}^{\mathcal{W}}\hat{\phi}_{\mathcal{B}_i}$ and ${}^{\mathcal{W}}\hat{\theta}_{\mathcal{B}_i}$ of its roll and pitch angles using the first two components of the IMU measurements ${}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i$, ${}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i$. This is achieved using a complementary filter (see [31], [32]).

Then, A_i is able to compute an estimate ${}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}$ of the rotation matrix ${}^{\mathcal{C}_i}\boldsymbol{R}_{\mathcal{B}_i}$ between the $\mathcal{B}_i$ and $\mathcal{C}_i$ plugging ${}^{\mathcal{W}}\hat{\phi}_{\mathcal{B}_i}$ and ${}^{\mathcal{W}}\hat{\theta}_{\mathcal{B}_i}$ in equation (3). By the use of ${}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}$, each A_i is able to express its own measurements in $\mathcal{C}_i$:

$$\hat{\boldsymbol{a}}_i = {}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}(0\ 0\ {}^{\mathcal{B}_i}\bar{a}_{i_Z})^T \tag{16}$$

$$\hat{\boldsymbol{\omega}}_i = {}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}(0\ 0\ {}^{\mathcal{B}_i}\bar{\omega}_{i_Z})^T \tag{17}$$

$$\hat{\boldsymbol{v}}_i = {}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}{}^{\mathcal{B}_i}\bar{\boldsymbol{v}}_i \qquad \text{in Scenario B} \tag{18}$$

$${}^{i}\hat{B} = \{\hat{\boldsymbol{b}}_h = {}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}\bar{\boldsymbol{b}}_h,\ \forall \bar{\boldsymbol{b}}_h \in {}^{\mathcal{B}_i}B\}. \tag{19}$$

Note that $\hat{\boldsymbol{a}}_i$ and $\hat{\boldsymbol{\omega}}_i$ are computed using only the third 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 avoid multiple use of the x and y components of ${}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i$ and ${}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i$, since they have been already used to compute the roll and pitch angels. For Example, in a typical quadrotor 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° [32]. More in general, an independent source should be used, either a different sensor or a model-based prediction using the control inputs. This is the case of Scenario B, in which we will not use $\hat{\boldsymbol{a}}_i$ during the motion update, being available $\hat{\boldsymbol{v}}_i$.

In addition, the system uses an estimate ${}^{\mathcal{W}}\hat{\dot{\psi}}_{\mathcal{B}_i}$ of the yaw rate:

$${}^{\mathcal{W}}\hat{\dot{\psi}}_{\mathcal{B}_i} = \begin{pmatrix} 0 & \dfrac{\sin {}^{\mathcal{W}}\hat{\phi}_{\mathcal{B}_i}}{\cos {}^{\mathcal{W}}\hat{\theta}_{\mathcal{B}_i}} & \dfrac{\cos {}^{\mathcal{W}}\hat{\phi}_{\mathcal{B}_i}}{\cos {}^{\mathcal{W}}\hat{\theta}_{\mathcal{B}_i}} \end{pmatrix} {}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i = \boldsymbol{f}_{\mathcal{B}_i}^T {}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i, \tag{20}$$

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

Once roll and pitch are computed and those rotations are applied, to solve Problem 2 the system can solve a simpler problem, consisting in retrieving the identities of the relative bearing measurements and estimating a *reduced relative con-*

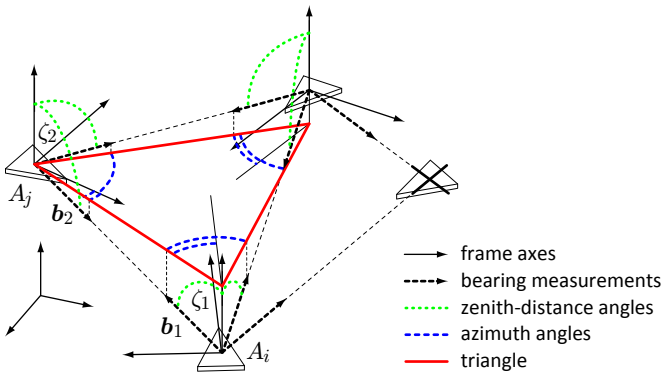


Fig. 10: Three robots measuring each other and the corresponding 2D triangle.

figuration ${}^i\boldsymbol{x}_j = \{{}^i\boldsymbol{p}_j, {}^i\boldsymbol{R}_j\}$, $j \in N_i^{1:t}$, where

$$ {}^i\boldsymbol{p}_j = {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{C}_i}^T({}^{\mathcal{W}}\boldsymbol{p}_j - {}^{\mathcal{W}}\boldsymbol{p}_i) \tag{21} $$

$$ {}^i\boldsymbol{R}_j = \boldsymbol{R}_Z({}^{\mathcal{W}}\psi_i)^T\boldsymbol{R}_Z({}^{\mathcal{W}}\psi_j). \tag{22} $$

From the corresponding estimates ${}^i\hat{\boldsymbol{x}}_j = \{{}^i\hat{\boldsymbol{p}}_j, {}^i\hat{\boldsymbol{R}}_j\}$, it is immediate to compute an estimate ${}^{\mathcal{B}_i}\hat{\boldsymbol{x}}_{\mathcal{B}_j} = \{{}^{\mathcal{B}_i}\hat{\boldsymbol{p}}_j, {}^{\mathcal{B}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_j}\}$ of the relative configuration required by Problem 2 by setting

$$ {}^{\mathcal{B}_i}\hat{\boldsymbol{p}}_j = {}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}^T\,{}^i\hat{\boldsymbol{p}}_j \tag{23} $$

$$ {}^{\mathcal{B}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_j} = {}^{\mathcal{C}_i}\hat{\boldsymbol{R}}_{\mathcal{B}_i}^T\,{}^i\hat{\boldsymbol{R}}_j. \tag{24} $$

Hence, we now need to address the following reformulation of Problem 2.

Problem 3: For $t = 1, 2, \ldots$ and $j \in N_i^{1:t}$, compute the following *belief*

$$ \mathrm{bel}({}^i\boldsymbol{x}_j) = P({}^i\boldsymbol{x}_j^t |\, \hat{\boldsymbol{m}}_i^{1:t}, {}^i\hat{B}^{1:t}, \{\hat{\boldsymbol{m}}_j^{1:t}, {}^j\hat{B}^{1:t}\}_{j\in N_i^{1:t}}). \tag{25} $$

with

$$ \hat{\boldsymbol{m}}_i^{1:t} = \begin{cases} \{\hat{\boldsymbol{a}}_i^{1:t}, \hat{\boldsymbol{\omega}}_i^{1:t}, {}^{\mathcal{W}}\hat{\psi}_i^{1:t}, \hat{\boldsymbol{v}}_i^{1:t}\} & \text{in Scenario B} \\ \{\hat{\boldsymbol{a}}_i^{1:t}, \hat{\boldsymbol{\omega}}_i^{1:t}, {}^{\mathcal{W}}\hat{\psi}_i^{1:t}\} & \text{in Scenario C.} \end{cases} \tag{26} $$

In order to solve Problem 3 we need to recover: (1) the identities of the measurements in $\hat{B}_i$, (2) the relative orientations ${}^i\boldsymbol{R}_j$, and (3) the relative distances $\|{}^i\boldsymbol{p}_j\|$. This will be done by a subsystem with structure as outlined in Fig. 9.

Hence, in this case the communication packet sent at time t by the generic robot A_i will contain, in addition to the robot signature,

1) the rotated measurement of the motion detector $\bar{\boldsymbol{m}}_i^t$;
2) the rotated feature set ${}^i\hat{B}^t$;
3) the estimates of roll and pitch ${}^{\mathcal{W}}\hat{\phi}_i$, ${}^{\mathcal{W}}\hat{\theta}_i$.

note that the last ones will only be needed to compute ${}^{\mathcal{B}_i}\hat{\boldsymbol{x}}_{\mathcal{B}_j}$, while they are not needed to compute ${}^i\boldsymbol{x}_j$.

B. Multiple Registration

Once the feature sets are expressed in $\mathcal{C}_1, \ldots, \mathcal{C}_n$, PMBR-3D exploits the same geometrical principle of its 2D version, hence many steps are the same. Its pseudo-code description is given in Algorithm 2, where we have highlighted the lines which differ or are added with respect to the 2D version.

The most important difference is that the third dimension, through the non-$\pi/2$ zenith-distance angles, offers an additional constraint. As highlighted in Fig. 10, the projection on any horizontal[1] plane of the bearing measurements of three robots measuring each other still draws a triangle whose internal angles are difference angles between the azimuth angles of the bearing sets. Hence, the triangle list can be computed as the 2D case considering only the azimuth measurements.

Assume now that in the triangle shown in Fig. 10 the bearing measurement $\boldsymbol{b}_1 \in {}^iB$ and $\boldsymbol{b}_2 \in {}^jB$ are respectively associated to A_j and A_i. The corresponding zenith-distance measurements ζ_1, ζ_2 must comply the constraint

$$ \zeta_1 + \zeta_2 = \pi \pm \tau_2 \tag{27} $$

as specified in line 2, where τ_2 is a tolerance to account for measurement noise. Although not relevant if the robots all lie at the same altitude, in the general case this additional constraint will consistently reduce the number of feasible triangles in the list, leading to lower ambiguity, lower computational time and improved performance. Note that each triangle, containing 3 pairs of associations, must satisfy three constraints in the form of eq. (27).

Another difference w.r.t. the 2D case is given in the rating of the triangles. In fact, two lines in space do not always intersect as it happens on the plane. Hence, it is possible to use the number of 2-intersections as rating instead of the number of 3-intersections (see lines 3 and 10). This tends to increase the number of features mutually seen in an elementary triangle, thus considerably improving the rating policy. The belief rating method must also be adapted as $\mathrm{beliefRating3D}(\diamond)$ (see lines 5 and 12), which uses the following metric function instead of equation (12):

$$ P(\{{}^i\hat{\boldsymbol{b}}_j, {}^i\hat{\boldsymbol{R}}_j\}) = \sum_{A_j \in T} \int p(\{{}^i\hat{\boldsymbol{b}}_j, {}^i\hat{\boldsymbol{R}}_j\} | {}^i\boldsymbol{x}_j)\,\overline{\mathrm{bel}}({}^i\boldsymbol{x}_j) d^i\boldsymbol{x}_j, \tag{28} $$

where $\overline{\mathrm{bel}}({}^i\boldsymbol{x}_j)$ comes from the corresponding particle filter. In eq. (28) the scale is selected so as to maximize the rating.

The basic steps of PMBR-3D are also shown in the illustrative example of Fig. 11. Consider the situation in Fig. 11a, where four robots are arranged in a 'square' formation with the opposite vertices at the same height; the corresponding feature sets are shown in Fig. 11b. The triangles finding process is conducted as explained above. Then, the 2-intersections rating is computed as the 2D case, but it is worth noting that in general an n-intersection accounts for $n!/2(n-2)! = n(n-1)/2$ 2-intersections. The triangles whose rating is above a certain threshold are collected in the triangle list $\mathcal{T}_{\mathrm{rat}}$ (Fig. 11c–e).

The rest of the algorithm follows the same conceptual steps of the 2D case. It extracts from $\mathcal{T}_{\mathrm{rat}}$ a maximal subset of irreconcilable triangles $\mathcal{T}_{\mathrm{irr}}$, and rates them through the use of the belief. Then, an adaptive threshold of these maximum values is enforced to select the triangles that better fit the belief. Those triangles are collected in a set $\mathcal{T}_{\mathrm{best}}$.

Each triangle of $\mathcal{T}_{\mathrm{best}}$ is the base of a branch of the algorithm and constitutes the partial solution S at the first step of its branch. In Fig. 11f the only branch has T_5 as first

[1] i.e., parallel to $X_{\mathcal{W}}Y_{\mathcal{W}}$.

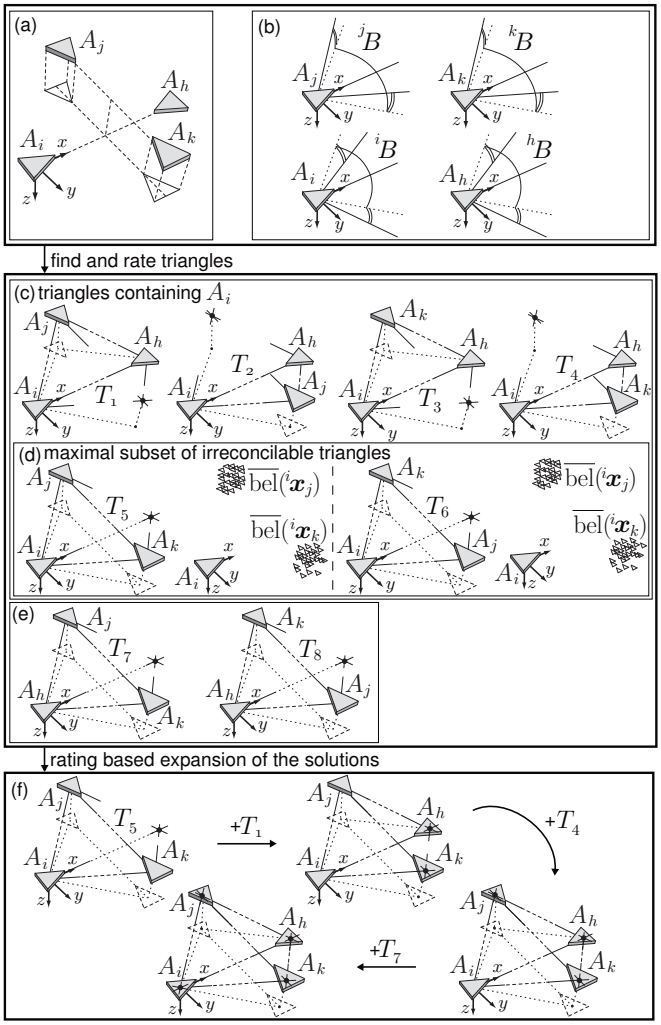


Fig. 11: Execution of PMBR-3D in an ambiguous situation: (a) true configuration (b) initial feature sets (c) triangle found in the first step containing the owner of the algorithm and their triple intersections (d) maximal subset of irreconcilable triangles and their comparison with the current belief (e) other triangles found in the first step of the algorithm and their triple intersections (f) expansion of the solution using the remaining triangles.

Algorithm 2: PMBR-3D - lines added with respect to Algorithm 1 are marked with a '+' on the right column; lines with minor differences are marked with an '*'.

input : feature sets ${}^iB^t$, $\{{}^jB^t\}_{j\in N_i^t}$, beliefs $\{\overline{\text{bel}}({}^i\boldsymbol{x}_j^t)\}_{j\in N_i^t}$ *
output: relative bearing-orientation estimates $\{{}^i\hat{\boldsymbol{\xi}}_j^t\}_{j\in N_i^t}$
init : $\mathcal{T}_{\text{raw}}=\varnothing$, $\mathcal{T}_{\text{rat}}=\varnothing$, $\mathcal{T}_{\text{irr}}=\varnothing$, $\mathcal{T}_{\text{best}}=\varnothing$, $\mathcal{S}=\varnothing$

foreach $j\in\{i\}\cup N_i^t$ **do** ${}^jD^t=\text{computeDiffAngleSet}({}^jB^t)$
foreach $\{D_1,D_2,D_3\}\subseteq\{{}^iD^t\}\cup\{{}^jD^t\}_{j\in N_i^t}$ **do**
  foreach $\{\boldsymbol{d}_1\in D_1,\boldsymbol{d}_2\in D_2,\boldsymbol{d}_3\in D_3\}$ **do**
    if $\boldsymbol{d}_1+\boldsymbol{d}_2+\boldsymbol{d}_3=\pi\pm\tau_1$ **then**
      $(T_a,T_b)=\text{computeTriangles}(\boldsymbol{d}_1,\boldsymbol{d}_2,\boldsymbol{d}_3)$
      if checkZenithConstraints(T_a) **then** +
        $\mathcal{T}_{\text{raw}}=\mathcal{T}_{\text{raw}}\cup\{T_a\}$ *
      if checkZenithConstraints(T_b) **then** +
        $\mathcal{T}_{\text{raw}}=\mathcal{T}_{\text{raw}}\cup\{T_b\}$ *
foreach $T\in\mathcal{T}_{\text{raw}}$ **do**
  if count2Intersections$(T)\geq\text{th}_1$ **then** $\mathcal{T}_{\text{rat}}=\mathcal{T}_{\text{rat}}\cup\{T\}$ *
foreach $T\in\mathcal{T}_{\text{rat}}$ **do**
  if $\boldsymbol{d}_1\in{}^iD^t$ **and** areIrreconcilable$(T,T'),\forall T'\in\mathcal{T}_{\text{irr}}$ **then**
    $\mathcal{T}_{\text{irr}}=\mathcal{T}_{\text{irr}}\cup\{T\}$
foreach $T\in\mathcal{T}_{\text{irr}}$ **do**
  if beliefRating3D$(T)\geq\text{th}_2$ **then** $\mathcal{T}_{\text{best}}=\mathcal{T}_{\text{best}}\cup\{T\}$ *
foreach $T\in\mathcal{T}_{\text{best}}$ **do** $\mathcal{S}=\mathcal{S}\cup\text{branch}(T,\mathcal{T}_{\text{rat}})$
foreach $j\in N_i^t$ **do** ${}^i\hat{\boldsymbol{\xi}}_j^t=\text{extractAziOri}(\mathcal{S})$

function branch(partialSolution S, triangleSet $\mathcal{T}_{\text{rat}}$):
init : $\mathcal{S}_{\text{new}}=\varnothing$, $\mathcal{S}_{\text{rat}}=\varnothing$, $\mathcal{S}_{\text{irr}}=\varnothing$, $\mathcal{S}_{\text{best}}=\varnothing$
foreach $T\in\mathcal{T}_{\text{rat}}$ **do**
  if commonSide(S,T) **and** $T\notin S$ **then**
    if checkZenitConstraints(S,T) **then** +
      $\mathcal{S}_{\text{new}}=\mathcal{S}_{\text{new}}\cup\{\text{join}(S,T)\}$
foreach $S\in\mathcal{S}_{\text{new}}$ **do**
  if count2Intersections$(S)\geq\text{th}_3$ **then** $\mathcal{S}_{\text{rat}}=\mathcal{S}_{\text{rat}}\cup\{S\}$ *
foreach $S\in\mathcal{S}_{\text{rat}}$ **do**
  if areIrreconcilable$(S,S'),\forall S'\in\mathcal{S}_{\text{irr}}$ **then** $\mathcal{S}_{\text{irr}}=\mathcal{S}_{\text{irr}}\cup\{S\}$
foreach $S\in\mathcal{S}_{\text{irr}}$ **do**
  if beliefRating3D$(S)\geq\text{th}_4$ **then** $\mathcal{S}_{\text{best}}=\mathcal{S}_{\text{best}}\cup\{S\}$ *
if $\mathcal{S}_{\text{best}}=\varnothing$ **then return** S
else foreach $S\in\mathcal{S}_{\text{best}}$ **do** $\mathcal{S}=\mathcal{S}\cup\text{branch}(S,\mathcal{T}_{\text{rat}})$

partial solution. Then the partial solution of each branch is expanded joining triangles which have common edges with it (see Fig. 11f). This recursive process is conducted in the same way of the 2D case, using 2-intersections and the metric in equation (28) as ratings.

In the case of Fig. 11f, 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.

In the end, each branch finds a solution, and the best of them are selected, again with the 2-intersection and belief criteria. Similarly to the 2D case, the result of the algorithm is a list of pairs ${}^i\hat{\boldsymbol{b}}_j, {}^i\hat{\boldsymbol{R}}_j$ for each A_j .

C. Filtering

The overall scheme of the filtering stage is in general equivalent to the 2D case. The same considerations on the choice of particle filters and separate belief holds in 3D. Hence, A_i runs one particle filter (PF_j) for each A_j to retrieve the missing relative distance $\|{}^i\boldsymbol{p}_j\|$. This is obtained by fusing the depthless quantities ${}^i\hat{\boldsymbol{b}}_j, {}^i\hat{\boldsymbol{R}}_j$ coming from PMBR-3D with the metric information provided by the motion detectors of A_i and A_j.

Since the two Scenarios varies for the presence of the measurements of the velocities, the filtering stage will vary accordingly. The kinematic equations of motion of the single robot are

$$ {}^i\dot{\boldsymbol{p}}_j = {}^i\boldsymbol{v}_j \tag{29}$$

$$ {}^i\dot{\boldsymbol{R}}_j = ({}^i\boldsymbol{R}_j[\boldsymbol{\omega}_j]_\times - [\boldsymbol{\omega}_i]_\times){}^i\boldsymbol{R}_j \tag{30}$$

where we denoted with ${}^i\boldsymbol{v}_j$ the velocity of $O_{\mathcal{C}_j}$ in $\mathcal{C}_i$ and

$$ [\boldsymbol{\omega}_i]_\times = \begin{pmatrix} 0 & -\omega_{iz} & \omega_{iy} \\ \omega_{iz} & 0 & \omega_{ix} \\ \omega_{iy} & -\omega_{ix} & 0 \end{pmatrix}. $$

Since

$$ {}^{i}\boldsymbol{R}_j = \boldsymbol{R}_Z(-{}^{\mathcal{W}}\psi_i)\boldsymbol{R}_Z({}^{\mathcal{W}}\psi_j) = \boldsymbol{R}_Z({}^{\mathcal{W}}\psi_j - {}^{\mathcal{W}}\psi_i), \quad (31) $$

we can replace (30) with

$$ {}^{i}\dot{\psi}_j = \dot{\psi}_j - \dot{\psi}_i = f_{\mathcal{B}_j}^{T\,\mathcal{B}_j}\boldsymbol{\omega}_j - f_{\mathcal{B}_i}^{T\,\mathcal{B}_i}\boldsymbol{\omega}_i, \quad (32) $$

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 (20).

In Scenario B an estimate ${}^{i}\hat{\boldsymbol{v}}_j$ of ${}^{i}\boldsymbol{v}_j$ is directly available from the estimates $\hat{\boldsymbol{v}}_i$, $\hat{\boldsymbol{v}}_j$ given by (18), through the following

$$ {}^{i}\hat{\boldsymbol{v}}_j = {}^{i}\hat{\boldsymbol{R}}_j\,\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_i\,. \quad (33) $$

Hence, the kinematic model given by equations (29) and (32) is sufficient for the motion update of the filter. Therefore, in Scenario B the state of each particle is the 4-dimensional tuple ${}^{i}\boldsymbol{\chi}_j = ({}^{i}\boldsymbol{p}_j, {}^{i}\psi_j) \in \mathbb{R}^3 \times \mathbb{S}^1$ including the relative position and yaw of A_j w.r.t. A_i, and the motion update step of the j-th particle filter is obtained by plugging ${}^{i}\hat{\boldsymbol{v}}_j, \hat{\boldsymbol{\omega}}_i, \hat{\dot{\psi}}_i, \hat{\dot{\psi}}_j$ in (29,32).

In Scenario C the velocity ${}^{i}\boldsymbol{v}_j$ is not available from the measurements, hence must be estimated in the filter: the state of the robot must be augmented in order to include ${}^{i}\boldsymbol{v}_j$. In particular, the equation of the update of the velocity is

$$ {}^{i}\dot{\boldsymbol{v}}_j = {}^{i}\boldsymbol{R}_j\boldsymbol{a}_j - \boldsymbol{a}_i + [\boldsymbol{\omega}_i]_\times {}^{i}\boldsymbol{v}_j \quad (34) $$

in which ${}^{i}\boldsymbol{R}_j$ can be computed using (31). Therefore, in Scenario C the state of each particle is the 7-dimensional tuple ${}^{i}\boldsymbol{\chi}_j = ({}^{i}\boldsymbol{p}_j, {}^{i}\boldsymbol{v}_j, {}^{i}\psi_j) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{S}^1$, and the motion update step of the j-th particle filter is obtained by plugging $\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_j, \hat{\boldsymbol{\omega}}_i, \hat{\dot{\psi}}_{\mathcal{B}_i}, \hat{\dot{\psi}}_{\mathcal{B}_j}$ in (29,32,34).

In both Scenarios the new state probability after the motion update is predicted by means of a Tustin integration of the motion measurements (velocity and acceleration respectively in Scenario B and C) with the knowledge of the measurement noise. Also in this case, a temporary or prolonged communication loss between two time instants (t_a, t_b) can be recovered by sending all the motion measurements happened between those times once the communication is restored.

As for the measurement update, which is the same in the two Scenarios, each solutions ${}^{i}\hat{\boldsymbol{b}}_j, {}^{i}\hat{\boldsymbol{R}}_j$ of PMBR-3D for A_j is approximated as a Gaussian measurement with a covariance proportional to its uncertainty. Therefore, similarly to the 2D case, the measurement model is given by the normalized sum of Gaussians centred at the solutions of PMBR-3D.

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

$$ P({}^{i}\boldsymbol{\chi}_j | {}^{i}\hat{\boldsymbol{b}}_j, {}^{i}\hat{\psi}_j) = FP({}^{i}\hat{\boldsymbol{b}}_j, {}^{i}\hat{\psi}_j | {}^{i}\boldsymbol{\chi}_j)P({}^{i}\boldsymbol{\chi}_j), \quad (35) $$

where F is a normalization factor.

In Scenario B also the velocities of the robots are available. Hence, another measurement update can be performed through

$$ P({}^{i}\boldsymbol{\chi}_j | {}^{i}\hat{\boldsymbol{v}}_j) = FP({}^{i}\hat{\boldsymbol{v}}_j | {}^{i}\boldsymbol{\chi}_j)P({}^{i}\boldsymbol{\chi}_j), \quad (36) $$

Note that the observability in both Scenarios is proved in [33].

VII. Computational complexity

Since PMBR and PMBR-3D are recursive algorithms which allow multiple branches, their execution time depends mostly on the number of branches and the depth of the recursion. In turn, these are functions of the geometric configuration of the multi-robot team. The dependency of the execution time of the algorithms with respect to the number of robots and false positives in two typical cases is reported in Table II and explained in the following of this section.

It is easy to show that the following methods can be executed in constant time:

- $\mathrm{computeTriangles}(\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3)$
- $\mathrm{checkZenithConstraints}(T)$
- $\mathrm{areIrreconcilable}(T, T')$
- $\mathrm{beliefRating2D}(T)$ and $\mathrm{beliefRating3D(T)}$.

Let $T, T', T'' \in \mathcal{T}_{\mathrm{rat}}$, $S' = \mathrm{join}(S, T')$ and $S'' = \mathrm{join}(S, T'')$. When performing the same methods on partial solutions rather than on triangles, the methods

- $\mathrm{commonSide}(S, T)$
- $\mathrm{checkZenithConstraints}(S')$
- $\mathrm{join}(S, T)$
- $\mathrm{areIrreconcilable}(S', S'')$
- $\mathrm{beliefRating2D}(S')$ and $\mathrm{beliefRating3D(S')}$

can be implemented iteratively, since only the new triangles joined to S need to be checked for common sides with other triangles, zenith-distance constraints, irreconcilability and belief rating. Hence, each single call of those methods can be executed in constant time.

Assume that N robots participate to the mutual localization, M false positives are present in the environment, and P is the mean value of number of features in each feature set. As explained in Sect. V-A, the corresponding sets will have $P(P-1)/2$ difference angles, so the time needed for $\mathrm{computeDiffAngleSet}({}^{j}B^t)$ is $P(P-1)/2 = O(P^2)$, while executing line 1 takes $NP(P-1)/2 = O(N, P^2)$ time.

In the triangle finding problem, there are $N!/(N-3)!3! = N(N-1)(N-2)/6$ combinations of three difference angles sets. For each of these combinations, since the difference angle inside each set can be chosen independently, there are P^3 possible combinations of difference angles to test. Then, the triangle finding loop of line 2 takes $NP^3(N-1)(N-2)/6 = O(N^3, P^3)$ time.

Now, let Q_{raw}, Q_{rat}, Q_{irr} and H be the cardinality of $\mathcal{T}_{\mathrm{raw}}$, $\mathcal{T}_{\mathrm{rat}}$, $\mathcal{T}_{\mathrm{irr}}$ and $\mathcal{S}$ respectively. In each triangle, there are $P-2$ rays starting from each vertex which are not associated. Then, there is a maximum of $3(P-2)^2$ 2-intersections to be checked, and 3-intersections can be checked by matching 2-intersections which are close to each other. Hence, the execution of line 3 takes $3Q_{\mathrm{raw}}(P-2)^2 = O(P^2, Q_{\mathrm{raw}})$ time. Similarly, $Q_{\mathrm{rat}}(Q_{\mathrm{rat}}-1)/2 = O(Q_{\mathrm{rat}}^2)$ calls of $\mathrm{areIrreconcilable}$ are needed in line 4, and Q_{irr} calls of $\mathrm{beliefRating2D/beliefRating3D}$ are executed in line 5, while line 7 takes $NH = O(N, H)$ time.

The execution time of the recursion in line 6 depends on the execution time of each call, the depth of each branch and the number of branches. Let $H_{n,\mathrm{new}}$, $H_{n,\mathrm{rat}}$ and $H_{n,\mathrm{irr}}$ be the cardinality of $\mathcal{S}_{\mathrm{new}}$, $\mathcal{S}_{\mathrm{rat}}$, $\mathcal{S}_{\mathrm{irr}}$ in the n-th recursion. Using the

TABLE II: Computational complexity of PMBR and PMBR-3D.

line (main)	nominal	worst single-branch
1	$O(N, P^2)$	$O(N^3, M^2)$
2	$O(N^3, P^3)$	$O(N^6, M^3)$
3	$O(P^2, Q_{\text{raw}})$	$O(N, P^2)$
4	$O(Q_{\text{rat}}^2)$	$O(N^6)$
5	$O(Q_{\text{irr}})$	1
6	eq. (37)	$O(N^9, M^2)$
7	$O(N, H)$	$O(N)$
line (branch)		
9	$O(N, P^2)$	$O(N^3, M^2)$
10	$O(P^2, H_{\text{new}})$	$O(N^5, M^2)$
11	$O(H_{\text{rat}}^2)$	$O(N^6)$
12	$O(H_{\text{irr}})$	1

previous results, and being $\max(H_{n,\text{irr}}) = H_{n,\text{rat}}$, it can be shown that the time needed by the n-th recursion in a branch is

$$(Q_{\text{rat}} - n) + O(P^2, H_{n,\text{new}}) + O(H_{n,\text{rat}}^2).$$

Let Q be the number of recursions in a branch. The total execution time of that branch is

$$\sum_{n=1}^{Q} [(Q_{\text{rat}} - n) + O(P^2, H_{n,\text{new}}) + O(H_{n,\text{rat}}^2)] \qquad (37)$$

A. Nominal case

In general, it is possible to consider P limited by the field of view of the robot sensor and the size of the robots, and N limited by the area or volume of the communication set of each robot. This is true in particular for large numbers of robots in the environment, such that not all the robots are close to each other.

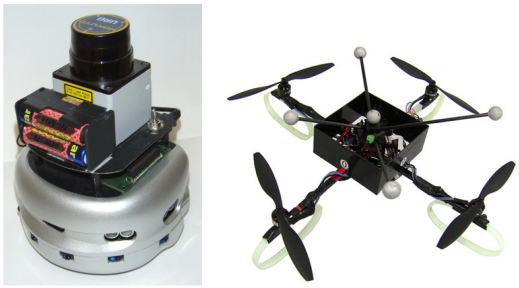
B. Worst single-branch case

In a worst case scenario (in terms of execution time), all robots and false positives are visible to all other robots, hence $P = N + M - 1$. In a formation without ambiguity, there are $N(N-1)(N-2)/6$ triangles, which results in at least twice the number of triangles in $\mathcal{T}_{\text{raw}}$ (see Sect. III), such that $Q_{\text{raw}} = N(N-1)(N-2)/3 = O(N^3)$. The intersection rating will be able to discriminate among the correctly matched and the incorrectly matched triangles, hence $Q_{\text{rat}} = Q_{\text{raw}}/2 = O(N^3)$, and only one irreconcilable triangle will exist, hence $Q_{\text{irr}} = 1$. Considering that in the recursion all Q_{rat} correctly matched triangles will be joined together, the single branch will have depth equal to $Q = Q_{\text{rat}} - 1 = O(N^3)$. Being in general, in the n-th recursion, $\max(H_{n,\text{new}}) = Q_{\text{raw}} - n$, $\max(H_{n,\text{rat}}) = Q_{\text{rat}} - n$, $H_{n,\text{irr}} = 1$, from eq. (37) the maximum possible execution time of the whole recursion is $O(N^9, M^2)$.

C. Ambiguous configurations

In ambiguous configurations, the recursive algorithm will have multiple branches, each with execution time similar that computed for the single-branch worst case scenario. However, since some call of $\text{branch}(\circ, \diamond)$ will be common among the different branches, the execution time of line 6 will be lower that the actual sum of the complexity of each branch. For example, if two branches lead to two different solutions at the n_1-th and n_2-th call respectively, and they originate by the same branch in the n-th call, the execution time of the two branches will be the sum of the first n calls plus the execution time of the $n_1 - n$ and $n_2 - n$ calls.

Unfortunately, ambiguity due to symmetric configurations usually leads to a factorial number of solutions, and hence to a factorial number of branches. The role of the belief rating is to avoid expansion of all these branches, allowing the algorithm to expand only those solutions that already fit the current belief. When no current belief is available (e.g., during the initialization of the localization system), use of a symmetry breaking control law [19]) is advisable.



(a) (b)

Fig. 12: Experimental platforms: (a) Khepera III robots were used in the 2D scenario, (b) Mikrokopter quadrotors were used in the 3D scenarios.

D. Impact of noise

In general, all the geometric constraints in triangle finding, rating and joining are checked using certain thresholds to account for measurement noise. On the one hand, if the noise on the measurements is larger than the thresholds, many triplets of difference angles originating from actual triangles in the formation may not respect the conditions (11) and (27). The intersection rating would also be affected in a similar way, leading to generally low ratings (hence less discriminative towards triangles which are not the result of correct associations). On the other hand, enlarging the thresholds will cause the algorithm to identify incorrectly matched triangles as feasible, with similar ratings to correctly matched ones. This will result in a degradation of the algorithm performance in terms of correct solutions found and a higher number of branches leading to wrong solutions.

VIII. Experimental results

We have implemented and tested the proposed mutual localization systems in simulations and experiments on various robots, with the aid of the multi-robot software platform MIP[2]. In particular, simulations are used to study its noise propagation characteristics, whereas experiments allow to show the performance of the proposed method on real data.

[2]See http://www.dis.uniroma1.it/~labrob/software/MIP.

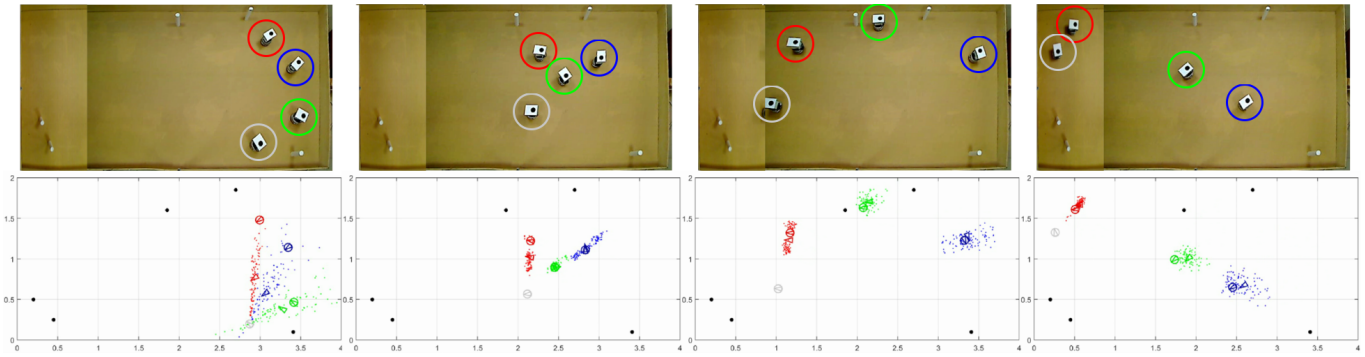


Fig. 13: Four snapshots of a 2D experiment and the corresponding estimates computed by A_1 (in gray).

For Scenario A we have employed a team of four Khepera III wheeled mobile robots (Fig. 12a), while Scenarios B and C are realized using a flock of eight Mikrokopter quadrotors (Fig. 12b).

A. Experimental results in 2D

We implemented and tested the proposed framework for the 2D localization problem on a team of four Khepera III robots. The robots are equipped with a 600 MHz processor and operates a minimal GNU/Linux distribution. They communicate through a wifi link with a base station desktop computer which performs the mutual localization off-board. Their motion capabilities are well modeled through the standard differential drive model, and motion measurements are provided by odometric integration of the readings of the wheel encoders at 20 Hz.

The robots are retrofitted with a Hokuyo URG-04LX laser sensor that provides range-bearing readings at 10 Hz within a 240 deg field of view, thus leaving a 120 deg blind zone behind the robot. The scans of the laser sensors are inspected by a simple feature extraction module looking for the indentation caused by a small cardboard "hat" mounted atop each robot. The bearing (and not the distance) of the extracted measurements are used as to come from the robot detector. Synchronization among the robots is not an issue, essentially because the communication delay is not significant at the typical robot speeds (some cm/s). The robots move with a pseudo-random navigation strategy incorporating obstacle avoidance. The ground truth is provided with cm accuracy by a feature extraction and tracking algorithm on the image streams of three USB cameras looking down on the arena.

We show in Fig. 13 four snapshots of an experiment with four robots and five static obstacles that generate false positive measurements. We used 500 particles for each filter and the corresponding particle clouds computed by A_1 (circled in gray in the snapshots) are shown in the bottom. In the beginning, the scale of the formation is unknown, hence the particle clouds are distributed in circular sectors. With time going on, also the scale of the formation is retrieved and the particle clouds assume a more spherical shape. In particular, the motion of A_3 (green) is perpendicular to the trajectory of A_1 from the very beginning, hence the estimate of its pose is the faster to converge, which happens already in the second snapshot. On the other hand, A_2 (red) moves parallel to A_1 for most of the experiment, hence the estimate of its pose is the slower to converge, which happens only in the last snapshot. The estimate of the pose of A_4 (blue) is in the middle and converges in the third snapshot.

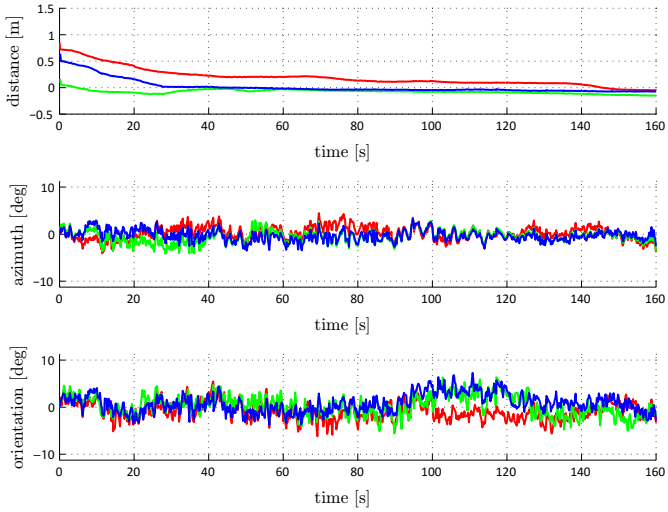


Fig. 14: Distance, azimuth and orientation errors on the estimates computed by A_4 during the planar experiment.

This behavior is clearly visible in the plots of the distance, azimuth and orientation errors affecting the estimates of the poses of the three robots shown in Fig. 14. Since the beginning of the experiment the error on the estimated distance of A_3 (green) is close to zero and around second 30 s most of the error on the distance of A_4 (blue) is reduced to zero. However, due to the unfavorable motion it takes up to 120-140 s for the error on the estimated distance of A_2 (red) to converge to zero. A video-clip of the experiment is shown in the first part of the attached video.

B. Simulation results in 2D

In order to further validate our mutual localization system, we have tested it with eight datasets collected in simulation with a varying number of robots (three to five) and a varying number of false positives (one to three). The simulation is performed using realistic 2D Khepera models in Player/Stage which provide simulated noise-free odometry. Each simulated robot is also equipped with a laser range finder and the

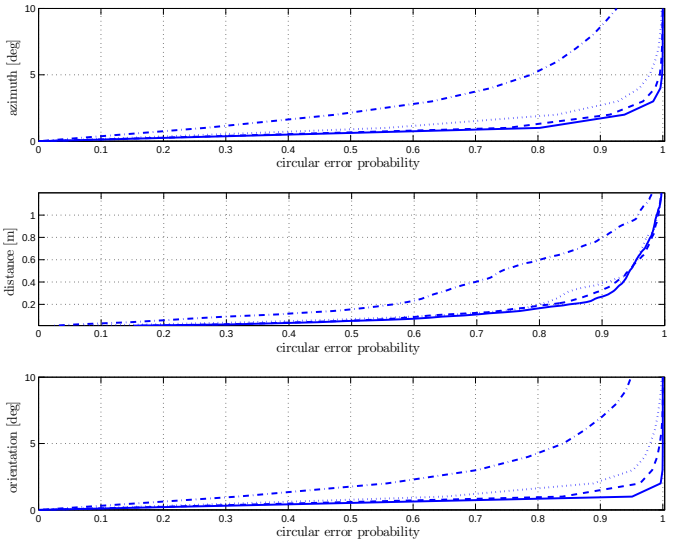


Fig. 15: Error probabilities (lower is better) for the estimated robots $A_2, \ldots, A_5$ of an aggregate of eight simulations for different values of covariance of the noise on the bearing measurements: solid 0 deg, dashed 5 deg, dotted 10 deg, dash-dotted 20 deg.

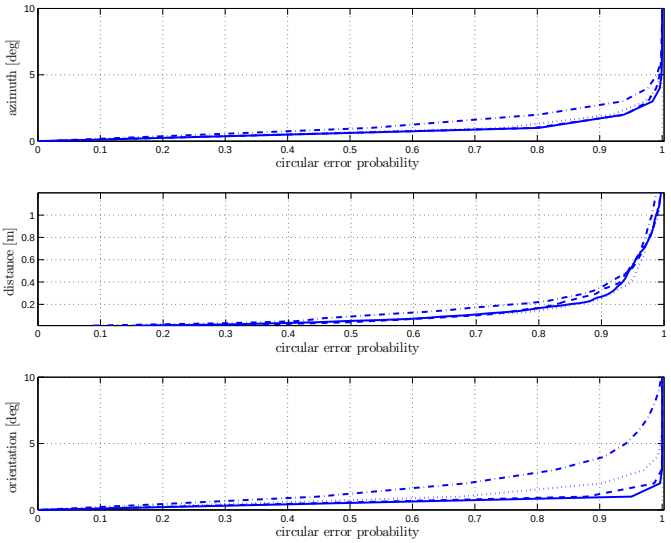


Fig. 16: Error probabilities (lower is better) for the estimated robots $A_2, \ldots, A_5$ of an aggregate of eight simulations for different values of covariance of the noise on the odometry (linear and angular velocities): solid 0 m/s and 0 deg/s, dashed 0.02 m/s and 5 deg/s, dotted 0.05 m/s and 10 deg/s, dash-dotted 0.1 m/s and 20 deg/s.

measurements are extracted with the same algorithm used in the experiments.

We have studied the noise propagation characteristics of the system and its robustness to noise by running each dataset with different additional zero-mean gaussian noise either on the measurements or on the odometry data. In particular, each simulation has been run with an additional noise on the measurements with covariance 5 deg, 10 deg and 20 deg. For the odometry, we have tested the system with the following noises respectively on the integrated linear and angular velocities: 0.02 m/s and 5 deg/s, 0.05 m/s and 10 deg/s, 0.1 m/s and 20 deg/s, for a total of 56 runs of the mutual localization system.

The results of this simulation campaign are presented in aggregated form in Fig. 15 and Fig. 16, which show the error probability for the different values of noise on the measurements and odometry respectively. Let e_β, e_d and e_α be the error on respectively orientation, distance and azimuth, their error probabilities are defined as the probabilities that they are equal or less than a parameter:

$$p\left(e_\beta < b\right) \quad p\left(e_d < d\right) \quad p\left(e_\alpha < a\right). \tag{38}$$

It is interesting to observe that the localization system in general shows low sensitivity to noise on the odometry, while it is more sensitive to high noise on the measurements. This is probably due to the fact that the multiple registration fails or finds many wrong geometric solutions when the noise on the measurements becomes too high. Comparatively, the filter is less sensible with respect to noise on the odometry, with only the error on the reconstructed distance increasing significantly.

C. Experimental results in 3D

The proposed localization system has been experimentally validated in Scenarios B and C using quadrotors[3] as mobile robots and an external Motion Capture System[4] (mocap) made up by 16 infrared cameras as ground truth, whose precision is about 1 mm for translations and 1 deg for rotations.

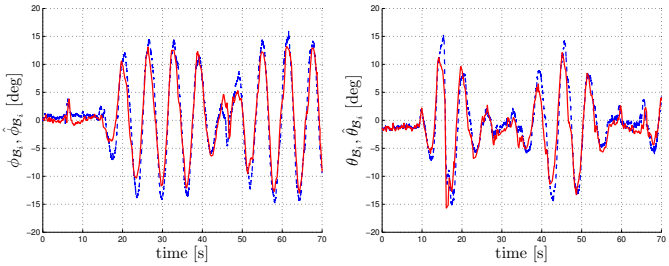
Each robot A_i carries an ATmega microcontroller (μC) board which performs the attitude control and interfaces with the hardware. An IMU composed by three-axis LIS344ALH linear MEMS accelerometer and three ADXRS610 angular-rate sensors orthogonally mounted on the μC board provides measurements of the linear acceleration ${}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i$ and angular velocity ${}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i$, both expressed in body frame. In order to give a characterization of the noise affecting the measurements provided by the IMU, we performed a statistical analysis over the values of ${}^{\mathcal{B}_i}\bar{\boldsymbol{a}}_i$, ${}^{\mathcal{B}_i}\bar{\boldsymbol{\omega}}_i$ registered during different experiments with the quadrotor in hovering. The analysis shows a precision of $[0.019, 0.019, 0.049]$ m/s^2, $[0.29, 0.29, 0.29]$ deg/s and variances $[0.1, 0.1, 0.6]$ m/s^2 and $[0.64, 0.64, 1.12]$ deg/s for accelerometer and gyroscope, respectively. The large variances for the accelerometer are also due to the high vibrations induced by the motors/propellers.

The microcontroller acquires the IMU data at about 400 Hz by means of a 10 bit digital converter and, at the same frequency, runs online a complementary filter in order to estimate the current attitude (see, e.g., [31]), i.e. the roll and pitch angles. The estimates have been compared to the values provided by the mocap system (see Fig. 17). The result of this analysis is a mean error of around 1.92° for roll and 2.67° for pitch.

The microcontroller can neither process nor store data; hence, the IMU readings, as well as the complementary filter output, have to be transferred to a GNU/Linux machine, through two serial connections. The usage of two separated connections, one for sending and one for receiving data to/from the quadrotor, allows a transmission with an average

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

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



(a) (b)

Fig. 17: True values (dashed blue) and estimates (solid red) of roll (a) and pitch (b) during a typical experiment.

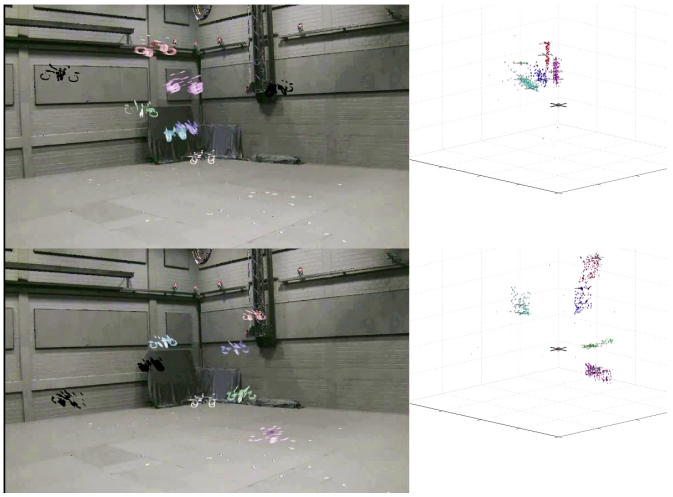


Fig. 18: Two snapshots of a 3D experiment and the corresponding estimates computed by A_1 in Scenario B.

rate of 370 Hz and a standard deviation of 4 ms. This is the main advantage w.r.t. our previous experimental setup used in [21], where the average IMU rate was 20 Hz due to a single serial connection. In Scenario C, this improvement will boost the performance of the localization system since the update step of the filtering phase can be run with higher frequency.

The main difference between Scenarios B and C is the device used as motion detector. In Scenario C, the mutual localization system will use the IMU readings of linear acceleration in order to feed the bank of particle filters.

On the contrary, in Scenario B the particle filters are fed by velocity measurements analytically emulated by adding zero-mean Gaussian noise with 0.25 m/s standard deviation on the velocity measurements provided by the mocap device. We emphasize that the same measurements can be obtained without the use of external sensors integrating IMU measurements with optical flow [28] or visual odometry algorithms [26], [27]. Note also that the IMU output has to be communicated also in this Scenario, since the knowledge of roll and pitch estimates is mandatory in order to solve the mutual localization problem.

To conclude, in both Scenarios the robot detector is emulated by analytically computing the relative bearing from the ground truth using equation (7). In order to have realistic measurements, a zero-mean 5 deg standard deviation Gaussian noise is added to the azimuth and zenith-distance angles.

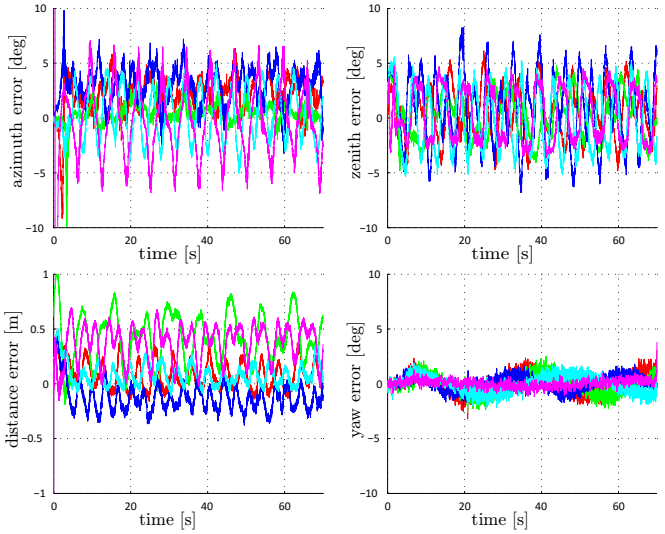


Fig. 19: Scenario C: error plots on azimuth, zenith-distance, distance and orientation estimates for A_2, ..., A_6 computed from A_1.

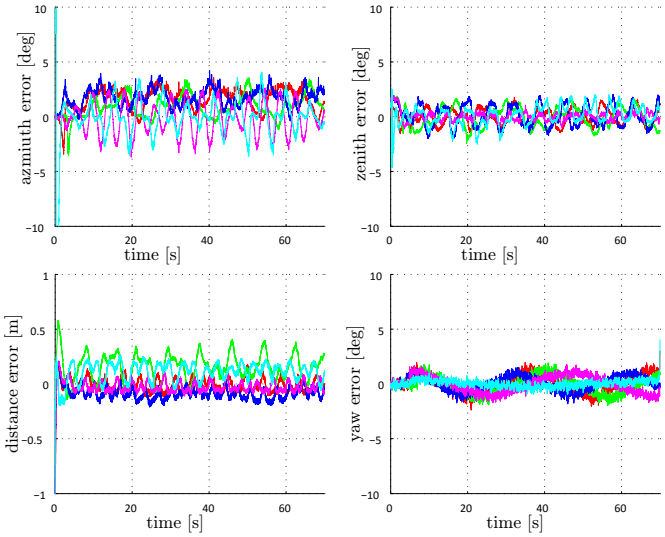


Fig. 20: Scenario B: error plots on azimuth, zenith-distance, distance and orientation estimates for A_2, ..., A_6 computed from A_1.

Moreover, random false negatives are added to prove the robustness of the proposed framework, while two quadrotors (A_7 and A_8) act as false positives, i.e., they are detected by the other robot detectors but do not communicate any information. Again, those measurements are compatible with the measurements provided by a visual tracking algorithm working on the images provided by a calibrated camera mounted on the robot.

During a typical experiment the quadrotors are driven to follow pre-defined trajectories through the use of the mocap, while the data provided to the GNU/Linux machine are stored. At the end, the data are subsequently synchronized and the estimation is conducted offline, but in real time. Because of this strategy, the data needed for one experiment can be collected by running the robots in a sequential way. In Fig. 18 we show two snapshots of an experiment, with the corresponding estimates computed in Scenario B.

The relative azimuth, zenith-distance, distance and orientation errors on the estimates computed in Scenario C are

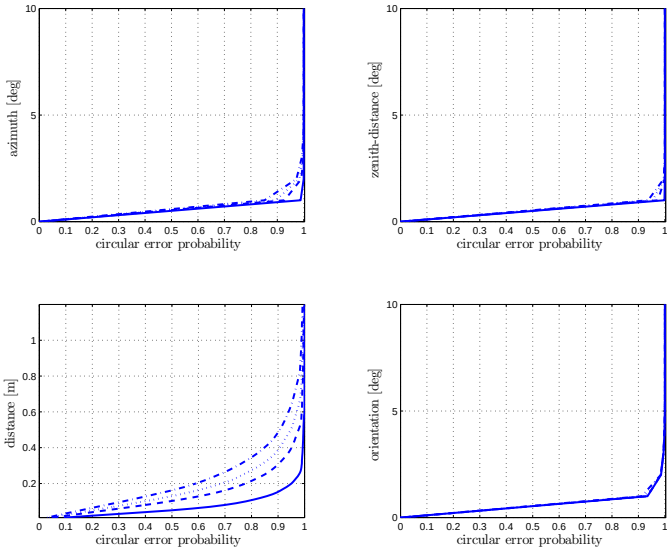


Fig. 21: Scenario B: error probabilities (lower is better) for the estimated robots $A_2, \ldots, A_6$ of an aggregate of eight simulations for different values of covariance of the noise on the bearing measurements: solid 0 deg, dashed 5 deg, dotted 10 deg, dash-dotted 20 deg.

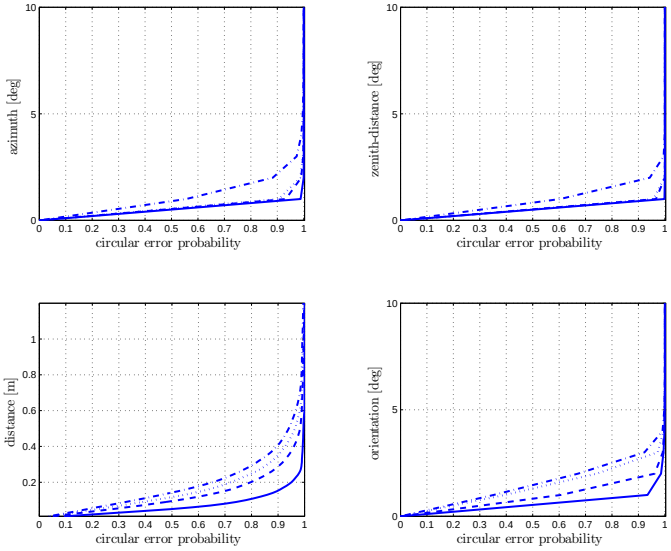


Fig. 22: Scenario B: error probabilities (lower is better) for the estimated robots $A_2, \ldots, A_6$ of an aggregate of eight simulations for different values of covariance of the noise on the motion measurements: solid 0 m/s and 0 deg/s, dashed 0.02 m/s and 5 deg/s, dotted 0.05 m/s and 10 deg/s, dash-dotted 0.1 m/s and 20 deg/s.

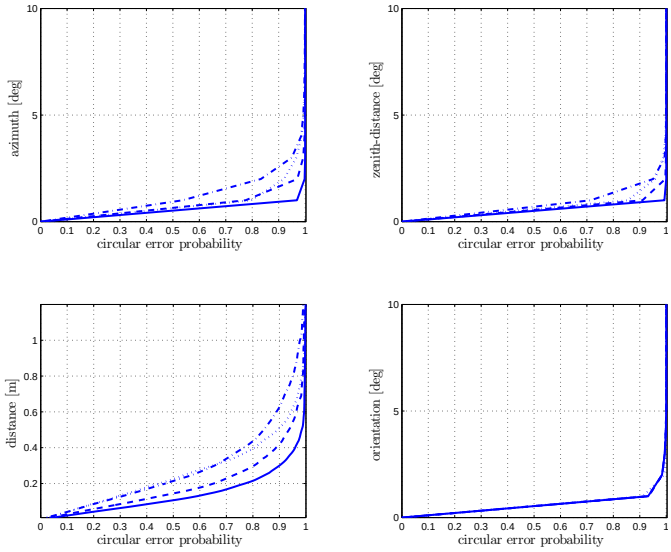


Fig. 23: Scenario C: error probabilities (lower is better) for the estimated robots $A_2, \ldots, A_6$ of an aggregate of eight simulations for different values of covariance of the noise on the bearing measurements: solid 0 deg, dashed 5 deg, dotted 10 deg, dash-dotted 20 deg.

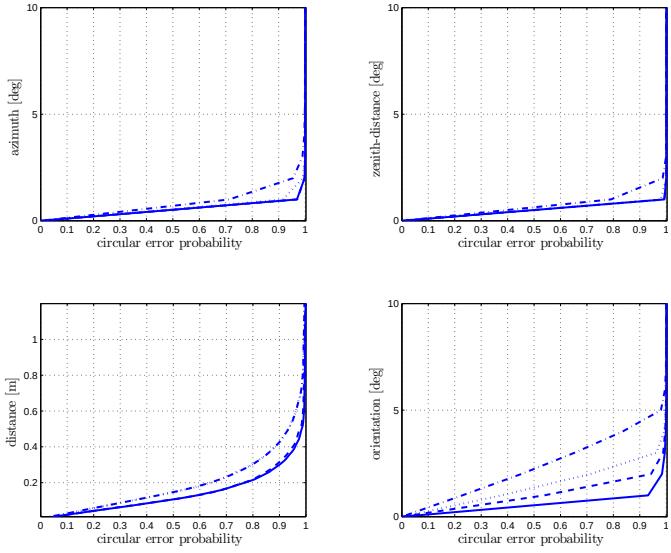


Fig. 24: Scenario C: error probabilities (lower is better) for the estimated robots $A_2, \ldots, A_6$ of an aggregate of eight simulations for different values of covariance of the noise on the motion measurements: solid 0 m/s^2 and 0 deg/s, dashed 2 m/s^2 and 5 deg/s, dotted 5 m/s^2 and 10 deg/s, dash-dotted 10 m/s^2 and 20 deg/s.

depicted in Fig. 19. The relative distances in the filters are initialized to 8 m w.r.t. a true distance of 3 m, showing that our mutual localization system is able not only to maintain the true distance but also to retrieve it. In addition, in the beginning of the experiment, when no belief is available, the multiple registration algorithm produces multiple solutions, causing the initial peak of the errors which is quickly recovered. The main improvement w.r.t. [21] is for the estimated relative angles. In fact, the error peaks for azimuth and zenith-distance drop from 10 deg in [21] to 5 deg of the current framework, while the orientation error decreases from 12 deg to 3 deg. This improvement in the performance is due to the higher frequency on the IMU output, as explained at the beginning of this section.

The plots of the errors affecting the estimates computed in Scenario B are presented in Fig. 20, and shows a similar behavior w.r.t. Scenario C. Although the greatest improvement in the performance is obtained on the estimates of the relative distances, whose maximum error drops at 0.26 m, w.r.t. 0.6 m of Scenario C, also the estimates of the angular quantities benefits from the employment of the velocity measurements. The improved performance can be explained considering that the update step of the PFs with the velocity needs only one integration in order to obtain the position, while the update

step of the PFs, if performed using the acceleration, needs a double integrator with a double integration of the noise acting on the measurements.

Video-clips of the experiment and the estimates computed in the two Scenarios are presented in the last part of the attached video.

D. Simulation results in 3D

Similarly to the 2D scenario, we have performed several simulations to study the behavior of our localization system with respect to noise. Also in this case, we have collected the data of eight simulations with varying number of robots (four to six) and run several times the mutual localization system with zero-mean and variable covariance additional noise either on the bearing or on the motion measurements.

Figures 21, 22, 23 and 24 present the aggregated results of those simulation campaigns for both Scenarios B and C. We have studied the behavior of the system in the presence of noise on the bearing or the motion measurements gathered by the robots, by adding zero-mean gaussian noise. In particular, we have added, on each simulation, noises with different levels of covariance (5 deg, 10 deg and 20 deg) for testing the robustness of the system w.r.t. noise on the bearing measurements. The results are shown in Figures 21 and 23. Regarding the robustness with respect to noise on the motion measurements, we have tested the system by varying the covariance on the linear and angular velocities in Scenario B (0.02 m/s and 5 deg/s, 0.05 m/s and 10 deg/s, 0.1 m/s and 20 deg/s) and on linear acceleration and angular velocities in Scenario C ($2\,\mathrm{m/s^2}$ and 5 deg/s, $5\,\mathrm{m/s^2}$ and 10 deg/s, $10\,\mathrm{m/s^2}$ and 20 deg/s). The results are shown in Figures 22 and 24. Altogether, this simulation campaign includes 56 runs of the mutual localization system for each Scenario. As it is evident from the plots, Scenario B outperforms Scenario C in terms of robustness and accuracy thanks to the introduction of the velocity measurements. The plots also show a behavior which is consistent with the 2D case, with higher sensitivity of the estimated distance to noise on the bearing measurements. However, the angular quantities are more sensitive to the noise on the motion measurements.

IX. Conclusions

In this paper, we have introduced the problem of mutual localization with anonymous relative-bearing measurements in multi-robot systems in three different operative scenarios, taking into consideration different platform (2D and 3D) and different sensors. The problem presents several challenges due to instantaneous ambiguities in the inversion of the measurement function in geometrically symmetric configurations. Hence, we have designed, implemented and tested a two-step system in order to solve the problem.

In the first step we apply geometrical considerations in order to reconstruct all the formations that are compatible with the measurements, up to an instantaneously unobservable scaling factor. Then, a bank of particle filters is meant to eliminate the non-time-consistent solutions and filter the noise, while reconstructing the missing scale of the true formation through integration with motion measurements. The system has been extensively tested through experiments in all the considered Scenarios.

For future work, we plan to integrate the developed mutual localization systems into working systems, closing the loop of the control on the produced estimates. In addition, we plan to further investigate the role of anonymity with a more thorough theoretical analysis and to study the noise propagation characteristics of the developed algorithms.

References

[1] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 962–975, 2009.

[2] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[3] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.

[4] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, 2010.

[5] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. on Automatic Control*, vol. 53, no. 11, pp. 2480–2496, 2008.

[6] 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*, Kobe, Japan, May 2009, pp. 3515–3522.

[7] 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*, Cancun, Mexico, Dec. 2008, pp. 3947–3952.

[8] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *1994 IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, May 1994, pp. 1250–1257.

[9] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[10] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[11] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland, Sep. 2002, pp. 434–439.

[12] A. Das, J. Spletzer, V. Kumar, and C. Taylor, "Ad hoc networks for localization and control," in *41th IEEE Conf. on Decision and Control*, Las Vegas, NV, Dec. 2002, pp. 2978–2983.

[13] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multi-robot cooperative localization," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

[14] 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*, Shanghai, China, May. 2011, pp. 4736–4743.

[15] N. Trawny, X. S. Zhou, K. Zhou, and S. I. Roumeliotis, "Inter-robot transformations in 3D," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 225–243, 2010.

[16] A. Martinelli, "Closed-form solutions for attitude, speed, absolute scale and bias determination by fusing vision and inertial measurements," Institut National de Recherche en Informatique et en Automatique., Tech. Rep., Feb. 2011.

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

[18] ——, "Mutual localization in multi-robot systems using anonymous relative measurements," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1302–1322, 2013.

[19] ——, "On the solvability of the mutual localization problem with anonymous position measures," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 3193–3199.
[20] 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*, San Francisco, CA, Sep. 2011, pp. 469–474.
[21] 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, pp. 791–798.
[22] M. Franceschelli and A.Gasparri, "On agreement problems with gossip algorithms in absence of common reference frames," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 4481–4486.
[23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
[24] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.
[25] 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*, vol. 5, no. 12, pp. 1663–1678, 2006.
[26] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I - the first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.
[27] P. Stegagno, M. Basile, H. H. Bülthoff, and A. Franchi, "A semi-autonomous UAV platform for indoor remote operation with visual and haptic feedback," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014.
[28] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano, "On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow," in *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 2012, pp. 491–497.
[29] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainly," *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
[30] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, Aug. 2005, pp. 1471–1476.
[31] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.
[32] P. Martin and E. Salaün, "The true role of accelerometer feedback in quadrotor control," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 1623–1629.
[33] A. Cristofaro and A. Martinelli, "3D cooperative localization and mapping: Observability analysis," in *2011 American Control Conference*, San Francisco, CA, June 2011, pp. 1630–1635.