

# Set-based Inverse Kinematics Control of an Anthropomorphic Dual Arm Aerial Manipulator

E. Cataldi<sup>1</sup>, F. Real<sup>2</sup>, A. Suarez<sup>2</sup>, P.A. Di Lillo<sup>1</sup>, F. Pierri<sup>3</sup>, G. Antonelli<sup>1</sup>, F. Caccavale<sup>3</sup>, G. Heredia<sup>2</sup>, A. Ollero<sup>2</sup>

**Abstract**—The paper presents a multiple task-priority inverse kinematics algorithm for a dual-arm aerial manipulator. Both tasks defined as equality constraints and inequality constraints are handled by means of a singularity robust method based on the Null-Space based Behavioral control. The proposed schema is constituted by the inverse kinematics control, that receives the desired behavior of the system and outputs the reference values for the motion variables, i.e. the UAV pose and the arm joints position, and a motion control, that computes the vehicle thrusts and the joint torques. The method has been experimentally validated on a system composed by an underactuated aerial hexarotor vehicle equipped with two lightweight 4-DOF manipulators, involved in operations requiring the coordination of the two arms and the vehicle.

## I. INTRODUCTION

Research interest on aerial manipulators is day by day increasing due to their capability to substitute the human personnel in works involving significant risks of accidents, including installation of equipment, inspection and maintenance of infrastructures and industrial process plants. In the last decade, aerial manipulators have evolved from using simple grippers [1] [2] to including multi-Degrees of Freedom (DOFs) manipulators. As an example, in [3] a robotic arm, with three active DOFs and a three-passive-DOFs end effector was designed, in [4] the first aerial robots with 6 and 7 DOFs robotic arms were developed and experimentally tested both indoors and outdoors, and in [5] a 6-DOF-parallel manipulator is used. A brief literature review can be found in [6]. In order to increase the dexterity, needed to deal with environment perturbations and to perform complex tasks, recently Unmanned Aerial Vehicles (UAVs) equipped by multiple arms have been proposed in [7], [8].

The presence of multi-DOFs arms confers redundancy to the system that can be exploited to perform several motion tasks simultaneously. In recent years a behavioral control scheme, based on the Null Space based Behavioral (NSB) approach [9], has been extended to single-arm aerial manipulators in [10] and [11]. More in detail, in

[10] the coordination between the aerial vehicle and the 6-DOF arm proposed in [4] has been considered by recurring to a multiple task priority inverse kinematics algorithm, while in [11] such a scheme has been extended to the case of multiple aerial manipulators. At the authors' best knowledge, these are the first works involving the use of aerial manipulators for the execution of multi-prioritized non-trivial tasks. In these papers, only equality constraints have been considered for the assigned tasks, while, in a robotic system, a task can be conveniently assigned by resorting to inequality constraints, i.e., control variables that need to be kept in a range of values instead of an exact one (hereafter *set-based* tasks). The problem of handling inequality constraints has been considered in the last years with different approaches, which propose to transform the inequality constraints into equality constraints or potential functions. In [12] both equality and inequality constraints are handled in a prioritized way via a hierarchical multiple least-square quadratic optimization problem. Similarly, in [13] the inequality constraints are transformed in an equality constraint by means of slack variables, which are, then, minimized together to the task errors in a task-priority architecture. The potential field approach is adopted in [14] where smooth potential fields are used to represent the set-based objectives and activation functions. Such an approach has been experimentally validated with an underwater vehicle-manipulator system, however strict priority among the tasks was lost due to the presence of smoothing functions introduced to avoid discontinuities. In order to overcome these drawbacks, in [15] the authors proposed a method that directly embeds the set-based tasks into a singularity-robust multiple task-priority inverse kinematics framework based on the NSB paradigm.

In this paper, the set-based inverse kinematics control proposed in [15] has been experimentally validated on a dual-arm aerial manipulator. At the authors' knowledge this is the first time in which a dual-arm aerial manipulator, indeed a novel platform still not well investigated, is tested for the execution of non-trivial mission requiring the coordination of the two arms. The proposed scheme is constituted by a trajectory generator, that outputs the desired motion of the manipulators' end-effectors, the set-based inverse kinematics control, and a motion control. In regard to the prioritized kinematic inversion, the equality tasks are always active and include tasks such as end-effector position, orientation and camera Field of View, whilst the set-based tasks are activated only if the task-variable is close to violate the bounds of its domain otherwise they are inactive and do not require any DOFs.

<sup>1</sup>G. Antonelli, E. Cataldi and P.A. Di Lillo are with University of Cassino and Southern Lazio, Cassino, Italy.

<sup>2</sup>A. Ollero, G. Heredia, F. Real and A. Suarez are with University of Sevilla, Sevilla, Spain.

<sup>3</sup>F. Caccavale and F. Pierri are with University of Basilicata, Potenza, Italy. Corresponding author: Francesco Pierri, e-mail: francesco.pierri@unibas.it.

This research has been supported 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).

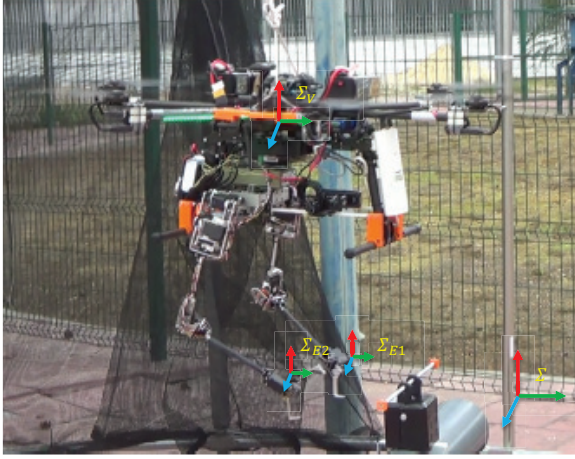


Fig. 1. Dual-arm aerial manipulator and considered coordinate frames

Such an approach permits to implement a number of tasks requiring more than the system's DOFs.

In regard to the motion control, the controller follows an inverse-dynamics approach, based on the dynamic model of the dual arm aerial manipulator [8], taking into account and compensating the movements of the arms and the interaction wrenches with the environment [16].

The experimental validation has been conducted in an indoor arena, available at the *Centro Avanzado de Tecnologías Aeroespaciales* of Seville, with a system composed by an underactuated aerial hexarotor vehicle equipped with two lightweight manipulators, each of them characterized by 4 DOFs, developed by the University of Seville.

## II. MODELING

Let us consider a UAV equipped with multiple robotic arms. For the sake of simplicity, let us assume that the UAV is a standard underactuated multi-rotor vehicle, characterized by 6 DOFs and only 4 control inputs, and the arms are only 2, each of them characterized, respectively, by  $n_1$  and  $n_2$  DOFs. However, the devised formulation can be easily generalized to the case of  $N$  manipulators and fully-actuated platforms, such as that in [17]. The following coordinate frames have been considered (see Fig. 1):

- the inertial frame  $\Sigma$ ;
- the frame  $\Sigma_V$  attached to the center of mass of the aerial vehicle, whose position and orientation w.r.t. the inertial frame are represented, respectively, by the vector  $\mathbf{p}_v \in \mathbb{R}^3$  and the orientation matrix  $\mathbf{R}_v \in \mathbb{R}^{3 \times 3}$ ;
- the frames  $\Sigma_{E_i}$  ( $i = 1, 2$ ) attached to the end-effector of the  $i$ th arm, whose position and orientation w.r.t. the frame  $\Sigma$  are represented, respectively, by the vectors  $\mathbf{p}_i \in \mathbb{R}^3$  and the orientation matrices  $\mathbf{R}_i \in \mathbb{R}^{3 \times 3}$ .

### A. Kinematics

For the  $i$ th ( $i = 1, 2$ ) arm, the end effector position and orientation are

$$\begin{cases} \mathbf{p}_i = \mathbf{p}_v + \mathbf{R}_v \mathbf{p}_{i,v}^V \\ \mathbf{R}_i = \mathbf{R}_v \mathbf{R}_{i,v}^V \end{cases} \quad (1)$$

where  $\mathbf{p}_{i,v}^V$  and  $\mathbf{R}_{i,v}^V$  are the relative position and orientation of the frame  $\Sigma_{E_i}$  w.r.t. the frame  $\Sigma_V$ , expressed in frame  $\Sigma_V$ .

By differentiating (1) and after some standard computation the linear,  $\dot{\mathbf{p}}_i$ , and angular,  $\boldsymbol{\omega}_i$  velocities of the  $i$ th manipulator end effector are given, respectively, by

$$\begin{cases} \dot{\mathbf{p}}_i = \dot{\mathbf{p}}_v - \mathbf{S}(\mathbf{R}_v \mathbf{p}_{i,v}^V) \boldsymbol{\omega}_v + \mathbf{R}_v \dot{\mathbf{p}}_{i,v}^V \\ \boldsymbol{\omega}_i = \boldsymbol{\omega}_v + \mathbf{R}_v \boldsymbol{\omega}_{i,v}^V \end{cases} \quad (2)$$

where  $\mathbf{S}(\cdot)$  is the skew symmetric operator performing the cross-product [18]. The relative velocities of  $\Sigma_{E_i}$  w.r.t. the frame  $\Sigma_V$  expressed in frame  $\Sigma_V$ ,  $\dot{\mathbf{p}}_{i,v}^V$  and  $\boldsymbol{\omega}_{i,v}^V$  can be rearranged as the standard manipulator differential kinematics

$$\mathbf{v}_{i,v}^V = \begin{bmatrix} \dot{\mathbf{p}}_{i,v}^V \\ \boldsymbol{\omega}_{i,v}^V \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{P_{i,v}}^V \\ \mathbf{J}_{O_{i,v}}^V \end{bmatrix} \dot{\mathbf{q}}_i = \mathbf{J}_{i,v}^V \dot{\mathbf{q}}_i, \quad (3)$$

with  $\mathbf{J}_{i,v}^V$  the manipulator Jacobian matrix expressed in frame  $\Sigma_V$ , and  $\mathbf{q}_i \in \mathbb{R}^{n_i}$  the vector of joint positions. From equations (2), the following form for the differential kinematics can be obtained

$$\begin{bmatrix} \dot{\mathbf{p}}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}(\mathbf{R}_v \mathbf{p}_{i,v}^V) \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_v \\ \boldsymbol{\omega}_v \end{bmatrix} + \begin{bmatrix} \mathbf{R}_v & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{R}_v \end{bmatrix} \mathbf{J}_{i,v}^V \dot{\mathbf{q}}_i, \quad (4)$$

where  $\mathbf{I}_m$  and  $\mathbf{O}_m$  are, respectively, the  $(m \times m)$  identity and null matrix. By denoting with

$$\mathbf{J}_v^i = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}(\mathbf{R}_v \mathbf{p}_{i,v}^V) \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix}$$

the Jacobian representing the vehicle contribution to the  $i$ th end-effector velocity and with

$$\mathbf{J}_{i,v} = \begin{bmatrix} \mathbf{R}_v & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{R}_v \end{bmatrix} \mathbf{J}_{i,v}^V$$

the geometric Jacobian of the  $i$ th manipulator in inertial frame  $\Sigma$ , (4) can be rewritten in compact form as

$$\mathbf{v}_i = \mathbf{J}_v^i \mathbf{v}_v + \mathbf{J}_{i,v} \dot{\mathbf{q}}_i, \quad (5)$$

where  $\mathbf{v}_* = [\dot{\mathbf{p}}_*^T \ \boldsymbol{\omega}_*^T]^T$ , ( $\star = 1, 2, v$ ), represent the generalized velocities in inertial coordinate frame.

By adopting the roll-pitch-yaw Euler angles for the UAV orientation, i.e.  $\boldsymbol{\eta}_v = [\varphi_v \ \vartheta_v \ \psi_v]^T$ , let us rewrite (5) by substituting the vehicle angular velocity with the Euler angles time derivative as

$$\mathbf{v}_i = \mathbf{J}_v^i \mathbf{T}_A(\boldsymbol{\eta}_v) \dot{\mathbf{x}}_v + \mathbf{J}_{i,v} \dot{\mathbf{q}}_i, \quad (6)$$

where  $\mathbf{x}_v = [\mathbf{p}_v^T \ \boldsymbol{\eta}_v^T]^T$  represents the UAV state and

$$\mathbf{T}_A(\boldsymbol{\eta}_v) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{T}(\boldsymbol{\eta}_v) \end{bmatrix},$$

with  $\mathbf{T}(\boldsymbol{\eta}_v)$  the matrix that relates the derivative of the chosen Euler angles with the angular velocity. Due to the assumption of underactuation for the UAV, 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  $\mathbf{x}_v$  in terms of controllable and uncontrollable variables as follows

$$\mathbf{x}_v = \begin{bmatrix} \chi_c \\ \chi_u \end{bmatrix}, \quad \chi_c = \begin{bmatrix} p_v \\ \psi_v \end{bmatrix}, \quad \chi_u = \begin{bmatrix} \varphi_v \\ \vartheta_v \end{bmatrix}.$$

Differential kinematics becomes

$$\mathbf{v}_i = \mathbf{J}_{\chi_c}^i(\mathbf{q}_i, \boldsymbol{\eta}_v) \dot{\chi}_c + \mathbf{J}_{\chi_u}^i(\mathbf{q}_i, \boldsymbol{\eta}_v) \dot{\chi}_u + \mathbf{J}_{i,v} \dot{\mathbf{q}}_i, \quad (7)$$

where  $\mathbf{J}_{\chi_c}^i$  and  $\mathbf{J}_{\chi_u}^i$  are the Jacobian referred to the controllable and uncontrollable variables, respectively, obtained from  $\mathbf{J}_v^i \mathbf{T}_A(\boldsymbol{\eta}_v)$  by selecting the opportune columns.

By considering both the end-effector velocities, the generalized end-effectors velocity  $\mathbf{v}_e = [\mathbf{v}_1^T \mathbf{v}_2^T]^T$  is given by the following relation

$$\mathbf{v}_e = \mathbf{J}_{\zeta_c}(\mathbf{x}_v, \mathbf{q}_1, \mathbf{q}_2) \dot{\zeta}_c + \mathbf{J}_{\zeta_u}(\mathbf{x}_v, \mathbf{q}_1, \mathbf{q}_2) \dot{\zeta}_u, \quad (8)$$

where

$$\zeta_c = \begin{bmatrix} \chi_c \\ \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix}, \quad \mathbf{J}_{\zeta_c} = \begin{bmatrix} \mathbf{J}_{\chi_c}^1 & \mathbf{J}_{1,v} & \mathbf{O}_{6 \times n_1} \\ \mathbf{J}_{\chi_c}^2 & \mathbf{O}_{6 \times n_2} & \mathbf{J}_{2,v} \end{bmatrix}, \quad \mathbf{J}_{\zeta_u} = \begin{bmatrix} \mathbf{J}_{\chi_u}^1 \\ \mathbf{J}_{\chi_u}^2 \end{bmatrix}.$$

### III. CONTROL SCHEME

The proposed scheme is a two-layer kinematic control aimed at coordinating the motion of the two arms and the aerial vehicle. The upper layer computes the reference trajectories for the arm joints as well as for the controlled variables of the vehicle (i.e., position and yaw angle). At this layer the kinematic redundancy is exploited to fulfill multiple tasks via a task-priority algorithm, based on the NSB control [19] [20]. The second layer is a motion controller aimed at ensuring the tracking of the computed reference values. The control law is an integral backstepping based on the dynamic model of the dual arm aerial manipulator and takes into account both the compensation of the arms' movements and the interaction wrenches with the environment. Such a controller has been described in [16] and here it is skipped for the sake of brevity.

#### A. Set-based Inverse Kinematics

A task assigned to the system is characterized by a task variable  $\boldsymbol{\sigma} \in \mathbb{R}^m$ , which is function of the system's configuration,  $\zeta$ , and by its Jacobian matrix  $\mathbf{J}_\sigma$ . The differential relationship between the task velocity and the system configuration is

$$\dot{\boldsymbol{\sigma}} = \mathbf{J}_\sigma(\zeta) \dot{\zeta}. \quad (9)$$

The kinematic control problem can be formulated as to find reference values for the controlled variables,  $\zeta_r$ , to be fed to a motion controller in order to ensure that the task variable reaches its desired value,  $\boldsymbol{\sigma}_d$ . The velocity reference,  $\dot{\zeta}_r$ , can be computed via a closed-loop algorithm [18] as

$$\dot{\zeta}_r = \mathbf{J}_\sigma^\dagger(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}), \quad (10)$$

where  $\mathbf{J}_\sigma^\dagger = \mathbf{J}_\sigma^T (\mathbf{J}_\sigma \mathbf{J}_\sigma^T)^{-1}$  is a right pseudo-inverse of  $\mathbf{J}_\sigma$ ,  $\boldsymbol{\Lambda}$  is a constant positive-definite matrix of gains and  $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$  is the task error. In the presence of under-actuated systems, such an aerial vehicle, the non-controlled variables can be taken into account in (10) as [10]

$$\dot{\zeta}_{c,r} = \mathbf{J}_{\sigma_c}^\dagger(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}} - \mathbf{J}_{\sigma_u} \dot{\zeta}_u), \quad (11)$$

where  $\mathbf{J}_{\sigma_c}$  and  $\mathbf{J}_{\sigma_u}$  are the task Jacobians referred to controlled and uncontrolled variables.

Since the system is characterized by  $n = 4 + n_1 + n_2$  DOFs, namely 4 DOFs for the quadrotor and  $n_i$  DOFs for each arm, if  $n$  is larger than the number of DOFs required by the main task function, the system is *kinematically redundant* and the redundant DOFs can be exploited to fulfill multiple tasks via a task-priority algorithm based on the NSB control. The overall reference velocity for the controlled variables can be obtained by merging the velocity due to each task (computed via (10)), in such a way that the lower-priority task contributions are projected onto the null space of the higher-priority ones. In this way, the velocity components that conflict with the higher priority tasks are removed, and, thus, the secondary tasks can be achieved only if they are compatible with those at higher priority.

By assigning  $n_b$  tasks with a given priority, the overall system velocity is given by the following recursive scheme:

$$\dot{\zeta}_r = \dot{\zeta}_1 + \sum_{k=2}^{n_b} \mathbf{N}_{1,k-1} \dot{\zeta}_k, \quad (12)$$

$$\mathbf{N}_{1,k} = \left( \mathbf{I} - \mathbf{J}_{1,k}^\dagger \mathbf{J}_{1,k} \right), \quad (13)$$

where the subscript  $k = 1, \dots, n_b$  represents the task priority,  $\mathbf{N}_{1,k}$  is a projector onto the null space of the augmented Jacobian  $\mathbf{J}_{1,k}$ , given by  $\mathbf{J}_{1,k} = [\mathbf{J}_1^T \mathbf{J}_2^T \dots \mathbf{J}_k^T]^T$ .

The above-described method has been developed for tasks characterized by a specific desired value,  $\boldsymbol{\sigma}_d$ , e.g. the desired end-effector position, hereafter mentioned as *equality tasks*. In the present paper, also the so-called *set-based tasks* have been considered, i.e. tasks whose variable must lie in a set of values instead of assuming a specific one, by resorting to the algorithm proposed in [15]. A set-based task can be still expressed via a task function and a Jacobian matrix defined via (9) but it cannot be directly inserted in the multiple task-priority inverse kinematics (10)-(12). In [15], the authors proposed a method which allows a general number of scalar set-based tasks to be handled with a given priority within a number of equality tasks. More in detail, when the set-based task variable assumes values inside the desired domain, it should not affect the behavior of the system and all the DOFs can be used to fulfill the equality tasks. On the contrary, if the set-based task variable is outside the desired domain, it is inserted into the active task stack to be handled.

Since a set-based task can be either active or inactive, a system with  $h$  set-based tasks is characterized

by  $2^h$  possible combinations of set-based tasks being active/inactive, referred to as modes of the system. The algorithm developed in [15] is in charge of switching between modes to fulfill the equality tasks while ensuring that the set-based tasks are not violated. The modes are sorted by increasing restrictiveness; the more set-based tasks are active in a mode, the more restrictive it is. Hence, in the first mode, no set-based tasks are active, i.e. only equality tasks are considered, while in the  $2^h$ -th mode all set-based tasks are active.

For the sake of simplicity, let us consider the case of a system with a single set based task,  $\sigma_s \in \mathbb{R}$ , satisfied in domain  $\mathcal{D}$ , and  $n_b$  equality tasks,  $\sigma_1, \dots, \sigma_{n_b}$ . In this case only two modes can be considered

- Ignoring the set-based task and considering only the equality tasks;
- Adding the set-based task at the stack of active tasks with its own priority.

The first mode is the normal operating condition, while the second one will be activated when it is necessary to prevent the inequality constraints to be violated, i.e. when the task variable  $\sigma_s$  is on the border of its domain  $\mathcal{D}$  and its time derivative is outside the *tangent cone* to the set  $\mathcal{D}$  at the point  $\sigma_s$  defined as

$$T_{\mathcal{D}}(\sigma_s) = \begin{cases} [0 + \infty) & \text{if } \sigma_s = \sigma_{min} \\ \mathbb{R} & \text{if } \sigma_s \in \mathcal{D} \\ (-\infty 0] & \text{if } \sigma_s = \sigma_{max} \end{cases} \quad (14)$$

where  $\sigma_{min}$  and  $\sigma_{max}$  are the left-end point and the right-end point of domain  $\mathcal{D}$ . It is worth noticing that if  $\dot{\sigma}_s \in T_{\mathcal{D}}(\sigma) \forall t > t_0$  then  $\sigma_s \in \mathcal{D} \forall t \geq t_0$ , moreover, if  $\sigma_s \in \mathcal{D}$  its derivative is always in the tangent cone, as it is the  $\mathbb{R}$ . If  $\sigma_s = \sigma_{min}$ , the task function is at lower border and if  $\dot{\sigma}_s \in T_{\mathcal{D}}(\sigma)$ , then  $\sigma_s$  will either stay on the border or move into the interior of  $\mathcal{D}$ . At same way, if  $\sigma_s = \sigma_{max}$ , and if  $\dot{\sigma}_s \in T_{\mathcal{D}}(\sigma)$ , the task function will not leave  $\mathcal{D}$ . Thus, when the second mode is activated, if the set-based task has higher priority with respect to the equality ones, it is guaranteed that the set-based task variable is frozen on the border of the admissible domain  $\mathcal{D}$ . On the contrary, if the set-based task has lower priority with respect to one or more equality tasks, then the activation of the second mode attempts to freeze it on the border of  $\mathcal{D}$ , but, since  $\dot{\sigma}_s$  is affected by the velocities of higher priority tasks, this cannot be guaranteed. The stability proof in the cases of high-priority and low-priority set-based tasks can be found in [15] and [21].

#### IV. SYSTEM DESCRIPTION

This section describes the dual arm aerial manipulation system employed to validate the control method described in Section III, including the dual arm, the aerial platform, and the software components and architecture.

##### A. Anthropomorphic Dual Arm

The manipulator integrated in the aerial platform is an anthropomorphic, compliant and lightweight dual arm [8] whose main parameters are summarized in Table I. Each of the arms provides four degrees of freedom for end effector positioning in a human-like kinematic



Fig. 2. Different poses of the anthropomorphic dual arm.

TABLE I  
MAIN SPECIFICATIONS OF THE ANTHROPOMORPHIC DUAL ARM.

Total weight	1.3 kg
Max. lift load per arm	0.3 kg
Upper arm/forearm lengths	25 cm
Separation between arms	30 cm
Shoulder pitch range	$\pm 90$ deg
Shoulder roll range	$[-30, 120]$ deg
Shoulder yaw range	$\pm 90$ deg
Elbow pitch range	$\pm 135$ deg

configuration, with three joints at the shoulder and one at the elbow. The arms are human-size, where the upper arm (from shoulder to elbow) and the forearm (from elbow to wrist) link lengths are 25 cm, whereas the separation between the shoulder joint of both arms is 30 cm. Figure 2 shows different poses that can be achieved with the arms. The anthropomorphic design approach is motivated by the convenience of replicating the manipulation capabilities of human operators during the realization of inspection and maintenance tasks in high altitude workspaces. The arms implement two protection mechanisms. On the one hand, the frame aluminum structure has been designed in such a way that the servo actuators are isolated against impacts, collisions and overloads, attaching polymer flange bearings that support the rotation of the links in the frame structure. On the other hand, and related with previous point, a spring-lever transmission mechanism is introduced between the servo shaft and the output link to provide compliance, so the peak torques associated to physical interactions are attenuated, allowing also the estimation and control of the forces and torques based on the measurement of the joint deflection angle [8], [16].

##### B. Aerial Platform

The aerial platform consists of an hexarotor manufactured by Drone Tools, whose main specifications are summarized in Table II. The hexarotor integrates an Intel NUC computer board, an Ubiquiti 5.8 GHz wireless link, along with the manipulator and the batteries. The arms are supported by a frame structure attached at the legs of the landing gear in such a way that the effective volume of operation of the arms is maximized, avoiding collisions with the landing gear or the propellers [8]. Figure 1 corresponds to the nominal operation position, while in the take-off and landing the arms rotate  $\pm 90$  deg around

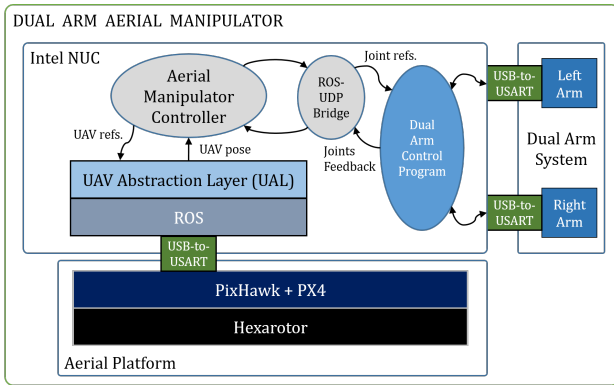


Fig. 3. Components and architecture of the aerial manipulator.

TABLE II  
MAIN SPECIFICATIONS OF THE AERIAL PLATFORM.

Total weight	9 kg
Max. flight time (nominal load)	15 cm
Max. payload	2.5 kg
Batteries	6S, 7000 mAh
Batteries weight	4 kg

the shoulder roll joint [7].

### C. Hardware/Software Architecture

The components and architecture of the dual arm aerial manipulator are represented in Figure 3. The low level control of the aerial platform is built in a PixHawk autopilot board running the PX4 flight stack that communicates with the Intel NUC computer board through a serial interface and the MAVROS protocol. The on-board computer executes three main software modules: the UAV Abstraction Layer (UAL), the dual arm control program, and the aerial manipulator control module which implements the methods described previously. The dual arm control program is developed in C/C++ and it is built around the Task Manager, where the functionalities of the manipulator are implemented. For the execution of the experiments presented below, three tasks were employed: go to rest position (take-off and landing phases), go to operation position (once the platform has taken-off), and external control model, in which the joints of the arms are commanded to move to the references received from the aerial manipulator controller, sending back the current position of the joints. The external control model included the inverse kinematics algorithm, written in C++ under ROS environment [22], with a library of equality and set-based tasks and a supervisor that, at each step, evaluates all possible  $2^h$  modes in order to select those which satisfy the task constraints and, among the selected ones, activates the less restrictive mode, i.e., that requiring less set-based tasks. More details about the algorithm implementation can be found in [15].

## V. EXPERIMENTS

A number of equality and set-based tasks have been implemented on the system for a total number of 23 DOFs, much greater than the number of available DOFs of the system. This is possible since the different tasks are never

activated all together but only a subset of them, based on the state of the system. More in detail, the following equality tasks have been implemented:

- Position and orientation trajectory tracking of the end-effector of the left arm. Such a task requires 6 DOFs.
- Field of View of the end effector of the right arm, equipped with a camera. Such a task requires 2 DOFs.
- Center of mass, this task is aimed at ensuring that the center of mass of the dual-arm system is, as much as possible, aligned with that of the UAV, in such a way to avoid to destabilize the flight and reduce the power consumption. Such a task requires 1 DOFs.

The following set-based tasks have been implemented:

- Joint limits: for each joint, upper and lower limits are set in order to avoid its mechanical limits. Such a task requires 1 DOF for each joint of the arms, thus the total number of required DOFs is 8.
- Virtual wall between the two arms: to avoid collisions between the two arms, a virtual wall is implemented in order to delimit their working spaces. Such a task requires 2 DOF.
- Virtual wall between the arms and the vehicle: to avoid collisions between the arms and the vehicle, virtual walls are implemented in order to delimit their working spaces. Such a task requires 1 DOF for each arm, thus the total number of required DOFs is 2.
- Manipulability, aimed at keeping the manipulators far enough from singular configurations, at which the structure loses mobility. Such a task requires 1 DOF for each arm, thus the total number of required DOFs is 2.

The position of the vehicle has been obtained via a motion capture system (VICON), while the joint positions are given by the servos. Some videos of the experiments are available at [www.elisabettaataldi.it/research/video/icra19.webm](http://www.elisabettaataldi.it/research/video/icra19.webm).

At the beginning, only the equality tasks are active and the end-effector trajectory tracking for the left arm and the Field of view for the right arm have been set as highest priority tasks, while the center of mass task was projected onto the null space of their Jacobians. During the motion, the set-based tasks are activated, those relative to the safety of the system, i.e. the joint limit and the virtual walls are always activated with higher priority than all the other tasks, while the manipulability is activated with lower priority than the end-effectors tasks.

Figure 4 reports the end-effector position and orientation of the left arm, in terms of planned trajectory (red dashed lines), reference trajectories computed from the inverse kinematics (blue dashed lines) and actual trajectories (continuous lines). It can be noticed that the reference trajectories and the actual ones are almost indistinguishable due to the accuracy of the low-level motion controller. On the other side, large errors are experienced between the planned and the reference trajectories in the first 60 s both for the position and the orientation: they are due to joint actuator saturations, which are activated due

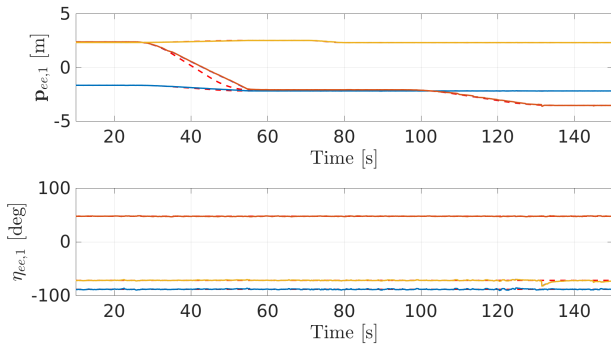


Fig. 4. Left end-effector position and orientation

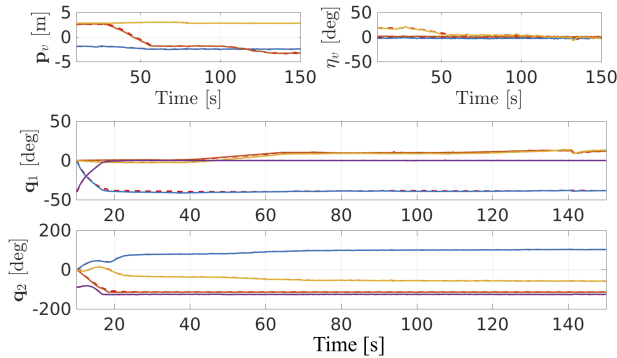


Fig. 5. Vehicle position (top left), vehicle yaw (top right), fist and second arm (bottom) tracking error

to the large acceleration required. The tracking errors of the motion variables, i.e. the vehicle and the arms' joints are reported in Fig. 5 and show the good performance of the low-level motion controllers.

Figure 6 reports the position for each manipulator joint: during the experiment, the second and the fourth joints, at the very beginning (around 15 s) try to exceed their joint limit, then both the tasks are activated and remain active during the whole experiment.

Figure 7 reports the task errors. It can be viewed that the virtual wall task (upper right corner) is never activated, while the manipulability (bottom left corner) is always active, and the manipulability measure is growing. In regard to the center of mass error (bottom right corner), it is reported the distance between the actual center of mass and the desired one and, even if the task is commanded with lower priority, it is slowly convergent to zero. Finally, at the upper left corner there is the Field of View task variable, whose explicit expression can be found in [11] and, roughly speaking, measures the angle between the outgoing vector of the camera and that of the left end-effector. As can be appreciated, its value quickly converges to  $\pi$  rad, i.e. the desired value, and, then, keeps this value during the whole experiment. In Fig. 8 some snapshots of the video are reported at different time instants with the camera view on the left upper corner: at the beginning the left end-effector was not in the field of view, while it appears after 20 s and then is kept for all the experiment.

## VI. CONCLUSIONS

A behavioral control for multiple-arms aerial manipulators, taking into account both equality and set-based tasks has been presented and experimentally validated.

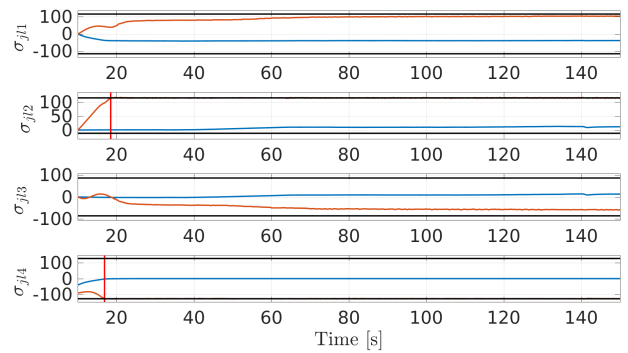


Fig. 6. Manipulator joint positions and their limits.

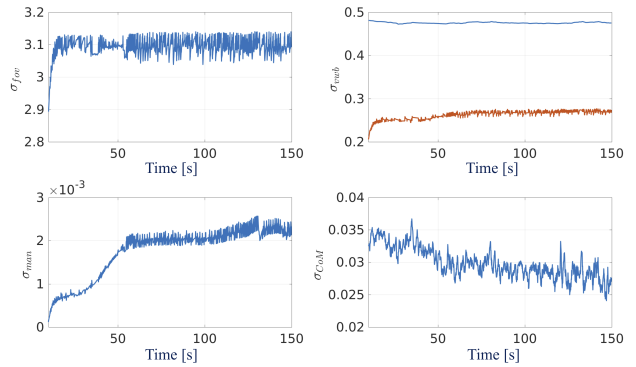


Fig. 7. Task variables: field of view (top left), virtual-wall base (top right), manipulability (bottom left), and center of mass (bottom right).

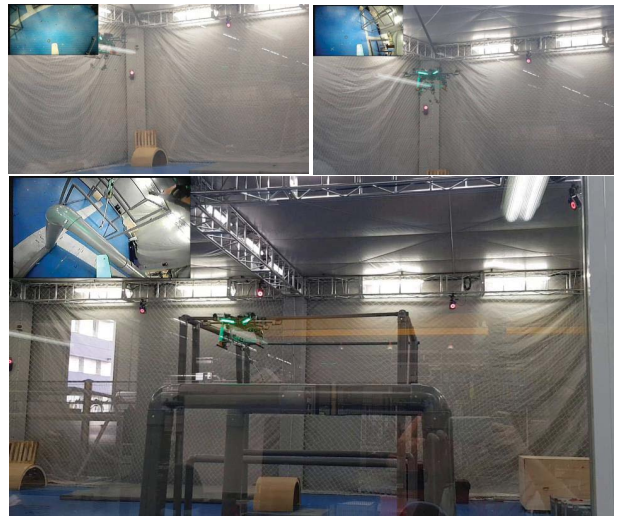


Fig. 8. Snapshots of the experiment: at the initial instant (upper left corner), after 10 s (upper right corner) and after 80 s (bottom).

The approach is based on a two-layer architecture, in which the upper layer is an inverse kinematics control which exploits the NSB paradigm to ensure the motion coordination and manage the priority between different tasks, while the bottom layer is a motion controller. The major contribution of the work is the validation of the algorithm on a real testbed, involving a dual-arm aerial manipulator, a novel platform never tested in the execution of complex multi-task missions.

## REFERENCES

- [1] P. Pounds, D. Bersak, and A. Dollar, "Grasping from the air: Hovering capture and load stability," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 2491–2498.
- [2] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modelling, estimation and control for aerial grasping and manipulation," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2668–2673.
- [3] A. Macchelli, F. Forte, A. Keemink, S. Stramigioli, R. Carloni, M. Fumagalli, R. Naldi, and L. Marconi, "Developing an aerial manipulator prototype," *IEEE Robot Autom Mag*, vol. 21, no. 3, pp. 41–55, 2014.
- [4] 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 *ROBOT 2013: First Iberian Robotics Conference, submitted*, 2013.
- [5] T. W. Danko, K. P. Chaney, and P. Y. Oh, "A parallel manipulator for mobile manipulating uavs," in *Technologies for Practical Robot Applications (TePRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.
- [6] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.
- [7] A. Suarez, A. Jimenez-Cano, V. Vega, G. Heredia, A. Rodriguez-Castaño, and A. Ollero, "Lightweight and human-size dual arm aerial manipulator," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 1778–1784.
- [8] A. Suarez, G. Heredia, and A. Ollero, "Design of an anthropomorphic, compliant and lightweight dual arm for aerial manipulation," *IEEE Access*, vol. 6, pp. 29 173–29 189, 2018.
- [9] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The Null-Space-based Behavioral control for autonomous robotic systems," *Journal of Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, Jan. 2008. [Online]. Available: <http://dx.doi.org/10.1007/s11370-007-0002-3>
- [10] K. Baizid, G. Giglio, F. Pierri, M. A. 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.
- [11] G. Muscio, F. Pierri, M. Trujillo, E. Cataldi, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Coordinated control of aerial robotic manipulators: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1406–1413, 2018.
- [12] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.
- [13] H. Azimian, T. Looi, and J. Drake, "Closed-loop inverse kinematics under inequality constraints: Application to concentric-tube manipulators," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 498–503.
- [14] 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.
- [15] S. Moe, G. Antonelli, A. Teel, K. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis and experimental results," *Frontiers in Robotics and AI*, vol. 3, p. 16, 2016.
- [16] A. Suarez, G. Heredia, and A. Ollero, "Physical-virtual impedance control in ultra-lightweight and compliant dual arm aerial manipulators," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2553–2560, 2018.
- [17] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, "6d physical interaction with a fully actuated aerial robot," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5190–5195.
- [18] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Verlag, 2009.
- [19] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1173–1182, Sept. 2009.
- [20] —, "The NSB control: a behavior-based approach for multi-robot systems," *Paladyn Journal of Behavioral Robotics*, vol. 1, no. 1, pp. 48–56, 2010.
- [21] S. Moe, A. Teel, G. Antonelli, and K. Pettersen, "Stability analysis for set-based control within the singularity-robust multiple task-priority inverse kinematics framework," in *54th IEEE Conference on Decision and Control and 8th European Control Conference*, Osaka, Jo, December 2015, pp. 171–178.
- [22] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng., "Ros: an open-source robot operating system," in *Open-source software workshop of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, J, 2009.