Experiments on coordinated motion of aerial robotic manipulators

G. Muscio[1], F. Pierri[1], M. A. Trujillo[2], E. Cataldi[3], G. Giglio[1], G. Antonelli[3], F. Caccavale[1], A. Viguria[2], S. Chiaverini[3] and A. Ollero[4]

***Abstract*— In this paper a three layer control architecture for multiple aerial robotic manipulators is presented. The top layer, on the basis of the desired mission, determines the end-effector desired trajectory for each manipulator, while the middle layer is in charge of computing the motion references in order to track such end-effectors trajectories coming from the upper layer. Finally the bottom layer is a low level motion controller, which tracks the motion references. The overall mission is decomposed in a set of elementary behaviors which are combined together, through the Null Space-based Behavioral (NSB) approach, into more complex compounds behaviors. The proposed framework has been tested conducting an experimental campaign.**

I. Introduction

Unmanned Aerial Vehicles (UAVs), over recent years, have been driving the researchers' attention for a wide range of applications involving inspection, surveillance, aerial video and photography. A new application field is the aerial manipulation, in which UAVs are equipped with grippers [1] and, more recently, multi-joint robotic manipulators [2] [3] (UAVM, Unmanned Aerial Vehicle-Manipulator systems).

In order to accomplish complex missions, which would be impossible for a single robot, teams of Unmanned Aerial Vehicles (UAVs) are often applied in operations such as surveillance, remote monitoring and mapping [4]. The adoption of multiple UAVMs in aerial manipulation tasks allows to overcome typical limitations (e.g., payload, battery autonomy, etc.) making them suitable for cooperative assembly of structures in remote or hazardous environments also involving cooperative transportation of large and/or heavy payloads [5].

Several control architectures for multi-robot systems have been proposed in literature [6]. More in detail, in [7] a flexible and reconfigurable platform for cooperative aerial multi-robot systems has been developed, while in [8] is presented a controller based on a decisional architecture for multi-UAV systems, that uses different control schemes depending on the current task status. In [9] the authors proposed a control framework for multiple aerial manipulators, in which the Null-Space based Behavioral (NSB) control [10] is exploited for complex mission achievement.

In this paper an experimental validation of a coordinated control framework for multiple aerial manipulators is proposed. In detail, a multi-layer approach has been adopted: the first layer is a centralized planner, that communicates with every vehicle and generates the trajectories for each end-effector; the second layer, local to each vehicle, through the Null Space-based Behavioral control, generates the references for the controlled variables; finally, the last layer ensures the tracking of the previous layer outputs. Preliminary experiments, on the single UAVM and involving the second and third layers only, have been presented in [11].

The effectiveness of the proposed architecture has been experimentally proven on a setup composed of two multi-rotor aerial platforms equipped with a 6 DOFs manipulator. The experimental setup has been developed and manufactured by CATEC (Centro Avanzado de Tecnologías Aeroespaciales) within the EU-funded project ARCAS [12] (Aerial Robotics Cooperative Assembly System), aimed at developing cooperative free-flying robot system for assembly and structure construction.

II. Modeling

Let us consider a team of aerial robotic manipulators, i.e., Unmanned Aerial Vehicles equipped with robotic Manipulators (UAVMs). The formulation includes two sets of robots:

- N_T UAVMs which are able to move in a coordinated way, e.g., to transport an object in a cooperative way: hereafter this set of UAVMs will be referred as *Transporting Robots* (TRs). The adopted task formulation does not strictly requires the presence of a physical object grasped by the TRs (e.g., in the case of simple coordinated motion of the end-effectors).
- N_A UAVMs, acting as *Auxiliary Robots* (ARs), whose motion has to be coordinated with that of the TRs. For example an AR could be equipped with a camera pointing at a given location.

Each manipulator (attached to the UAV base) has n_{Mi} (n_{Mj}) DOFs (hereafter $i = 1, \ldots, N_T$ and $j = 1, \ldots, N_A$), while each flying vehicle has n_{Vi} (n_{Vj}) actuated DOFs (e.g., n_{Vi}=4 for a standard quadrotor). Hence, each aerial manipulator has $n_i = n_{Mi} + n_{Vi}$ ($n_j = n_{Mj} + n_{Vj}$) DOFs, and the total number of DOFs of the system is given by $n = \sum_{i=1}^{N_T} n_i + \sum_{j=1}^{N_A} n_j$. Of course, if a given object is grasped by the TRs, the total number of DOFs reduces to $n = \sum_{i=1}^{N_T} n_i + \sum_{j=1}^{N_A} n_j - n_O$, being n_O the mechanical constraints imposed by the object. In the following, it is assumed that $n_i \geq 6$ ($n_j \geq 6$).

The relevant coordinate frames are (Figure 1):

- the common base frame $\mathcal{F}$ (fixed and inertial);
- for the TRs, N_T end-effector frames, $\mathcal{F}_{ETi}$, and N_T frames attached to the vehicle body with origin in the center of mass, $\mathcal{F}_{VTi}$ ($i = 1, \ldots, N_T$);

[1] F. Caccavale, G. Giglio, G. Muscio and F. Pierri are with University of Basilicata, Potenza, Italy. Corresponding author: Francesco Pierri, e-mail: francesco.pierri@unibas.it.

[2] G. Antonelli, E. Cataldi and S. Chiaverini are with University of Cassino and Southern Lazio, Cassino, Italy.

[3] M.A. Trujillo and A. Vigura are with Center for Advanced Aerospace Technologies (CATEC), Sevilla, Spain.

[4] A. Ollero is with University of Sevilla, Sevilla, Spain.

This research has been supported by the European Community's 7th Framework Program under grant agreement No.287617 (IP project ARCAS - Aerial Robotics Cooperative Assembly system) and grant agreement No. 608849 (IP project EUROC - European Robotics Challenges), and by the European Community's Horizon 2020 Program under grant agreement No. 644271 (IP project AEROARMS - AErial RObotic system integrating multiple ARMS and advanced manipulation capabilities for inspection and maintenance).

Preprint version of the paper: G. Muscio, F. Pierri, M.A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, A. Ollero, "Experiments on coordinated motion of aerial robotic manipulators." IEEE International Conference on Robotics and Automation, pp. 1224-1229, 2016. DOI: 10.1109/ICRA.2016.7487252.

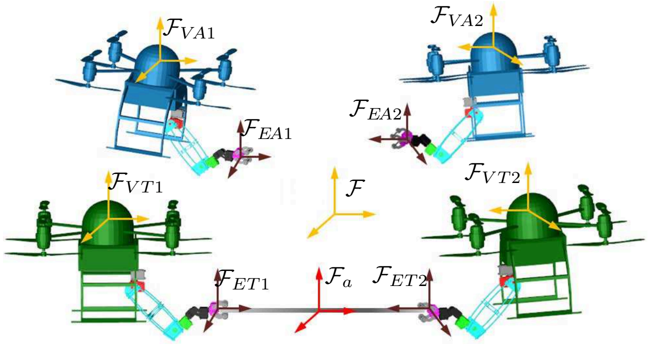


Fig. 1. An example of coordinated multiple vehicle-manipulator system with $N_T = 2$ and $N_A = 2$.

- for the ARs, N_A end-effector frames, $\mathcal{F}_{EAj}$, and N_A frames attached to the vehicle body with origin in the center of mass, $\mathcal{F}_{VAj}$ $(j = 1, \dots, N_A)$;
- the *absolute frame* $\mathcal{F}_a$, whose motion is assigned by the motion planner; this frame describes the absolute motion of the TRs (e.g., an object grasped by the TRs).

A. Kinematics of individual UAVM system

Let $\mathcal{F}_V$ the vehicle-fixed reference frame of an aerial manipulator (the subscripts T and A, as well as the indexes i and j, are dropped for notation compactness). The position of $\mathcal{F}_V$ with respect to the inertial reference frame $\mathcal{F}$ is given by the (3×1) vector $\boldsymbol{p}_V$, while its orientation is given by the rotation matrix $\boldsymbol{R}_V(\boldsymbol{\phi}_V)$ where $\boldsymbol{\phi}_V = [\psi_V \ \theta_V \ \varphi_V]^T$ is the triple of ZYX yaw-pitch-roll angles [13].

Let $\mathcal{F}_E$ be the frame attached to the end-effector of the UAVM, its position and orientation with respect to the common base frame are given by

$$\begin{cases} \boldsymbol{p}_E = \boldsymbol{p}_V + \boldsymbol{R}_V \boldsymbol{p}_{E,V}^V \\ \boldsymbol{R}_E = \boldsymbol{R}_V \boldsymbol{R}_E^V, \end{cases} \tag{1}$$

where the vector $\boldsymbol{p}_{E,V}^V$ and the matrix $\boldsymbol{R}_E^V$ describe the position and the orientation of $\mathcal{F}_E$ with respect to $\mathcal{F}_V$, respectively. The linear, $\dot{\boldsymbol{p}}_E$, and angular, $\boldsymbol{\omega}_E$, velocities of $\mathcal{F}_E$ are obtained by differentiating eq. (1)

$$\begin{cases} \dot{\boldsymbol{p}}_E = \dot{\boldsymbol{p}}_V - \boldsymbol{S}(\boldsymbol{R}_V \boldsymbol{p}_{E,V}^V)\boldsymbol{\omega}_V + \boldsymbol{R}_V \dot{\boldsymbol{p}}_{E,V}^V \\ \boldsymbol{\omega}_E = \boldsymbol{\omega}_V + \boldsymbol{R}_V \boldsymbol{\omega}_{E,V}^V, \end{cases} \tag{2}$$

where $\boldsymbol{S}(\cdot)$ is the (3×3) skew-symmetric matrix operator performing the cross product [14]. Let $\boldsymbol{q}$ be the $(n_M \times 1)$ vector of manipulator joint coordinates; the (6×1) generalized velocity of the end-effector relative to $\mathcal{F}_V$, $\boldsymbol{v}_{E,V}^V = \left[\dot{\boldsymbol{p}}_{E,V}^{V\,\mathrm{T}} \ \boldsymbol{\omega}_{E,V}^{V\,\mathrm{T}}\right]^{\mathrm{T}}$, can be expressed in terms of the joint velocities $\dot{\boldsymbol{q}}$ via the manipulator Jacobian $\boldsymbol{J}_{E,V}^V$, i.e.,

$$\boldsymbol{v}_{E,V}^V = \boldsymbol{J}_{E,V}^V(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3}$$

On the basis of (2) and (3), the generalized end-effector velocity, $\boldsymbol{v}_E = \left[\dot{\boldsymbol{p}}_E^{\mathrm{T}} \ \boldsymbol{\omega}_E^{\mathrm{T}}\right]^{\mathrm{T}}$, can be expressed as

$$\boldsymbol{v}_E = \boldsymbol{G}_V^T(\boldsymbol{R}_V, \boldsymbol{q})\boldsymbol{v}_V + \boldsymbol{J}_{E,V}(\boldsymbol{R}_V, \boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4}$$

where

$$\boldsymbol{G}_V = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{R}_V \boldsymbol{p}_{E,V}^V) & \boldsymbol{I}_3 \end{bmatrix}, \boldsymbol{J}_{E,V} = \begin{bmatrix} \boldsymbol{R}_V & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{R}_V \end{bmatrix} \boldsymbol{J}_{E,V}^V, \tag{5}$$

with $\boldsymbol{I}_m$ and $\boldsymbol{O}_m$ denoting the $(m \times m)$ identity and null matrix, respectively. If the attitude of the vehicle is expressed in terms of yaw-pitch-roll angles, equation (4) becomes

$$\boldsymbol{v}_E = \boldsymbol{G}_V^T(\boldsymbol{q}, \boldsymbol{\phi}_V)\bar{\boldsymbol{T}}(\boldsymbol{x}_V)\dot{\boldsymbol{x}}_V + \boldsymbol{J}_{E,V}(\boldsymbol{R}_V, \boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{x}_V, \boldsymbol{q})\dot{\boldsymbol{\zeta}}, \tag{6}$$

where

$$\boldsymbol{x}_V = \begin{bmatrix} \boldsymbol{p}_V \\ \boldsymbol{\phi}_V \end{bmatrix}, \ \bar{\boldsymbol{T}}(\boldsymbol{x}_V) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}(\boldsymbol{\phi}_V) \end{bmatrix}, \tag{7}$$

$\boldsymbol{T}(\boldsymbol{\phi}_V)$ is the matrix relating angular velocity to yaw-pitch-roll angles rate [13] and

$$\boldsymbol{J} = \left[\boldsymbol{G}_V^T \bar{\boldsymbol{T}}(\boldsymbol{x}_V) \ \boldsymbol{J}_{E,V}\right], \ \boldsymbol{\zeta} = \left[\boldsymbol{x}_V^{\mathrm{T}} \ \ \boldsymbol{q}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{8}$$

In the case of a standard quadrotor-arm system, the vehicle is an under-actuated system, i.e., only 4 independent control inputs are available against the 6 degrees of freedom, the position and the yaw angle are usually the controlled variables, while pitch and roll angles are used as intermediate control inputs for position control. Hence, it is worth rewriting the vector $\boldsymbol{x}_V$ as follows

$$\boldsymbol{x}_V = \left[\boldsymbol{x}_{cV}^{\mathrm{T}} \ \boldsymbol{x}_{uV}^{\mathrm{T}}\right]^{\mathrm{T}}, \boldsymbol{x}_{cV} = \left[\boldsymbol{p}_V^{\mathrm{T}} \ \psi_V^{\mathrm{T}}\right]^{\mathrm{T}}, \boldsymbol{x}_{uV} = [\theta_V \ \varphi_V]^{\mathrm{T}}.$$

Thus, the differential kinematics (6) can be rearranged as

$$\boldsymbol{v}_E = \boldsymbol{J}_c(\boldsymbol{\zeta}_c, \boldsymbol{\zeta}_u)\dot{\boldsymbol{\zeta}}_c + \boldsymbol{J}_u(\boldsymbol{\zeta}_c, \boldsymbol{\zeta}_u)\dot{\boldsymbol{\zeta}}_u, \tag{9}$$

where $\boldsymbol{\zeta}_c$ and $\boldsymbol{\zeta}_u$ are the vectors of controlled and uncontrolled variables, respectively, given by

$$\boldsymbol{\zeta}_c = \left[\boldsymbol{x}_{cV}^{\mathrm{T}} \ \boldsymbol{q}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \boldsymbol{\zeta}_u = \boldsymbol{x}_{uV}, \tag{10}$$

while $\boldsymbol{J}_c$ and $\boldsymbol{J}_u$ are obtained by $\boldsymbol{J}$ as

$$\boldsymbol{J}_c = [\boldsymbol{J}_{cV} \quad \boldsymbol{J}_{E,V}], \quad \boldsymbol{J}_u = \boldsymbol{J}_{uV},$$

with $\boldsymbol{J}_{cV}$ composed by the first 4 columns of $\boldsymbol{G}_V^T \bar{\boldsymbol{T}}$ (i.e., the columns related to $\boldsymbol{x}_{cV}$) and $\boldsymbol{J}_{uV}^T$ composed by the last 2 columns of $\boldsymbol{G}_V^T \bar{\boldsymbol{T}}$ (i.e., the columns corresponding to $\boldsymbol{x}_{uV}$).

III. Control Architecture

The proposed control architecture is a partially decentralized scheme, based on the following three-layer hierarchical architecture:

- The first layer is centralized, i.e., it needs to communicate with the motion planner of the whole team as well as with the controller of each individual UAVM. It receives, from the motion planner, the desired behavior of the whole system, in terms of suitable variables describing the coordinated task, and computes the corresponding reference trajectories for each aerial manipulator.
- The second layer is local to each aerial manipulator: it receives the motion references from the upper layer and computes the reference motion in terms of the controlled variables, $\boldsymbol{\zeta}_c$, via an inverse kinematic algorithm with redundancy resolution, based on the NSB paradigm.
- The third layer is local to each UAVM as well. In this layer, the motion control law for the aerial manipulator is computed, so as to guarantee that the motion references computed by the second layer are tracked as accurately as possible.

IV. First layer: multi-vehicle coordinated control

This layer is in charge of computing the reference motion, in the Cartesian space, of each UAVM end-effector, on the basis of the given coordinate task. A task for such a multi-robot system could be specified by assigning the desired trajectories of the following set of variable:

- the absolute motion, which can be expressed in terms of position, $\boldsymbol{p}_a$, and orientation, $\boldsymbol{R}_a$, of the absolute frame, $\mathcal{F}_a$, with respect to the base frame $\mathcal{F}$;
- the relative motion, which is described by the relative position and orientation of each TR end-effector frame, $\mathcal{F}_{ETi}$ ($i = 1, \ldots, N_T$), with respect to $\mathcal{F}_a$, i.e.,
$$\begin{cases} \boldsymbol{p}^a_{Ei,a} = \boldsymbol{R}^T_a(\boldsymbol{p}_{Ei} - \boldsymbol{p}_a) \\ \boldsymbol{R}^a_{Ei} = \boldsymbol{R}^T_a \boldsymbol{R}_{Ei} \end{cases} \qquad i = 1, \ldots, N_T, \quad (11)$$
where the superscript a denotes that all quantities have been referred to the absolute frame. For example, when the TRs grasp a rigid object in a rigid way, the relative variables, $\boldsymbol{p}^a_{Ei,a}$ and $\boldsymbol{R}^a_{Ei}$, have to be kept constant, according to the grasp geometry;
- the task of the ARs could be specified by assigning the relative trajectories between each AR tool frame, $\mathcal{F}_{EAj}$, and the absolute frame, $\mathcal{F}_a$,
$$\begin{cases} \boldsymbol{p}^a_{Ej,a} = \boldsymbol{R}^T_a(\boldsymbol{p}_{Ej} - \boldsymbol{p}_a) \\ \boldsymbol{R}^a_{Ej} = \boldsymbol{R}^T_a \boldsymbol{R}_{Ej} \end{cases} \qquad j = 1, \ldots, N_A, \quad (12)$$
where all quantities have been referred to the absolute frame.

The motion planner provides the desired trajectories for all the variables described above ($i=1, \ldots, N_T$, $j=1, \ldots, N_A$) in terms of position, orientation and velocities ($* = i, j$)

$$\boldsymbol{p}_{a\,d},\ \boldsymbol{R}_{a\,d},\ \dot{\boldsymbol{p}}_{a\,d},\ \boldsymbol{\omega}_{a\,d},\ \boldsymbol{p}^a_{E*,a\,d},\ \boldsymbol{R}^a_{E*\,d},\ \dot{\boldsymbol{p}}^a_{E*,a\,d},\ \boldsymbol{\omega}^a_{E*,a\,d}.$$

In normal operating conditions, this layer simply computes the reference motion of each end-effector, from (11) and (12), both in terms of position and orientation

$$\begin{cases} \boldsymbol{p}_{E*\,d} = \boldsymbol{p}_{a\,d} + \boldsymbol{R}_{a\,d}\boldsymbol{p}^a_{E*,a\,d} \\ \boldsymbol{R}_{E*\,d} = \boldsymbol{R}_{a\,d}\boldsymbol{R}^a_{E*\,d}, \end{cases} \quad (13)$$

and velocities

$$\begin{cases} \dot{\boldsymbol{p}}_{E*\,d} = \dot{\boldsymbol{p}}_{a\,d} - \boldsymbol{S}(\boldsymbol{R}_{a\,d}\boldsymbol{p}^a_{E*,a\,d})\boldsymbol{\omega}_{a\,d} + \boldsymbol{R}_{a\,d}\dot{\boldsymbol{p}}^a_{E*,a\,d} \\ \boldsymbol{\omega}_{E*\,d} = \boldsymbol{\omega}_{a\,d} + \boldsymbol{R}_{a\,d}\boldsymbol{\omega}^a_{E*\,d}, \end{cases} \quad (14)$$

where $* = \{i, j\}$, $i = 1, \ldots, N_T$ and $j = 1, \ldots, N_A$.

In this layer the case of avoidance of unexpected obstacles (i.e., obstacles not considered at the motion planning level) is properly handled. If an unexpected obstacle is detected, the whole team of UAVMs has to avoid the obstacle while keeping the formation. To this aim, the desired absolute trajectory $\boldsymbol{p}_{a\,d}$ must be replaced by a modified trajectory, $\bar{\boldsymbol{p}}_{a\,d}$, computed by resorting to the NSB control approach [10], via the task function

$$\sigma_o(\boldsymbol{p}_a) = \frac{1}{2}\left\|\boldsymbol{p}_a - \boldsymbol{p}_o\right\|^2, \quad (15)$$

where $\boldsymbol{p}_o \in \mathbb{R}^3$ is the position of the obstacle. The modified absolute frame trajectory is obtained by projecting onto the null space of the Jacobian of the task σ_o, $\boldsymbol{J}_o = (\boldsymbol{p}_a - \boldsymbol{p}_o)^{\mathrm{T}} \in \mathbb{R}^{1\times 3}$, the desired path tracking

$$\dot{\bar{\boldsymbol{p}}}_{a\,d} = k_o\boldsymbol{J}^\dagger_o(\boldsymbol{\sigma}_{o\,d} - \boldsymbol{\sigma}_o(\bar{\boldsymbol{p}}_{a\,d})) + \boldsymbol{N}_{J_o}(\dot{\boldsymbol{p}}_{a\,d} + k_{pa}(\boldsymbol{p}_{a\,d} - \bar{\boldsymbol{p}}_{a\,d})), \quad (16)$$

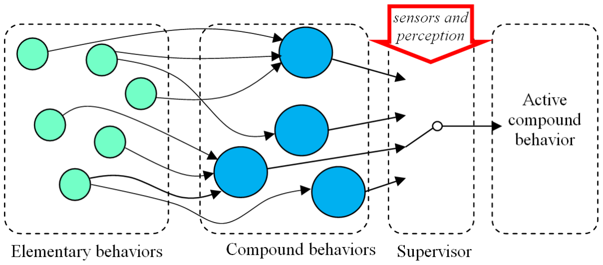


Fig. 2. Sketch illustrating the relationship between elementary, compound behaviors and supervisor.

where $\boldsymbol{J}^\dagger_o = \boldsymbol{J}^{\mathrm{T}}_o\left(\boldsymbol{J}_o\boldsymbol{J}^{\mathrm{T}}_o\right)^{-1}$ is a right pseudo-inverse of $\boldsymbol{J}_o$, $\boldsymbol{N}_{J_o} = \boldsymbol{I}_3 - \boldsymbol{J}^\dagger_o\boldsymbol{J}_o$ is a projector onto the null space of $\boldsymbol{J}_o$, $\boldsymbol{\sigma}_{o\,d}$ is a given desired value for the task function (suitably chosen to ensure the appropriate level of safety of the system), k_o and k_{pa} are constant gains.

Once the modified trajectory is computed, the motion references for the transporting and auxiliary robots are defined via (13), in such a way to move the whole system away from the obstacle and, at same time, keep the relative motion.

V. Second layer: aerial manipulator coordinated control

The second layer, local to each UAVM, is in charge of computing the reference trajectories for the controlled variables $\boldsymbol{\zeta}_c$, i.e., the manipulator joints as well as the position and the yaw angle of the vehicle. The NSB control approach has been adopted to handle the kinematic redundancy of the system and fulfill multiple tasks arranged in a priority order.

Therefore, also the coordinated control of the individual UAVM can be seen as a three-level scheme (see Fig. 2):

- *Elementary behaviors*: the atomic task functions to be controlled at kinematic level.
- *Compound behaviors*: combinations of several elementary behaviors, arranged in a given priority order.
- *Supervisor*: in charge of switching between the defined compound behaviors, on the basis of the state of the multi-UAVMs system.

A. Elementary behaviors

A specific elementary behavior for a UAVM can be analytically described through a task variable $\boldsymbol{\sigma} \in \mathbb{R}^m$ to be controlled. Let $\boldsymbol{f}$ be the configuration-dependent task function, that represents the relationship between the task variable and the state vector $\boldsymbol{\zeta}$, defined in (8), i.e., $\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{\zeta})$. The task Jacobian matrix $\boldsymbol{J}_\sigma \in \mathbb{R}^{m\times(6+n_M)}$ is defined as

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{f}(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\dot{\boldsymbol{\zeta}} = \boldsymbol{J}_\sigma(\boldsymbol{\zeta})\dot{\boldsymbol{\zeta}} = \boldsymbol{J}_{c,\sigma}(\boldsymbol{\zeta})\dot{\boldsymbol{\zeta}}_c + \boldsymbol{J}_{u,\sigma}(\boldsymbol{\zeta})\dot{\boldsymbol{\zeta}}_u, \quad (17)$$

where $\boldsymbol{J}_{c,\sigma} \in \mathbb{R}^{m\times(4+n_M)}$ is obtained by merging the first 4 columns with the last n_M columns of $\boldsymbol{J}_\sigma$, while $\boldsymbol{J}_{u,\sigma}\mathbb{R}^{m\times(2+n_M)}$ is given by extracting the 5th and 6th column of $\boldsymbol{J}_\sigma$.

Let $\boldsymbol{\sigma}_d$ be the desired value for the task variable, output by the first layer. The motion references, on the basis of (17), are computed via a closed-loop inverse kinematics algorithm with compensation of not actuated motion variables [13] as

$$\dot{\boldsymbol{\zeta}}_{c,r} = \boldsymbol{J}^\dagger_{c,\sigma}(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}} - \boldsymbol{J}_{u,\sigma}\dot{\boldsymbol{\zeta}}_u), \quad (18)$$

Preprint version of the paper: G. Muscio, F. Pierri, M.A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, A. Ollero, "Experiments on coordinated motion of aerial robotic manipulators." IEEE International Conference on Robotics and Automation, pp. 1224-1229, 2016. DOI: 10.1109/ICRA.2016.7487252.

where $\boldsymbol{J}^{\dagger}_{c,\sigma} = \boldsymbol{J}^{\mathrm{T}}_{c,\sigma}\left(\boldsymbol{J}_{c,\sigma}\boldsymbol{J}^{\mathrm{T}}_{c,\sigma}\right)^{-1}$ is a right pseudo-inverse of $\boldsymbol{J}_{c,\sigma}$, $\boldsymbol{\Lambda}$ is a positive-definite matrix of gains and $\widetilde{\boldsymbol{\sigma}}$ is the task error. The pseudo-inverse in (18) requires that $\boldsymbol{J}_{c,\sigma}$ is full rank, i.e., $\mathrm{rank}(\boldsymbol{J}_{\sigma}(\boldsymbol{\zeta})) = m$, and $m \leq (4 + n_M)$.

Several elementary behaviors can be defined [15]; they can be roughly classified into two categories: the behaviors related to the control of the manipulator (e.g., position and/or orientation of the end-effector frame, tasks aimed at keeping the arm in a particular configuration or avoiding the violation of constraints, such as the mechanical joint limits) and those related to the control of the vehicle (mostly tasks aimed at avoiding the collision between UAVs and obstacles or neighboring robots).

B. Compound behaviors

In case the DOFs of the aerial manipulator are more than those required by the task function, the system is kinematically redundant and the redundant DOFs can be exploited to achieve secondary tasks, by resorting to a task-priority approach, such as the NSB control [10].

The overall system velocity is obtained by properly merging the velocity vectors computed for each behavior as if it is acting alone; then, a lower-priority behavior is projected onto the null space of the Jacobian of the higher-priority behaviors so as to remove the conflicting velocities components. Thus, the overall system velocity is given by

$$\dot{\boldsymbol{\zeta}}_r = \dot{\boldsymbol{\zeta}}_1 + \sum_{k=2}^{N_b} \boldsymbol{N}_{1,k-1}\dot{\boldsymbol{\zeta}}_k, \tag{19}$$

where the subscript k denotes the task priority level, N_b is the number of behaviors to be fulfilled, $\boldsymbol{N}_{1,k} = \boldsymbol{I} - \boldsymbol{J}^{\dagger}_{1,k}\boldsymbol{J}_{1,k}$ is a projector onto the null space of the augmented Jacobian $\boldsymbol{J}_{1,k}$, defined as $\boldsymbol{J}_{1,k} = \left[\begin{array}{cccc} \boldsymbol{J}^{\mathrm{T}}_1 & \boldsymbol{J}^{\mathrm{T}}_2 & \ldots & \boldsymbol{J}^{\mathrm{T}}_k \end{array}\right]^{\mathrm{T}}$.

To accomplish complex missions, the elementary behaviors can be hierarchically combined in *compound behaviors*. The priority order between elementary behaviors depends on practical consideration (e.g., safety behaviors as obstacles avoidance have always higher priority). Not all the elementary behaviors can be combined with each others, since some of them are not compatible; in such a case the lower priority tasks will be not achieved at all. A rigorous analysis about the compatibility can be found in [16], where it is proven that the solution (19) gives rise to stable and convergent error dynamics under mild conditions on the Jacobians.

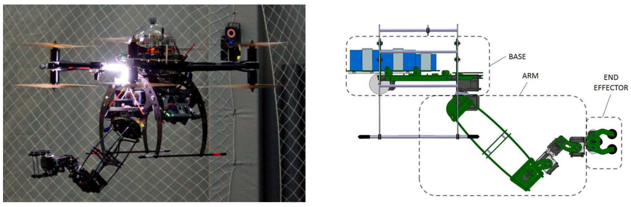
A supervisor is in charge of monitoring the current state of both the mission and the robotic system, and choosing, via real-time switching, the suitable compound behavior to be activated. Thus, smooth switching laws should be designed to ensure continuity of the velocity commands for the third layer (see, e.g., [17]).

VI. Third layer: low-level control

Once the motion references for the controlled variables, $\boldsymbol{\zeta}_{c,r}$, have been determined by the second layer, any kind of motion controller can be adopted to ensure motion tracking. In this paper, the control architecture exploited for the experimental campaign is described in Section VII.

VII. Case Study

The effectiveness of the proposed architecture has been proven via experimental tests on the ARCAS (Aerial Robotics Cooperative Assembly System project) setup [12] consisting of two aerial manipulators equipped with 6 DOFs robotic arms involved in coordinated motion tasks.



(a) (b)

Fig. 3. ARCAS indoor setup: (a) aerial platform, (b) manipulator.

The ARCAS multicopters are eight rotor aircrafts in coaxial configuration with a tip-to-tip wingspan of 105 cm, 13 inches propellers, height of 50 cm and mass of 8.2 kg, including the Lithium Polymer batteries and the robotic arm, as shown in Fig.3(a). Each aerial platform is equipped with a 6-DOFs robotic arm [18] with all revolute joints and a gripper mounted on the end-effector, as depicted in Fig. 3(b).

The ARCAS platform counts two processing units: an autopilot, developed by CATEC (Centro Avanzado de Tecnologas Aeroespaciales), and on-board computer, integrated into a common framework structured on the following levels:

- Control level: this level, running on the autopilot, includes the integration of the control algorithms for the aerial platform and the robotic arm. It has been developed using a Model-Based Design (MBD) methodology [19], based on Simulink code generation tools that have proved to be very reliable, fast and convenient.
- Functional level: this level integrates the perception and cooperation algorithms that run on-board the aerial robot. A Linux processing unit, namely an i7 Asctec Mastermind with ROS [20], is used as the framework for the integration of the different functionalities.
- Multi-vehicle level: this level includes the integration of the software modules that require the information from multiple vehicles.

At the Control level, a special control architecture has been developed [21]. The aerial manipulator controller is challenging since the arm weight corresponds to 17% of the total mass, which means that a significant part of the weight is moving while flying. The control architecture is composed by 4 modules. A specific module has been designed for the robotic arm, while the remaining ones are standard for multirotor control, suitably modified to be adapted to the ARCAS platforms. A moving tray moves the batteries in order to compensate the center of mass displacements. The key Controller module is the robotic arm compensator [21], which is located inside the Attitude Controller Module and modifies the position of the batteries tray, the thrust and torques references according to the arm state.

The ROS middleware is used to handle the communication among the functional modules as well as the multi-vehicle communication. The proposed algorithms have been developed in C++ under ROS environment and were running at 50 Hz receiving, as feedback data, platforms position and attitude as well as joint positions of both the arms. The ROS master was running on the on-board PC of first UAVM (UAVM1), together with the first layer, computing motion references of both the UAVMs' end-effectors, and the two lower layers modules; the second UAVM (UAVM2), hosted only its own second and third layer controllers. Vehicle positions have been provided at 100 Hz by the Vicon system, while the attitude is obtained through the IMU.

A. Behaviors

In the experimental campaign four elementary behaviors have been considered: End-Effector Configuration (EEC), Inter-Vehicle distance (IV), relative Field of View (FoV) and Mechanical Joint Limit avoidance (MJL).

1) End Effector Configuration: The objective is that of tracking a desired end-effector trajectory, both in terms of position and orientation. The task function can be stated as $\boldsymbol{\sigma}_{EEC} = [\boldsymbol{p}_E \ \mathcal{Q}_E] \in \mathbb{R}^7$, where index i has been dropped for notation compactness, $\mathcal{Q}_E$ is the unit quaternion extracted by the rotation matrix $\boldsymbol{R}_E$ [14] . The task Jacobian is the matrix $\boldsymbol{J}$ defined in (8), namely $\boldsymbol{J}_{EEC} = \boldsymbol{J} \in \mathbb{R}^{6\times 6+n_M}$. In details, the inverse kinematics (18) can be particularized as

$$\dot{\boldsymbol{\zeta}}_{c,r} = \boldsymbol{J}_c^{\dagger}(\dot{\boldsymbol{v}}_{E,d} + \boldsymbol{\Lambda}_{EEC}\tilde{\boldsymbol{\sigma}}_{EEC} - \boldsymbol{J}_u\dot{\boldsymbol{\zeta}}_u), \tag{20}$$

where the task error $\tilde{\boldsymbol{\sigma}}_{EEC}$ can be computed as [14]

$$\tilde{\boldsymbol{\sigma}}_{EEC} = \begin{bmatrix} \boldsymbol{e}_P \\ \boldsymbol{e}_O \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{E,d} - \boldsymbol{p}_E(\boldsymbol{\zeta}) \\ \tilde{\boldsymbol{\epsilon}}(\boldsymbol{\zeta}) \end{bmatrix}, \tag{21}$$

where $\tilde{\boldsymbol{\epsilon}}$ is the vectorial part of the unit quaternion extracted from the mutual orientation matrix $\boldsymbol{R}_{Ed}\boldsymbol{R}_E^{\mathrm{T}}(\boldsymbol{\zeta})$.

2) Inter-Vehicle distance: In order to avoid collisions between UAVMs, it could be useful to keep a safety distance between each couple of UAVMs. This behavior is characterized by the task function $\sigma_{IV_{ij}} = \|\boldsymbol{p}_{V_i} - \boldsymbol{p}_{V_j}\|^2 \in \mathbb{R}$, $\forall\,(i,j) \in \{1,\ldots,N\} \times \{1,\ldots,N\}$, $i \neq j$. The Jacobian matrices for the two UAVMs are given by

$$\boldsymbol{J}_{IV_i} = -\boldsymbol{J}_{IV_j} = 2(\boldsymbol{p}_{V_i} - \boldsymbol{p}_{V_j})^{\mathrm{T}}[\boldsymbol{I}_3 \ \boldsymbol{O}_{3\times 3+n_M}] \in \mathbb{R}^{3\times 6+n_M}.$$

The desired value for the task function is the square of the safety distance, determined in order to keep the vehicles safe and so as to minimize the mutual aerodynamics disturbances.

3) Relative Field of View: Directional devices or sensors mounted on the end-effector, such as, e.g., a laser or a videocamera, do not need to constrain the whole end-effector orientation, but only the outgoing unit vector, which is required to point toward a given location $\boldsymbol{p}_o \in \mathbb{R}^3$. The task function can be defined as follows $\boldsymbol{\sigma}_{FoV} = [\sigma_{FoV1} \ \sigma_{FoV2}]^{\mathrm{T}} \in \mathbb{R}^2$

$$\sigma_{FoVk} = \pi/2 - \arccos\left(r_k^E/\|\boldsymbol{r}^E\|\right), \ k = 1, 2\,, \tag{22}$$

where $\boldsymbol{r}^E = \boldsymbol{R}_E^{\mathrm{T}}(\boldsymbol{p}_E - \boldsymbol{p}_o) \in \mathbb{R}^3$. From (22), the task jacobian, $\boldsymbol{J}_{FoV} = [\boldsymbol{J}_{FoV1}^{\mathrm{T}} \ \boldsymbol{J}_{FoV2}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2\times 6+n_M}$, is given by

$$\boldsymbol{J}_{FoVk} = \frac{\boldsymbol{e}_k^{\mathrm{T}}}{\sqrt{\|\boldsymbol{r}^E\|^2 - |r_k^E|^2}}\left[\boldsymbol{P}_r^{\perp}\boldsymbol{R}_E^{\mathrm{T}} \quad \boldsymbol{S}(\boldsymbol{r}^E)\right]\boldsymbol{J}, \tag{23}$$

where $\boldsymbol{P}_r^{\perp} = \boldsymbol{I}_3 - \boldsymbol{r}^E(\boldsymbol{r}^E)^{\mathrm{T}}/\|\boldsymbol{r}^E\|^2 \in \mathbb{R}^{3\times 3}$, r_k^E is the the kth component of $\boldsymbol{r}^E$, $\boldsymbol{e}_k$ is the kth unit vector of the canonical base spanning $\mathbb{R}^3$ and $\boldsymbol{J}$ is the jacobian in (8).

4) Mechanical Joint Limit Avoidance: Any manipulator exhibits mechanical limits for the joint mobility. It might be appropriate to define a task that avoids the violation of those limits. In the literature, a number of joint limit task functions could be found, in this paper the following choice has been considered [15] $\boldsymbol{\sigma}_{MJL} = \sum_{i=1}^{n_M} l_i$ where

$$l_i(q_i) = \begin{cases} (\underline{q}_i - q_i)^2/(2n_M), & \text{if} \ \ q_i \leq \underline{q}_i, \\ 0, & \text{if} \ \ \underline{q}_i < q_i \leq \overline{q}_i, \\ (\overline{q}_i - q_i)^2/(2n_M), & \text{if} \ \ q_i > \overline{q}_j, \end{cases}$$

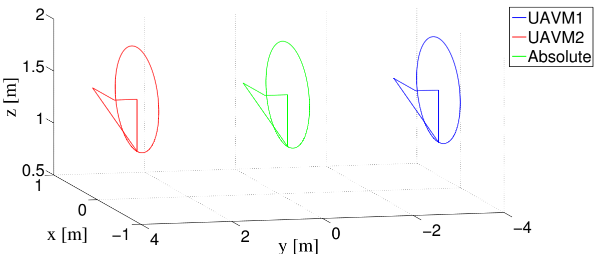


Fig. 4. Desired trajectories involved in the coordinated motion task: absolute motion (green line) UAVM1 and UAVM2 end-effector motions (blue and red lines respectively).

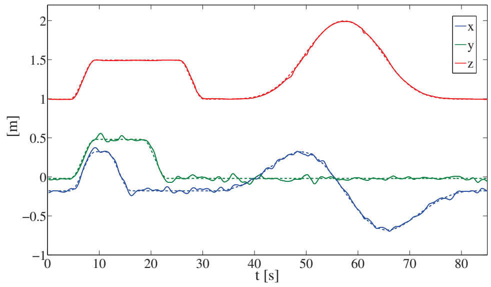


Fig. 5. Comparison between the desired absolute motion (dashed lines) and the effective ones (solid lines).

where $\overline{q}_j$ and $\underline{q}_j$ are the lower and upper joint limit respectively. The task Jacobian is $\boldsymbol{J}_{MJL} = [\boldsymbol{0}_{1\times 6} \ \boldsymbol{J}_l] \in \mathbb{R}^{1\times 6+n_M}$ with $\boldsymbol{J}_{l_i} = \partial l_i/\partial q_i$ $(i = 1, \ldots, n_M)$.

The defined elementary behaviors have been combined into three compound behaviors: EEC+MJL, FoV, IV+FoV. The ECC and MJL behaviors are not fully compatible: in executing EEC+MJL the priority is given to ECC and MJL acts in such a way to prevent joint limit violations. As concerns FoV and IV, they are fully compatible since the first one constraints the end-effector orientation while the second one constraints only vehicles' relative position.

VIII. Experimental Results

Two experiments have been performed on the ARCAS setup. In the first one, a desired motion is assigned to the absolute frame, located at centroid of the two end-effectors. In the second experiment, UAVM1 is required to move toward UAVM2, which, in turn, is in charge of pointing a camera, mounted on its end-effector, at the other one. Moreover, the supervisor monitors the distance between the two vehicles and, if it becomes lower than a defined threshold, switches to a safer compound behavior (IV+FoV).

A. First Experiment

The trajectory depicted in Fig. 4 is assigned to the absolute frame: a straight line displacements of 0.5 m along each axis, then a movement of −0.5 m along the three axes separately and finally a circle of 0.5 m radius in the $x - z$ plane. The relative motion variables have been kept constant $\boldsymbol{p}_{E1,a\,d}^a = [0 \ -3 \ 0]^{\mathrm{T}}$ m, $\boldsymbol{p}_{E2,a\,d}^a = [0 \ 3 \ 0]^{\mathrm{T}}$ m, $\boldsymbol{R}_{E1\,d}^a = \boldsymbol{R}_{E2\,d}^a = \boldsymbol{I}_3$. Figure 5 depicts the comparison between desired and effective absolute frame motion components: some small oscillations are present, especially along x and y direction, mainly due to aerodynamics interaction. The maximum tracking error, in Fig. 6, is about 8 cm in transient phase and around 4 cm at steady state. As concerns the orientation error, quaternion representation (21) has been adopted, and it is always below 0.06.

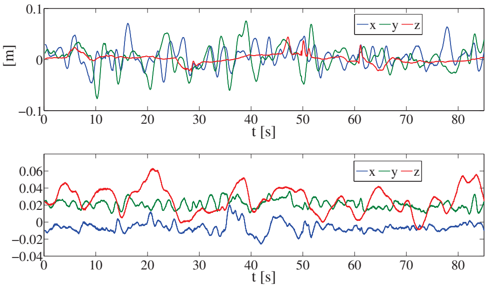


Fig. 6. Error between the desired absolute motion and the effective one both in terms of position (top) and orientation (bottom).

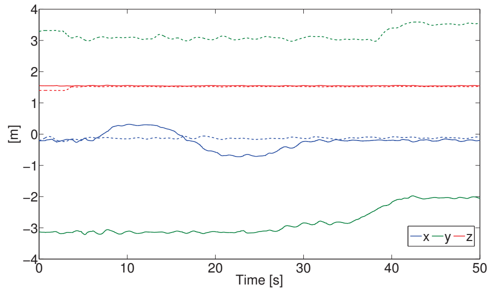


Fig. 7. End-effector trajectories of UAVM1 (solid lines) and UAVM2 (dashed lines).

B. Second Experiment

In the second experiment, UAVM1 is in charge of tracking a given trajectory, while UAVM2, which plays the role of AR, is tasked of pointing a camera toward the UAVM1's end-effector. More in detail, UAVM1 and UAVM2 end-effectors move according to trajectories depicted in Fig. 7; starting from 38s UAVM1 moves toward UAVM2 along the y axis and the mutual distance becomes lower than the safety one, set to 6 m since the vehicle-vehicle airflow disturbances are magnified in an indoor environment. Thus, it can be seen in Fig.7 that UAVM2 moves along the y direction and keeps the safety distance. Figure 8 reports the mutual distance, which reaches the minimum at about 38 s: as soon as the safety distance constraint is violated, UAVM2, which was only in Field of View (FoV) mode, reacts by moving backward, still keeping the end-effector frame in the desired orientation. This is automatically done by the supervisor, which switches between the compound behaviors FoV and IV+FoV.

In order to measure the performance of the Field of View task, let us define an opportune index given by the projection of the z axis of UAVM2 end-effector frame onto the vector joining the two end-effectors, $\boldsymbol{r}^{E2} = \boldsymbol{R}_{E2}^{\mathrm{T}}(\boldsymbol{p}_{E1} - \boldsymbol{p}_{E2})$, i.e.,

$$\mathcal{I}_{FoV} = [0\ 0\ 1]\,\boldsymbol{r}^{E2}/\|\boldsymbol{r}^{E2}\|. \tag{24}$$

The time history of such index is shown in Fig. 8: the experiment starts with the two vectors orthogonal (i.e., an initial value of $\mathcal{I}_{FoV} \simeq 0$ is experienced), then, the desired FoV is reached and kept during the entire flight (i.e., $\mathcal{I}_{FoV} \simeq 1$).

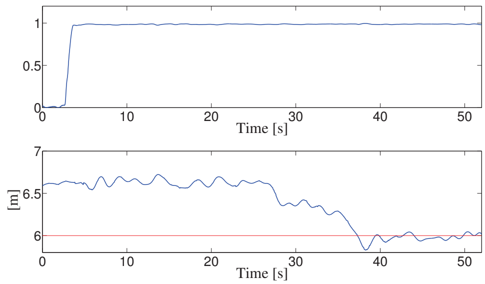


Fig. 8. Field of view performance index (top) and distance between the two UAV (bottom).

REFERENCES

[1] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modelling, estimation and control for aerial grasping and manipulation," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 2668–2673.

[2] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Keemink, S. Stramigioli, R. Carloni, and L. Marconi, "Developing an aerial manipulator prototype: Physical interaction with the environment," *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 41–50, 2014.

[3] K. Kondak, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach, M. Laiacker, I. Maza, A. Rodriguez-Castano, and A. Ollero, "Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks," *International Journal of Advanced Robotic Systems*, vol. 10, no. 145, pp. 1–9, 2013.

[4] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[5] I. Maza, K. Kondak, M. Bernard, and A. Ollero, "Multi-UAV cooperation and control for load transportation and deployment," *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 417–449, 2010.

[6] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized centroid and formation control for multi-robot systems," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 3511–3516.

[7] S. Nestinger and H. Cheng, "Mobile-R: A reconfigurable cooperative control platform for rapid deployment of multi-robot systems," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 52–57.

[8] J. Gancet, G. Hattenberger, R. Alami, and S. Lacroix, "Task planning and control for a multi-UAV system: architecture and algorithms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 1017–1022.

[9] G. Antonelli, K. Baizid, F. Caccavale, G. Giglio, and F. Pierri, "CAVIS: a control software architecture for cooperative multi-unmanned aerial vehicle-manipulator systems," in *19th World Congress The International Federation of Automatic Control*, August 2014, pp. 1108–1113.

[10] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The NSB control: a behavior-based approach for multi-robot systems," *Paladyn Journal of Behavioral Robotics*, vol. 1, no. 1, pp. 48–56, 2010.

[11] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on behavioral coordinated control of an unmanned aerial vehicle manipulator system," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4680–4685.

[12] ARCAS, "Aerial Robotics Cooperative Assembly System," 2010. [Online]. Available: http://www.arcas-project.eu

[13] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri, "Control of quadrotor aerial vehicles equipped with a robotic arm," in *Proc. of 21th Mediterranean Conference on Control and Automation*, 2013, pp. 1174–1180.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics – Modelling, Planning and Control*. London, UK: Springer, 2009.

[15] G. Antonelli, K. Baizid, F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Control software architecture for cooperative multiple unmanned aerial vehicle-manipulator systems," *Journal of Software Engineering for Robotics*, pp. 123–134, 2014.

[16] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic system," *IEEE Transactions on Robotics*, vol. 25, pp. 985–994, 2009.

[17] F. Caccavale, V. Lippiello, G. Muscio, F. Pierri, F. Ruggiero, and L. Villani, "Grasp planning and parallel control of a redundant dual-arm/hand manipulation system," *Robotica*, vol. 31, pp. 1169–1194, 2013.

[18] R. Cano, C. Pérez, F. Pruaño, A. Ollero, and G. Heredia, "Mechanical design of a 6-DOF aerial manipulator for assembling bar structures using UAVs," in *2nd RED-UAS 2013 Workshop on Research, Education and Development of Unmanned Aerial Systems*, 2013.

[19] D. Santamaría, F. Alarcón, A. Jiménez, A. Viguria, M. Béjar, and A. Ollero, "Model-based design, development and validation for uas critical software," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 103–114, 2012.

[20] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[21] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Perez, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor UAVs equipped with a servo robot arm," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4014–4020.