

Relative Localization and Identification in a Heterogeneous Multi-Robot System

Paolo Stegagno, Marco Cognetti, Lorenzo Rosa, Pietro Peliti, Giuseppe Oriolo

Abstract—We develop a localization method for a single-UAV/multi-UGV heterogeneous system of robots. Considering the natural supervisory role of the UAV and the challenging (but realistic) assumption that the UAV-to-UGV measurements do not include the identities of the UGVs, we have adopted the PHD filter as a multi-target tracking technique. However, the standard version of this filter does not take into account odometric information coming from the targets, nor does it solve the problem of estimating their identities. Hence, we design ID-PHD, a modification of the PHD filter that is able to reconstruct the identities of the targets by incorporating odometric data. The proposed localization method has been successfully validated through experiments. Some preliminary results of a localization-based control scheme for the multi-robot system are also presented.

I. INTRODUCTION

Heterogeneous systems are a particular case of multi-robot systems whose team members are not equal each other. This strategy is usually employed to specialize each agent for a specific role in the team, so that each robot can perform better its specific task. For this reason, heterogeneous systems are raising a growing interest among researchers.

In particular, mixing Unmanned Air (UAV) and Ground (UGV) Vehicles can be particularly useful, because it allows to mix the localization capabilities and the on-site point of view of the UGVs with the more complete vision of the environment offered by the UAVs.

In this context, we want to design an heterogeneous system in which a multitude of UGVs lies on a 2D plane (that can easily be identified as the floor of a building) and an UAV performs the role of supervisor flying over them, providing higher level information to the team.

In this scheme, the knowledge of each others' relative configuration is mandatory for the members of the system in order to exchange data, plan the action and cooperate to perform higher level tasks. It can be achieved either by external sensors, such as camera systems or Ground Positioning System (GPS) like in [1], or on-board exteroceptive sensors such as cameras and laser range finders. However, assuming a general non structured environment leads to the assumption that no external systems are available. This means that the team can rely only on the measurements gathered from the sensors of its members in order to estimate their relative configurations. The relative localization of an heterogeneous UAVs/UGVs system is faced in [2] where the authors used a Kalman filter for the estimation in an outdoor experimental

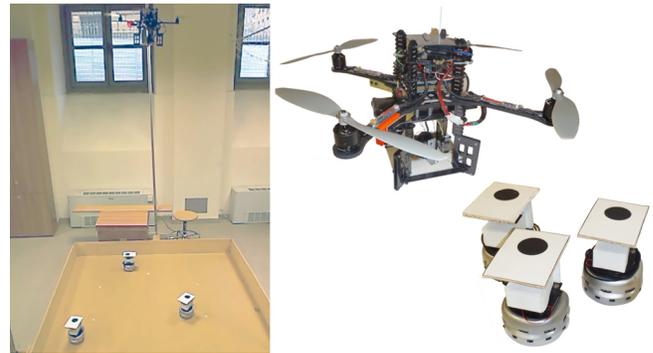


Fig. 1: The heterogeneous multi-robot system considered in this paper consists of a single UAV (Pelican by AscTec) and multiple UGVs (Khepera III by K-Team).

setup. In [3], a single user controls multiple robots, using a particle filter for the estimation process with good results. Similar techniques have been applied also using climbing robots instead of UAVs [4].

A reasonable assumption which we will take is that the UAV is able to gather measurements over the relative configuration of the UGVs. While in normal operation it is usually possible to discriminate among the measured robots (e.g.: applying tagging), there are several adverse environmental conditions (low illumination, smoky environment, etc..) under which the identification of the measured robots may not be feasible or reliable. For this reason, we will assume that the UAV is able to identify the UGVs in its field of view, without detecting their identities. In previous works on multi-robot system, this hypothesis is usually called unknown data association [5], or anonymous measurements [6], [7], [8]. However, our situation is different from the one described in those works since only one robot is in charge of the localization, being able to gather relative measurements.

In this work we tackle the unknown data association issue by employing a technique called Probability Hypothesis Density (PHD) filter, which was first developed to solve the multi-target tracking problem by Mahler in [9]. It intrinsically solves the data association by estimating the PHD of the targets in a given region of the space. However, since the PHD filter is designed to allow the identification of generic targets, it does not take into account two facts.

The first is the possibility that the targets are cooperative and send their odometries to the agent performing the estimation. In fact, the PHD filter is designed to work with generic targets whose time update is performed using a motion model with unknown input. On the contrary, in robotics each UGV can usually measure its own displacement through the use of

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy. E-mail: {stegagno,cognetti,rosa,peliti,oriolo}@dis.uniroma1.it

encoders on the wheels, so that the motion, hence the input for the time update, is different and known for each UGV.

The second fact is the necessity to reconstruct not only the state of each UGV in the team, but also its identity. Being the localization provided from the UAV to the UGVs as a set of relative configurations, each robot needs to recognize itself in the set to compute its own control. For the same reasons, our work differs from SLAM with moving objects approaches like in [10]. In view of these additional characteristics, we will develop a modification of the PHD filter which is able to incorporate the odometries of the UGVs and use them to reconstruct their identities directly in the filtering step.

Once the localization issue has been tackled, it is possible to apply a control law performing higher level tasks. The heterogeneity of the system allows the design of several tasks relying on the robots coordination, such as exploration, patrolling or deployment. Furthermore the large number of robots gives to the system a high degree of redundancy, hence the ability to execute several tasks at the same time.

The rest of the paper is organized as follows. After presenting an overview on the system in Section II, we formally state the main assumptions of the localization problem in Section III. The relative localization system is presented in Section IV, and its simulation and experimental results are addressed in Section V and VI. An outline and preliminary results of suitable control law for the system is discussed in Section VII. Section VIII concludes the paper.

II. SYSTEM OVERVIEW

During the design of the multi-robot system, we have recognized the necessity to solve two main problems. The first is the relative localization problem. Each robot, in order to cooperate with the others, needs the knowledge — or an estimate — of the relative configuration of the other members of the team w.r.t. itself. The second is the control problem, in which that knowledge can be used to actually perform collective tasks.

As stated before, the specialization of the agents leads to a task allocation that depends on the characteristics of the various robots. Given its dominant point of view over the ground, the natural role of the UAV is to perform relative localization of the UGVs acting as supervisor; obviously, it will also need to move so as to maintain optimal visibility of the ground robots. In view of its role, the UAV is typically endowed with a rich sensor equipment and high computational power. In our case, an Inertial Measurement Unit (IMU) measures the attitude and velocities of the UAV; whereas exteroceptive sensing is obtained by an on-board camera pointing downwards. Finally, a height sensor provides an information which will be important to reconstruct the scale of relative configurations. For simplicity, in the following we will consider all sensors to be centered in the UAV reference frame. Nonzero roto-translations among the various frames, invariably present in practice, can be easily taken into account by kinematic transformations.

The task of the UGVs is operative, in the sense that they must carry out the required activities on the ground. Given

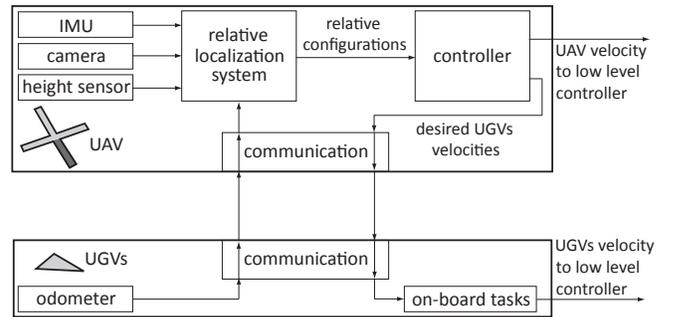


Fig. 2: The heterogeneous multi-robot system with the allocation of tasks and the information flow among the robots.

their number, and considering that the UAV is in charge of relative localization, the sensing equipment of the UGVs can be quite basic. This is important because it will result in a relatively low cost. In particular, we will assume that each UGV is only equipped with an *odometer* that measures its incremental motion. Other exteroceptive sensors may be present to perform specific on-board tasks: e.g., range finders for obstacle avoidance. However, we shall not consider the availability of these sensors for the localization method.

Figure 2 illustrates the resulting control architecture with the flow of information within the multi-robot system. The relative configurations (with identities) reconstructed by the localization system are sent to a centralized controller, which computes motion commands for the UGVs (depending on the ground task) and the UAV itself (to maintain the UGVs in its field of view). The UGV reference velocities are then sent to the UGVs, which may modify them in real time to implement reactive behaviors. This control architecture may be modified in order to introduce more decentralization. For example, the localization data may be directly made available to the robots, which can then compute their own control action on an individual basis.

III. PROBLEM SETTING

We consider a system consisting of a single UAV \mathcal{R}_0 and n UGVs $\{\mathcal{R}_1, \dots, \mathcal{R}_n\}$, with n unknown and variable over time. The generic robot \mathcal{R}_i , $i = 0, \dots, n$ is modeled as a rigid body moving in the 3D space and is equipped with a reference frame $\mathcal{F}_i = \{\mathcal{O}_i, X_i, Y_i, Z_i\}$ attached to a representative point of the robot. Hence, the configuration of \mathcal{R}_i is uniquely determined in a world frame $\mathcal{W} = \{\mathcal{O}, X, Y, Z\}$ by the position ${}^{\mathcal{W}}\mathbf{p}_i \in \mathbb{R}^3$ of \mathcal{O}_i and the rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{F}_i}$ between \mathcal{F}_i and \mathcal{W} . The latter can be computed by the roll ${}^{\mathcal{W}}\phi_i$, pitch ${}^{\mathcal{W}}\theta_i$ and yaw ${}^{\mathcal{W}}\psi_i$ angles of \mathcal{F}_i in \mathcal{W} .

In general, the left superscript and the right subscript will indicate the frame of reference in which a variable is expressed and the entity to which that variable refers, respectively. The UAV frame conforms to the North-East-Down (NED) convention (Z -axis pointing down), as customary in the aerospace field, while the UGVs adopt the North-West-Up (NWU) frame (Z -axis pointing up), as shown in Fig. 3.

Denote with $\mathbf{R}_X(\alpha)$, $\mathbf{R}_Y(\alpha)$, $\mathbf{R}_Z(\alpha)$ the elementary rotation matrix around the X , Y and Z axes of an angle α . We define $\mathcal{G}_0 = \{\mathcal{O}_{\mathcal{G}_0}, X_{\mathcal{G}_0}, Y_{\mathcal{G}_0}, Z_{\mathcal{G}_0}\}$ as the frame having

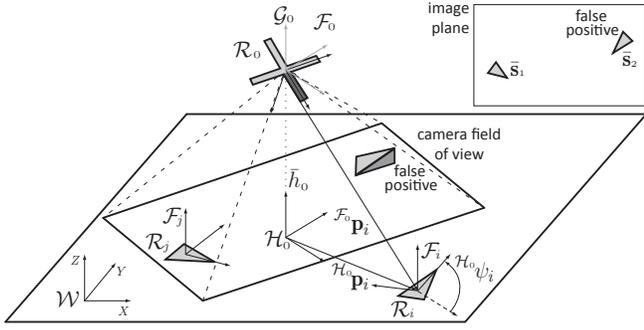


Fig. 3: Relative localization with anonymous measurements. Triangles are UGVs with their attached frames. The image plane is also shown with the set of detected features ${}^{\mathcal{F}_0}\bar{S}_0$.

the same origin of \mathcal{F}_0 and such that ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{G}_0} = \mathbf{R}_Z({}^{\mathcal{W}}\psi_0)$, as in Fig. 3 (the axis $Z_{\mathcal{G}_0}$ is parallel to Z). Being ${}^{\mathcal{G}_0}\mathbf{R}_{\mathcal{F}_0} = \mathbf{R}_X(\pi)\mathbf{R}_Y({}^{\mathcal{W}}\theta_0)\mathbf{R}_X({}^{\mathcal{W}}\phi_0)$, we have

$${}^{\mathcal{G}_0}\mathbf{R}_{\mathcal{F}_0} = \begin{pmatrix} c^{\mathcal{W}}\theta_0 & s^{\mathcal{W}}\phi_0 s^{\mathcal{W}}\theta_0 & c^{\mathcal{W}}\phi_0 s^{\mathcal{W}}\theta_0 \\ 0 & -c^{\mathcal{W}}\phi_0 & s^{\mathcal{W}}\phi_0 \\ s^{\mathcal{W}}\theta_0 & -s^{\mathcal{W}}\phi_0 c^{\mathcal{W}}\theta_0 & -c^{\mathcal{W}}\phi_0 c^{\mathcal{W}}\theta_0 \end{pmatrix} \quad (1)$$

where $s\alpha = \sin \alpha$, $c\alpha = \cos \alpha$ and the rotation matrix

$$\mathbf{R}_X(\pi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

represents the rotation from NED to NWU reference frames.

Furthermore, we define $\mathcal{H}_0 = \{O_{\mathcal{H}_0}, X_{\mathcal{H}_0}, Y_{\mathcal{H}_0}, Z_{\mathcal{G}_0}\}$ the frame with the origin in the projection of \mathcal{O}_0 on the XY plane, and such that ${}^{\mathcal{H}_0}\mathbf{R}_{\mathcal{G}_0} = \mathbf{I}_3$, where \mathbf{I}_m is the $m \times m$ identity matrix. Hence, as depicted in Fig. 3, \mathcal{G}_0 and \mathcal{H}_0 are the same frame moved along $Z_{\mathcal{G}_0}$, and ${}^{\mathcal{H}_0}\mathbf{R}_{\mathcal{F}_0} = {}^{\mathcal{G}_0}\mathbf{R}_{\mathcal{F}_0}$.

The IMU provides at each time t the measurements ${}^{\mathcal{F}_0}\hat{\mathbf{a}}_0$, ${}^{\mathcal{F}_0}\hat{\boldsymbol{\omega}}_0$ of the linear acceleration ${}^{\mathcal{F}_0}\mathbf{a}_0$ and angular velocity ${}^{\mathcal{F}_0}\boldsymbol{\omega}_0$. Moreover, it provides, at each time instant, an estimate $({}^{\mathcal{W}}\hat{\phi}_0, {}^{\mathcal{W}}\hat{\theta}_0, {}^{\mathcal{W}}\hat{\psi}_0)^T$ of the attitude of \mathcal{R}_0 . We will use the right superscript to express a time dependency (e.g.: \mathbf{a}^τ is quantity \mathbf{a} at time τ). However, we will omit the time dependency t , so that each variable without a right superscript is implicitly referred to time t (e.g. ${}^{\mathcal{F}_0}\hat{\mathbf{a}}_0 = {}^{\mathcal{F}_0}\hat{\mathbf{a}}_0^t$).

The images from the camera are scanned by a feature extraction algorithm which is able to recognize the UGVs but is not able to discriminate among the different UGVs and recognize their identities. For this reason, its output is a set ${}^{\mathcal{F}_0}\bar{S}_0 = \{\bar{s}_1, \dots, \bar{s}_m\}$ of unlabeled points on the image plane corresponding to the \mathcal{R}_i 's, $i > 0$ that are in the field of view of the camera. In addition to a gaussian noise on the measured points, we also consider false positives (objects in the image that look like robots) and false negatives (robots in the image not recognized as such).

Assume that all the objects in the scene are stationary except for the feature set ${}^{\mathcal{F}_0}\bar{S}_0$. By removing this set from the images, the Optical Flow [11] is computed on the remaining set of M features ${}^{\mathcal{F}_0}\hat{\Sigma}_0$, obtaining an estimate of the velocity ${}^{\mathcal{F}_0}\hat{\Sigma}_0$ of those features on the image plane, which is influenced only by the camera ego-motion.

The height sensor provides at each time instant a measure \bar{h}_0 of the distance $\|{}^{\mathcal{F}_0}\mathbf{p}_{\mathcal{H}_0}\|$ between the UAV and its

projection on the XY plane.

Although the generic UGV \mathcal{R}_i , $i = 1, \dots, n$ is described as a 3-dimensional robot, it lies on the XY plane, so that the third coordinate (along the axis Z) of ${}^{\mathcal{W}}\mathbf{p}_i$ is always zero and the $X_i Y_i$ plane is coincident with the XY plane for all $i > 0$. Hence, the rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{F}_i}$, $i > 0$ is an elementary rotation around the axis Z .

The odometer of \mathcal{R}_i , $i > 0$ provides, at each time t , a measure ${}^{\mathcal{F}_i}\hat{\mathbf{u}}_i \in \mathbb{R}^2 \times S^1$ of ${}^{\mathcal{F}_i}\mathbf{u}_i$, the robot displacement between two consecutive sampling instants $t-1$ and t on the plane XY .

The robots of the team communicate through a wireless communication channel. In particular, each \mathcal{R}_i , $i > 0$ sends as message to \mathcal{R}_0 containing (1) the robot signature (the index i), (2) the total displacement ${}^{\mathcal{F}_i}\hat{\mathbf{q}}_i \in \mathbb{R}^2 \times S^1$ of the robot between 0 and t on the plane XY , computed by composing the elementary displacements ${}^{\mathcal{F}_i}\hat{\mathbf{u}}_1^1, \dots, {}^{\mathcal{F}_i}\hat{\mathbf{u}}_i^i$. Let be N_0 the set of robots whose odometry message is received by \mathcal{R}_0 at time t , and let be $N_0^{1:t} = N_0^1 \cup \dots \cup N_0^t$.

The UAV \mathcal{R}_0 runs the relative localization system using its own sensor perceptions and the odometries gathered and communicated by \mathcal{R}_i , $i > 0$. It conducts the estimation from a robo-centric point of view, computing at each time a belief over the position ${}^{\mathcal{F}_0}\mathbf{p}_i$ and the rotation matrix ${}^{\mathcal{F}_0}\mathbf{R}_{\mathcal{F}_i}$ of the generic \mathcal{R}_i . Hence, it solves the following problem:

Problem 1: (Relative Localization) For $t = 1, 2, \dots$ and $i \in N_0^{1:t}$, compute the belief $\text{bel}({}^{\mathcal{F}_0}\mathbf{p}_i, {}^{\mathcal{F}_0}\mathbf{R}_{\mathcal{F}_i})$ given ${}^{\mathcal{F}_0}\hat{\Sigma}_0^\tau$, ${}^{\mathcal{W}}\hat{\phi}_0^\tau$, ${}^{\mathcal{W}}\hat{\theta}_0^\tau$, ${}^{\mathcal{W}}\hat{\psi}_0^\tau$, ${}^{\mathcal{F}_0}\bar{S}_0^\tau$, \bar{h}_0^τ , $\{{}^{\mathcal{F}_i}\hat{\mathbf{q}}_i^\tau\}_{i \in N_0^{1:t}}$, $\tau = 1, \dots, t$.

IV. RELATIVE LOCALIZATION SYSTEM

The scheme of our estimation algorithm is shown in Fig. 4. The first step is to fuse the depth information \bar{h}_0 given by height sensor with ${}^{\mathcal{F}_0}\bar{S}_0$ and the camera calibration matrix \mathbf{K} through (assume ${}^{\mathcal{W}}\hat{\phi}_0, {}^{\mathcal{W}}\hat{\theta}_0 \neq 0$)

$${}^{\mathcal{F}_0}\bar{P}_0 = \left\{ \frac{\bar{h}_0}{c^{\mathcal{W}}\hat{\phi}_0 c^{\mathcal{W}}\hat{\theta}_0} \mathbf{K}^{-1} \begin{pmatrix} \bar{s}_j \\ 1 \end{pmatrix}, \forall \bar{s}_j \in {}^{\mathcal{F}_0}\bar{S}_0 \right\} \quad (2)$$

where $\bar{s}_j \in {}^{\mathcal{F}_0}\bar{S}_0$ is the generic j -th element of the camera observation set. The result is a set of relative position measurements ${}^{\mathcal{F}_0}\bar{P}_0$.

In order to compute the velocities ${}^{\mathcal{F}_0}\mathbf{v}_0$ of the UAV, we consider the relation between the velocities of the camera and the features in the set ${}^{\mathcal{F}_0}\bar{S}_0$:

$${}^{\mathcal{F}_0}\hat{\Sigma}_0 = \mathbf{J}_s({}^{\mathcal{F}_0}\bar{S}_0, {}^{\mathcal{W}}\hat{\phi}_0, {}^{\mathcal{W}}\hat{\theta}_0, \bar{h}_0) {}^{\mathcal{F}_0}\hat{\mathbf{v}}_0 \quad (3)$$

The so-called *image Jacobian* \mathbf{J}_s [12] is a rectangular matrix of size $2M \times 6$. Equation (3) can be inverted by computing the left pseudo-inverse of \mathbf{J}_s to obtain a least-square estimate ${}^{\mathcal{F}_0}\hat{\mathbf{v}}_0$ of the UAV velocities using ${}^{\mathcal{F}_0}\hat{\Sigma}_0$.

If the UAV is in hovering (roll and pitch angles equal to 0), Problem 1 can be simplified since the planes $X_0 Y_0$ and $X_i Y_i$ become parallel. Moreover, since \mathcal{H}_0 and \mathcal{G}_0 differs only for a known translation along the axis Z , the problem reduces in the estimation of the relative quantities $({}^{\mathcal{H}_0}\mathbf{p}_i, {}^{\mathcal{H}_0}\psi_i)^T$, $i > 0$. Note that the third coordinate of ${}^{\mathcal{H}_0}\mathbf{p}_i$ is always zero and ${}^{\mathcal{H}_0}\mathbf{R}_{\mathcal{F}_i} = \mathbf{R}_Z({}^{\mathcal{H}_0}\psi_i)$, since $Z_{\mathcal{H}_0} \parallel Z_i, \forall i > 0$.

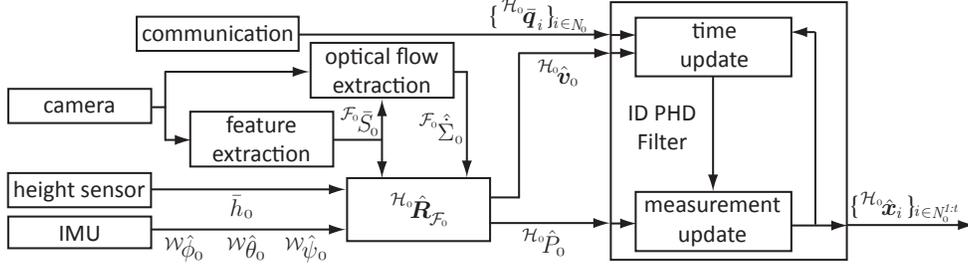


Fig. 4: The relative localization system.

When \mathcal{R}_0 is not in hovering, it is convenient to rewrite the problem as if it was, by multiplying the UAV velocity and position measurement set ${}^{\mathcal{F}_0}\hat{P}_0$ for ${}^{\mathcal{H}_0}\hat{R}_{\mathcal{F}_0}$ via (1), available from roll/pitch estimates (${}^{\mathcal{W}}\hat{\phi}_0, {}^{\mathcal{W}}\hat{\theta}_0$), and applying the translation of \bar{h}_0 along the axis Z . The results are the velocity ${}^{\mathcal{H}_0}\hat{v}_0$ of \mathcal{H}_0 on the plane XY and position measurement set ${}^{\mathcal{H}_0}\hat{P}_0$ expressed in \mathcal{H}_0 .

Then Problem 1 is simplified as

Problem 2: (Reduced Relative Localization) For $t > 0$ and $i \in N_0^{1:t}$, compute the belief $\text{bel}({}^{\mathcal{H}_0}\mathbf{p}_i, {}^{\mathcal{H}_0}\psi_i)$ given ${}^{\mathcal{H}_0}\hat{P}_0^\tau, {}^{\mathcal{H}_0}\hat{v}_0^\tau, {}^{\mathcal{W}}\hat{\phi}_0^\tau, \{\mathcal{F}_i \bar{\mathbf{q}}_i^\tau\}_{i \in N_0^{1:t}}, \tau = 1, \dots, t, i > 0$.

Note that if Problem 2 is solved, also Problem 1 is solved. In fact, the other relative quantities are easily recoverable. For the relative attitude we have $({}^{\mathcal{F}_0}\hat{\phi}_i, {}^{\mathcal{F}_0}\hat{\theta}_i)^T = (-{}^{\mathcal{W}}\hat{\phi}_0, -{}^{\mathcal{W}}\hat{\theta}_0)^T$, since the attitude angles of the UGVs are equal to zero. In addition, ${}^{\mathcal{F}_0}\mathbf{p}_i$ is retrieved via

$${}^{\mathcal{F}_0}\mathbf{p}_i = {}^{\mathcal{H}_0}\hat{R}_{\mathcal{F}_0}^T {}^{\mathcal{H}_0}\mathbf{p}_i + {}^{\mathcal{F}_0}\bar{\mathbf{p}}_{\mathcal{H}_0}, \quad (4)$$

where ${}^{\mathcal{F}_0}\bar{\mathbf{p}}_{\mathcal{H}_0} = (0, 0, \bar{h}_0)^T$ from Fig.3.

In order to solve Problem 2 we need to recover (i) the identities of the measurements in the set ${}^{\mathcal{H}_0}\hat{P}_0$ (ii) the relative positions ${}^{\mathcal{H}_0}\mathbf{p}_i$ (iii) the relative orientation ${}^{\mathcal{H}_0}\psi_i$.

By the definition of the *reduced relative configuration* $\mathbf{x}_i = ({}^{\mathcal{H}_0}p_{i_x}, {}^{\mathcal{H}_0}p_{i_y}, {}^{\mathcal{H}_0}\psi_i)^T \in \mathbb{R}^2 \times S^1$ of \mathcal{R}_i in \mathcal{H}_0 we can further simplify Problem 2 reformulating it in 2D space.

Problem 3: (2D Reduced Relative Localization) For $t > 0$ and $i \in N_0^{1:t}$, compute the belief $\text{bel}(\mathbf{x}_i)$ given ${}^{\mathcal{H}_0}\hat{P}_0^\tau, {}^{\mathcal{H}_0}\hat{v}_0^\tau, {}^{\mathcal{W}}\hat{\phi}_0^\tau, \{\mathcal{F}_i \bar{\mathbf{q}}_i^\tau\}_{i \in N_0^{1:t}}, \tau = 1, \dots, t, i > 0$.

The solution of Problem 3 is delegated to a modification of the standard Gaussian Mixture PHD filter [13].

A. Interpretation of the PHD filter

The PHD filter is a generic filter that estimates the PHD of targets in the space [9]. Assuming n targets, with n unknown and variable over time, the PHD $D(\mathbf{x}')$ of generic targets is defined as the function such that its integral over any subset $S \subseteq \mathcal{X}$ of the state space of the targets is the expected number of targets $N(S)$ in those subset. Its mathematical definition is given by the following relationship:

$$N(S) = \int_S D(\mathbf{x}') d\mathbf{x}' \quad (5)$$

Note that we can always factorize the probability hypothesis density of generic targets in a sum of elements

$$D(\mathbf{x} \cdot) = \sum_{h=1}^n d^h(\mathbf{x} \cdot) = \sum_{h=1}^n P_E^h(\cdot) p^h(\mathbf{x} \cdot) \quad (6)$$

each one of them representing a target with probability of existence

$$P(E^h | \cdot) = P_E^h(\cdot) = \int_{\mathcal{X}} d^h(\mathbf{x} | \cdot) d\mathbf{x}, \quad h = 1, \dots, n \quad (7)$$

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

$$p^h(\mathbf{x} | \cdot) = d^h(\mathbf{x} | E^h, \cdot) \quad (8)$$

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

The derivation of the filter is based on random finite set theory (RFS), and is somehow long and complicated. However, the resulting recursive filter is composed by two steps – time and measurement updates – whose interpretation [14] is straightforward.

Denote with $D_{k|k}(\mathbf{x})$ an estimate of the PHD at a given time kT computed using all measurements up to kT , and assume that the next measurements are available at the time $(k+1)T$. The time update is performed between k and $k+1$ to propagate over time the estimate $D_{k|k}(\mathbf{x})$ using the motion model of the targets. The result is an estimate $D_{k+1|k}(\mathbf{x})$ of the PHD at the step $k+1$, given all the measurements up to step k . The generic equation of the time update is

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \int [p_S(\mathbf{x}') f_{k+1|k}(\mathbf{x} | \mathbf{x}') + b_{k+1|k}(\mathbf{x} | \mathbf{x}')] D_{k|k}(\mathbf{x}') d\mathbf{x}' \quad (9)$$

where $b_{k+1|k}(\mathbf{x})$ is the probability that a new target appears in \mathbf{x} between k and $k+1$, $p_S(\mathbf{x}')$ is the probability that a target with state \mathbf{x}' at step k will survive into step $k+1$, $b_{k+1|k}(\mathbf{x} | \mathbf{x}')$ is the probability that a new target spawns in \mathbf{x} at $k+1$ from a target that is in \mathbf{x}' at k , and $f_{k+1|k}(\mathbf{x} | \mathbf{x}')$ is the single-target Markov transition density.

Using an interpretation of (6), assuming a constant $p_S(\mathbf{x}) = p_S$ and assuming no target spawning, equation (9) becomes

$$\begin{aligned} D_{k+1|k}(\mathbf{x}) &= \\ &= b_{k+1|k}(\mathbf{x}) + \int f_{k+1|k}(\mathbf{x} | \mathbf{x}') \sum_{h=1}^n p_S P_E^h p^h(\mathbf{x}') d\mathbf{x}' = \\ &= b_{k+1|k}(\mathbf{x}) + \sum_{h=1}^n p_S P_E^h \int f_{k+1|k}(\mathbf{x} | \mathbf{x}') p^h(\mathbf{x}') d\mathbf{x}' \quad (10) \end{aligned}$$

showing that in this case each component of the probability hypothesis density is governed by the same law of motion governing the time evolution of the posterior probability density function of a single target in the multi-target system.

Assume that at step $k+1$ a set of measurements Z (whose generic element is z) is gathered. It is used to compute an estimate $D_{k+1|k+1}(\mathbf{x})$ of the PHD at $k+1$ incorporating all the measurements up to $k+1$. Its generic equation is

$$D_{k+1|k+1}(\mathbf{x}) = D_{k+1|k}(\mathbf{x}) [1 - p_D(\mathbf{x}) + \sum_{z \in Z} \frac{p_D(\mathbf{x})g(z|\mathbf{x})}{\lambda c(z) + \int p_D(\mathbf{x}')g(z|\mathbf{x}')D_{k+1|k}(\mathbf{x}')d\mathbf{x}'}] \quad (11)$$

where $g(z|\mathbf{x})$ is the sensor likelihood function, $p_D(\mathbf{x})$ is the probability that an observation is collected from a target with state \mathbf{x} and $\lambda c(z)$ expresses the probability that a given measurement z is a false alarm.

Using (6), assuming no false alarms ($\lambda = 0$) and a constant probability of detection for each component $p_D(\mathbf{x}) = p_D^h$, equation (11) becomes

$$D_{k+1|k+1}(\mathbf{x}) = \sum_{h=1}^n P_E^h p^h(\mathbf{x}) [1 - p_D^h + \sum_{z \in Z} \frac{p_D^h g(z|\mathbf{x})}{\int g(z|\mathbf{x}') \sum_{h=1}^n p_D^h P_E^h p^h(\mathbf{x}') d\mathbf{x}'}] = \sum_{h=1}^n \left[(1 - p_D^h) P_E^h p^h(\mathbf{x}) + \sum_{z \in Z} w^h(z) p^h(\mathbf{x}) g(z|\mathbf{x}) \right] \quad (12)$$

with $w^h(z) = \frac{p_D^h P_E^h}{\int g(z|\mathbf{x}') \sum_{h=1}^n p_D^h P_E^h p^h(\mathbf{x}') d\mathbf{x}'}$.

Hence, the PHD filter exploits the data association by associating each measurement with each component of the PHD and giving a weight $w^h(z)$ to each component–measurement pair on a probability basis. Also the case of no measurement from a given component is taken into account by the first term, that is weighted by $(1 - p_D^h)$.

In this way the PHD filter keeps track of all possible associations without computational explosion, magnifying the most likely data associations through the weights. Moreover, the update rule for the single component $p^h(\mathbf{x})g(z|\mathbf{x})$ expresses a Bayes-like paradigm.

If all components $p^h(\mathbf{x})$, the Markov model of the targets $f_{k+1|k}(\mathbf{x}|\mathbf{x}', \mathbf{u})$ and the measurement model $g(z|\mathbf{x}')$ are (or can be approximated as) gaussian functions, as the case of unicycles moving on a plane, the resulting update step for each component is the time update of an (extended) Kalman filter. This principle is at the basis of the Gaussian Mixture implementation of the PHD filter, first proposed in [13]. In this work, we will make use of the gaussian mixture paradigm for our modification of the PHD filter. Hence, in the rest of the paper we will assume all $p^h(\mathbf{x})$ to be gaussian functions.

B. The ID-PHD filter

In our framework, we have used the PHD filter to estimate the probability hypothesis density of the robots. Hence, the

state space is $\mathcal{X} \equiv \mathbb{R}^2 \times S^1$, while the measurements Z are the position set ${}^{\mathcal{H}_0} \hat{P}_0$. Note that the elements of ${}^{\mathcal{H}_0} \hat{P}_0$ are 3D vectors, but the third component is always equal to zero, hence in the following of the paragraph we are allowed to consider and use them as if they were elements of \mathbb{R}^2 .

The main difference between the capabilities of a standard PHD filter and the features required by our problem is the necessity to reconstruct the identities of the robots. At this aim, we have implemented some significant modifications to the time update of the filter in order to introduce in (10) the odometries communicated by \mathcal{R}_i , $i > 0$, and use them to retrieve the identities.

First, we define the *tagging* of the components of a PHD $D_{k+1|k} = \sum_{h=1}^n P_E^h p^h(\mathbf{x})$ as the function $\text{tag} : \{p^h(\mathbf{x}), h = 1, \dots, n\} \rightarrow \mathbb{Z}$ such that

$$\text{tag}(p^h(\mathbf{x})) = \begin{cases} -1 & \text{if } p^h(\mathbf{x}) \text{ is not associated} \\ i & \text{if } p^h(\mathbf{x}) \text{ is associated to } \mathcal{R}_i \end{cases} \quad (13)$$

Hence, the tagging of a PHD is the association of identities to the components of the PHD itself. We use the integer -1 to conventionally indicate that a component is not associated to any robot.

Note that one component can be associated to at most one integer, but more components may be associated to the same integer, leading to an intrinsic multi-hypothesis filtering on the reduced relative configuration of each \mathcal{R}_i . In fact, the mean value of each component $p^h(\mathbf{x}) : \text{tag}(p^h(\mathbf{x})) = i, i > 0$ represents an hypothesis on the estimate over the relative configuration of \mathcal{R}_i . Each hypothesis is then weighted by the probability of existence of that component P_E^h , and the mean value of the component with the highest weight associated to \mathcal{R}_i is chosen as estimate of its 2-D reduced relative configuration.

Using the concept of tagging, we can now express the modification of the time update step (10) which enables the estimation of the identities, that is

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \sum_{h=1}^n \int \text{up}[f_{k+1|k}(\mathbf{x}|\mathbf{x}', Q) p_S P_E^h p^h(\mathbf{x}')] d\mathbf{x}' \quad (14)$$

where

$$\begin{aligned} \text{up}[f_{k+1|k}(\mathbf{x}|\mathbf{x}', Q) p_S P_E^h p^h(\mathbf{x}')] &= \\ &= \begin{cases} f_{k+1|k}(\mathbf{x}|\mathbf{x}', {}^{\mathcal{F}_i} \mathbf{q}_i) p_S P_E^h p^h(\mathbf{x}') & \text{if } \text{tag}(p^h(\mathbf{x}')) \neq -1 \\ \sum_{i \in N_0^{1:t}} f_{k+1|k}(\mathbf{x}|\mathbf{x}', {}^{\mathcal{F}_i} \mathbf{q}_i) \frac{p_S P_E^h}{|N_0^{1:t}|} p^h(\mathbf{x}') & \text{otherwise.} \end{cases} \end{aligned} \quad (15)$$

and $Q = \{{}^{\mathcal{F}_i} \mathbf{q}_i, \forall i \in N_0^t\}$ is the set of the odometries communicated to \mathcal{R}_0 .

If a component $p^h(\mathbf{x}')$ is associated to \mathcal{R}_i , then it is updated using the odometry ${}^{\mathcal{F}_i} \mathbf{q}_i$. Otherwise, $p^h(\mathbf{x}')$ is split in $|N_0^{1:t}|$ components equal to the original component with weight equal to $P_E^h/|N_0^{1:t}|$. Each of them is associated to a different $i \in N_0^{1:t}$ and updated using the odometry ${}^{\mathcal{F}_i} \mathbf{q}_i$. Once the odometry of \mathcal{R}_i is used to update a non previously associated component, that component is then automatically

associated to i for all future steps. Newly introduced components $b_{k+1|k}(\mathbf{x})$ are assigned $\text{tag}(b_{k+1|k}(\mathbf{x})) = -1$, so that the filter will automatically select a proper id.

Since our assumption of false positive measurements, we conventionally associate those objects to the id -2 , and add to the set Q the null odometry $\mathcal{F}^{-2}\mathbf{q}_{-2} = (0\ 0\ 0)^T$. Hence, object mistaken for robots will be included in the estimation of the PHD, but can be ignored in a second moment since their estimated ids will be -2 .

The initialization is conducted by generating in the state space random components with very high variance and very low weight and associating them to -1 .

Each single component update of equation (15) is the update step of a Bayesian recursive estimator, hence it can be implemented as the update step of a standard EKF. Denote with $\mathcal{F}_i \Delta \mathbf{q}_i^{k+1} = (\Delta \mathbf{q}_{p,i}^{k+1} \Delta \mathbf{q}_{\psi,i}^{k+1})^T$ the position and yaw displacement of \mathcal{R}_i in the time interval $[kT, (k+1)T]$ on the plane XY , which can be computed from $\mathcal{F}_i \mathbf{q}_i^{(k+1)T}$ and $\mathcal{F}_i \mathbf{q}_i^{kT}$.

Since the estimation is conducted in \mathcal{H}_0 , we need to take into account also its own motion in all single components updates. The position displacements of \mathcal{H}_0 on the XY plane can be computed as integration of $\mathcal{H}_0 \mathbf{v}_0$ with the Tustin method

$$\mathcal{H}_0 \Delta \mathbf{q}_{p,0}^{k+1} = T/2[\mathcal{H}_0 \mathbf{v}_0^{kT} + \mathbf{R}_Z(\mathcal{W} \Delta \psi_0^{k+1}) \mathcal{H}_0 \mathbf{v}_0^{(k+1)T}] \quad (16)$$

where the yaw displacement is easily computed as

$$\mathcal{H}_0 \Delta \mathbf{q}_{\psi,0}^{k+1} = \mathcal{W} \Delta \psi_0^{k+1} = \mathcal{W} \hat{\psi}_0^{(k+1)T} - \mathcal{W} \hat{\psi}_0^{kT}. \quad (17)$$

Hence, the time update of the single component is

$$\mathcal{H}_0 \mathbf{x}_i^{k+1|k} = \begin{pmatrix} \mathbf{R}_Z^T(\mathcal{H}_0 \Delta \mathbf{q}_{\psi,0}^{k+1}) [\mathcal{H}_0 \mathbf{p}_i^{k|k} - \mathcal{H}_0 \Delta \mathbf{q}_{p,0}^{k+1} + \\ + \mathbf{R}_Z(\mathcal{H}_0 \psi_i^{k|k}) \mathcal{F}_i \Delta \mathbf{q}_{p,i}^{k+1}] \\ \mathcal{F}_i \Delta \mathbf{q}_{\psi,i}^{k+1} + \mathcal{H}_0 \psi_i^{k|k} - \mathcal{H}_0 \Delta \mathbf{q}_{\psi,0}^{k+1} \end{pmatrix} \quad (18)$$

$$\mathbf{C}_{k+1|k} = \mathbf{F}_i^k \mathbf{C}_{k|k} \mathbf{F}_i^k + \mathbf{J}_i^{k+1} \mathbf{C}_i \mathbf{J}_i^{k+1} + \mathbf{J}_0^{k+1} \mathbf{C}_0 \mathbf{J}_0^{k+1} \quad (19)$$

where $\mathcal{H}_0 \mathbf{x}_i^{k+1|k}$ and $\mathbf{C}_{k+1|k}$ are mean and covariance of the prevision at time $(k+1)T$, $\mathcal{H}_0 \mathbf{x}_i^{k|k} = (\mathcal{H}_0 \mathbf{p}_i^{k|k} \mathcal{H}_0 \psi_i^{k|k})^T$ and $\mathbf{C}_{k|k}$ are mean and covariance of the estimate at time kT , \mathbf{F}_i^k , \mathbf{J}_i^{k+1} and \mathbf{J}_0^{k+1} are the Jacobian matrices of $\mathcal{H}_0 \mathbf{x}_i^{k+1|k}$ w.r.t. $\mathcal{H}_0 \mathbf{x}_i^{k|k}$, $\mathcal{F}_i \Delta \mathbf{q}_i^{k+1}$ and $\mathcal{H}_0 \Delta \mathbf{q}_0^{k+1}$ respectively (omitted for brevity). $\mathbf{C}_i, i \geq 0$ are constant matrices representing the noise on the movement of \mathcal{R}_i .

The measurement model of the single robot is

$$\mathbf{z} = \mathbf{H} \mathcal{H}_0 \mathbf{x}_i, \quad \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (20)$$

and the measurement update for each component is

$$\begin{aligned} \mathcal{H}_0 \mathbf{x}_i^{k|k} &= \mathcal{H}_0 \mathbf{x}_i^{k|k-1} + \mathbf{K}_k (\mathbf{z} - \mathcal{H}_0 \mathbf{x}_i^k) \\ \mathbf{C}_{k|k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{C}_{k|k-1} \end{aligned} \quad (21)$$

where \mathbf{K}_k is the well known Kalman gain.

The probability of detection p_D^h of each component can be computed by considering whether or not its mean falls in the field of view of the camera (FOV).

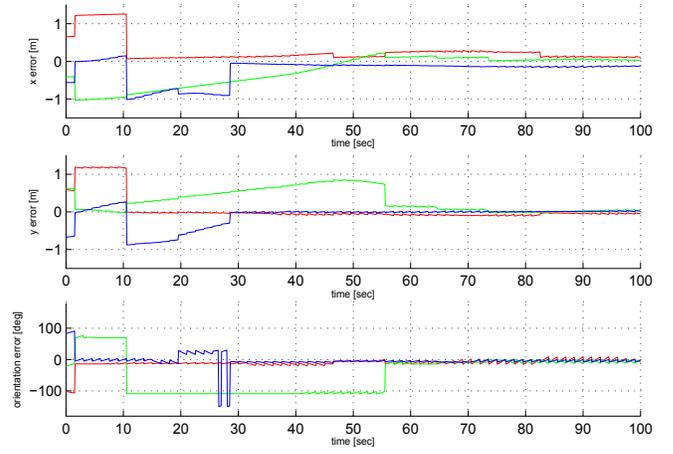


Fig. 5: Errors on x, y, ψ on the estimates of the three robots.

V. SIMULATION RESULTS

During the development of the system we used the Gazebo-ROS environment to simulate robots motion and gather all data needed by the algorithm. Realistic 3D models of all robots were used as well as simulated noisy sensors that emulate the real ones used in the experiments.

To simplify the robot detection, the UGVs are covered with simple markers (a white rectangle with a black dot inside). Hence, we perform a simple image segmentation using both color and shape data to extract robots measurements. More sophisticated methods may be used and the markers may be removed to recover a more general setting.

A typical simulation starts with the UAV in flight above three UGVs, a position control keeps it still in hovering for all the time. The UGVs randomly move performing obstacle avoidance. All the information as well as ground truth given by the simulator are collected together with common timing.

Fig. 5 shows the results of a typical simulation. In particular, the figure shows the errors in the x, y , and ψ components of the best hypothesis for all the three ground robots. The estimates converge in less than one minute. Note that during the transient, the changes in the best hypothesis lead to some instantaneous changes in the errors values. The final error is less than 12 cm for the position components while it is less than 10° for the angular one.

VI. EXPERIMENTAL RESULTS

After testing the system in simulation, we performed some experiments with real robots. The team of robots is composed by three unicycle-like Khepera III from K-Team and a quadrotor Pelican from Ascending Technologies.

The Khepera III robots are small sized wheeled mobile robots (14 cm diameter). They are a common platform used to develop and test swarm control algorithms. Each UGV is equipped with wheel encoders, a standard wireless connection and an ARM-7 CPU, powerful enough to collect sensor data, manage TCP/UDP net connection and perform some simple control algorithm. A low-level speed control was implemented on-board in order to manage trajectory following.

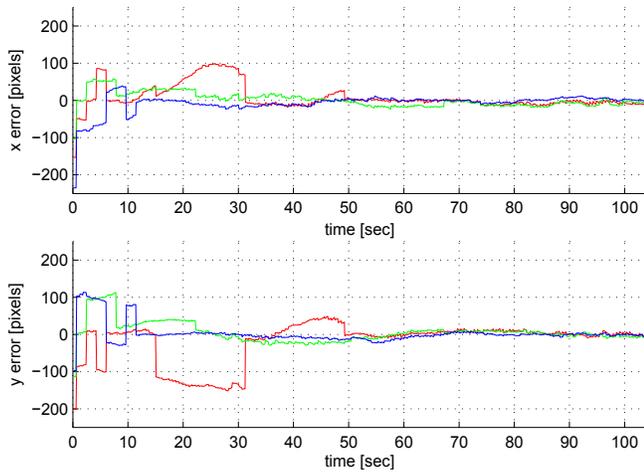


Fig. 6: Errors on the x and y components of the re-projection on the image plane of the estimates of the three UGVs w.r.t. corresponding features in a typical experiment.

The Pelican UAV comes with an IMU sensor (angular velocities and linear accelerations), a magnetometer (heading) and two ARM-7 CPUs used to collect and filter data from sensors and perform attitude stabilization. A sonar pointing along the positive $Z_{\mathcal{F}_0}$ axis was added to provide reliable indoor height measurements, while the camera is a standard USB webcam (25 Hz framerate). An embedded board with a standard wireless link and a 1.66GHz Atom CPU with GNU-Linux OS is used to manage data collection and transmission.

Since the UAV height controller is not implemented yet, during a typical experiment the quadrotor hangs from the ceiling with a wire. However, the propellers are on and turning in order to introduce all realistic noises in the IMU and camera measurements, while the yaw angle is manually driven. Two wheeled robots are performing some pre-defined closed trajectories (circle, eight) by using an open-loop control, and the third is manually driven to loiter around. The odometries of the UGVs are transmitted to a remote station and collected with their timestamps. Similarly, data from the UAV are collected on-board with their timestamps. All clocks are synchronized by using the Network Time Protocol (NTP) in order to have a common timing for all the measurements.

At the current stage, all the computations needed by the filter are performed offline. Note that the computational cost is quadratic with the number of robots as stated in [13], [9] and every computation can easily be performed online on-board the UAV to be used for control purpose.

Since our estimates are based on features positions in the image plane, in order to evaluate the performances of the system we can re-project the estimated positions of the UGVs on the image plane and calculate the errors w.r.t. them. Each feature is labeled and tracked in the video flow. The algorithm is run on the data set and robots identities and the estimates are collected. At the end each label is assigned to the robot that is closest in the final part (after the algorithm has converged) and the association is back-propagated till the begin of the experiment. In this way we can plot the estimates errors flows during the whole experiment.

Fig. 6 shows the plots built in the above described way for

a typical experiment. After the transient initial phase (around 50 seconds), the algorithm converges and the errors remain bounded under 20 pixels (corresponding to about 10 cm on the plane XY) till the end of the experiment. For a clip of the experiment see the attached video.

VII. APPLICATION IN A CONTROL PROBLEM

Once the relative configurations among the team members are retrieved by the localization system, it is possible to apply a control scheme to execute some desired tasks. To test the potentialities of our system, we are developing a control scheme for a set of tasks representing the basic bricks to build some intelligent behavior, whose design is still in progress. Each task can be represented with an appropriate function, whose Jacobian w.r.t. the robot velocities gives the relation between the robots movement and task accomplishment.

To combine the tasks we resorted to redundancy resolution techniques such as task augmentation and null space projection [15]. For each task we define a priority level according to its effect on overall system performances: low-priority tasks must not affect the execution of the high-priority ones.

Since the UGVs are moving in an unstructured environment, the highest priority task T_0 is obstacle avoidance. For safety reasons it is designed to rely on on-board proximity sensors only (e.g. a laser range finder). Each UGV runs this task independently and the influence on the overall system can be seen as a disturbance for all the lower-priority tasks.

Maintaining the UGVs in the camera field of view is another fundamental requirement for the whole system to work. In fact, even if the filter can cope with a temporary loss of view of some UGVs, the estimates on the configurations will drift if new measurements can not be collected. To increase the probability of detecting the largest number of robots, the UAV should hover above the robots centroid and the dispersion of the robots along the image plane must be kept inside the limits of the camera field of view. The dynamic of the feature associated with each UGV depends on both the UAV and UGV motion itself. The centroid of all the UGV features can be used to bring the camera view in the best position to observe all the robots.

We designed two visual based tasks T_1 and T_2 to take into account the motion of the features in the image plane. Task T_1 consists in driving the UAV with an Image Based Visual Servoing [16] scheme which generates linear velocities in order to maintain the centroid of the features from the UGVs at the center of the image plane. Task T_2 consists in keeping the dispersion of the features from the UGVs inside a predefined limit.

The choice of an IBVS scheme makes the execution of T_1 and T_2 robust to error in the robot localization and decouples them by the filter estimation. The UGVs can cooperate in the execution of these tasks with a controller based on both image information and relative configuration estimates.

Once the safety of the system has been assured, one may look forward to accomplish some high level tasks, as setting a particular formation for the UGVs or driving the whole team to a relevant spot. Those problems can be addressed

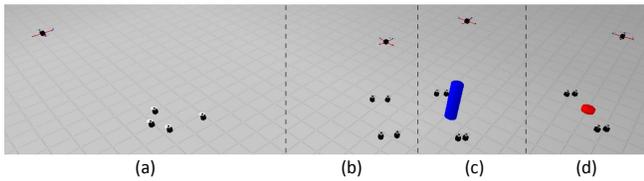


Fig. 7: Snapshots from numeric simulation: (a) initial state, (b) formation achieved, (c) obstacle (blue cylinder) avoidance is active, (d) final state (red dot is the goal).

by choosing a direction of motion, by assigning a given position as goal or by designing a trajectory for it. We chose to consider the centroid of the UGVs: task T_3 is designed to drive it toward a fixed goal.

If our formation is deployed to explore an unknown environment it may be useful to uniformly space the UGVs along the unknown area in an almost rigid formation, moving them as a platoon. To this aim we chose to deploy the robots on a circle centered into the UGVs' centroid: task T_4 is designed to force the UGVs to converge on assigned positions on the circle.

Note that T_3 and T_4 have a lower priority w.r.t. T_1 and T_2 , since they do not affect the localization system dependability. For this reason they are augmented and projected into the null space of tasks T_1, T_2 , similar to the work in [17].

In Fig. 7, we show four snapshots taken from a numeric simulation, while in Fig. 8 we show the normalized errors in the execution of the tasks. After starting the simulation in a random configuration (Fig. 7a), thanks to its redundancy the system is able to execute all the tasks (Fig. 7b), while all errors tends to zero. The rate of convergence and the oscillations depend on the priority of each specific task in the overall control scheme. When an obstacle is detected (Fig. 7c) the robots formation changes its shape and spreads, increasing the errors of the lower priority tasks (gray area in Fig. 8). The increase of the dispersion is compensated by increasing the altitude of the UAV and, thus, the camera field of view. Remember that the obstacle avoidance can be considered as a disturbance acting on the system: as soon as the perturbation is over the circular formation is restored and the error for task T_4 goes back to zero. Then the team reaches the goal (Fig. 7d).

VIII. CONCLUSIONS

We have presented a heterogeneous system including a single UAV and multiple UGVs. Its design required the solution of two main problems. The first is the estimation by the UAV of the relative configuration of the robots using anonymous measurements. For its solution, we have developed a relative localization system which employs a modification of the PHD filter to recover the identities of the robots. The system has been validated through both simulations and experiments.

The computed estimates can be applied in a control law as that outlined in Sect. VII, for which we presented some preliminary results obtained in simulation. We plan to extend this part in future works performing experiments. Another interesting challenge would be the use of multiple UAVs at the same time, in order to make the system more reliable.

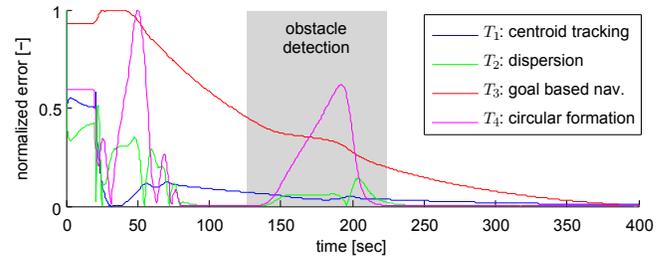


Fig. 8: Normalized task errors for high priority (T_1, T_2) and low priority (T_3, T_4) tasks. The gray area highlights the activity of obstacle avoidance controllers.

REFERENCES

- [1] R. Vaughan, G. Sukhatme, F. Mesa-Martinez, and J. Montgomery, "Fly spy: lightweight localization and target tracking for cooperating air and ground robots," in *Int. Symp. Distributed Autonomous Robot Systems*, 2000.
- [2] R. Simmons, D. Apfelbaum, D. Fox, R. Goldman, K. Haigh, D. Musliner, M. Pelican, and S. Thrun, "Coordinated deployment of multiple, heterogeneous robots," in *2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Takamatsu, Japan, 30 Oct. – 5 Nov. 2000, pp. 2254 – 2260 vol.3.
- [3] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, pp. 16–26, Sep. 2006.
- [4] Y. Feng, Z. Zhu, and J. Xiao, "Self-localization of a heterogeneous multi-robot team in constrained 3d space," in *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, USA, 29 Oct. – 2 Nov. 2007, pp. 1343 – 1350 vol.3.
- [5] R. Aragues, E. Montijano, and C. Sagues, "Consistent data association in multi-robot systems with limited communications," in *Robotics: Science and Systems*, Zaragoza, Spain, June 2010, pp. 97–104.
- [6] A. Franchi, P. Stegagno, and G. Oriolo, "Probabilistic mutual localization in multi-agent systems from anonymous position measures," in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 6534–6540.
- [7] P. Stegagno, M. Cagnetti, A. Franchi, and G. Oriolo, "Probabilistic mutual localization in multi-agent systems from anonymous bearing-only measures," in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011.
- [8] M. Cagnetti, P. Stegagno, A. Franchi, and G. Oriolo, "3-d mutual localization with anonymous bearing measurements," in *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, Minnesota, May 2012.
- [9] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [10] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *2003 IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 842–849.
- [11] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastri, *An invitation to 3D vision*. Springer, 2004.
- [12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. Springer, 2009.
- [13] B. Vo and W. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4091–4103, 2006.
- [14] O. Erdinc, P. Willett, and Y. Bar-Shalom, "The bin-occupancy filter and its connection to the phd filters," *IEEE Trans. on Signal Processing*, vol. 57, no. 11, pp. 4232–4246, 2009.
- [15] S. Chiaverini, G. Oriolo, and I. Walker, "Chapter 11: Kinematically redundant manipulators," in *Handbook of Robotics*, O. Khatib and B. Siciliano, Eds. Springer, 2009, pp. 245–268.
- [16] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 82–90, 2006.
- [17] G. Antonelli, F. Arrichiello, S. Chakraborti, and S. Chiaverini, "Experiences of formation control of multi-robot systems with the null-space-based behavioral control," in *2007 IEEE Int. Conf. on Robotics and Automation*, Roma, Italy, April 2007.