

Ground and Aerial Mutual Localization Using Anonymous Relative-Bearing Measurements

Paolo Stegagno, *Member, IEEE*, Marco Cognetti, *Student Member, IEEE*, Giuseppe Oriolo, *Senior Member, IEEE*, Heinrich H. Bühlhoff, *Member, IEEE*, and Antonio Franchi, *Senior Member, IEEE*

Abstract—We present a decentralized algorithm for estimating mutual poses (relative positions and orientations) in a group of mobile robots. The algorithm uses relative-bearing measurements, which, for example, can be obtained from onboard cameras, and information about the motion of the robots, such as inertial measurements. It is assumed that all relative-bearing measurements are anonymous; i.e., each specifies a direction along which another robot is located but not its identity. This situation, which is often ignored in the literature, frequently arises in practice and remarkably increases the complexity of the problem. The proposed solution is based on a two-step approach: in the first step, the most likely unscaled relative configurations with identities are computed from anonymous measurements by using geometric arguments, while in the second step, the scale is determined by numeric Bayesian filtering based on the motion model. The solution is first developed for ground robots in SE (2) and then for aerial robots in SE (3). Experiments using Khepera III ground mobile robots and quadrotor aerial robots confirm that the proposed method is effective and robust w.r.t. false positives and negatives of the relative-bearing measuring process.

Index Terms—Aerial robots, anonymous measurements, bearing-only sensors, ground robots, multirobot systems, mutual localization, unknown data-association.

I. INTRODUCTION

THE designer of a decentralized multirobot system must consider the problem of 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 it 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 study 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]–[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*; i.e., to estimate the change of coordinates among its fixed frame and the fixed frames of the other robots. Otherwise, given the assumption that each robot is equipped with its own attached frame, one can define the *relative mutual localization* (RML) as the problem of estimating the change of coordinates among those attached 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 3-D 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] and [18], we have addressed the problem of 2-D 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 of 1) a multiple registration algorithm that computes all the possible changes of coordinates among the agents' relative frames using geometrical arguments

Manuscript received April 11, 2015; accepted May 27, 2016. Date of publication September 7, 2016; date of current version September 30, 2016. This paper was recommended for publication by Associate Editor D. Scaramuzza and Editor T. Murphey upon evaluation of the reviewers' comments.

P. Stegagno and H. H. Bühlhoff are with the Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany (e-mail: paolo.stegagno@tuebingen.mpg.de; hhb@tuebingen.mpg.de).

M. Cognetti and G. Oriolo are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, 00185 Roma, Italy (e-mail: cognetti@diag.uniroma1.it; oriolo@diag.uniroma1.it).

A. Franchi is with LAAS-CNRS, Université de Toulouse, CNRS, 31031 Toulouse CEDEX 4, France (e-mail: afranchi@laas.fr).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>. The multimedia attachment for the paper "Ground and Aerial Mutual Localization using Anonymous Relative Bearing Measurements" consists of a single 10 MB movie file in MPEG-4 format. The file shows three experiments of relative mutual localization performed with ground and aerial vehicles. This file has been tested in Quicktime media player as well as the VLC player, among others. The video clip presents three experiments of mutual localization using anonymous bearing measurements in different operative conditions: 1) 2-D ground robots with limited field of view, 2) 3-D aerial robots with full field of view and linear velocity measurements, and 3) 3-D aerial robots with full field of view and linear acceleration measurements.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2016.2593454

to invert the measurement map and 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 to further generalize in [20] and [21], we have considered 2-D and 3-D extensions to the case of bearing-only (rather than range-and-bearing) measurements, to allow the use of nondepth sensors, such as simple cameras, whose employment widens the field of applicability of our system. The objective of this study is to offer a comprehensive point of view on bearing-only mutual localization with anonymous measurements. To this aim, we will tackle the problem both in 2-D and 3-D worlds considering three among the most significant settings for robotics applications, defined as follows:

Scenario A (2-D, 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 (3-D, linear and angular velocity measurements):

The robots live in a 3-D world and are able to measure both their own linear and angular velocity in body frame.

Scenario C (3-D, linear acceleration and angular velocity measurements): The robots live in a 3-D 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 3-D case. Moreover, the introduction of Scenario A is functional to the explanation of the general case and provides more insights into some design choices.

Preliminary works regarding only Scenarios A and C appeared in our conference papers [20] and [21]. With respect to these works, we also address in this paper 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, and we present new experiments comparing 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 mutual localization 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. In Section V, we address Scenario A, and the developed system is then modified in Section VI to tackle the increased dimensionality of Scenarios B and C. 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 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 (mocap), and more in general off-board sensors, 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 (2-D and 3-D).

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

- 1) A *motion detector*, which provides some proprioceptive measurements of its own motion.
- 2) 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; 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 by 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 also addresses 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 case of anonymous measurements.

Most previous works in mutual localization apply Bayesian filters and/or geometrical considerations to perform the estimation 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 study, 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, a foregoing step is required 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 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. Each robot collects at each time-step the sensor perception

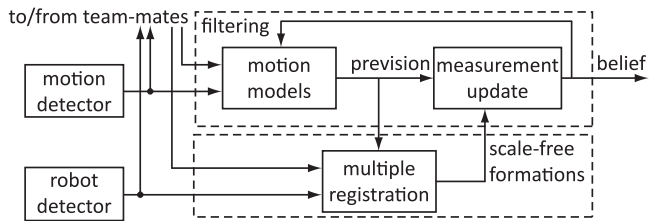


Fig. 1. Working principle of the mutual localization system.

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 registrations 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 Section 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, the reconstruction of the scale of the formation and the noise rejection are demanded to the filtering. While the time update in the filtering stage 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 filter is used in the multiple registration to help in the reconstruction of the measurements/robots correspondences and to prune the multiple solutions before the filtering stage.

III. CONSEQUENCES OF ANONYMITY

A key point of our mutual localization system is the design of a multiple registration algorithm that reconstructs the correspondences among the measurements and the measured robots on the basis of geometric considerations. It is well known from the literature that this problem admits in general multiple solutions [25] even in the simplified situation of nonanonymous position measurements and 2-D world. In such a case, frameworks (i.e., spatial configuration and measurement graph) that admit a single solution are said to be rigid, and there are known conditions and algorithms to individuate rigid and nonrigid 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 nonanonymous 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

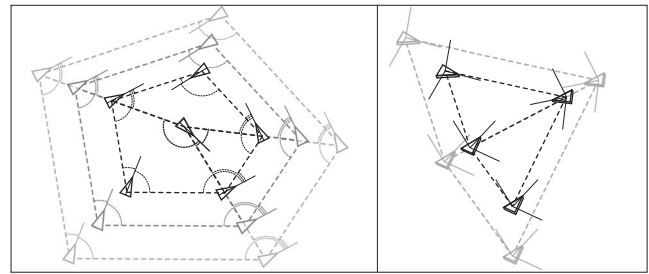


Fig. 2. Examples of 2-D (left) and 3-D (right) formations up to a scaling factor.

algorithm. In the rest of the paper, 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 2-D and 3-D (see 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 an MRAB problem always admits infinite solutions. In the following analysis, we will not consider the infinite solutions derived from 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 scaling factor.

Property 2: A solution of the MRAP problem is also a solution of the equivalent MRAB problem, whereas the opposite is not necessarily true. In fact, the additional distance information contained in the position measurements constitutes additional constraints that are not present in the case of bearing-only 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_j . 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-only measurements.

Property 3: Consider the 2-D MRAB problem. If the measurement sets of two robots coincide up to a 2-D rotation of all the measurements, then the 2-D 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 to create a different solution. This property can be generalized to the 3-D MRAB problem: if the measurement sets of two robots coincide up to a 3-D rotation of all the measurements, then the 3-D MRAB

	spatial arrangement	position measurements	solution(s)	bearing measurements	solution(s)
equilateral triangle					
isosceles triangle					
scalene triangle					

Fig. 3. Comparison of the solvability of the MRAP and MRAB problems in case of triangular formation.

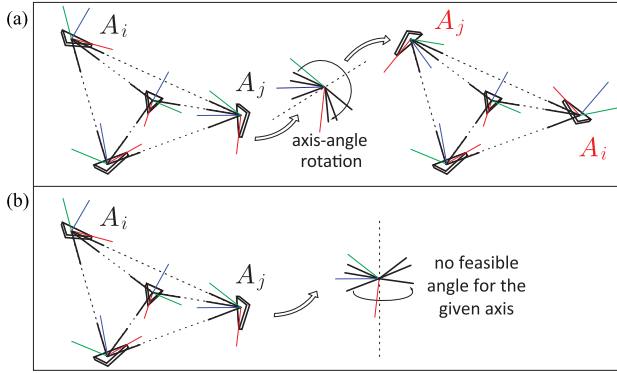


Fig. 4. (a) Example of axis-angle rotation in the 3-D MRAB problem; (b) same, with a common direction in space available to the robots.

problem admits multiple solutions. Fig. 4(a) shows an example of such situation.

Note that in the 3-D 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 2-D [see Fig. 4(b)]. In particular, if the sensor equipment includes an inertial measurement unit (IMU), each robot is able to estimate the gravity vector and to 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 3-D MRAB problem can be restricted to the horizontal plane. The simplification introduced by this approach is only one of two advantages. The second is that, as we will show in detail in Section VI, once the problem is reduced to 2-D, 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 3-D Case

By using 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

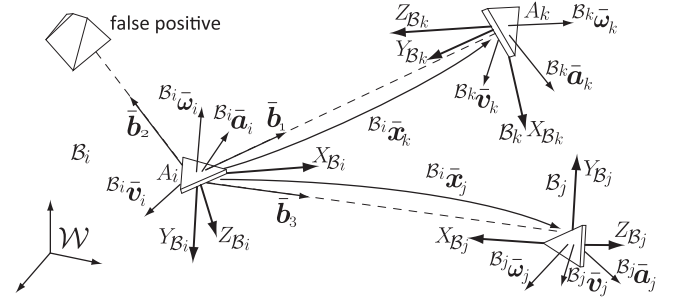


Fig. 5. Mutual localization with anonymous bearing measurements. Triangles are robots with their attached frame, velocity, acceleration, angular velocity measurements, and robot detectors.

bodies in $SE(3)$. Denote by $\mathcal{W} : \{O_{\mathcal{W}}, X_{\mathcal{W}}, Y_{\mathcal{W}}, Z_{\mathcal{W}}\}$ and $\mathcal{B}_i : \{O_{\mathcal{B}_i}, X_{\mathcal{B}_i}, Y_{\mathcal{B}_i}, Z_{\mathcal{B}_i}\}$, the inertial (world) frame, and the body frame attached to the center of mass of A_i , respectively. 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 ${}^{\mathcal{W}}\mathbf{p}_i \in \mathbb{R}^3$ of $O_{\mathcal{B}_i}$ in \mathcal{W} and the rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}_i} \in SO(3)$ between \mathcal{W} and \mathcal{B}_i . Denote with $\mathbf{R}_X(\cdot), \mathbf{R}_Y(\cdot), \mathbf{R}_Z(\cdot)$, the canonical rotation matrices about the axes X, Y, Z , respectively. Then, ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}_i}$ can be written as ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}_i} = \mathbf{R}_Z({}^{\mathcal{W}}\psi_i)\mathbf{R}_Y({}^{\mathcal{W}}\theta_i)\mathbf{R}_X({}^{\mathcal{W}}\phi_i)\mathbf{R}_X(\pi)$, 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, $\mathbf{R}_X(\pi)$ the rotation matrix between an NED frame to its corresponding North-West-Up (NWU) 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:

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

$${}^{\mathcal{B}_i}\mathbf{R}_{\mathcal{B}_j} = {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}_j}^T {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}_i} \quad (2)$$

and we denote by ${}^{\mathcal{B}_i}\mathbf{x}_{\mathcal{B}_j} = \{{}^{\mathcal{B}_i}\mathbf{p}_j, {}^{\mathcal{B}_i}\mathbf{R}_{\mathcal{B}_j}\}$ the *full relative configuration* of A_j in \mathcal{B}_i . Computing an estimate of ${}^{\mathcal{B}_i}\mathbf{x}_{\mathcal{B}_j}$ 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 origin as \mathcal{B}_i ($O_{\mathcal{C}_i} \equiv O_{\mathcal{B}_i}$) and such that ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}_i} = \mathbf{R}_Z({}^{\mathcal{W}}\psi_i)$. Hence, the rotation matrix between \mathcal{B}_i and \mathcal{C}_i is

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

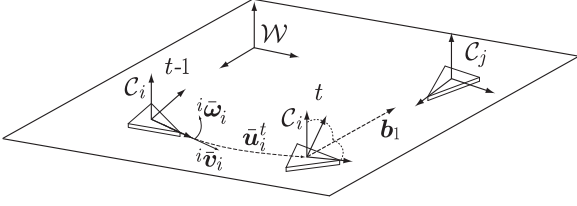


Fig. 6. Mutual localization with anonymous bearing measurements. The differences of the 2-D case w.r.t. the other scenarios.

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 2-D case. Note that the plane $X_{C_i}Y_{C_i}$ is parallel to the plane X_WY_W , or equivalently $Z_{C_i} \parallel Z_W$. In general, we will use the left superscripts $^{B_i}(\cdot)$ and $^i(\cdot)$ to denote quantities (\cdot) expressed in B_i and C_i , respectively.

B. Particularization to the 2-D Case

Relevant quantities in Scenario A are depicted in Fig. 6. If the robots lie on the X_WY_W plane, the third component of $^W p_i$, as well as the roll $^W \phi_i$ and pitch $^W \theta_i$ angles are equal to zero. Hence, the configuration of A_i is completely determined as $^W \tilde{x}_i = (^W \tilde{p}_i^T, ^W \psi_i)^T$, where $^W \tilde{p}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} ^W p_i \in \mathbb{R}^2$. Therefore, (3) reduces to

$$^{C_i} R_{B_i} = R_X(\pi). \quad (4)$$

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: $^i \bar{v}_i$, the linear velocity of A_i in the X_WY_W plane in C_i ; and $^i \bar{\omega}_i$, the angular rate about the axis Z_W . This can be easily done, e.g., by means the use of encoders on the wheels. The integration of $^i \bar{v}_i, ^i \bar{\omega}_i$ between two sampling instants $t-1$ and t , provides a measurement of the robot displacement \bar{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 that is usually present on those platforms is the IMU that provides to A_i noisy measurements $^{B_i} \bar{a}_i, ^{B_i} \bar{\omega}_i$ of its proper acceleration $^{B_i} a_i$, and angular velocity $^{B_i} \omega_i$ in body frame, defined by

$$^{B_i} a_i = ^W R_{B_i}^T (^W \ddot{p}_i - g e_3) \quad (5)$$

$$^{B_i} \omega_i = ^W R_{B_i}^T ^W \omega_i \quad (6)$$

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

In Scenario B only, we assume that in addition to the IMU, measurements $^{B_i} \bar{v}_i$ of the linear velocity $^{B_i} v_i$ 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]–[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*

$$^{B_i} b_j = ^W R_{B_i}^T \frac{^W p_j - ^W p_i}{\| ^W p_j - ^W p_i \|} \quad (7)$$

of all robots A_j in the perception set D_p attached to B_i .

The relative bearing $^{B_i} b_j \in \mathbb{S}^2$ is the unit-norm vector pointing toward the center of mass of A_j , expressed in 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 A_i .

Note that a relative bearing 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

$$b = (\sin \zeta \cos \alpha \quad \sin \zeta \sin \alpha \quad \cos \zeta)^T. \quad (8)$$

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 $^{B_i} B^t = \{\bar{b}_1, \bar{b}_2, \dots\}$ 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 $^{B_i} B^t$ 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{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 C_i using the rotation matrix given by (4), while in Scenarios B and 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 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, provided that it lies in the *communication set* D_c rigidly attached to 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 changes with 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 RML 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

TABLE I
MAIN SYMBOLS WITH THE SCENARIOS WHERE THEY ARE USED

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
${}^{\mathcal{F}}\mathbf{p}_i$	position of A_i in \mathcal{F}	ABC
${}^{\mathcal{F}}\mathbf{R}_{\mathcal{G}}$	rotation matrix from the frame \mathcal{G} to the frame \mathcal{F}	ABC
w_{ψ_i}	yaw angle of A_i	ABC
w_{θ_i}, w_{ϕ_i}	pitch and roll angles of A_i	BC
${}^{\mathcal{F}}\mathbf{x}_{\mathcal{G}}$	configuration (position and orientation) of \mathcal{G} in \mathcal{F}	BC
${}^{\mathcal{F}}\mathbf{p}_{\mathcal{G}}$	reduced position of \mathcal{G} in \mathcal{F}	BC
${}^{\mathcal{F}}\tilde{\mathbf{x}}_{\mathcal{G}}$	reduced configuration (${}^{\mathcal{F}}\tilde{\mathbf{p}}_{\mathcal{G}}, {}^{\mathcal{F}}\psi_{\mathcal{G}}$) of \mathcal{G} in \mathcal{F}	BC
\mathbf{v}_i	linear velocity of A_i	ABC
$\boldsymbol{\omega}_i$	angular velocity of A_i	ABC
\mathbf{u}_i^t	displacement on the plane of A_i between $t-1$ and t	A
\mathbf{a}_i	linear acceleration of A_i	BC
iB	feature set relative bearing of A_j	ABC
\mathbf{b}_j	relative bearing of A_j	ABC
\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

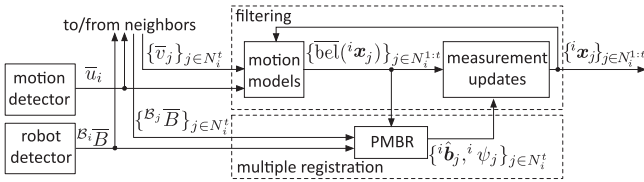


Fig. 7. Scheme of the mutual localization system that runs on A_i in Scenario A.

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 $B_i \mathbf{x}_j$ of all A_j that have been in communication with A_i , (i.e., such that $j \in N_i^{1:t} = N_i^1 \cup N_i^2 \cup \dots \cup N_i^t$) 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 \mathbf{u}_i^t ;
- 2) the feature set ${}^iB^t$;

and the RML problem can be formalized as

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

Algorithm 1: PMBR.

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

- 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** $\{d_1 \in D_1, d_2 \in D_2, d_3 \in D_3\}$ **do**
 - if** $d_1 + d_2 + d_3 = \pi \pm \tau_1$ **then**
 - $(T_a, T_b) = \text{computeTriangles}(d_1, d_2, d_3)$
 - $\mathcal{T}_{\text{raw}} = \mathcal{T}_{\text{raw}} \cup \{T_a, T_b\}$
- 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** $d_1 \in {}^iD^t$ **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** ${}^i\hat{\xi}_j^t = \text{extractAziOri}(\mathcal{S})$

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

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

- 9 **foreach** $T \in \mathcal{T}_{\text{rat}}$ **do**
 - if** $\text{commonSide}(S, T)$ **and** $T \notin S$ **then**
 - $\mathcal{S}_{\text{new}} = \mathcal{S}_{\text{new}} \cup \{\text{join}(S, T)\}$
- 10 **foreach** $S \in \mathcal{S}_{\text{new}}$ **do**
 - if** $\text{count3Intersections}(S) \geq \text{th}_3$ **then** $\mathcal{S}_{\text{rat}} = \mathcal{S}_{\text{rat}} \cup \{S\}$
- 11 **foreach** $S \in \mathcal{S}_{\text{rat}}$ **do**
 - if** $\text{areIrreconcilable}(S, S'), \forall S' \in \mathcal{S}_{\text{irr}}$ **then** $\mathcal{S}_{\text{irr}} = \mathcal{S}_{\text{irr}} \cup \{S\}$
- 12 **foreach** $S \in \mathcal{S}_{\text{irr}}$ **do**
 - if** $\text{beliefRating2D}(S) \geq \text{th}_4$ **then** $\mathcal{S}_{\text{best}} = \mathcal{S}_{\text{best}} \cup \{S\}$
- 13 **if** $\mathcal{S}_{\text{best}} = \emptyset$ **then return** \mathcal{S}
- 14 **else foreach** $S \in \mathcal{S}_{\text{best}}$ **do** $\mathcal{S} = \mathcal{S} \cup \text{branch}(S, \mathcal{T}_{\text{rat}})$

belief:

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

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

A. Multiple Registration

In this scenario, the multiple registration is performed by PMBR, a probabilistic 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 “viewpoints” consist only of bearing angles, the scale of the relative poses cannot be recovered. In particular, given the sets of features ${}^iB^t, \{{}^jB^t\}_{j \in N_i^t}$ and the current beliefs $\{\overline{\text{bel}}({}^i\tilde{\mathbf{x}}_j^t)\}_{j \in N_i^t}$ computed in the particle filters through the motion model of the robots, PMBR derives an estimate ${}^i\hat{\xi}_j^t$ 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.

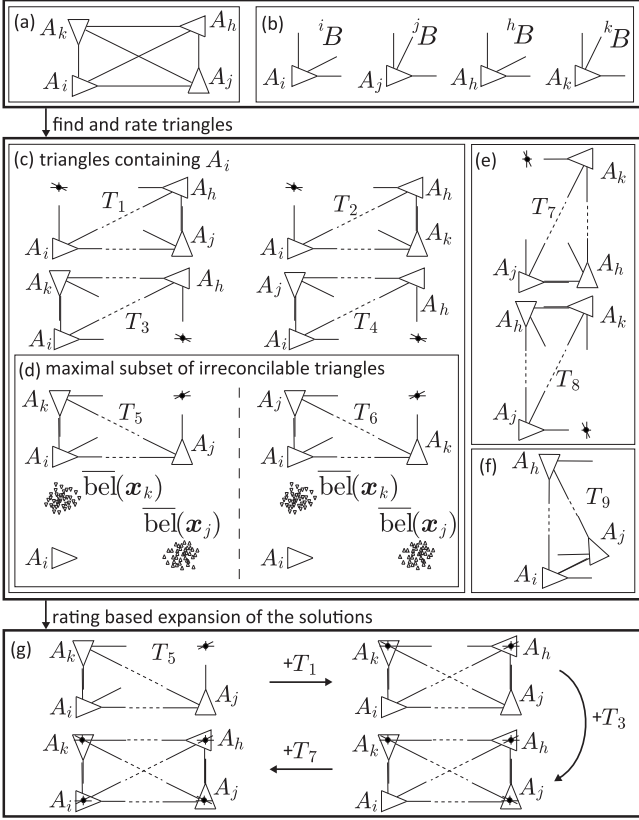


Fig. 8. Illustration of the basic steps of PMBR in a simple situation: (a) true configuration, (b) feature sets, (c) all the triangles having one three-intersection and containing A_i , (d) one choice for the maximal subset \mathcal{T}_{irr} and comparison with the current belief, (e) all the triangles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

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

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

and in the pseudo-code is computed through the function `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 $\{\mathbf{b}_r, \mathbf{b}_s\}$, and each pair identifies two difference angles, one larger and one smaller than π , with only the second being part of D . If $\mathbf{b}_r - \mathbf{b}_s = \pi$, 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) $\mathbf{d}_1 \in D_i, \mathbf{d}_2 \in$

$D_j, \mathbf{d}_3 \in D_k$ whose sum is π , with a certain tolerance τ_1 needed to account for measurement noise:

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

As explained in Section 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. 8(c) and T_9 in Fig. 8(f). Note that triangles include the identities of the robots at their vertices. Once appropriate triplets are identified, the triangle pairs are computed through the function `computeTriangles(\diamond, \square, \circ)` and collected in a set \mathcal{T}_{raw} .

Line 3) Three-Intersection Rating: When three robots of a triangle see a fourth robot [e.g., A_h in Fig. 8(a)], their sets of features include three rays (one from each feature set) that intersect in a single point (we call this a *three-intersection*). Based on this idea, the algorithm rates all triangles by counting their three-intersections (function `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}_{rat} . Note that a simple two-intersection does not provide any useful information, since two nonparallel lines on the plane will always intersect.

Line 4) Irreconcilable Triangles: From the set \mathcal{T}_{rat} , one extracts a maximal subset \mathcal{T}_{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 `Irreconcilable(\diamond, \circ)` to test whether two triangles are irreconcilable. The result of the triangle finding and rating process is illustrated in Fig. 8(c)–(f).

Line 5) Belief Rating: With the function `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

$$P({}^i \hat{\mathbf{x}}_j^t) = \sum_{A_j \in T} \int p({}^i \hat{\mathbf{x}}_j^t \mid {}^i \tilde{\mathbf{x}}_j^t) \overline{\text{bel}}({}^i \tilde{\mathbf{x}}_j^t) d^i \tilde{\mathbf{x}}_j^t \quad (12)$$

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 `branch(\diamond, \circ)`. Each triangle in $\mathcal{T}_{\text{best}}$ is the base of a branch of the algorithm, being selected as 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 three-intersections. In Fig. 8(g), the only branch of the algorithm has T_5 as first partial solution. 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 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} come in principle from many different branches, it is possible to rate them again (with three-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}_{rat} that have 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}_{new} of partial solutions is created.

Each partial solution is 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}_{new} is rated counting out its total number of three-intersections (line 10). Having more triangles joined together, it is possible that more than three rays intersect in the same point. In general, an n -intersection counts as $n!/3!(n-3)! = n(n-1)(n-2)/6$ three-intersections. Among the best rated partial solutions of each branch, collected in \mathcal{S}_{rat} , the algorithm selects a maximal subset of irreconcilable solutions \mathcal{S}_{irr} (line 11). Among those, only the solutions that fit with the current belief according to (12) are collected in $\mathcal{S}_{\text{best}}$ (line 12) and used as partial solutions in the following step, expanding a branch for each of them (line 14).

In the example of Fig. 8(g), 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 implies 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 multimodal that is desirable since multiple solutions may be returned from the multiple registration algorithm.

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

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

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

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

$${}^i\tilde{\mathbf{x}}_j = ({}^i\tilde{\mathbf{x}}_j \ominus (\mathbf{u}_i \oplus n_{\mathbf{u}_i})) \oplus (\mathbf{u}_j \oplus n_{\mathbf{u}_j})$$

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

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 $\bar{\mathbf{u}}_j^{t_a}, \dots, \bar{\mathbf{u}}_j^{t_b}$, 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 $\{\text{bel}({}^i\tilde{\mathbf{x}}_j^t)\}$.

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

$$p({}^i\tilde{\mathbf{x}}_j|\hat{\xi}_j) = F p({}^i\hat{\xi}_j|{}^i\tilde{\mathbf{x}}_j)p({}^i\tilde{\mathbf{x}}_j) \quad (13)$$

where F is another normalization factor. Equation (13) allows the computation of the posteriors $\{\text{bel}({}^i\tilde{\mathbf{x}}_j^t)\}$ as depicted in Fig. 7 by using the beliefs $\{\text{bel}({}^i\tilde{\mathbf{x}}_j^t)\}$ 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). Martinelli and Siegwart [30] proved that the system is observable; hence, our combination of inputs and measurements is suitable to fulfill this task.

Standard expedients can be applied to improve the performance of the filter. The initial prior distribution can be generated using the first measurements. Similarly, if new robots are encountered during operation, 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 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: (3-D RML with anonymous bearing measurements) For $t = 1, 2, \dots$ and $j \in N_i^{1:t}$, compute the following belief:

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

with

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

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 2-D to 3-D. In particular, the additional nonzero 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 2-D 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 an 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 Scenarios 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 ${}^{\mathcal{W}} \hat{\phi}_{B_i}$ and ${}^{\mathcal{W}} \hat{\theta}_{B_i}$ of its roll and pitch angles using the first two components of the IMU measurements ${}^{B_i} \bar{\mathbf{a}}_i$, ${}^{B_i} \bar{\boldsymbol{\omega}}_i$. This is achieved using a complementary filter (e.g., see [31], [32]).

Then, A_i is able to compute an estimate ${}^{C_i} \hat{\mathbf{R}}_{B_i}$ of the rotation matrix ${}^{C_i} \mathbf{R}_{B_i}$ between the B_i and C_i by plugging ${}^{\mathcal{W}} \hat{\phi}_{B_i}$ and ${}^{\mathcal{W}} \hat{\theta}_{B_i}$ in (3). By the use of ${}^{C_i} \hat{\mathbf{R}}_{B_i}$, each A_i is able to express its own measurements in C_i as follows:

$$\hat{\mathbf{a}}_i = {}^{C_i} \hat{\mathbf{R}}_{B_i} (0 \ 0 \ {}^{B_i} \bar{a}_{i_z})^T \quad (16)$$

$$\hat{\boldsymbol{\omega}}_i = {}^{C_i} \hat{\mathbf{R}}_{B_i} (0 \ 0 \ {}^{B_i} \bar{\omega}_{i_z})^T \quad (17)$$

$$\hat{\mathbf{v}}_i = {}^{C_i} \hat{\mathbf{R}}_{B_i} {}^{B_i} \bar{\mathbf{v}}_i \text{ in Scenario B} \quad (18)$$

$${}^i \hat{B} = \{ \hat{\mathbf{b}}_h = {}^{C_i} \hat{\mathbf{R}}_{B_i} \bar{\mathbf{b}}_h \ \forall \bar{\mathbf{b}}_h \in {}^{B_i} B \}. \quad (19)$$

Note that $\hat{\mathbf{a}}_i$ 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 ${}^{B_i} \bar{\mathbf{a}}_i$ and ${}^{B_i} \bar{\boldsymbol{\omega}}_i$, since they have been already used to compute the roll and pitch angles. For example, this approximation is reasonable for a typical quadrotor if 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 $\hat{\mathbf{a}}_i$ during the motion update, since $\hat{\mathbf{v}}_i$ is available.

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

$${}^{\mathcal{W}} \hat{\psi}_{B_i} = \begin{pmatrix} 0 & \sin {}^{\mathcal{W}} \hat{\phi}_{B_i} & \cos {}^{\mathcal{W}} \hat{\phi}_{B_i} \\ \cos {}^{\mathcal{W}} \hat{\theta}_{B_i} & \cos {}^{\mathcal{W}} \hat{\theta}_{B_i} & \end{pmatrix} {}^{B_i} \bar{\boldsymbol{\omega}}_i = \mathbf{f}_{B_i}^T {}^{B_i} \bar{\boldsymbol{\omega}}_i \quad (20)$$

where $\mathbf{f}_{B_i}^T$ is the covector that transforms the angular velocity in body frame into the yaw rate.

Once roll and pitch are computed and those rotations are applied, the system can solve a simpler problem w.r.t. Problem 2, consisting in retrieving the identities of the relative bearing measurements and estimating a *reduced relative configuration* ${}^i \mathbf{x}_j = \{ {}^i \mathbf{p}_j, {}^i \mathbf{R}_j \}$, $j \in N_i^{1:t}$, where

$${}^i \mathbf{p}_j = {}^{\mathcal{W}} \mathbf{R}_{C_i}^T ({}^{\mathcal{W}} \mathbf{p}_j - {}^{\mathcal{W}} \mathbf{p}_i) \quad (21)$$

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

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

$${}^{B_i} \hat{\mathbf{p}}_j = {}^{C_i} \hat{\mathbf{R}}_{B_i}^T {}^i \hat{\mathbf{p}}_j \quad (23)$$

$${}^{B_i} \hat{\mathbf{R}}_{B_j} = {}^{C_i} \hat{\mathbf{R}}_{B_i}^T {}^i \hat{\mathbf{R}}_j. \quad (24)$$

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

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

$$\text{bel}({}^i \mathbf{x}_j) = P({}^i \mathbf{x}_j^t \mid \hat{\mathbf{m}}_i^{1:t}, {}^i \hat{B}^{1:t}, \{ \hat{\mathbf{m}}_j^{1:t}, {}^j \hat{B}^{1:t} \}_{j \in N_i^{1:t}}) \quad (25)$$

with

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

To solve Problem 3, we need to recover the identities of the measurements in \hat{B}_i , the relative orientations ${}^i \mathbf{R}_j$, and the relative distances $\| {}^i \mathbf{p}_j \|$ using the system 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 $\hat{\mathbf{m}}_i^t$;
- 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 be needed only to compute ${}^{B_i} \hat{\mathbf{x}}_{B_j}$, while they are not needed to compute ${}^i \mathbf{x}_j$.

B. Multiple Registration

Once the feature sets are expressed in $\mathcal{C}_1, \dots, \mathcal{C}_n$, PMBR-3D exploits the same geometrical principle of its 2-D version; hence, many steps are the same. Its pseudo-code description is

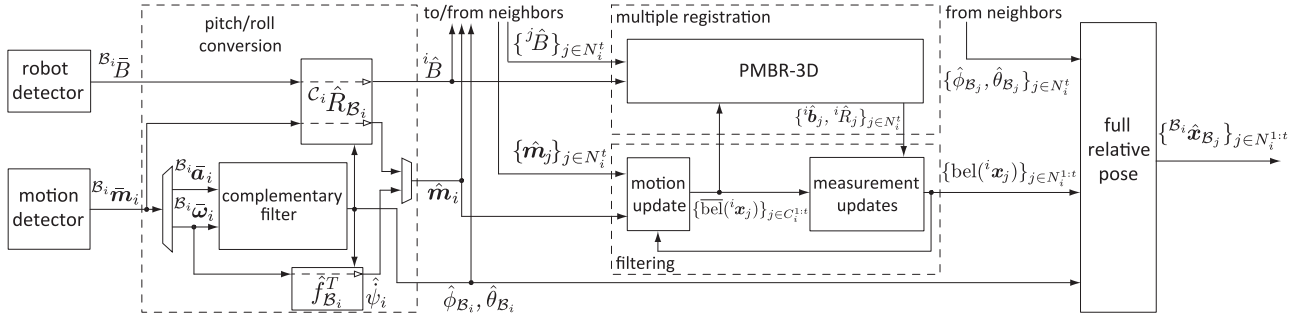


Fig. 9. Scheme of the mutual localization system that runs on A_i in Scenarios B and C.

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  ${}^i B^t, \{^j B^t\}_{j \in N_i^t}$ , beliefs  $\{\text{bel}({}^i x_j^t)\}_{j \in N_i^t}$  *
output: relative bearing-orientation estimates  $\{\hat{\xi}_j^t\}_{j \in N_i^t}$ 
init :  $\mathcal{T}_{\text{raw}} = \emptyset, \mathcal{T}_{\text{rat}} = \emptyset, \mathcal{T}_{\text{irr}} = \emptyset, \mathcal{T}_{\text{best}} = \emptyset, \mathcal{S} = \emptyset$ 
1 foreach  $j \in \{i\} \cup N_i^t$  do  ${}^j D^t = \text{computeDiffAngleSet}({}^j B^t)$ 
2 foreach  $\{D_1, D_2, D_3\} \subseteq \{^i D^t\} \cup \{^j D^t\}_{j \in N_i^t}$  do
   foreach  $\{d_1 \in D_1, d_2 \in D_2, d_3 \in D_3\}$  do
     if  $d_1 + d_2 + d_3 = \pi \pm \tau_1$  then
        $(T_a, T_b) = \text{computeTriangles}(d_1, d_2, d_3)$ 
       if  $\text{checkZenithConstraints}(T_a)$  then +
          $\mathcal{T}_{\text{raw}} = \mathcal{T}_{\text{raw}} \cup \{T_a\}$  *
       if  $\text{checkZenithConstraints}(T_b)$  then +
          $\mathcal{T}_{\text{raw}} = \mathcal{T}_{\text{raw}} \cup \{T_b\}$  *
3 foreach  $T \in \mathcal{T}_{\text{raw}}$  do
   if  $\text{count2Intersections}(T) \geq \text{th}_1$  then  $\mathcal{T}_{\text{rat}} = \mathcal{T}_{\text{rat}} \cup \{T\}$  *
4 foreach  $T \in \mathcal{T}_{\text{rat}}$  do
   if  $d_1 \in {}^i D^t$  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  $\text{beliefRating3D}(T) \geq \text{th}_2$  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  ${}^i \hat{\xi}_j^t = \text{extractAziOri}(\mathcal{S})$ 
8 function  $\text{branch}(\text{partialSolution } S, \text{triangleSet } \mathcal{T}_{\text{rat}})$ :
   init :  $\mathcal{S}_{\text{new}} = \emptyset, \mathcal{S}_{\text{rat}} = \emptyset, \mathcal{S}_{\text{irr}} = \emptyset, \mathcal{S}_{\text{best}} = \emptyset$ 
9 foreach  $T \in \mathcal{T}_{\text{rat}}$  do
   if  $\text{commonSide}(S, T)$  and  $T \notin \mathcal{S}$  then
     if  $\text{checkZenitConstraints}(S, T)$  then +
        $\mathcal{S}_{\text{new}} = \mathcal{S}_{\text{new}} \cup \{\text{join}(S, T)\}$  *
10 foreach  $S \in \mathcal{S}_{\text{new}}$  do
   if  $\text{count2Intersections}(S) \geq \text{th}_3$  then  $\mathcal{S}_{\text{rat}} = \mathcal{S}_{\text{rat}} \cup \{S\}$  *
11 foreach  $S \in \mathcal{S}_{\text{rat}}$  do
   if  $\text{areIrreconcilable}(S, S'), \forall S' \in \mathcal{S}_{\text{irr}}$  then  $\mathcal{S}_{\text{irr}} = \mathcal{S}_{\text{irr}} \cup \{S\}$ 
12 foreach  $S \in \mathcal{S}_{\text{irr}}$  do
   if  $\text{beliefRating3D}(S) \geq \text{th}_4$  then  $\mathcal{S}_{\text{best}} = \mathcal{S}_{\text{best}} \cup \{S\}$  *
13 if  $\mathcal{S}_{\text{best}} = \emptyset$  then return  $\mathcal{S}$ 
14 else foreach  $S \in \mathcal{S}_{\text{best}}$  do  $\mathcal{S} = \mathcal{S} \cup \text{branch}(S, \mathcal{T}_{\text{rat}})$ 

```

given in Algorithm 2, where we have highlighted the lines that differ or are added with respect to the 2-D 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

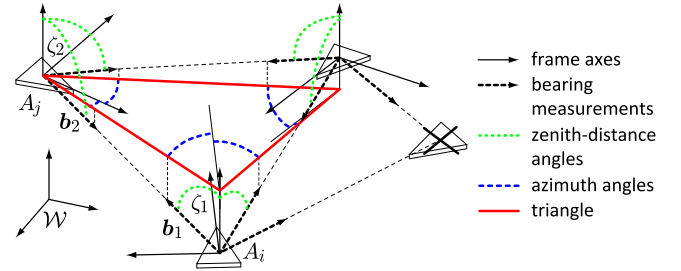


Fig. 10. Three robots measuring each other and the corresponding 2-D triangle.

horizontal¹ 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 2-D case considering only the azimuth measurements.

Assume now that in the triangle shown in Fig. 10, the bearing measurements $b_1 \in {}^i B$ and $b_2 \in {}^j B$ are, respectively, associated with A_j and A_i . The corresponding zenith-distance measurements ζ_1, ζ_2 must comply the constraint

$$\zeta_1 + \zeta_2 = \pi \pm \tau_2 \quad (27)$$

as specified in line 2, where τ_2 is a tolerance to account for measurement noise. Although this additional constraint is not relevant if the robots all lie at the same altitude, in general case, it 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 three pairs of associations, must satisfy three constraints in the form of (27).

Another difference w.r.t. the 2-D 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 two-intersections as rating policy instead of the number of three-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 $\text{beliefRating3D}(\diamond)$ (see lines 5 and 12), which uses the following metric function

¹i.e., parallel to $X_W Y_W$.

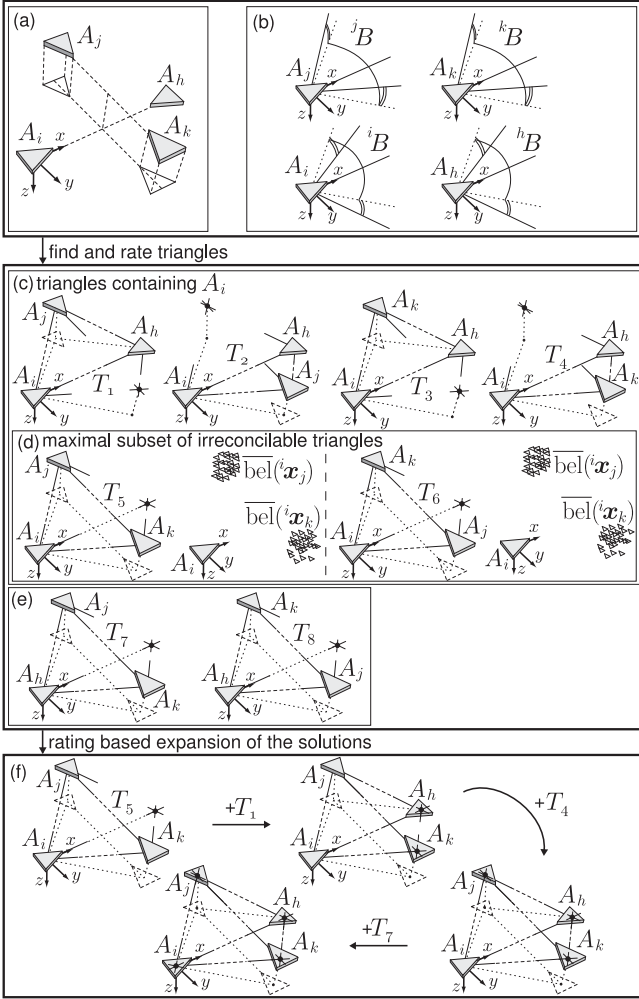


Fig. 11. Execution of PMBR-3D in an ambiguous situation: (a) true configuration, (b) initial feature sets, (c) triangle found in the first step containing the owner of the algorithm and their triple intersections, (d) maximal subset of irreconcilable triangles and their comparison with the current belief,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

Instead of (12)

$$P(\{\hat{\mathbf{b}}_j, \hat{\mathbf{R}}_j\}) = \sum_{A_j \in T} \int p(\{\hat{\mathbf{b}}_j, \hat{\mathbf{R}}_j\} | \mathbf{x}_j) \overline{\text{bel}}(\mathbf{x}_j) d^3 \mathbf{x}_j \quad (28)$$

where $\overline{\text{bel}}(\mathbf{x}_j)$ comes from the corresponding particle filter. In (28), the scale is selected to maximize the rating.

The basic steps of PMBR-3D are also shown in the illustrative example in Fig. 11. Consider the situation in Fig. 11(a), in which four robots are arranged in a “square” formation with the opposite vertices at the same height; the corresponding feature sets are shown in Fig. 11(b). The triangles finding process is conducted as explained above. Then, the two-intersections rating is computed as the 2-D case, but it is worth to highlight that, in general, an n -intersection accounts for $n!/2(n-2)! = n(n-1)/2$ two-intersections. The triangles whose rating is above a certain threshold are collected in the triangle list \mathcal{T}_{rat} [see Fig. 11(c)–(e)].

The rest of the algorithm follows the same conceptual steps in the 2-D case. It extracts from \mathcal{T}_{rat} a maximal subset of irreconcilable triangles \mathcal{T}_{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 $\mathcal{T}_{\text{best}}$.

Each triangle in $\mathcal{T}_{\text{best}}$ is the base of a branch of the algorithm and is selected as the partial solution S at the first step of its branch. In Fig. 11(f), there is only one branch that has T_5 as first partial solution. Then, the partial solution of each branch is expanded joining triangles that have common edges with it [see Fig. 11(f)]. This recursive process is conducted in the same way of the 2-D case, using two-intersections and the metric in (28) as ratings. In the case of Fig. 11(f), the algorithm expands a partial solution by joining to 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 two-intersection and belief criteria. Similarly to the 2-D case, the result of the algorithm is a list of pairs $\{\hat{\mathbf{b}}_j, \hat{\mathbf{R}}_j\}$ for each A_j .

C. Filtering

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

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

$${}^i \dot{\mathbf{p}}_j = {}^i \mathbf{v}_j \quad (29)$$

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

where we denoted with ${}^i \mathbf{v}_j$ the velocity of O_{C_j} in 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

$${}^i \mathbf{R}_j = \mathbf{R}_Z(-{}^w \psi_i) \mathbf{R}_Z({}^w \psi_j) = \mathbf{R}_Z({}^w \psi_j - {}^w \psi_i) \quad (31)$$

we can replace (30) with

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

being ${}^i \psi_j = \psi_{B_j} - \psi_{B_i}$ and $f_{B_i}^T$, $f_{B_j}^T$ defined by (20).

In Scenario B, an estimate ${}^i \hat{\mathbf{v}}_j$ of ${}^i \mathbf{v}_j$ is directly available from the estimates $\hat{\mathbf{v}}_i$, $\hat{\mathbf{v}}_j$ given by (18) as ${}^i \hat{\mathbf{v}}_j = {}^i \hat{\mathbf{R}}_j \hat{\mathbf{v}}_j - \hat{\mathbf{v}}_i$. Hence, the kinematic model given by (29) and (32) is sufficient for the motion update of the filter. Therefore, in Scenario B, the state of each particle is the four-dimensional tuple ${}^i \boldsymbol{\chi}_j = ({}^i \mathbf{p}_j, {}^i \psi_j) \in \mathbb{R}^3 \times \mathbb{S}^1$, 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 ${}^i\hat{\mathbf{v}}_j, {}^i\hat{\boldsymbol{\omega}}_i, {}^i\hat{\psi}_i, {}^i\hat{\psi}_j$ in (29) and (32).

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

$${}^i\dot{\mathbf{v}}_j = {}^i\mathbf{R}_j\mathbf{a}_j - \mathbf{a}_i + [\boldsymbol{\omega}_i]_{\times}{}^i\mathbf{v}_j \quad (33)$$

in which ${}^i\mathbf{R}_j$ can be computed using (31). Therefore, in Scenario C, the state of each particle is the seven-dimensional tuple ${}^i\boldsymbol{\chi}_j = ({}^i\mathbf{p}_j, {}^i\mathbf{v}_j, {}^i\psi_j) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{S}^1$, and the motion update step of the j th particle filter is obtained by plugging ${}^i\hat{\mathbf{a}}_i, {}^i\hat{\mathbf{a}}_j, {}^i\hat{\boldsymbol{\omega}}_i, {}^i\hat{\psi}_{B_i}, {}^i\hat{\psi}_{B_j}$ in (29), (32), and (33).

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 Scenarios 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 between those times after the communication is restored.

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

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

$$P({}^i\boldsymbol{\chi}_j | {}^i\hat{\mathbf{b}}_j, {}^i\hat{\psi}_j) = FP({}^i\hat{\mathbf{b}}_j, {}^i\hat{\psi}_j | {}^i\boldsymbol{\chi}_j)P({}^i\boldsymbol{\chi}_j) \quad (34)$$

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

$$P({}^i\boldsymbol{\chi}_j | {}^i\hat{\mathbf{v}}_j) = FP({}^i\hat{\mathbf{v}}_j | {}^i\boldsymbol{\chi}_j)P({}^i\boldsymbol{\chi}_j). \quad (35)$$

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

VII. EXECUTION TIME

Since PMBR and PMBR-3D are recursive algorithms that allow multiple branches, their execution times depend mostly on the number of branches and the depth of the recursion. In turn, these are functions of the geometric configuration of the multirobot 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 section.

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

- 1) computeTriangles($\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$);
- 2) checkZenithConstraints(T);
- 3) areIrreconcilable(T, T');
- 4) beliefRating2D(T) and beliefRating3D(T).

TABLE II
EXECUTION TIMES 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. (36)	$O(N^3, M^2)$
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

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

- 1) commonSide(S, T);
- 2) checkZenithConstraints(S');
- 3) join(S, T);
- 4) areIrreconcilable(S', S'');
- 5) beliefRating2D(S') and 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 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 Section V-A, the corresponding sets will have $P(P-1)/2$ difference angles, so the time needed for computeDiffAngleSet(${}^iB^i$) 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, 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 $Q_{\text{raw}}, Q_{\text{rat}}, Q_{\text{irr}}$, and H be the cardinality of $\mathcal{T}_{\text{raw}}, \mathcal{T}_{\text{rat}}, \mathcal{T}_{\text{irr}}$, and \mathcal{S} , respectively. In each triangle, there are $P-2$ rays starting from each vertex that are not associated. Then, there is a maximum of $3(P-2)^2$ two-intersections to be checked, and three-intersections can be checked by matching two-intersections that are close to each other. Hence, the execution of line 3 takes $3Q_{\text{raw}}(P-2)^2 = O(P^2, Q_{\text{raw}})$ time. Similarly, $Q_{\text{rat}}(Q_{\text{rat}}-1)/2 = O(Q_{\text{rat}}^2)$ calls of areIrreconcilable are needed in line 4, and Q_{irr} calls of 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 $H_{n, \text{new}}, H_{n, \text{rat}}$, and $H_{n, \text{irr}}$ be the

cardinality of \mathcal{S}_{new} , \mathcal{S}_{rat} , \mathcal{S}_{irr} in the n th recursion. Using the previous results, and being $\max(H_{n, \text{irr}}) = H_{n, \text{rat}}$, we can show that the time for 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

$$\sum_{n=1}^Q [(Q_{\text{rat}} - n) + O(P^2, H_{n, \text{new}}) + O(H_{n, \text{rat}}^2)]. \quad (36)$$

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}_{raw} (see Section 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; therefore, $Q_{\text{irr}} = 1$. 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 the n th recursion $\max(H_{n, \text{new}}) = Q_{\text{raw}} - n$, $\max(H_{n, \text{rat}}) = Q_{\text{rat}} - n$, and $H_{n, \text{irr}} = 1$, from (36) 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 to the one computed for the single-branch worst-case scenario. However, since some calls of $\text{branch}(\circ, \diamond)$ will be common among the different branches, the execution time of line 6 will be lower than 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 have been generated 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 the 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), the use of a symmetry breaking control law [19] is advisable.



Fig. 12. Left: Khepera III robots used in the 2-D scenario; right: Mikrokopter quadrotors used in the 3-D 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 toward triangles that 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 w.r.t. the 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 multirobot software platform MIP². In particular, simulations are used to study its noise propagation characteristics, whereas experiments allow us to show the performance of the proposed method on real data. For Scenario A, we have employed a team of four Khepera III wheeled robots [see Fig. 12(a)], while Scenarios B and C were realized using a flock of eight Mikrokopter quadrotors [see Fig. 12(b)].

A. Experimental Results in 2-D

We implemented and tested the proposed framework for the 2-D localization problem on a team of four Khepera III robots. The robots are equipped with a 600 MHz processor and run a minimal GNU/Linux distribution. They communicate through a WiFi link with a base station desktop computer that 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

²See <http://www.dis.uniroma1.it/labrob/software/MIP/>

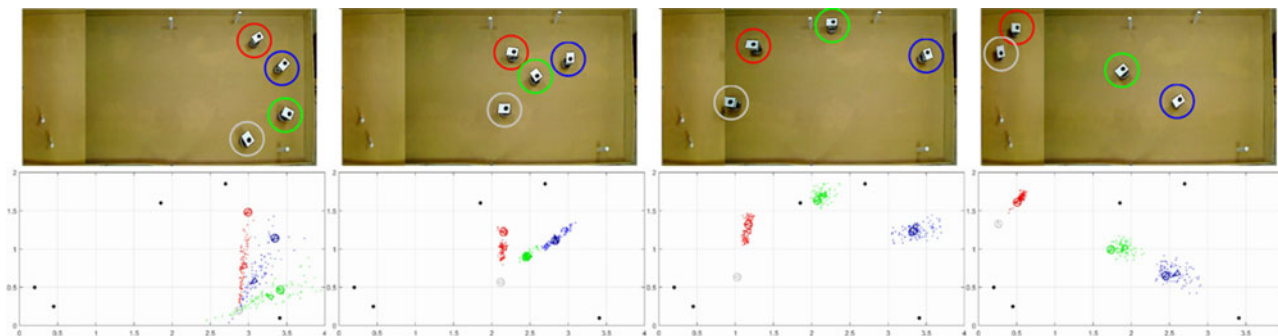


Fig. 13. Four snapshots of a 2-D experiment and the corresponding estimates computed by A_1 (in gray).

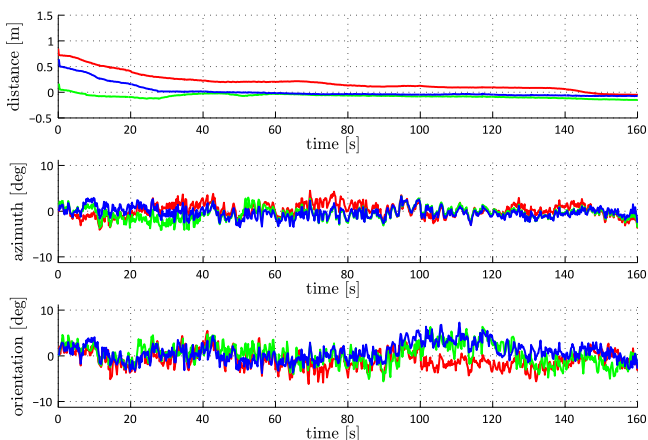


Fig. 14. Scenario A: error plots on distance, azimuth, and orientation estimates for A_2 , A_3 , A_4 computed by A_1 , respectively.

a 240° field of view, thus leaving a 120° 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 in order to emulate the measurements from a bearing-only 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 in a $4\text{ m} \times 2\text{ m}$ arena with a pseudo-random navigation strategy incorporating obstacle avoidance. The ground truth is provided with 1 cm accuracy by a feature extraction and tracking algorithm on the image streams of three USB cameras looking down on the arena.

Fig. 13 shows 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. As time progresses, 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 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.

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. From the start of the experiment, the error on the estimated distance of A_3 (green) is close to zero, and around the second 30 s, most of the error on the distance of A_4 (blue) is reduced to zero. However, due to the unfavorable motion, the error on the estimated distance of A_2 (red) converges to zero only after 120–140 s. A video clip of the experiment is shown in the first part of the attached multimedia material.

B. Simulation Results in 2-D

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 2-D Khepera models in Player/Stage which provides simulated noise-free odometry. Each simulated robot is also equipped with a laser range finder and the 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 additive zero-mean Gaussian noise either on the measurements or on the odometry data. In particular, each simulation has been run with an additive noise on the measurements with covariance 5° , 10° , and 20° . For the odometry, we have tested the system with the following noises on the integrated linear and angular velocities, respectively: 0.02 m/s and $5^\circ/\text{s}$, 0.05 m/s and $10^\circ/\text{s}$, 0.1 m/s and $20^\circ/\text{s}$. Overall, we have collected data from 56 runs of the mutual localization system.

The results of this simulation campaign are presented in aggregated form in Figs. 15 and 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 orientation, distance, and azimuth, respectively. Their error probabilities are defined as the probabilities that they are equal

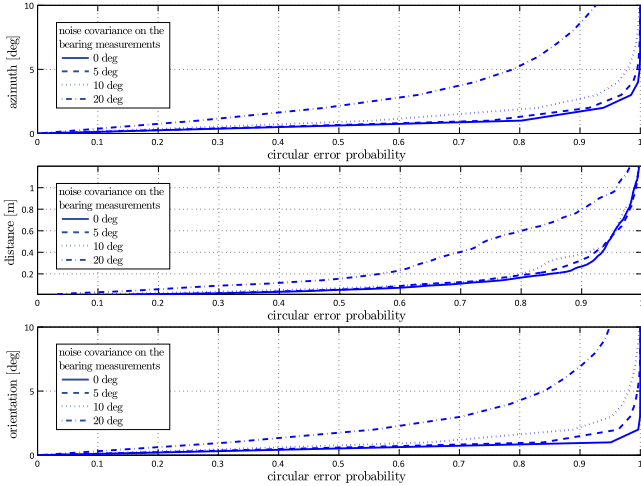


Fig. 15. Scenario A: error probabilities (lower is better) for the estimated robots A_2, \dots, A_5 of an aggregate of eight simulations for different values of covariance of the noise on the bearing measurements: solid 0° , dashed 5° , dotted 10° , dash-dotted 20° .

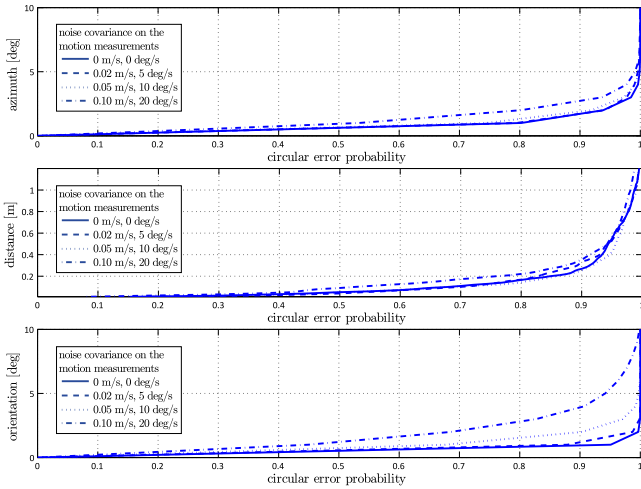


Fig. 16. Scenario A: error probabilities (lower is better) for the estimated robots A_2, \dots, A_5 of an aggregate of eight simulations for different values of covariance of the noise on the odometry (linear and angular velocities): solid 0 m/s and 0° /s, dashed 0.02 m/s and 5° /s, dotted 0.05 m/s and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

or less than a parameter

$$p(e_\beta < b) p(e_d < d) p(e_\alpha < a). \quad (37)$$

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

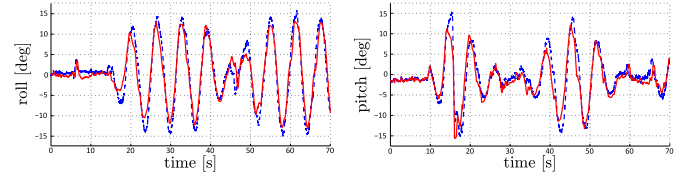


Fig. 17. True values (dashed blue) and estimates (solid red) of roll and pitch during a typical experiment.

C. Experimental Results in 3-D

The proposed localization system has been experimentally validated in Scenarios B and C using quadrotors as mobile robots and an external mocap made up by 16 infrared cameras as ground truth, whose precision is about 1 mm for translations and 1° for rotations.

Each robot A_i carries an ATmega microcontroller (μ C) board that 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 μ C board provides measurements of the linear acceleration ${}^{B_i}\bar{a}_i$ and angular velocity ${}^{B_i}\bar{\omega}_i$, both expressed in body frame. To give a characterization of the noise affecting the measurements provided by the IMU, we performed a statistical analysis over the values of ${}^{B_i}\bar{a}_i$, ${}^{B_i}\bar{\omega}_i$ registered during different experiments with the quadrotor in hovering. The analysis shows a precision of $[0.019, 0.019, 0.049]$ m/s², $[0.29, 0.29, 0.29]^\circ$ /s and variances $[0.1, 0.1, 0.6]$ m/s² and $[0.64, 0.64, 1.12]^\circ$ /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 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), showing 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 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 to feed the bank of particle filters. On the contrary, in Scenario B, the

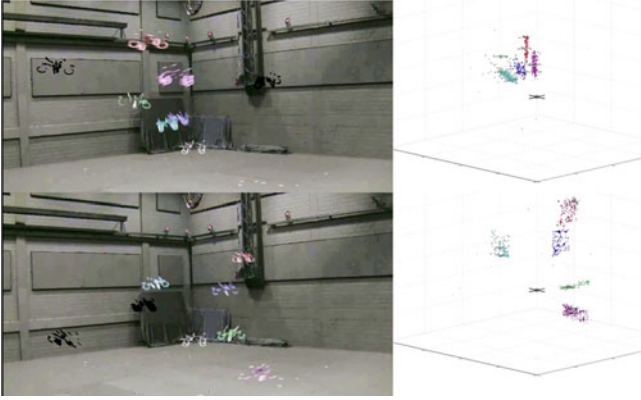


Fig. 18. Two snapshots of a 3-D experiment and the corresponding estimates computed by A_1 in Scenario B.

particle filters are fed by velocity measurements analytically emulated by adding the 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 (7). To simulate realistic measurements, a zero-mean 5° standard deviation Gaussian noise is added to the azimuth and zenith-distance angles. 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 predefined 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 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

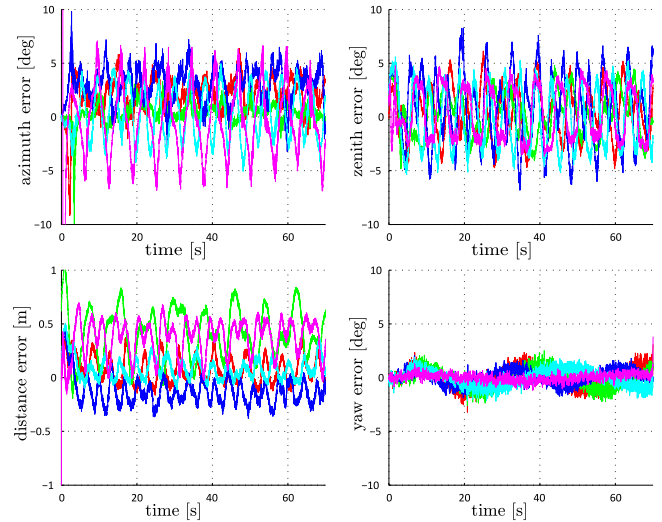


Fig. 19. Scenario C: error plots on azimuth, zenith-distance, distance, and orientation estimates for A_2, \dots, A_6 computed by A_1 .

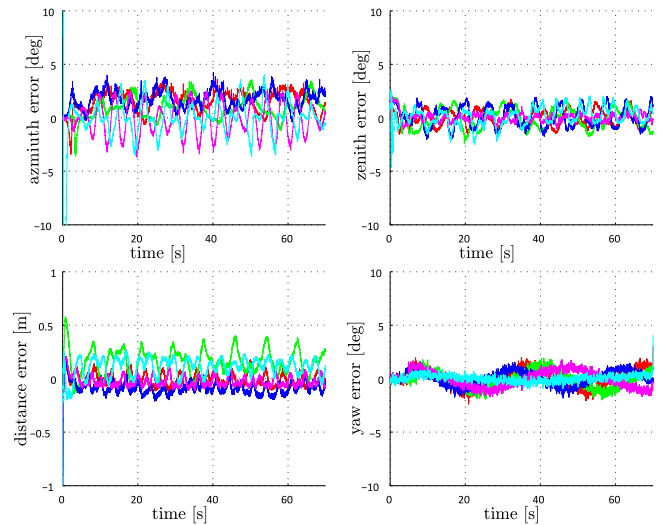


Fig. 20. Scenario B: error plots on azimuth, zenith-distance, distance, and orientation estimates for A_2, \dots, A_6 computed by A_1 .

improvement w.r.t. [21] is for the estimated relative angles. In fact, the error peaks for azimuth and zenith-distance drop from 10° in [21] to 5° , while the orientation error decreases from 12° to 3° . 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 in Scenario B are presented in Fig. 20, and show a similar behavior w.r.t. Scenario C. 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, but also the estimates of the angular quantities benefit from the employment of the velocity measurements. The performance is improved because the update step of the PFs with the velocity needs only one integration to obtain the position, whereas

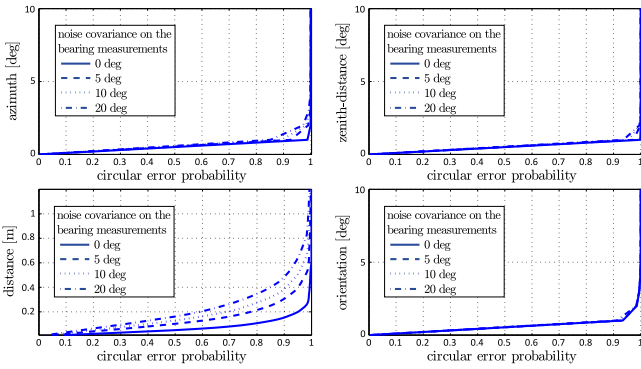


Fig. 21. Scenario B: error probabilities (lower is better) for the estimated robots A_2, \dots, A_6 of an aggregate of eight simulations for different values of covariance of noise on the bearing measurements: solid 0° , dashed 5° , dotted 10° , dash-dotted 20° .

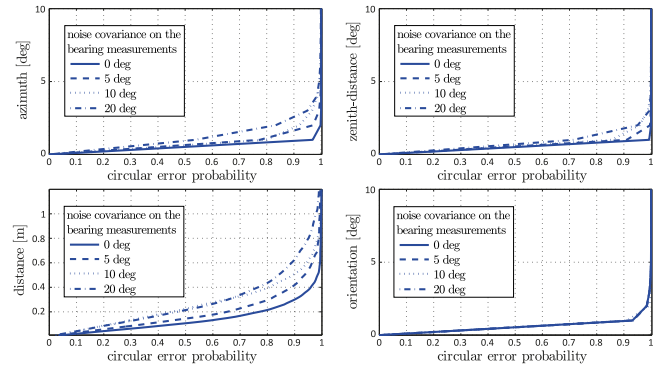


Fig. 23. Scenario C: error probabilities (lower is better) for the estimated robots A_2, \dots, A_6 of an aggregate of eight simulations for different values of covariance of noise on the bearing measurements: solid 0° , dashed 5° , dotted 10° , dash-dotted 20° .

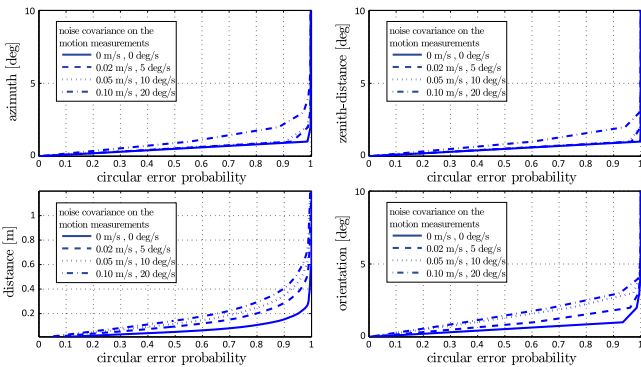


Fig. 22. Scenario B: error probabilities (lower is better) for the estimated robots A_2, \dots, A_6 of an aggregate of eight simulations for different values of covariance of noise on the motion measurements: solid 0 m/s and $0^\circ/\text{s}$, dashed 0.02 m/s and $5^\circ/\text{s}$, dotted 0.05 m/s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

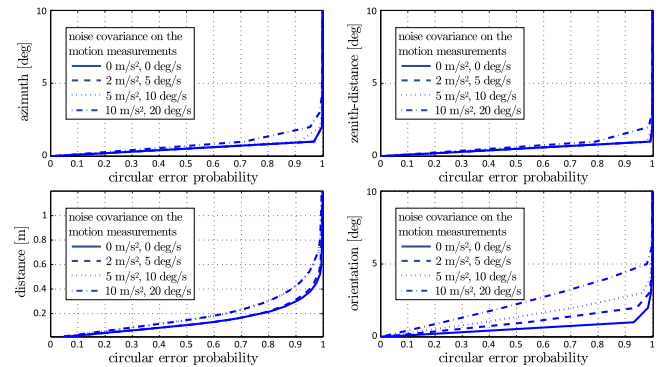


Fig. 24. Scenario C: error probabilities (lower is better) for the estimated robots A_2, \dots, A_6 of an aggregate of eight simulations for different values of covariance of noise on the motion measurements: solid 0 m/s^2 and $0^\circ/\text{s}$, dashed 2 m/s^2 and $5^\circ/\text{s}$, dotted 5 m/s^2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

the update step of the PFs with the acceleration, involves a double integration of the noise acting on the measurements. A video clip of the experiment is presented in the last part of the multimedia material provided.

D. Simulation Results in 3-D

Similar to the 2-D 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 we have run several times the mutual localization system with zero-mean and variable covariance additive noise either on the bearing or on the motion measurements.

Figs. 21–24 present the aggregated results of those simulation campaigns for both Scenario 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° , 10° , and 20°) for testing the robustness of the system w.r.t. noise on the bearing measurements (see Figs. 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 $5^\circ/\text{s}$, 0.05 m/s and $10^\circ/\text{s}$, 0.1 m/s and $20^\circ/\text{s}$) and on linear acceleration and angular velocities in Scenario C (2 m/s^2 and $5^\circ/\text{s}$, 5 m/s^2 and $10^\circ/\text{s}$, 10 m/s^2 and $20^\circ/\text{s}$). The results are shown in Figs. 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. Consistently with the 2-D case, the estimated distance is more sensitive to noise on the bearing measurements, whereas the angular quantities are more sensitive to the noise on the motion measurements.

IX. CONCLUSION

In this paper, we have introduced the problem of mutual localization with anonymous bearing-only measurements in multi-robot systems in three different operative scenarios, taking into consideration different platforms (2-D and 3-D) 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 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 nontime-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 and simulations 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.

REFERENCES

- [1] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *Int. J. Robot. Res.*, 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," *Int. J. Robot. Res.*, vol. 25, nos. 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," *Int. J. Robot. Res.*, 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. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, Dec. 2008.
- [6] M. Schwager, B. J. Julian, and D. Rus, "Optimal coverage for multiple hovering robots with downward facing cameras," in *Proc. IEEE Int. Conf. Robot. Autom.*, 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 *Proc. 47th IEEE Conf. Decis. Control*, Cancun, Mexico, Dec. 2008, pp. 3947–3952.
- [8] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Diego, CA, USA, May 1994, pp. 1250–1257.
- [9] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Auton. Robots*, vol. 8, no. 3, pp. 325–344, 2000.
- [10] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 781–795, Oct. 2002.
- [11] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 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 *Proc. 41th IEEE Conf. Decis. Control*, Las Vegas, NV, USA, Dec. 2002, pp. 2978–2983.
- [13] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 666–681, Aug. 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 *Proc. IEEE Int. Conf. Robot. Autom.*, 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. Robot.*, vol. 26, no. 2, pp. 225–243, Apr. 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, Inria Grenoble—Rhône-Alpes, Montbonnot-Saint-Martin, France, Tech. Rep. inria-00569083, Feb. 2011.
- [17] A. Franchi, G. Oriolo, and P. Stegagno, "Probabilistic mutual localization in multi-agent systems from anonymous position measures," in *Proc. 49th IEEE Conf. Decis. Control*, Atlanta, GA, USA, Dec. 2010, pp. 6534–6540.
- [18] A. Franchi, G. Oriolo, and P. Stegagno, "Mutual localization in multi-robot systems using anonymous relative measurements," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1302–1322, 2013.
- [19] A. Franchi, G. Oriolo, and P. Stegagno, "On the solvability of the mutual localization problem with anonymous position measures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 3193–3199.
- [20] P. Stegagno, M. Cagnetti, A. Franchi, and G. Oriolo, "Mutual localization using anonymous bearing-only measures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 469–474.
- [21] M. Cagnetti, P. Stegagno, A. Franchi, G. Oriolo, and H. H. Bühlhoff, "3-D mutual localization with anonymous bearing measurements," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, USA, May 2012, pp. 791–798.
- [22] M. Franceschelli and A. Gasparri, "On agreement problems with gossip algorithms in absence of common reference frames," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 4481–4486.
- [23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
- [24] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.
- [25] J. Aspnes *et al.*, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.
- [26] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I - the first 30 years and fundamentals," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.
- [27] P. Stegagno, M. Basile, H. H. Bühlhoff, and A. Franchi, "A semi-autonomous UAV platform for indoor remote operation with visual and haptic feedback," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, May 2014, pp. 3862–3869.
- [28] V. Grabe, H. H. Bühlhoff, and P. Robuffo Giordano, "On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, USA, May 2012, pp. 491–497.
- [29] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.
- [30] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 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. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.
- [32] P. Martin and E. Salaün, "The true role of accelerometer feedback in quadrotor control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 1623–1629.
- [33] A. Cristofaro and A. Martinelli, "3D cooperative localization and mapping: Observability analysis," in *Proc. Amer. Control Conf.*, San Francisco, CA, USA, Jun. 2011, pp. 1630–1635.



Paolo Stegagno (S'09–M'14) received the Ph.D. degree in System Engineering from Sapienza University of Rome, Italy, in 2012.

In 2010 he was Visiting Scholar at the University of Minnesota, Minneapolis, MN, USA. From 2012 to 2013 he was a Post-Doc at Sapienza University of Rome. Since July 2013, he has been a Research Scientist in the Autonomous Robotics and Human-Machine Systems group at the Max Planck Institute for Biological Cybernetics (MPIKYB), Tübingen, Germany. Since 2014, he has also been the Project

Leader of the group. His main research interests include mobile robotics, multi-robot systems, and unmanned aerial vehicles, with a focus on sensing, estimation and localization.

Since 2014, has been an Associate Editor for the IEEE/RSJ International Conference on Intelligent Robots and Systems.



Marco Cognetti (S'14) received the Ph.D. degree in System Engineering from Sapienza University of Rome, Italy, in 2016.

In 2011, he was Visiting Scholar at the Human-Robot Interaction group at the Max Planck Institute for Biological Cybernetics (MPI-KYB), Tübingen, Germany. In 2015, he was Visiting Scholar at the Personal Robotics Lab of The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include multi-robot control and localization and whole-body motion planning and

control for humanoid robots.



Giuseppe Oriolo (S'89–M'92–SM'02) received the Ph.D. degree in control engineering in 1992 from Sapienza University of Rome, Italy.

He is currently with the Department of Computer, Control and Management Engineering (DIAG) of the same university, where he teaches courses on automatic control and robotics. He is also the coordinator of the DIAG Robotics Laboratory. His research interests are in the area of planning and control of robotic systems, in which he has published two books and more than 170 papers.

Dr. Oriolo has been an Associate Editor of the *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION* from 2001 to 2005, and Editor of the *IEEE TRANSACTIONS ON ROBOTICS* from 2009 to 2013.



Heinrich H. Bülthoff (M'95) received the Ph.D. degree in Natural Sciences from Eberhard Karls University, Tübingen, Germany, in 1980.

From 1980 to 1985, he was a Research Scientist with the Max Planck Institute for Biological Cybernetics (MPIKYB), Tübingen, Germany, and from 1985 to 1988, he was a Visiting Scientist at the Massachusetts Institute of Technology, Cambridge, MA, USA. Between 1988 and 1993, he was an Assistant Professor, an Associate Professor, and finally a Full Professor of Cognitive Science at Brown University,

Providence, RI, USA. In 1993, he became the Director of the Department of Human Perception, Cognition, and Action at the MPI-KYB, and a Scientific Member of the Max Planck Society. Since 1996, he has been an Honorary Professor with the Eberhard Karls University, and has been an Adjunct Professor with Korea University, Seoul, South Korea. His current research interests include object recognition and categorization, perception, and action in virtual environments, human-robot interaction.



Antonio Franchi (S'07–M'11–SM'16) received the Ph.D. degree in System Engineering from Sapienza University of Rome, Italy, in 2010.

In 2009, he was a Visiting Scholar at the University of California at Santa Barbara, Santa Barbara, CA, USA. From 2010 to 2012, he was a Research Scientist in the Human-Robot Interaction group at the Max Planck Institute for Biological Cybernetics (MPI-KYB), Tübingen, Germany. From 2012 to 2013, he was a Senior Research Scientist and the Project Leader of the Autonomous Robotics and

Human-Machine Systems group at the MPIKYB. Since 2014, he has been a Permanent CNRS Researcher in the RIS group at LAAS-CNRS, Toulouse, France. His main research interests include autonomous systems and robotics, with a special regard to control, planning, estimation, human-machine systems, haptics, and hardware/software architectures. His main areas of expertise are multiple-robot systems and aerial robotics. He published more than 70 papers in peer-reviewed international journals and conferences.

In 2010, he was awarded the IEEE RAS ICYA Best Paper Award. He is an Associate Editor of the *IEEE Robotics & Automation Magazine*. He is the co-founder and co-chair of the IEEE RAS Technical Committee on Multiple Robot Systems.