Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-5 (2015) 053–058

Multi-task Cooperative Control in a Heterogeneous Ground-Air Robot Team

Lorenzo Rosa, Marco Cognetti, Andrea Nicastro, Pol Alvarez, Giuseppe Oriolo

Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Rome, Italy,
{*rosa,cognetti,oriolo*}*@diag.uniroma1.it,*{*and.nicastro,pol.avms*}*@gmail.com*

Abstract: For a heterogeneous multi-robot system composed by an Unmanned Aerial Vehicle (UAV) equipped with a camera and several Unmanned Ground Vehicles (UGVs), we design a cooperative control scheme aimed at realizing a primary visual task, specifically designed to allow relative localization of the UGVs by the UAV, along with a generic secondary task to be carried out by the UGVs. The stability of the closed-loop system is formally proven. Inclusion of other behaviors, such as system navigation and ground obstacle avoidance, is considered. The theoretical results are validated by simulations in MATLAB and Gazebo/ROS.

Keywords: heterogeneous team, multi-robot system, multi-task control, stability analysis.

1. INTRODUCTION

In recent years, heterogeneous multi-robot systems have attracted increasing attention, thanks to the versatility that can be achieved by taking advantage of the different capabilities of each robot. For example the combination of Unmanned Aerial (UAV) and Ground (UGV) Vehicles is interesting, because the latter can perform tasks on the ground missions while the former monitor the mission from a vantage point. In particular, we will consider a heterogeneous system where a single UAV flies over several UGVs, acting as a supervisor and possibly providing look-ahead information to the ground robots, such as the presence of dangerous areas, obstacles, and so on.

The idea of controlling a multi-robot system to accomplish several tasks has been often considered in the literature. Antonelli and Chiaverini (2006) proposed a multi-task control scheme for a platoon of autonomous vehicles; this framework was validated via experimental results in Antonelli et al. (2007). Garcia et al. (2013) have attempted the fusion of different behaviors, such as individual, collective and social. Rahimi et al. (2014) have addressed a cooperative formation control problem for a multi-agent system employed in rescue and surveillance tasks. Saska et al. (2014) have used model predictive control to realize several simultaneous tasks including formation control, trajectory tracking and obstacle avoidance. Gans et al. (2009) have considered a system of multiple UAVs equipped with gimbaled cameras and propose a task-priority approach to coordinate its motion. Finally, Gans et al. (2011) have studied the problem of tracking multiple non-cooperative moving targets using an actuated camera.

In a previous work, we have addressed the problem of estimating relative poses within a heterogeneous multi-robot system composed by several UGVs and a single UAV (Stegagno et al., 2013). A preliminary multi-task control scheme making uses of these estimates was then developed and analyzed (Cognetti et al., 2014). In this paper, we further develop the same approach, and propose a general framework from which different control schemes may be derived, with various degrees of cooperation. The stability of the closed-loop system is carefully analyzed, with particular attention to perturbation terms due to the interaction among the tasks. A successful preliminary validation of the proposed control scheme is achieved via simulation results.

The paper is organized as follows. Section 2 gives an overview of the heterogeneous team, while in Section 3 we describe a rather general multi-task cooperative control approach. The stability of the closed-loop system is analyzed in Section 4. Section 5 discusses some specific choices of additional task and introduces an adaptive strategy for distributing the control effort among the robots. Simulation results of the proposed method are presented in Section 6. Section 7 concludes the paper.

2. THE HETEROGENEOUS TEAM

In this work, we consider a multi-robot system composed by several (henceforth, N) UGVs and a single UAV. The latter is capable of hovering and equipped with a vision sensor. The basic idea is that the UGVs will carry out motion tasks on the ground, while the UAV will fly above the UGVs in order to monitor their mission and, in particular, perform localization (see Fig. 1).

As UAV, we consider a rotary wing aircraft, and in particular a quadrotor. Considering the typical equipment of these systems, we postulate the presence on-board of an Inertial Measurement Unit (IMU), providing measurements of orientation, angular velocities and Cartesian accelerations, as well as of an altitude sensor, such as a barometer or, for indoor flying, a range finder. It is also assumed that the UAV carries a video camera attached to its bottom. In the following, we consider for simplicity the case of a rigid connection between the UAV and the camera, but the presence of an actuated gimbal can be easily accommodated.

10.1016/j.ifacol.2015.06.463

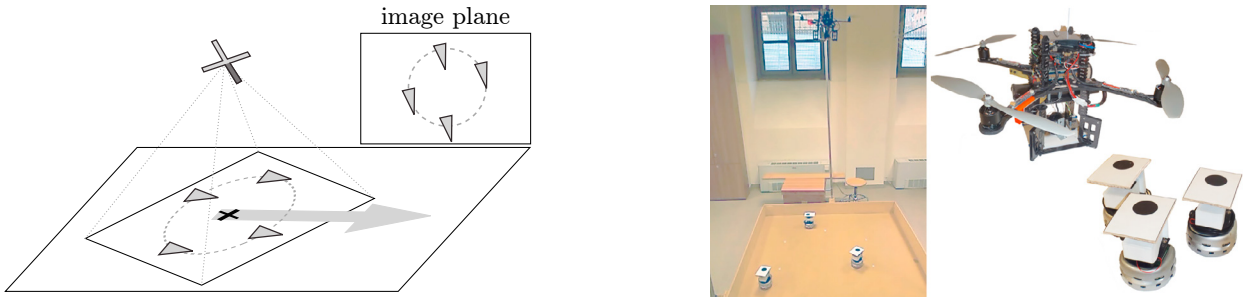

Fig. 1. Left: The heterogeneous team includes a single UAV and many UGVs. Right: The experimental set-up available in our lab consists of a Pelican quadrotor (Ascending Technologies) and several Khepera III robots (K-TEAM).

It is assumed that a Cartesian velocity controller is available as a basic component of the UAV control suite. Due to the inherent underactuation of quadrotors, Cartesian accelerations are generated via angular velocities that modify the aircraft orientation (Hua et al., 2013). Constant Cartesian velocity is achieved by keeping the UAV orientation fixed; this includes the case of zero Cartesian velocity (hovering), in which the UAV is kept horizontal.

As UGVs, we consider wheeled mobile robots and assume that they are equipped with wheel encoders. The latter provide measurements of the robot driving and steering velocity, which are used for odometric computations.

We assume that each ground robot has a low-level feedback loop for tracking desired Cartesian velocities. To achieve this in spite of the nonholonomy of wheeled robots, one may use a dynamic feedback linearization scheme (Oriolo et al., 2002). The high-level control inputs for our multi-robot system will therefore be the Cartesian velocities of the UAV and the UGVs, respectively a vector in $\mathbb{R}^3$ and N vectors in $\mathbb{R}^2$. The low-level controllers of each robot will automatically realize the high-level command by generating appropriate actuator inputs.

A fundamental component of our multi-robot system is a *relative localization* module that reconstructs the relative pose (i.e., position and orientation) of each UGV with respect to the UAV. An algorithm [1] for relative localization under the above sensing assumptions was developed in a previous work (Stegagno et al., 2013). Therefore, in the rest of the paper, we can consider relative localization data as available and focus on the control problem only.

3. MULTI-TASK COOPERATIVE CONTROL

In this section we define the control problem by setting up appropriate tasks for realizing the coordinated multi-robot system behavior described in the previous section. A solution is then composed by carefully analyzing the dynamics of each task.

3.1 Primary task

For the correct operation of our multi-robot system, and in particular for relative localization, it is vital that all the UGVs are maintained in the field of view of the UAV camera. Assuming that the UGVs appear as point features in the camera image, the above condition can be guaranteed by keeping (1) the centroid of the UGVs at the center of the image and (2) their dispersion to an appropriately chosen value. We then define [2] as a *primary* task function the composition of the UGV centroid $\boldsymbol{\sigma} \in \mathbb{R}^2$ and variance $\gamma \in \mathbb{R}$. Denoting the corresponding error by $\boldsymbol{e}_p = (\boldsymbol{e}_\sigma^T \;\; e_\gamma)^T$, the dynamics of $\boldsymbol{e}_p$ is

$$\dot{\boldsymbol{e}}_p = \begin{pmatrix} \dot{\boldsymbol{e}}_\sigma \\ \dot{e}_\gamma \end{pmatrix} = - \begin{pmatrix} \boldsymbol{J}_\sigma \\ \boldsymbol{J}_\gamma \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_A \\ \boldsymbol{v}_G \end{pmatrix} - \boldsymbol{J}_\omega \boldsymbol{\omega}_A = - \boldsymbol{J}_p \, \boldsymbol{v} - \boldsymbol{J}_\omega \boldsymbol{\omega}_A . \quad (1)$$

Here, $\boldsymbol{J}_\sigma$ and $\boldsymbol{J}_\gamma$ are the Jacobian matrices of $\boldsymbol{\sigma}$ and γ, respectively, with respect to the control input $\boldsymbol{v}$, which consists of the UAV Cartesian velocity $\boldsymbol{v}_A$ and the UGV Cartesian velocities $\boldsymbol{v}_G$ (we use the A subscript for the UAV and the G subscript for the UGVs). Note that the dynamics of $\boldsymbol{\sigma}$ and γ, and therefore of $\boldsymbol{e}_p$, are also influenced by the UAV angular velocity $\boldsymbol{\omega}_A$ through the Jacobian matrix $\boldsymbol{J}_\omega$. However, the corresponding term in (1) must be considered as an exogenous perturbation, because the UAV angular velocities $\boldsymbol{\omega}_A$ are generated by the low-level Cartesian velocity controller.

Now consider the $3 \times (3 + 2N)$ primary task Jacobian $\boldsymbol{J}_p$ in (1), and partition it column-wise as $(\boldsymbol{J}_{p,A} \;\; \boldsymbol{J}_{p,G})$. It is easy to prove that the 3×3 matrix $\boldsymbol{J}_{p,A}$ is always invertible, which implies that $\boldsymbol{J}_p$ is full row rank. Therefore, the following choice of control input in (1) renders the primary task error dynamics decoupled and globally exponentially stable:

$$\boldsymbol{v} = \boldsymbol{v}_p = \boldsymbol{G}(\boldsymbol{K}_p \boldsymbol{e}_p - \boldsymbol{J}_\omega \, \boldsymbol{\omega}_A), \quad (2)$$

where $\boldsymbol{G}$ is any generalized inverse of $\boldsymbol{J}_p$ and $\boldsymbol{K}_p$ is a positive definite diagonal gain matrix. In fact, since $\boldsymbol{J}_p$ is full row rank, any generalized inverse $\boldsymbol{G}$ of $\boldsymbol{J}_p$ is also a right inverse, i.e., $\boldsymbol{J}_p \boldsymbol{G} = \boldsymbol{I}$.

The choice of $\boldsymbol{G}$ in (2) allows to modulate the distribution of the control effort among the robots. For example, one may set

$$\boldsymbol{G} = \begin{pmatrix} (\boldsymbol{J}_{p,A})^{-1} \\ \boldsymbol{0} \end{pmatrix}, \quad (3)$$

which corresponds to assigning full responsibility for the primary task to the UAV and leaving the UGVs completely

[1] A remarkable feature of this algorithm is that no UGV tagging is required: i.e., the ground robots can be indistinguishable one from the other in the UAV camera image (*anonymous* measurements). The identity of each robot will be reconstructed by the algorithm.

[2] Refer to (Cognetti et al., 2014) for detailed expressions.

free to pursue other goals. Another possibility is to define a weighting matrix $\boldsymbol{W} = \text{diag}\{\boldsymbol{w}_A, \boldsymbol{w}_G\}$, with $\boldsymbol{w}_A$ and $\boldsymbol{w}_G$ weight vectors for the UAV and the UGV velocities, and use a weighted pseudoinverse of $\boldsymbol{J}_p$:

$$\boldsymbol{G} = (\boldsymbol{J}_p)^{\dagger}_W = \boldsymbol{W}^{-1}\boldsymbol{J}_p^T(\boldsymbol{J}_p\boldsymbol{W}^{-1}\boldsymbol{J}_p^T)^{-1}, \tag{4}$$

producing the velocity input $\boldsymbol{v}_p$ with minimum weighted norm $\boldsymbol{v}_p^T\boldsymbol{W}\boldsymbol{v}_p$. Setting $\boldsymbol{W} = \boldsymbol{I}$ corresponds to distributing the control effort related to the primary task as evenly as possible among all the robots, whereas different effort distributions are achieved by choosing non-uniform weights. In particular, one may show that when all the components of $\boldsymbol{w}_G$ tend to infinity the weighted pseudoinverse $(\boldsymbol{J}_p)^{\dagger}_W$ tends to the particular generalized inverse (3).

3.2 Secondary task

Assume now that the UGVs are assigned one or more motion tasks to be carried out on the ground (e.g., formation control, navigation, etc.). Thanks to the redundancy of the multi-robot system, it is possible to stack these into a secondary task to be pursued beside the primary task. As for the latter, we suppose that the objective is to regulate the secondary task function to a constant desired value.

Denoting with $\boldsymbol{e}_s$ the error on the *secondary* task, its dynamics depends only on the UGV velocities $\boldsymbol{v}_G$ and is generically written as

$$\dot{\boldsymbol{e}}_s = -\boldsymbol{J}_s\,\boldsymbol{v}_G, \tag{5}$$

where $\boldsymbol{J}_s$ is the Jacobian matrix of the secondary task function. If we were interested in this task only, setting the UGV input velocities to

$$\boldsymbol{v}_G = \boldsymbol{v}_{G,s} = \boldsymbol{J}_s^{\dagger}\boldsymbol{K}_s\boldsymbol{e}_s \tag{6}$$

would guarantee exponential convergence of $\boldsymbol{e}_s$ to zero while minimizing the overall UGV control effort, provided that $\boldsymbol{J}_s$ is full rank and $\boldsymbol{K}_s$ is positive definite.

However, since in general the UGVs must also cooperate to the execution of the primary task, we must add the UGV control input $\boldsymbol{v}_{G,s}$ given by (6) to the primary UGV control input $\boldsymbol{v}_{G,p}$ specified by the lower components of the vector in (2):

$$\boldsymbol{v}_G = \boldsymbol{v}_{G,p} + \boldsymbol{v}_{G,s}.$$

This produces a perturbation on both tasks.

The perturbation in the dynamics of the primary task can be compensated by adding a suitable UAV velocity $\boldsymbol{v}_{A,s}$ to the primary UGV control input $\boldsymbol{v}_{A,p}$ specified by (2):

$$\boldsymbol{v}_A = \boldsymbol{v}_{A,p} + \boldsymbol{v}_{A,s}.$$

The dynamics of the primary task error become

$$\dot{\boldsymbol{e}}_p = -\boldsymbol{K}_p\boldsymbol{e}_p + \boldsymbol{J}_{p,A}\,\boldsymbol{v}_{A,s} + \boldsymbol{J}_{p,G}\,\boldsymbol{v}_{G,s},$$

and exponential stability of the primary task error is preserved by choosing the UAV compensation velocity as

$$\boldsymbol{v}_{A,s} = -(\boldsymbol{J}_{p,A})^{-1}\boldsymbol{J}_{p,G}\,\boldsymbol{v}_{G,s}. \tag{7}$$

Wrapping up, we obtain two different multi-task control schemes depending on the choice of $\boldsymbol{G}$ in (2). The first (*Scheme 1*) is obtained using the generalized inverse (3) and leads to the following velocity inputs:

$$\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_A \\ \boldsymbol{v}_G \end{pmatrix} = \begin{pmatrix} (\boldsymbol{J}_{p,A})^{-1}(\boldsymbol{K}_p\boldsymbol{e}_p - \boldsymbol{J}_\omega\,\boldsymbol{\omega}_A) + \boldsymbol{v}_{A,s} \\ \boldsymbol{v}_{G,s} \end{pmatrix},$$

with $\boldsymbol{v}_{A,s}$ and $\boldsymbol{v}_{G,s}$ respectively given by (7) and (6).

The second scheme (*Scheme 2*) is obtained using the generalized inverse (4). The resulting velocity inputs are

$$\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_A \\ \boldsymbol{v}_G \end{pmatrix} = (\boldsymbol{J}_p)^{\dagger}_W(\boldsymbol{K}_p\boldsymbol{e}_p - \boldsymbol{J}_\omega\,\boldsymbol{\omega}_A) + \begin{pmatrix} \boldsymbol{v}_{A,s} \\ \boldsymbol{v}_{G,s} \end{pmatrix}, \tag{8}$$

with the same $\boldsymbol{v}_{A,s}$ and $\boldsymbol{v}_{G,s}$ of Scheme 1.

Scheme 1 represents a *non-cooperative* control approach, in which the UGVs address only the secondary task and the UAV alone is in charge of the primary task, whereas Scheme 2 is *cooperative* because the UGVs contribute also to the primary task. As a consequence, in Scheme 2 one may expect a more uniform distribution of the control effort among the robots.

Another difference between the two schemes is related to the dynamics of the secondary task (recall that correct execution of the primary task is guaranteed in both schemes, thanks to the compensation term $\boldsymbol{v}_{A,s}$). If Scheme 1 is used, then the UGV control velocities consist only of the term $\boldsymbol{v}_{G,s}$, producing no perturbation on the secondary task error dynamics which remains decoupled and globally exponentially stable. When using Scheme 2, however, the UGV control velocities include the term $\boldsymbol{v}_{G,p}$ that contributes to realizing the primary task. This introduces a perturbation on the dynamics of the secondary task error, which become

$$\dot{\boldsymbol{e}}_s = -\boldsymbol{K}_s\boldsymbol{e}_s - \boldsymbol{J}_s\boldsymbol{v}_{G,p}.$$

In the next section, we prove that the perturbation is asymptotically rejected by the controller.

4. STABILITY ANALYSIS

Denote by $\boldsymbol{e} = (\boldsymbol{e}_p^T, \boldsymbol{e}_s^T)^T$ the composite (primary and secondary) task error. Before analyzing its dynamics, we introduce two preliminary assumptions.

As already mentioned in Section 2, the UAV is driven by a low-level controller which uses the angular velocities $\boldsymbol{\omega}_A$ to realize the Cartesian velocity input specified by the multi-task controller. This input aims at zeroing the primary task error, and it vanishes at steady state. Based on this, we will assume that

$$\|\boldsymbol{\omega}_A\| \leq \beta\|\boldsymbol{e}_p\|, \qquad \beta > 0 \tag{9}$$

in a neighborhood N of the origin.

The second assumption concerns the Jacobian matrix $\boldsymbol{J}_s$ of the so far general secondary task, which we will suppose to be bounded in norm.

We can now state the following proposition.

Proposition 1. Under the above assumptions, the control scheme (8), with $\boldsymbol{v}_{G,s}$ as in (6) and $\boldsymbol{v}_{A,s}$ as in (7), renders the error dynamics exponentially stable around $\boldsymbol{e} = \boldsymbol{0}$.

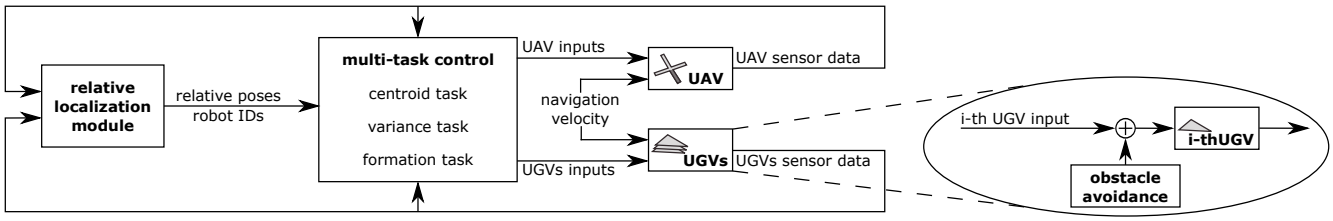

Fig. 2. The structure of the proposed multi-task control scheme. Note the central role of the relative localization module.

Proof. The closed-loop dynamics of $\boldsymbol{e}$ is

$$\dot{\boldsymbol{e}} = \begin{pmatrix} \dot{\boldsymbol{e}}_p \\ \dot{\boldsymbol{e}}_s \end{pmatrix} = - \begin{pmatrix} \boldsymbol{K}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_s \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_s \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{J}_s \, \boldsymbol{v}_{G,p} \end{pmatrix} , \quad (10)$$

which can be interpreted as a system globally exponentially stable at the origin, perturbed by the term $\boldsymbol{J}_s \, \boldsymbol{v}_{G,p}$. Isolating $\boldsymbol{v}_{G,p}$ from (8), using (9), and taking into account the boundedness of $\boldsymbol{J}_s$, one readily obtains

$$\| \boldsymbol{J}_s \, \boldsymbol{v}_{G,p} \| \leq \gamma \| \boldsymbol{e} \|,$$

in the same neighborhood N where (9) is valid. At this point, using the Lyapunov function

$$V = \alpha \| \boldsymbol{e} \|^2 , \qquad \alpha > 0,$$

and applying well-known results (Khalil, 2002, Lemma 9.1), it is possible to prove local exponential stability of the closed-loop system (10). Note that if (9) holds globally, then stability is global. □

Perturbation terms unrelated to task errors may be considered; for example, this may be the case of obstacle avoidance actions performed by the UGVs (see the next section). Since in general these perturbations are not structurally zero at the desired regulation point, they must be considered as persistent perturbations. If their norm is bounded, however, ultimate boundedness arguments (Khalil, 2002, Chap. 9) can be used to show that the induced steady-state error is limited. As soon as the perturbation vanishes, however, exponential convergence of both the primary and the additional tasks is recovered.

5. RESULTING CONTROL SCHEME

In this section, a specific form is chosen for the additional task. This will lead to the multi-task control scheme shown in Fig. 2, whose structure will be discussed in detail.

A typical task of interest for ground robots is formation control. For example, assume that we would like all the UGVs to remain at a distance R from their centroid $\boldsymbol{\mu}$. A suitable definition of the error for this task is

$$\boldsymbol{e}_s = \boldsymbol{e}_\phi = \begin{pmatrix} R^2 \\ \vdots \\ R^2 \end{pmatrix} - \boldsymbol{\phi} = \begin{pmatrix} R^2 \\ \vdots \\ R^2 \end{pmatrix} - \begin{pmatrix} (\boldsymbol{x}_1 - \boldsymbol{\mu})^T (\boldsymbol{x}_1 - \boldsymbol{\mu}) \\ \vdots \\ (\boldsymbol{x}_N - \boldsymbol{\mu})^T (\boldsymbol{x}_N - \boldsymbol{\mu}) \end{pmatrix} ,$$

where $\boldsymbol{x}_i$ is the position of the i-th UGV. One may easily show that $\boldsymbol{e}_s$ can be computed from the relative positions of the UGVs with respect to the camera frame, which are provided by our localization system.

The secondary task error dynamics is in the form (5), with $\boldsymbol{J}_s = \boldsymbol{J}_\phi$ directly computed from the above task function. The additional velocity for the UGV is then

$$\boldsymbol{v}_{G,s} = \boldsymbol{J}_\phi^\dagger \boldsymbol{K}_s \boldsymbol{e}_s \, ,$$

and the associated compensation velocity for the UAV is computed as in (7). At this point, we may use either of the two control schemes (1 and 2, non-cooperative and cooperative) discussed in Section 3.

Note that a common horizontal velocity can be freely added to the multi-robot system, i.e., to the UAV as well as to the UGVs. In fact, it is easy to verify that such a drift term perturbs neither the primary nor the additional task, and can therefore be used to assign a high-level navigation velocity to the whole system.

Another important remark is related to obstacle avoidance, an essential feature for any multi-robot system including ground robots. In particular, we assume that every UGV is equipped with a local control module that monitors data from an on-board sensor (e.g., a laser range finder) and, when necessary, computes a velocity term $\boldsymbol{v}_{G,oa}$ aimed at avoiding collisions with nearby obstacles. Independently from the specific realization of this module, which is outside the scope of this paper, $\boldsymbol{v}_{G,oa}$ is a locally generated command that is added to the UGV velocity input. In general, this clearly results in a transient perturbation on both the primary and the additional task. The perturbation on the primary task can be cancelled by adding to the UAV velocity input a further compensation term computed from (7) with $\boldsymbol{v}_{G,oa}$ in place of $\boldsymbol{v}_{G,s}$. In any case, as soon as the obstacle avoidance action disappears, the multi-robot system resumes normal operation in which both the primary and the additional task error exponentially converge to zero.

At a closer look, however, neither Scheme 1 nor 2 are completely satisfactory if we take into account the occasional necessity of obstacle avoidance. Scheme 1 assigns full responsibility for the primary task to the UAV, leaving the UGVs completely free to execute the additional task and, when necessary, obstacle avoidance; however, during normal operation this non-cooperative scheme results in a larger control effort (hence, faster energy consumption) for the UAV. On the other hand, the cooperative approach of Scheme 2 achieves better control effort distribution during normal operation but requires that the UGVs always contribute to the primary task, therefore disturbing obstacle avoidance whenever this becomes necessary.

More flexibility is achieved by adopting Scheme 2 with a variable weighting matrix $\boldsymbol{W}$: in particular, one may use $\boldsymbol{W} = \boldsymbol{I}$ during normal operation to distribute the control effort evenly among the robots, and quickly increase the UGV weight vector $\boldsymbol{w}_G$ when one or more UGVs are performing obstacle avoidance so as to approach the per-

formance of Scheme 1. With this strategy, during obstacle avoidance the responsibility of the primary task is rapidly transferred to the UAV, guaranteeing its execution, while the UGVs are released from this task, resulting at the same time in a more effective obstacle avoidance action.

In practice, the above mechanism may be realized by letting the weight vector $\boldsymbol{w}_G$ vary as a function of the UGV obstacle avoidance term: One possible choice is

$$\boldsymbol{w}_G = (1 + k_{oa}||\boldsymbol{v}_{G,oa}||_M)\,\mathbf{1},$$

where k_{oa} is a positive constant, $||\boldsymbol{v}_{G,oa}||_M$ denotes the norm of the largest obstacle avoidance velocity term among all UGVs, and $\mathbf{1}$ is a $2N$-vector with unit elements.

6. SIMULATION RESULTS

As a preliminary step towards experimental validation, the performance of the proposed control scheme has been evaluated on a simulated version of our experimental set-up shown in Fig. 1. The quadrotor simulation accounts for rotational dynamics, aerodynamic disturbances, rotor dynamics, as well as the presence of low-level control loops. The UGVs are simulated as differential-drive robots equipped with a low-level control loop based on feedback linearization for velocity tracking.

Two simulation environments have been used: MATLAB and Gazebo/ROS. In the MATLAB simulations, all sensors (camera, IMU, altimeter, encoders) are assumed to be ideal, while in Gazebo/ROS a realistic simulation of the actual sensors available on our robots is included.

In the first MATLAB simulation, the multi-robot system is assigned only the primary task. Starting from a initial arrangement yielding a nonzero task error, the system should converge to one in which the centroid of the UGVs is at the center of the image and their variance matches a desired value. Figure 3 shows the results obtained by using Schemes 1 and 2 (with $\boldsymbol{W} = \boldsymbol{I}$). As expected, the performance of the two schemes is indistinguishable from the viewpoint of the task error, that converges exponentially to zero. The distribution of the control effort is however different: in fact, Scheme 1 requires a much higher velocity input for the UAV, to which primary task execution is completely delegated.

The second MATLAB simulation deals with a more complete scenario, in which the multi-robot system must satisfy the primary task as well as the formation task defined in Section 3.2. Moreover, a high-level navigation velocity is specified, that leads the UGV towards a ground obstacle. Scheme 2 with variable $\boldsymbol{W}$ is adopted for multi-task control, using the weight adjustment mechanism described at the end of the previous section. Results are shown in Fig. 4. Note how both the visual and formation tasks are quickly realized in the initial part of the simulation, with a rather uniform distribution of the control effort among the robots. At $t = 13$ s, the obstacle avoidance phase starts and responsibility for the primary task is progressively transferred to the UAV. Thanks to the compensation term, the execution of the primary task is not perturbed, whereas an error on the formation tasks appears, indicating that the UGVs temporarily break the desired formation to avoid a collision. At circa $t = 30$ s the obstacle is overcome,

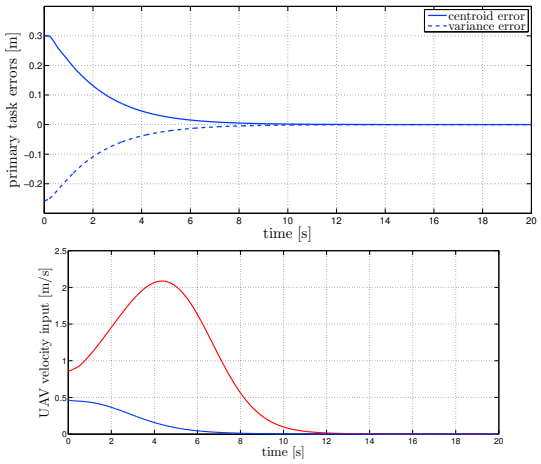

Fig. 3. MATLAB simulation 1 (red: Scheme 1; blue: Scheme 2). Top: primary task errors. Bottom: norm of the UAV velocity input.

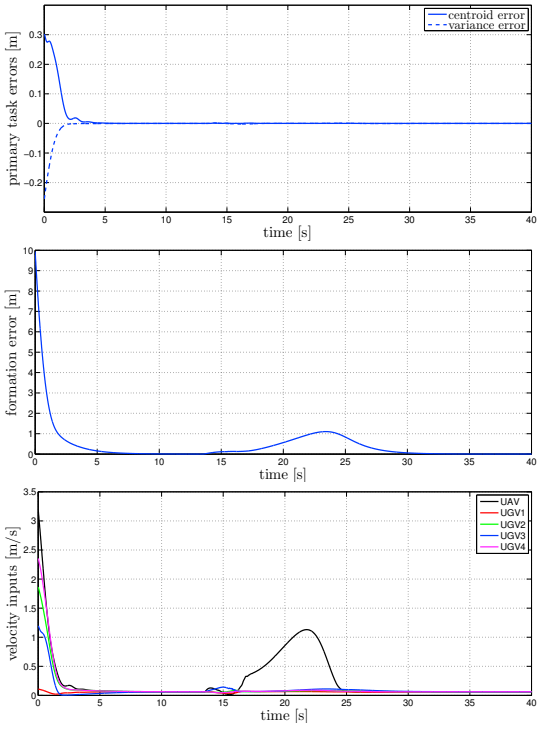

Fig. 4. MATLAB simulation 2. Top: primary task errors. Center: norm of the formation error. Bottom: norm of the velocity input for each robot.

the perturbation on the formation task vanishes, and the corresponding error rapidly returns to zero.

Finally, Fig. 5 shows some snapshots from a simulation in Gazebo/ROS, which considers a scenario similar to the second MATLAB simulation and uses the same control scheme. Task errors are reported in Fig. 6, where the effect of the sensor noise is clearly visible. Obstacle avoidance takes place approximately between $t = 17$ s and $t = 120$ s; as before, only the formation task is temporarily perturbed by this action.

See http://www.diag.uniroma1.it/labrob/research/heter.html for additional details and video clips.

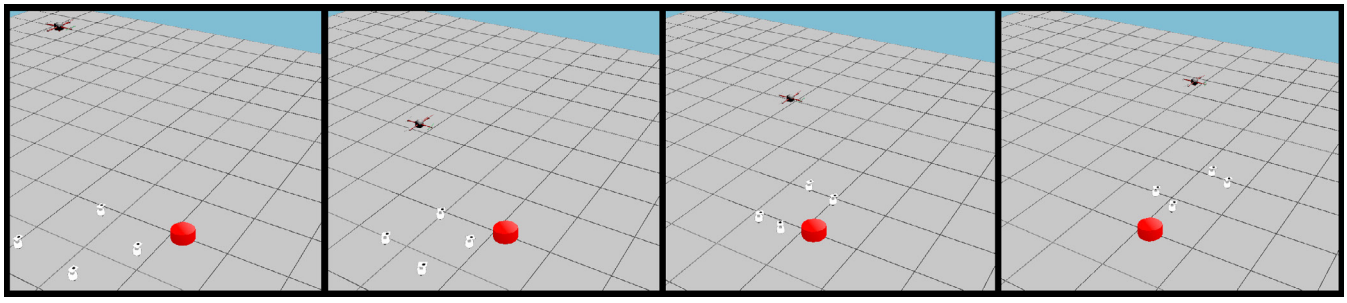

Fig. 5. Snapshots from the Gazebo/ROS simulation. The red cylinder represents a ground obstacle.

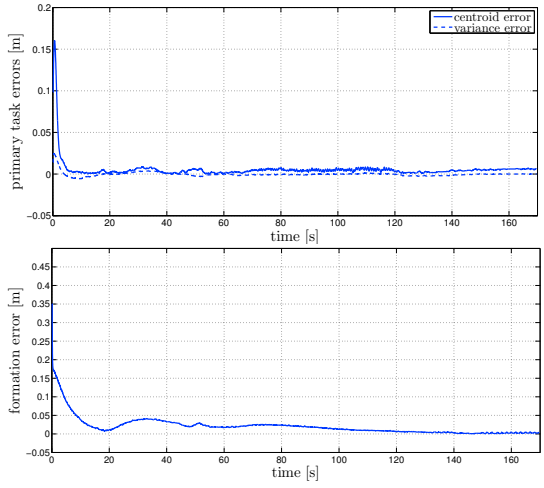

Fig. 6. Gazebo/ROS simulation. Top: primary task errors. Bottom: norm of the formation error.

7. CONCLUSIONS

We have described a cooperative control framework for a multi-robot system composed by a single UAV and several UGVs. Thanks to its redundancy, the system can pursue multiple tasks. In particular, we have considered visual tracking from the air and formation control on the ground as tasks to be executed. The mutual interaction between these tasks has been studied, and the stability of the closed-loop system has been analyzed. Moreover, we have discussed the perturbation introduced by temporary obstacle avoidance actions required for the ground robots. Realistic simulations of the proposed control scheme corroborate the theoretical findings. This constitutes a first encouraging step towards experimental validation, which is the next objective of our research.

REFERENCES

Antonelli, G., Arrichiello, F., Chakraborti, S., and Chiaverini, S. (2007). Experiences of formation control of multi-robot systems with the Null-Space-based Behavioral Control. In *2007 IEEE Int. Conf. on Robotics and Automation*, 1068–1073. Roma, Italy.

Antonelli, G. and Chiaverini, S. (2006). Kinematic Control of Platoons of Autonomous Vehicles. *IEEE Transactions on Robotics*, 22(6), 1285–1292.

Cognetti, M., Oriolo, G., Peliti, P., Rosa, L., and Stegagno, P. (2014). Cooperative Control of a Heterogeneous Multi-Robot System based on Relative Localization. In *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 350 – 356. Chicago, IL, USA.

Gans, N., Curtis, J., Barooah, P., Shea, J., and Dixon, W. (2009). Balancing Mission Requirement for Networked Autonomous Rotorcrafts Performing Video Reconnaissance. In *2009 AIAA Guidance, Navigation, and Control Conference*, 1–14. Chicago, IL, USA.

Gans, N.R., Hu, G., Nagarajan, K., and Dixon, W.E. (2011). Keeping Multiple Moving Targets in the Field of View of a Mobile Camera. *IEEE Transactions on Robotics*, 27(4), 822–828.

Garcia, P., Caamano, P., Duro, R.J., and Bellas, F. (2013). Scalable Task Assignment for Heterogeneous Multi-Robot Teams. *2013 International Journal of Advanced Robotic Systems*, 10, 1–10.

Hua, M., Hamel, T., Morin, P., and Samson, C. (2013). Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Control Systems Magazine*, 33(1), 61–75.

Khalil, H.K. (2002). *Nonlinear Systems*. Prentice-Hall.

Oriolo, G., De Luca, A., and Vendittelli, M. (2002). WMR control via dynamic feedback linearization: Design, implementation and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6), 835–852.

Rahimi, R., Abdollahi, F., and Naqshi, K. (2014). Time-varying formation control of a collaborative heterogeneous multi agent system. *Robotics and Autonomous Systems*, 62, 1799–1805.

Saska, M., Vonásek, V., Krajník, T., and Přeučil, L. (2014). Coordination and Navigation of Heterogeneous MAV-UGV Formations Localized by a 'Hawk-eye'-like Approach Under a Model Predictive Control Scheme. *The International Journal of Robotics Research*, 33(10), 1393–1412.

Stegagno, P., Cognetti, M., Rosa, L., Peliti, P., and Oriolo, G. (2013). Relative Localization and Identification in a Heterogeneous Multi-Robot System. In *2013 IEEE Int. Conf. on Robotics and Automation*, 1857–1864. Karlsruhe, Germany.