

Coordinated control of aerial robotic manipulators: theory and experiments

G. Muscio¹, F. Pierri¹, M. A. Trujillo², E. Cataldi³, G. Antonelli³,
F. Caccavale¹, A. Viguria², S. Chiaverini³ and A. Ollero⁴

Abstract—This paper presents a three-layer control architecture for coordinated control of multiple aerial manipulators (UAVMs): the centralized top layer plans the end-effector desired trajectories of each UAVM; the middle layer, local to each vehicle, computes the corresponding motion references; the bottom layer is a low level dynamic motion controller, which tracks the motion references. At second layer, the overall mission is hierarchically decomposed in a set of elementary behaviors, which are combined together, via the Null Space-based Behavioral approach, into more complex compound behaviors. For each UAVM, the suitable compound behavior to be executed is selected by a supervisor. The proposed framework has been tested through an experimental campaign.

I. INTRODUCTION

Multiple mobile robots have raised the interest of the scientific community due to their wide application domain as well as their flexibility and capacity to accomplish complex tasks impossible for single robots. Among mobile robots, Unmanned Aerial Vehicles (UAVs) have been involved in a wide number of applications, including aerial manipulation, in which UAVs are equipped with grippers or, more recently, multi-joints robotic manipulators (UAVMs, Unmanned Aerial Vehicle-Manipulator systems) [1]. Design and control of aerial manipulators are handled in [2], where the design, modeling and control of a quadrotor helicopter endowed with a 3-DOFs delta structure and a 3-DOFs end-effector, is proposed, or in [3] where a quadrotor vehicle with a 2-DOFs manipulator is adopted for an experimental validation of an adaptive control. In [4] a behavioral control for a multi-rotor vehicle with a 6-DOFs arm is presented and experimentally validated.

Since aerial manipulators present limitations in terms of payload, team of UAVMs can be adopted to carry out complex missions such as, e.g., cooperative transportation of large and/or heavy payloads [5] and cooperative assembly of structures in remote or hazardous environments. Several frameworks for multi-robot aerial systems have been proposed in literature: [6] proposes a decisional architecture for multi-UAVs systems, that uses different control schemes depending on the status of the current task, in [7] coordination of swarms of UAVs is achieved via a fuzzy control methodology, and in [8] a multi-layer control scheme is proposed to guide a formation

of three UAVs in trajectory tracking missions. Multiple robots allow to achieve complex missions by assigning the single sub-tasks to different vehicles. Distributed task allocation for a multi-UAV sensor network is dealt with in [9], where each UAV distributes its remaining tasks dynamically to teammates based on its own resource availability. Multi-priority control has been exploited for floating-base manipulation in the last years: experimental results have been obtained for underwater systems in grasping operations [10] with a system characterized by 13 degrees of freedom taking into account several prioritized tasks run by means of proper activating functions. In [11], a control framework for multiple UAVMs, in which the Null-Space based Behavioral (NSB) control [12] is exploited for fulfilling multiple tasks in complex missions is proposed.

In this paper, a three-layers coordinated control for multiple UAVMs is presented and experimentally validated. The first layer is centralized, it communicates with every vehicles and generates the desired trajectories of each end-effector; the second layer, local to each vehicle, computes the motion references; the last layer is a motion controller ensuring the tracking of the previous layer outputs. At the second layer, the overall mission is hierarchically decomposed in a set of basic sub-tasks, called *elementary behaviors*, which are combined together, in a prioritized way, into more complex tasks, called *compound behaviors*, by exploiting the NSB paradigm. The compound behavior to be assigned to each UAVM is selected via a local supervisor, designed via a Finite State Automata. Experiments have been conducted on a test-bed composed of two multi-rotor aerial platforms equipped with a 6 DOFs manipulator, developed by CATEC (Centro Avanzado de Tecnologías Aeroespaciales) within the EU-funded project ARCAS (Aerial Robotics Cooperative Assembly System, <http://www.arcas-project.eu>), aimed at developing cooperative free-flying robot system for assembly and structure construction. Developing a framework for multi-UAVMs coordination is a necessary step toward the goal of cooperative load transportation and structure assembly, since only a strict coordination together with an accurate motion control could allow to handle rigid object with rigid contact. At our best knowledge, this is the first time that a complete control framework for coordinated motion control of multiple aerial robots has been proposed and experimentally tested. A coordinated motion control framework for a single UAVM is presented in [4], where a preliminary version of the two lowest layers used in this paper is developed. In [13], the coordinated control for multiple UAVMs has been sketched, while in this paper the design of the local supervisor is formalized and more technical details, regarding, e.g. the smooth transition

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

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

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

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

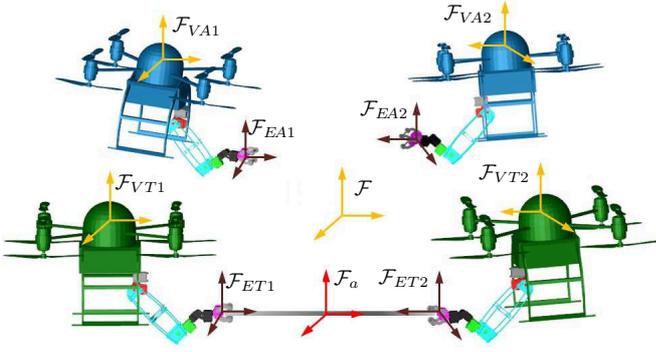


Fig. 1. Multiple aerial manipulator system with $N_T = 2$ and $N_A = 2$.

between different compound behaviors, are reported. Although the proposed framework is partly based on already established techniques in robotics, such techniques have been modified for the case of aerial manipulators, e.g. by properly taking into consideration the underactuation of the floating base.

II. MODELING

Let us consider a team of N aerial manipulators (UAVMs) including two sets of robots:

- N_T UAVMs, referred as *Transporting Robots* (TRs), moving in a coordinated fashion, for example, in order to cooperatively transport an object. The proposed task formulation does not strictly require the presence of a physical object grasped by TRs, e.g. in the case in which simple coordination of end-effector motion is required;
- N_A UAVMs, acting as *Auxiliary Robots* (ARs), whose motion has to be coordinated with that of the TRs in order to perform auxiliary tasks. For instance, an AR could be equipped with a camera pointing at a given location.

Each robotic arm has n_{Mi} (n_{Mj}) DOFs (hereafter $i = 1, \dots, N_T$ and $j = 1, \dots, N_A$), while each flying platform has n_{Vi} (n_{Vj}) actuated DOFs (in the case of standard quadrotors $n_{Vi} = n_{Vj} = 4$). Hence, each aerial robotic manipulator has $n_i = n_{Mi} + n_{Vi}$ ($n_j = n_{Mj} + n_{Vj}$) DOFs. A task in the operational space may require at most 6 DOFs. In the following, it is assumed that $n_i \geq 6$ ($n_j \geq 6$), i.e. each aerial manipulator is ensured to be, at least, a fully actuated system.

Let us define the following coordinate frames (see Fig. 1):

- the common, fixed and inertial, base frame \mathcal{F} ;
- \mathcal{F}_{ETi} and \mathcal{F}_{VTi} , attached, respectively, to the end-effector and to the UAV center of mass of the i th TR;
- \mathcal{F}_{EAj} and \mathcal{F}_{VAj} , attached, respectively, to the end-effector and to the UAV center of mass of the j th AR;
- the *absolute frame* \mathcal{F}_a , attached to an object grasped by the TRs or simply located at the centroid of the TRs' end-effectors.

A. Kinematics of individual UAVM system

In the following, let us drop, for notation compactness, the subscripts T and A , as well as the indexes i and j . The position of the UAV, i.e. of reference frame \mathcal{F}_V , expressed in the inertial reference frame \mathcal{F} , is given by the (3×1) vector \mathbf{p}_V , while its orientation is expressed via the rotation matrix

\mathbf{R}_V , whose expression can be found in [14], or, by using the triple of ZYX yaw-pitch-roll angles $\phi_V = [\psi_V \ \theta_V \ \varphi_V]^T$.

Position and orientation of the end-effector, i.e. of frame \mathcal{F}_E , with respect to frame \mathcal{F} , are given by the following equations

$$\begin{cases} \mathbf{p}_E = \mathbf{p}_V + \mathbf{R}_V \mathbf{p}_{E,V}^V \\ \mathbf{R}_E = \mathbf{R}_V \mathbf{R}_E^V, \end{cases} \quad (1)$$

where the (3×1) vector $\mathbf{p}_{E,V}^V$ and the (3×3) matrix \mathbf{R}_E^V describe the position and the orientation of \mathcal{F}_E relative to \mathcal{F}_V , respectively. By computing the time derivative of (1), the generalized (6×1) end-effector velocity, $\mathbf{v}_E = [\dot{\mathbf{p}}_E^T \ \boldsymbol{\omega}_E^T]^T$, where $\dot{\mathbf{p}}_E$ and $\boldsymbol{\omega}_E$ are the linear and angular velocities, after some algebraic steps, that can be found in [4], is given by

$$\mathbf{v}_E = \mathbf{G}_V^T(\mathbf{R}_V, \mathbf{q}) \mathbf{v}_V + \mathbf{J}_{E,V}(\mathbf{R}_V, \mathbf{q}) \dot{\mathbf{q}}, \quad (2)$$

where $\mathbf{J}_{E,V}$ is the Jacobian matrix of the manipulator expressed in the inertial frame, \mathbf{q} ($\dot{\mathbf{q}}$) are the joint positions (velocities), $\mathbf{v}_V = [\dot{\mathbf{p}}_V^T \ \boldsymbol{\omega}_V^T]^T$ is the (6×1) vector of vehicle generalized velocity and

$$\mathbf{G}_V = \begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{S}(\mathbf{R}_V \mathbf{p}_{E,V}^V) & \mathbf{I}_3 \end{bmatrix}, \quad (3)$$

with $\mathbf{S}(\cdot)$ representing the (3×3) skew-symmetric matrix operator [15] and \mathbf{I}_m and \mathbf{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, ϕ_V , the state of the UAVM system can be defined as

$$\boldsymbol{\zeta} = [\mathbf{p}_V^T \ \phi_V^T \ \mathbf{q}^T]^T. \quad (4)$$

In the case of a standard quadrotor, the vehicle is an underactuated system, characterized by only 4 independent control inputs against the 6 degrees of freedom. Hence, it is worth partitioning the vehicle state into the controlled variables, $\boldsymbol{\zeta}_c$, including position and yaw angle, and the uncontrolled variables, $\boldsymbol{\zeta}_u$, including roll and pitch angles, i.e.,

$$\boldsymbol{\zeta}_c = \begin{bmatrix} \mathbf{p}_V \\ \psi_V \\ \mathbf{q} \end{bmatrix}, \quad \boldsymbol{\zeta}_u = \begin{bmatrix} \theta_V \\ \varphi_V \end{bmatrix}. \quad (5)$$

Therefore (2) can be rewritten as

$$\mathbf{v}_E = \mathbf{J}(\boldsymbol{\zeta}_c, \boldsymbol{\zeta}_u) \dot{\boldsymbol{\zeta}} = \mathbf{J}_c(\boldsymbol{\zeta}_c, \boldsymbol{\zeta}_u) \dot{\boldsymbol{\zeta}}_c + \mathbf{J}_u(\boldsymbol{\zeta}_c, \boldsymbol{\zeta}_u) \dot{\boldsymbol{\zeta}}_u, \quad (6)$$

where

$$\mathbf{J} = [\mathbf{G}_V^T \bar{\mathbf{T}}(\phi_V) \mathbf{J}_{E,V}], \quad \bar{\mathbf{T}}(\mathbf{x}_V) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{T}(\phi_V) \end{bmatrix}, \quad (7)$$

$\mathbf{T}(\phi_V)$ is the matrix relating angular velocity to yaw-pitch-roll angles rate [14], while \mathbf{J}_c and \mathbf{J}_u are extracted from \mathbf{J} , by considering the columns related to the controlled and uncontrolled variables, respectively.

III. CONTROL ARCHITECTURE

In the proposed control framework, a global mission, to be achieved by the team, is described in terms of absolute coordinate frame motion and team formation. It is a partially decentralized scheme, based on a three-layer architecture:

- The first layer is centralized: it communicates with the motion planner of the whole team, providing the desired motion of the absolute frame \mathcal{F}_a and the team formation, as well as with each UAVM. Notice that the TRs must fulfill the formation constraint, while not necessarily the ARs are required to keep the formation. Thus, for all the TRs and the subset of ARs which are required to fulfill the formation constraints the output of this layer is the end-effector reference trajectories, while for the remaining ARs, the layer outputs the specific desired behaviors required to that robot.
- The second layer is local to each UAVM: it receives the references output by the upper layer and computes the reference motion for the controlled variables, ζ_c , by resorting to an inverse kinematic algorithm with redundancy resolution, based on the NSB paradigm.
- The third layer is local to each UAVM as well, and consists of a motion controller aimed at tracking the motion references output by the second layer. Any kind of controller can be adopted, as an example, the control architecture exploited for the experimental campaign is described in Section VI-A.

IV. FIRST LAYER: MULTI-VEHICLE COORDINATED CONTROL

The global mission is specified in terms of position and orientation of the absolute frame and the team formation, by assigning the desired time profiles of the following variables:

- the absolute motion, i.e. position, \mathbf{p}_a , and orientation, \mathbf{R}_a , of the absolute frame, \mathcal{F}_a , with respect to frame \mathcal{F} ;
- the UAVMs formation, described by the relative position and orientation of the end-effector frames of all the TRs and of those ARs involved in the formation, \mathcal{F}_{ETi} and \mathcal{F}_{ETj} , with respect to the absolute frame, \mathcal{F}_a

$$\begin{cases} \mathbf{p}_{E^*,a}^a = \mathbf{R}_a^T (\mathbf{p}_{E^*} - \mathbf{p}_a) \\ \mathbf{R}_{E^*}^a = \mathbf{R}_a^T \mathbf{R}_{E^*} \end{cases} \quad (8)$$

$*$ = $\{i, j\}$, with $i = 1, \dots, N_T$ and $j = 1, \dots, N_A$. The superscript a denotes the quantities referred to the \mathcal{F}_a . In the case of transportation of a rigid object grasped in a rigid way, the relative variables, $\mathbf{p}_{Ei,a}^a$ and \mathbf{R}_{Ei}^a , have to be kept constant, according to the grasp geometry.

Thus, the motion planner has to provide the desired trajectories for all the variables described above in terms of position, orientation and the corresponding linear and angular velocities.

This layer computes, from (8), the reference motion, in the Cartesian space, of each UAVM's end-effector

$$\begin{cases} \mathbf{p}_{E^*d} = \mathbf{p}_a d + \mathbf{R}_a d \mathbf{p}_{E^*,a d}^a \\ \mathbf{R}_{E^*d} = \mathbf{R}_a d \mathbf{R}_{E^*d}^a \end{cases} \quad (9)$$

and the corresponding velocities

$$\begin{cases} \dot{\mathbf{p}}_{E^*d} = \dot{\mathbf{p}}_a d - \mathbf{S}(\mathbf{R}_a d \mathbf{p}_{E^*,a d}^a) \boldsymbol{\omega}_a d + \mathbf{R}_a d \dot{\mathbf{p}}_{E^*,a d}^a \\ \boldsymbol{\omega}_{E^*d} = \boldsymbol{\omega}_a d + \mathbf{R}_a d \boldsymbol{\omega}_{E^*d}^a \end{cases} \quad (10)$$

where the subscript d denotes the desired values.

Remark 4.1: At this layer the presence of unexpected obstacles, not taken into account during the planning phase, can

be handled by considering a safety sphere, including the all the UAVMs of the team, with center in the origin of frame \mathcal{F}_a and radius representing the safety distance to be ensured between the obstacle and the absolute frame. Then, when an obstacle is detected, the desired absolute trajectory $\mathbf{p}_a d$ can be re-planned by resorting to the NSB control approach [12], while the formation is kept constant. More details about this approach can be found in [16].

V. SECOND LAYER: UAVM BEHAVIORAL CONTROL

The second layer, local to each UAVM, computes the reference trajectories for the controlled variables ζ_c . Since the UAVM is a kinematically redundant system, the NSB control approach is adopted to exploit the redundant DOFs in order to achieve multiple tasks, arranged in a given priority order [12]. The proposed control scheme [4] is characterized by:

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

A. Elementary behaviors

An elementary behavior is described by a configuration-dependent task function, $\sigma(\zeta) \in \mathbb{R}^m$, and the corresponding task Jacobian matrix $\mathbf{J}_\sigma \in \mathbb{R}^{m \times (6+n_M)}$, defined as

$$\dot{\sigma} = \frac{\partial \sigma(\zeta)}{\partial \zeta} \dot{\zeta} = \mathbf{J}_\sigma(\zeta) \dot{\zeta} = \mathbf{J}_{c,\sigma}(\zeta) \dot{\zeta}_c + \mathbf{J}_{u,\sigma}(\zeta) \dot{\zeta}_u, \quad (11)$$

where $\mathbf{J}_{c,\sigma} \in \mathbb{R}^{m \times (4+n_M)}$ is obtained by merging the first 4 columns with the last n_M columns of \mathbf{J}_σ , while $\mathbf{J}_{u,\sigma} \in \mathbb{R}^{m \times (2+n_M)}$ is given by the 5th and 6th column of \mathbf{J}_σ . Let σ_d be the desired value for the task function; from (11), the references for the motion controller are computed through a closed-loop inverse kinematics algorithm with compensation of not actuated motion variables [14] as

$$\dot{\zeta}_{c,r} = \mathbf{J}_{c,\sigma}^\dagger (\dot{\sigma}_d + \boldsymbol{\Lambda} \tilde{\sigma} - \mathbf{J}_{u,\sigma} \dot{\zeta}_u), \quad (12)$$

where $\mathbf{J}_{c,\sigma}^\dagger = \mathbf{J}_{c,\sigma}^T (\mathbf{J}_{c,\sigma} \mathbf{J}_{c,\sigma}^T)^{-1}$ is a right pseudo-inverse of $\mathbf{J}_{c,\sigma}$, $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains and $\tilde{\sigma}$ is the task error. It is worth pointing out that the pseudo-inverse in (12) requires that $\mathbf{J}_{c,\sigma}$ is full rank, i.e., $\text{rank}(\mathbf{J}_\sigma(\zeta)) = m$, and $m \leq (4 + n_M)$.

Elementary behaviors can be roughly classified into behaviors related to the control of the manipulator (e.g., end-effector pose, arm configuration, mechanical joint limits) and those related to the vehicle control (e.g., vehicle position and yaw, collision avoidance with obstacles or neighboring robots).

B. Compound behaviors

To accomplish complex missions, the elementary behaviors are hierarchically combined in *compound behaviors*. A compound behavior is a set of fixed elementary behaviors arranged in a given priority order. Thus, the elementary behaviors can

be seen as a library of atomic control objectives used to build more complex control objectives. The priority composition is built by resorting to the NSB paradigm: the overall velocity is obtained by merging the velocities computed for each elementary behavior as if it is acting alone; then, the lower-priority behaviors are projected onto the null space of the Jacobian of the higher-priority behaviors so as to remove the conflicting velocity components. Thus, the overall system velocity is computed as

$$\dot{\zeta}_r = \dot{\zeta}_1 + \sum_{k=2}^{N_b} N_{1,k-1} \dot{\zeta}_k, \quad (13)$$

where the subscript k denotes the task priority level, N_b is the number of behaviors to be fulfilled, $N_{1,k} = \mathbf{I} - \mathbf{J}_{1,k}^\dagger \mathbf{J}_{1,k}$ is a projector onto the null space of the augmented Jacobian $\mathbf{J}_{1,k}$, defined as $\mathbf{J}_{1,k} = [\mathbf{J}_1^T \quad \mathbf{J}_2^T \quad \dots \quad \mathbf{J}_k^T]^T$.

Priority order between elementary behaviors is *a priori* assigned, on the basis of practical consideration (e.g., safety behaviors, as obstacles avoidance, have always highest priority) or design choices. Not all the elementary behaviors can be combined with each others, since some of them are not compatible an, thus, the lower priority task will be not achieved at all. A rigorous analysis about the compatibility of elementary behaviors has been conducted in [17], where it has been proven that the solution (13) gives rise to stable and convergent error dynamics under mild conditions on the Jacobians \mathbf{J}_k and $\mathbf{J}_{1,k}$.

C. Supervisor

The adoption of the NSB paradigm and the definition of a set of compound behaviors for the proposed control framework, implies that a supervisor, implemented via a Finite State Machine [18], has to monitor the current state of both the mission and the robotic system and manage the real-time switching between the designed compound behaviors. Despite the supervisor is implemented on board each vehicle, it requires information from all the other vehicles, to accomplish, e.g., collision avoidance between the teammates. For supervisor design, let us classify the compound behaviors between *safety behaviors*, including at highest priority an elementary behavior related to the safety of the system, and *functional behaviors*, aimed at assigning a motion to the system or at optimizing some properties (e.g., robot manipulability) [11]. Adoption of a Finite State Automata requires the definition of a finite set of allowed compound behaviors (states of the Automata), mutually exclusive, as well as the set of causes forcing the UAVM to change its state (i.e., the switching rules between the different behaviors). The set of functional behaviors strictly depends on the assigned mission, thus also the supervisor design must be customized for the mission. The safety behaviors include elementary behaviors as obstacle avoidance, mechanical joint limits avoidance, inter-vehicle distance, and must be always included in the set of possible states.

Remark 5.1: Continuous and smooth reference time profiles must be fed to the motion controller, but switching between

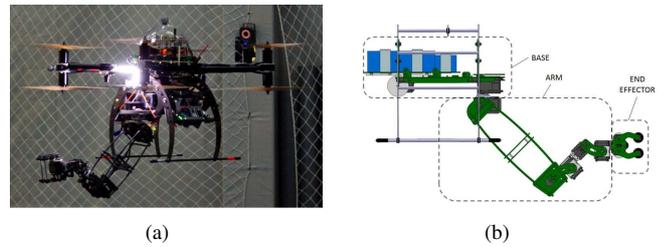


Fig. 2. ARCAS indoor setup: (a) aerial platform, (b) manipulator.

different behaviors can lead to velocity discontinuities. Possible solutions consists in filtering the task function error as in [19], or adopting a scaling filter to ensure continuity and smoothness of reference velocities [20] (see Section VI-C).

VI. CASE STUDY

An experimental campaign has been conducted on the ARCAS setup, consisting of two aerial manipulators equipped with 6 DOFs robotic arms. In the following, a case study requiring the activation of four compound behaviors and designed to intentionally activate the intervention of safety behaviors, is described. Some videos showing this experiment as well as other experiments not reported here for brevity can be found at <http://www2.unibas.it/automatica/multimedia.html>.

A. Setup and software architecture

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.2. Each platform is equipped with a 6-DOFs robotic arm [21] and a 2-DoFs gripper mounted on the end-effector. The ARCAS platform counts two processing units, an autopilot and an on-board computer, integrated into a common framework structured on the following levels:

- Control level, running on the autopilot and including 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 [22], based on Simulink code generation tools.
- Functional level, including the integration of the perception and cooperation algorithms running on-board the aerial robot. A Linux processing unit, namely an i7 Asctec Mastermind with ROS [23], is used for the integration of different functionalities.
- Multi-vehicle level, that integrates the software modules requiring information from multiple UAVMs.

At the Control level, a custom controller, detailed in [24], has been developed to properly handle the dynamic coupling between the arm and the UAV: this is a challenging issue since the arm weight corresponds to 17% of the total mass, which means that a significant part of the weight can change its distribution while flying. The key feature of the controller is the robotic arm compensator, which modifies the position of the batteries tray, in order to compensate for the changes of center of mass position due to the arm motion.

The ROS middleware is used to handle the communication among the functional modules as well as the multi-vehicle communication. The algorithms have been developed in C++ under ROS environment, running at 50 Hz. The ROS master is running on the on-board PC of the first UAVM, hereafter called UAVM1, together with the centralized layer, and the two lower layers modules. The second UAVM, hereafter called UAVM2, hosted only its own second and third layer controllers. Vehicle positions have been provided by the Vicon Motion System (<http://www.vicon.com>), running at 100 Hz, while the attitude is obtained through an estimator exploiting the IMU data.

B. Behaviors

Four elementary behaviors have been considered during the experiment: End-Effector Configuration (EEC), Inter-Vehicle distance (IV), relative Field of View (FoV) and Mechanical Joint Limit avoidance (MJL). It is worth noticing that the proposed control architecture is designed in such a way it can be adopted in semi-structured environments, in which unexpected changes of the operational scenario (e.g., unexpected obstacles) need to resort to on-line re-planning of the motion. Thus, safety behaviors, such as, e.g., MJL, become of the utmost importance.

1) *End Effector Configuration*: The goal is tracking a desired end-effector trajectory, both in terms of position and orientation. The task function can be written as $\sigma_{EEC} = [\mathbf{p}_E \ \mathcal{Q}_E] \in \mathbb{R}^7$, where \mathcal{Q}_E is the unit quaternion extracted by the rotation matrix \mathbf{R}_E describing the end-effector orientation [15]. The task Jacobian is the matrix \mathbf{J} defined in (7), namely $\mathbf{J}_{EEC} = \mathbf{J} \in \mathbb{R}^{6 \times 6 + n_M}$. The inverse kinematics law (12) becomes

$$\dot{\zeta}_{c,r} = \mathbf{J}_c^{\dagger}(\mathbf{v}_{E,d} + \Lambda_{EEC} \tilde{\sigma}_{EEC} - \mathbf{J}_u \dot{\zeta}_u), \quad (14)$$

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

$$\tilde{\sigma}_{EEC} = \begin{bmatrix} e_P \\ e_O \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{E,d} - \mathbf{p}_E(\zeta) \\ \tilde{\epsilon}(\zeta) \end{bmatrix}, \quad (15)$$

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

2) *Inter-Vehicle distance*: During the achievement of coordinated tasks, UAV-to-UAV collision avoidance must be ensured. The safety distance could be designed in such a way to minimize vehicle-to-vehicle airflow disturbances.

The Inter-Vehicle distance behavior is described by the following task function

$$\sigma_{IV_{ij}} = \|\mathbf{p}_{V_i} - \mathbf{p}_{V_j}\|^2 \in \mathbb{R},$$

where $\forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, N\}, i \neq j$. The Jacobians for the two UAVMs are given by

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

The desired value for the task function is the square of the safety distance.

3) *Mechanical Joint Limit Avoidance*: Any manipulator exhibits mechanical limits for the joint mobility. Each joint is usually allowed to move in a range; ignoring this limit may cause the robot to incur in an emergency stop. Different joint limit task functions could be found in literature [15], here the following choice has been considered [11]

$$\sigma_{MJL} = \sum_{i=1}^{n_M} l_i, \quad l_i(q_i) = \begin{cases} \frac{(q_i - \underline{q}_i)^2}{2n_M}, & \text{if } q_i \leq \underline{q}_i, \\ 0, & \text{if } \underline{q}_i < q_i \leq \bar{q}_i, \\ \frac{(\bar{q}_i - q_i)^2}{2n_M}, & \text{if } q_i > \bar{q}_i, \end{cases}$$

with \bar{q}_i and \underline{q}_i are software joint limits far enough from the lower and upper mechanical limit of the i th joint. In other words, \bar{q}_i and \underline{q}_i define an acceptable region in which the task function is null, when one of the joints violates such a safety region, even-though the physical limits are not violated, the task function increases so as to drive the joint back in the acceptable region. In the experiments the software limits have been chosen 5 deg less and 5 deg over the upper and the lower mechanical limit, respectively.

4) *Relative Field of View*: Directional devices or sensors mounted on the end-effector, such as, e.g., a video-camera, 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 $\mathbf{p}_l \in \mathbb{R}^3$. The task function is

$$\sigma_{FoV} = \begin{bmatrix} \sigma_{FoV1} \\ \sigma_{FoV2} \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} - \arccos\left(\frac{r^E}{\|r^E\|}\right) \\ \frac{\pi}{2} - \arccos\left(\frac{r^E}{\|r^E\|}\right) \end{bmatrix} \in \mathbb{R}^2, \quad (16)$$

where $r^E = \mathbf{R}_E^T(\mathbf{p}_E - \mathbf{p}_l) \in \mathbb{R}^3$. The task jacobian, $\mathbf{J}_{FoV} = [\mathbf{J}_{FoV1}^T \ \mathbf{J}_{FoV2}^T]^T \in \mathbb{R}^{2 \times 6 + n_M}$, can be computed from (16) as ($k = 1, 2$)

$$\mathbf{J}_{FoV k} = \frac{e_k^T}{\sqrt{\|r^E\|^2 - |r_k^E|^2}} \left[\mathbf{S}(r^E) - \mathbf{P}_r^\perp \mathbf{R}_E^T \right] \mathbf{J}, \quad (17)$$

where $\mathbf{P}_r^\perp = \mathbf{I}_3 - r^E (r^E)^T / \|r^E\|^2 \in \mathbb{R}^{3 \times 3}$, r_k^E is the k th component of r^E , e_k is the k th unit vector of the canonical base spanning \mathbb{R}^3 and \mathbf{J} is the jacobian defined by eq. (7).

Above-defined elementary behaviors have been combined in a set of compound behaviors. The EEC and MJL elementary behaviors are not fully compatible: in executing the compound behavior EEC+MJL the priority is given to EEC and MJL acts in such a way to prevent, if possible, joint limit violations. On the contrary, the compound behavior MJL+EEC assigns priority to the MJL task, in such a way to guarantee the respect of the joint limit at expense of a certain error in end-effector trajectory tracking. Behaviors FoV and IV are fully compatible since the first one constraints the end-effector orientation while the second constraints only vehicles' relative position.

C. Smooth transition

The switching among the different compound behaviors can generate discontinuities in the controlled variables' reference

velocities. In order to avoid such discontinuities, the references coming out from the second layer are stacked in the vector

$$\zeta_{c,r}^* = \left[\zeta_{c,r}^{1T}, \zeta_{c,r}^{2T}, \dots, \zeta_{c,r}^{NT} \right]^T,$$

which can be filtered via the following scaling law

$$\ddot{\zeta}_{c,r}^* = \text{sat}(\gamma \ddot{\zeta}_{c,r}^* + \mathbf{A}_v(\gamma \dot{\zeta}_{c,r}^* - \dot{\zeta}_{c,r}^*)), \quad (18)$$

where $\text{sat}(\cdot)$ is a saturation function ensuring that the accelerations will not exceed defined limits, and $\dot{\zeta}^*$ is the vector of the scaled reference velocities of all the UAVMs. \mathbf{A}_v is a diagonal matrix of positive gains, γ is the scalar scaling factor, computed according to the updating law [20]

$$\begin{aligned} \gamma &= 1 + \alpha_\gamma x_f & (19) \\ \dot{x}_f &= \begin{cases} (-\alpha_f x_f + 1) \|\dot{\zeta}_{c,r}^*\| - \gamma \|\dot{\zeta}_{c,r}^*\|, & \gamma \|\dot{\zeta}_{c,r}^*\| \geq \|\dot{\zeta}_{c,r}^*\| \\ -\alpha_f x_f \|\dot{\zeta}_{c,r}^*\|, & \gamma \|\dot{\zeta}_{c,r}^*\| < \|\dot{\zeta}_{c,r}^*\| \end{cases}, \end{aligned}$$

where x_f is a scalar filter variable, α_γ and α_f are positive scalar gains and the operator $\|\cdot\|$ represents the infinity norm. The key idea is that of adjusting the time profile of the reference velocities during the acceleration phases. When the acceleration of at least one UAVM reaches the imposed the saturation limits, the norm of the current velocities of all the UAVMs in the team $\|\dot{\zeta}_{c,r}^*\|$ results to be lower than the nominal time profile $\gamma \|\dot{\zeta}_{c,r}^*\|$, thus, the adaptive law (19) acts in such a way to reduce x_f , thereby reducing γ , until the nominal time profile approaches the current one.

VII. EXPERIMENTAL RESULTS

In the proposed experiment, UAVM2, which plays the role of AR, is in charge of pointing a camera, mounted on its end-effector, at UAVM1 which, in turn, is commanded to move toward the other aerial manipulator. More in detail, UAVM1 executes the compound behavior EEC+MJL, while UAVM2 executes the compound behavior FoV+MJL. Figure 3, reports both the end-effector and the vehicles positions: starting from 26s UAVM1 moves toward UAVM2 along the y axis and, as result, the mutual distance becomes lower than the safety one, set to 6 m. Such a value depends on the vehicle-vehicle airflow disturbances, which, in an indoor environment, tends to be magnified by the constrained space, and on the need of ensuring a reasonable manoeuvre space to the safety pilot to react and take manual control. Therefore, from Fig.3(b), it can be seen that UAVM2, starting from 38s moves along the y direction in such a way to restore the safety distance. Figure 4(a) reports the mutual distance, which reaches the minimum value of about 5.85 m: when the safety distance constraint is violated, the UAVM2 switches to compound behavior IV+FoV+MJL and reacts by moving backward, still keeping the end-effector frame in the desired orientation. The orientation of the UAVM2 end-effector, in terms of yaw, pitch and roll angles, is reported in Fig. 5(b).

Regarding the Field of View, an opportune performance index has been defined as

$$\mathcal{I}_{FoV} = [0 \ 0 \ 1] \frac{\mathbf{r}^{E2}}{\|\mathbf{r}^{E2}\|}, \quad (20)$$

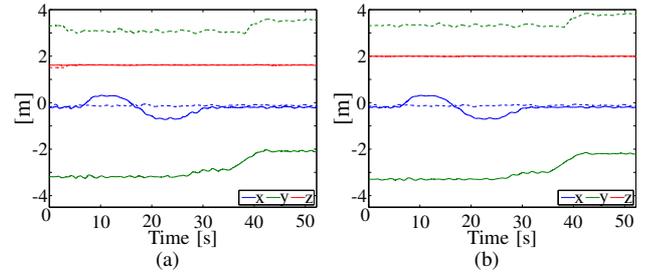


Fig. 3. Trajectories of the end-effectors (a) and vehicle center of mass (b) referred to UAVM1 (solid lines) and UAVM2 (dashed lines).

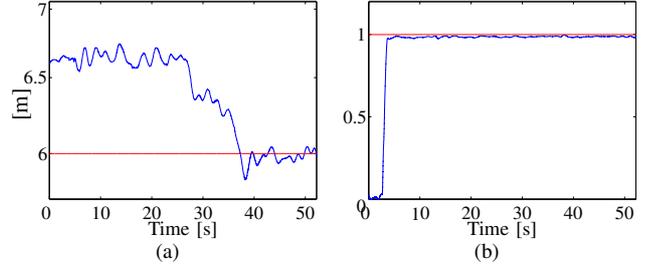


Fig. 4. Distance between the two UAVs (a) and field of view performance index (b).

where $\mathbf{r}^{E2} = \mathbf{R}_{E2}^T(\mathbf{p}_{E1} - \mathbf{p}_{E2})$ represents the vector joining the two end-effectors. This index measures the projection of the z axis of UAVM2 end-effector frame onto the vector \mathbf{r}^{E2} . Its time history is shown in Fig. 4(b): 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., $\mathcal{I}_{FoV} \simeq 1$).

Finally, the end-effector position and orientation (in terms of yaw pitch and roll angles) tracking error of UAVM1 are reported in Figs. 6(a) and 6(b). It can be noticed that the errors remain always limited during the whole mission.

Figure 7 shows some snapshots of the experiment. The initial configuration of the UAVMs with the camera attached to the end effector of UAVM2 pointed to the ground (see the image in the right bottom corner) is reported in Fig. 7(a). Fig. 7(b) shows a intermediate configuration, in which the UAVM2 end-effector is moving up in order to include the UAVM1 in the camera field of view. Finally, in Fig. 7(c), the desired field of view is reached, thus the UAVM1 is in the camera view. During the experiment, due to the FoV behavior, z axis of UAVM2 end-effector frame (in blue in the Figure) modified its orientation of about 90 degrees. Such rotation around the x axis of frame \mathcal{F} can be appreciated in Fig. 5(b).

A. Supervisor

The supervisor implemented on board each UAVM for the experimental campaign is reported in Fig. 8. As seen in Section V-C, the supervisor includes a set of functional behaviors and a set of safety behaviors: the functional behavior set includes only a compound behavior, which is EEC+MJL for UAVM1 and FoV+MJL for UAVM2. In Fig. 8, for the sake of compactness, the notation FB represents EEC or FoV, thus the initial state is set to FB+MJL. Since in the experiments, for safety reasons, the UAVM take off and landing are manually managed, they are not taken into account in the supervisor.

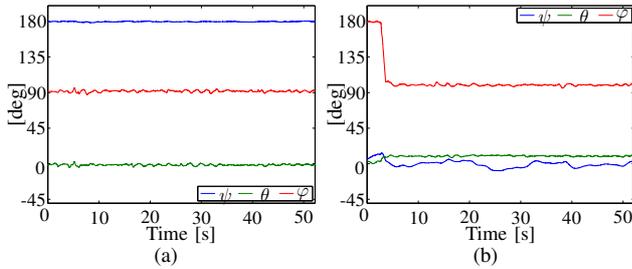


Fig. 5. End effector orientation of UAVM1 (a) and UAVM2 (b).

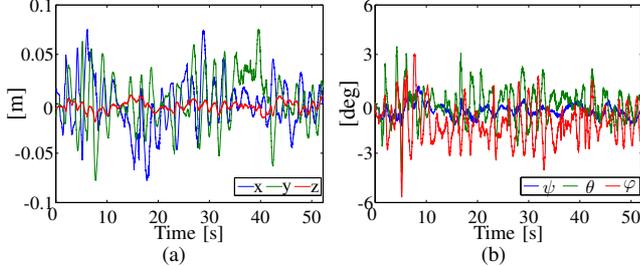


Fig. 6. Position (a) and orientation (b) end effector tracking error of UAVM1.

Two safe operation modes have been considered: the first one includes the IV at highest priority, while the second case presents the inversion of priorities between the MJL and the functional behavior. This last safe mode could occur in the case the MJL at lower priority is insufficient to avoid the violation of the mechanical limits, thus, its priority is increased by defining an opportune compound behavior (MJL+FB).

The transition between the state FB+MJL and the state MJL+FB is commanded when the MJL task in second priority does not allow to drive to zero the task function σ_{MJL} for a certain time interval Δt . It can be summarized by the rule:

$$R1: \sigma_{MJL} > 0 \quad \forall t \in [t^*, t^* + \Delta t],$$

where t^* is the first time instant such that $\sigma_{MJL} > 0$. The behavior FB+MJL is activated again when the function σ_{MJL} is driven to zero and is kept to zero for a certain time interval, in order to avoid chattering, namely the rule R2 is

$$R2: \sigma_{MJL} = 0 \quad \forall t \in [t^*, t^* + \Delta t].$$

Activation and deactivation of the IV safe mode for the UAVM1, are commanded when the following conditions hold

$$R3: \|\mathbf{p}_{V_1} - \mathbf{p}_{V_2}\| < d_{IV} \text{ AND } \{\dot{\mathbf{p}}_{V_1}^T(\mathbf{p}_{V_2} - \mathbf{p}_{V_1}) > 0 \text{ OR } \dot{\mathbf{p}}_{V_2}^T(\mathbf{p}_{V_1} - \mathbf{p}_{V_2}) > 0\};$$

$$R4: \|\mathbf{p}_{V_1} - \mathbf{p}_{V_2}\| \geq d_{IV} \text{ AND } \dot{\mathbf{p}}_{V_1}^T(\mathbf{p}_{V_2} - \mathbf{p}_{V_1}) \leq 0 \text{ AND } \dot{\mathbf{p}}_{V_2}^T(\mathbf{p}_{V_1} - \mathbf{p}_{V_2}) \leq 0,$$

The same rules can be defined for the UAVM2 by replacing index 1 with index 2. In other words, the Inter Vehicle behavior is activated when the distance between the two vehicles is less than a safety value, d_{IV} , and at least one of the vehicles is moving toward the other. On the other side, the safety behavior is deactivated when the distance is equal or bigger than the safety one and the vehicles moves away from each other.

It is easy to verify that the supervisor is consistent, namely only a single state can be active at each time instant, and complete, namely it operates correctly for all possible input/state sequences [25].

In the experiment, not all the described states have been activated: the behavior MJL+FB was never necessary, since

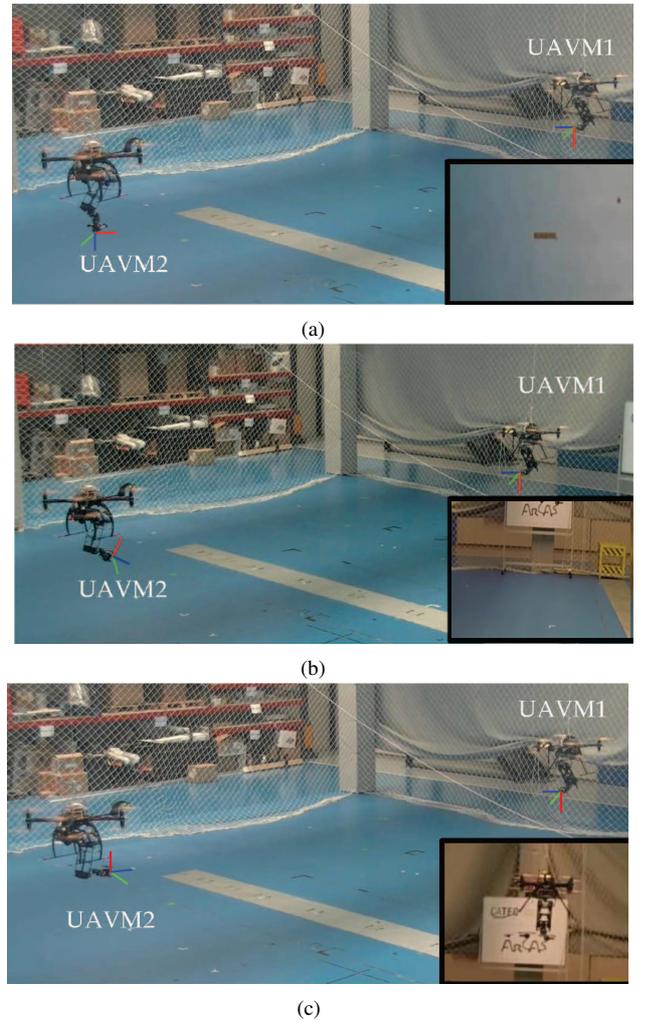


Fig. 7. Some snapshots of the second experiment. On the right bottom corner, there are the corresponding snapshots of the camera attached to the UAVM2.

the condition in rule R1 was never verified. However such behavior must be implemented in order to ensure the completeness of the automata.

VIII. CONCLUSIONS

A general framework for multiple aerial manipulators coordinated control has been presented. The approach is based on a hierarchical architecture, in which the overall mission is split among the robots of the team and decomposed in a set of tasks, called compound behaviors, assigned to each teammate. The NSB paradigm has been adopted in order to manage the priority between different tasks. The method has been successfully validated in a real test-bed, involving two aerial vehicles equipped with robotic arms. At the the best of authors' knowledge, these are the first experiments involving multiple UAVM executing complex multi-task missions.

ACKNOWLEDGMENTS

This work was supported by the European Commission under the 7th Framework Program, grant agreement No.287617 (ARCAS Aerial Robotics Cooperative Assembly system) and No. 608849 (EUROC - European Robotics Challenges), and

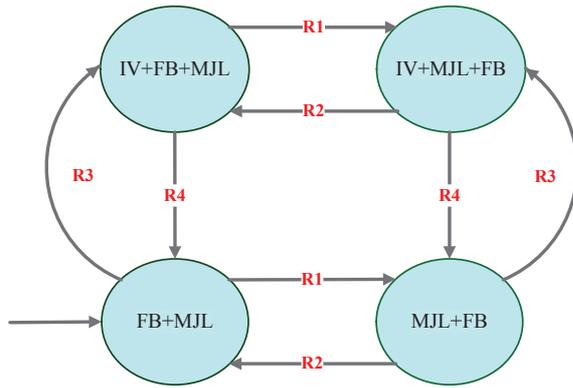


Fig. 8. Automata modeling the supervisor implemented on board the aerial manipulators during the experiments.

under the Horizon 2020 Program, grant agreement No. 644271 (AEROARMS - Aerial Robotic system integrating multiple ARMS and advanced manipulation capabilities for inspection and maintenance).

REFERENCES

- [1] 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," *Int. Journal of Advanced Robotic Systems*, vol. 10, no. 145, pp. 1–9, 2013.
- [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] S. Kim, S. Choi, and H. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int. Conf. on*, 2013, pp. 4990–4995.
- [4] K. Baizid, G. Giglio, F. Pierri, M. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Behavioral control of unmanned aerial vehicle manipulator systems," *Autonomous Robots*, vol. 41, no. 5, pp. 1203–1220, 2017.
- [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] J. Gancet, G. Hattenberger, R. Alami, and S. Lacroix, "Task planning and control for a multi-uav system: architecture and algorithms," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ Int. Conf. on*, Aug 2005, pp. 1017–1022.
- [7] M. Ceberio, L. Valera, O. Kosheleva, and R. Romero, "Model reduction: Why it is possible and how it can potentially help to control swarms of unmanned aerial vehicles (uavs)," in *Annual Conf. of the North American Fuzzy Information Processing Society (NAFIPS)*, 2015, pp. 1–6.
- [8] A. Brandao, J. Barbosa, V. Mendoza, M. Sarcinelli-Filho, and R. Carelli, "A multi-layer control scheme for a centralized uav formation," in *Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 1181–1187.
- [9] S. Simi, R. Kurup, and S. Rao, "Distributed task allocation and coordination scheme for a multi-uav sensor network," in *10th Int. Conf. on Wireless and Optical Communications Networks*, July 2013, pp. 1–5.
- [10] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project," *Journal of Field Robotics*, vol. 31(3), pp. 364–385, 2013.
- [11] 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 Int. Federation of Automatic Control*, 2014, pp. 1108–1113.
- [12] 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.
- [13] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on coordinated motion of aerial robotic manipulators," in *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1224–1229.
- [14] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri, "Control of quadrotor aerial vehicles equipped with a robotic arm," in *Proc. of 21th Mediterranean Conf. on Control and Automation*, 2013, pp. 1174–1180.
- [15] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics – Modelling, Planning and Control*. London, UK: Springer, 2009.
- [16] K. Baizid, F. Caccavale, S. Chiaverini, G. Giglio, and F. Pierri, "Safety in coordinated control of multiple unmanned aerial vehicle manipulator systems: Case of obstacle avoidance," in *Control and Automation (MED), 2014 22nd Mediterranean Conference of*. IEEE, 2014, pp. 1299–1304.
- [17] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic system," *IEEE Trans. on Robotics*, vol. 25, pp. 985–994, 2009.
- [18] R. Alur and M. Yannakakis, "Model checking of hierarchical state machines," *ACM Transactions on Programming Languages and Systems*, vol. 23, no. 3, pp. 273–303, 2001.
- [19] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *Robotics, IEEE Trans. on*, vol. 23, no. 1, pp. 60–72, 2007.
- [20] O. Dahl and L. Nielsen, "Torque-limited path following by online trajectory time scaling," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 5, pp. 554–561, 1990.
- [21] 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.
- [22] D. Santamaria, F. Alarcon, A. Jimenez, A. Viguria, M. Bejar, and A. Ollero, "Model-based design, development and validation for uas critical software," *Journal of Intelligent & Robotic Systems*, vol. 65, pp. 103–114, 2012.
- [23] 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.
- [24] 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 *Robotics and Automation (ICRA), 2015 IEEE Int. Conf. on*, 2015, pp. 4014–4020.
- [25] M. Ouimet and K. Lundqvist, "Automated verification of completeness and consistency of abstract state machine specifications using a SAT solver," *Electronic Notes in Theoretical Computer Science*, vol. 190, no. 2, pp. 85 – 97, 2007.