

A Framework for Set-based Kinematic Control of Multi-robot Systems^{*}

Paolo Di Lillo^{a,*}, Francesco Pierri^b, Gianluca Antonelli^a, Fabrizio Caccavale^b and Anibal Ollero^c

^aDepartment of Electrical and Information Engineering
University of Cassino and Southern Lazio
via Di Biasio 43, 03043, Cassino, Italy

^bAREA Laboratory
School of Engineering, University of Basilicata
via dell'Ateneo Lucano 10, 85100, Potenza, Italy

^cUniversity of Seville, 41004 Seville, Spain

ARTICLE INFO

Keywords:

Multi-robot systems
Multi-arm systems
Set-based kinematic control

ABSTRACT

In this paper, a novel control framework for coordinated motion for kinematically redundant multi-robot systems is developed. The framework embeds both tasks expressed as equality constraints and set-based tasks, i.e., tasks expressed via inequality constraints, in a task-priority kinematic control scheme. The effectiveness of the approach is experimentally demonstrated on two different multi-arm systems. The first system, aimed at operating in household tasks, is composed by two arms mounted on a mobile holonomic base, the second system is mounted on an aerial vehicle and is conceived to perform inspection and maintenance tasks.

1. Introduction

Many robotic tasks require the adoption of multiple robotic units. Notable examples are multi-arm systems, either ground-fixed or mounted on a mobile base (wheeled or aerial), and fleets of mobile agents. Thus, the coordinated motion control of the involved robotic agents for the execution of complex tasks has become subject of active research since many years.

When dealing with multiple robotic systems, proper handling of kinematic redundancy is needed to achieve safe and robust mission accomplishment, especially in environments co-habitated by humans and robots. Redundancy resolution methods have been widely studied since decades, both at kinematic and at dynamic level (Hollerbach and Ki Suh, 1987; Khatib, 1987). The classical approach is to exploit the system redundancy to perform multiple tasks arranged in a hierarchy with decreasing priority order (Nakamura et al., 1987; Siciliano and Slotine, 1991). In (Chiaverini, 1997) an approach that avoids algorithmic singularities has been proposed, then extended to handle multiple tasks in (Antonelli et al., 2008) with the Null Space based Behavioral (NSB) approach. It has been widely adopted both for manipulators and mobile robots (Antonelli et al., 2010). More recently, it was extended and experimentally validated to aerial manipulators in Baizid et al. (2017) and Muscio et al. (2018) to


achieve coordination between an aerial vehicle and a 6-DOF arm.


However, in the above mentioned contributions, the mission for the robotic system is specified by a set of tasks defined in terms of equality constraints. On the other hand, several tasks related to the safety of the robotic systems (e.g., joint limits avoidance) and/or of the surrounding environment and humans (e.g., virtual walls), can be more conveniently expressed via inequality constraints (hereafter *set-based* tasks), i.e. the value of the task function has to be kept in a range of values.

Handling of inequality constraints has been tackled in different ways, e.g. a widely adopted approach transforms inequality constraints into equivalent equality constraints or potential functions. In Escande et al. (2014) both equality and inequality constraints are solved by resorting to a hierarchical multiple least-squares quadratic programming (QP) problem. d QP-based algorithms represent an effective and powerful tool to tackle high-dimensional control problems for complex articulated systems (e.g., control of humanoid robots). However, since the solution is computed in an iterative way, a suitable strategy should be devised to tackle task switching, i.e. inserting and deleting tasks in the hierarchy and consequently changing the priority levels among them. In Azimian et al. (2014), the transformation of the inequality constraints into equality ones is achieved via slack variables. In Simetti et al. (2013) smooth potential fields are used to represent the set-based objectives and activation functions for an underwater vehicle-manipulator system, but strict priority among the tasks is not ensured, due to the presence of smoothing functions introduced to avoid discontinuities. In order to overcome these drawbacks, in Moe et al. (2016), a method that directly embeds set-based tasks into a singularity-robust multiple task-priority inverse kinematics framework, based on the NSB paradigm, was introduced and experimentally validated for a fixed-base manipulator with 7 DOFs. A recent extension on stability proof for the tracking

* Conflict of interest - none declared

* Corresponding author

 pa.dilillo@unicas.it (P. Di Lillo); francesco.pierri@unibas.it (F. Pierri); antonelli@unicas.it (G. Antonelli); fabrizio.caccavale@unibas.it (F. Caccavale); aollero@us.es (A. Ollero)

 www.paolodilillo.wordpress.com (P. Di Lillo);
<http://www2.unibas.it/automatica/staff.html> (F. Pierri);
<http://pages.unicas.it/gianluca-antonelli> (G. Antonelli);
<http://www2.unibas.it/automatica/staff.html> (F. Caccavale);
<https://grvc.us.es/head-of-the-group> (A. Ollero)

ORCID(s): 0000-0003-2083-1883 (P. Di Lillo); 0000-0002-8267-0512 (F. Pierri); 0000-0002-5511-0165 (G. Antonelli); 0000-0003-4225-0107 (F. Caccavale); 0000-0003-2155-2472 (A. Ollero)

problem has been presented in Arbo and Gravdahl (2018). The use of control barrier functions is proposed in Notomista et al. (2020) and in Basso and Pettersen (2020). More recently, set-based task-priority inverse kinematics has been proven to be effective on a dual-arm aerial manipulator in Cataldi et al. (2019).

In this paper, starting from the approach developed in Moe et al. (2016), a control framework for coordinated motion of a kinematically redundant multi-robot system is developed. The overall mission of a robotic system composed by subsystems (agents) is defined via a set of elementary tasks, arranged in a suitable priority order, classified in terms of their cooperative nature and their functional role. In detail, three different classes of tasks are defined (non-cooperative tasks, requiring loose and tight cooperation), depending on how they affect the motion of multiple robotic subsystems. Then, the tasks are grouped into three categories, depending on their functional role: safety, operational and optimization tasks (Di Lillo et al., 2018). Safety tasks are aimed at avoiding damage of the system and/or of the surrounding environment, and, thus, a higher priority with respect to the other tasks is usually assigned; safety tasks are set-based tasks activated only when the corresponding value exceeds a pre-defined activation threshold. Operational tasks, usually expressed as equality constraints, are those strictly required to achieve the assigned mission. Finally, optimization tasks are aimed at optimizing various indexes (e.g., dexterity for a robotic arm) useful to improve overall quality of the mission execution.

The adopted approach is a kinematic control scheme, i.e., the set-based task-priority inverse kinematics algorithm is used to compute the motion references for the low-level motion controller of the single robotic agents of the system. The choice of this approach is due to the specific features of many commercially available robotic systems, that do not provide torque interfaces, as those considered in the experimental study carried out in this paper. For a deep and complete comparison between torque-based and kinematic control approaches the interested reader is referred to Nakanishi et al. (2008).

The developed control framework is tested on two multi-robot systems, both involving a dual-arm system but addressing quite different environment to indirectly prove generalization of the proposed approach. The first system is an aerial dual-arm aerial manipulator, mounted on a multi-rotor vehicle, designed for inspection and maintenance tasks. The second system is a dual-arm mobile manipulator with holonomic mobile base operating in a domestic scenario.

In sum, a first contribution of this paper is the definition of a framework for coordinated control of multi-robot systems, based the approach developed in Moe et al. (2016), in order to solve the specific challenges related to multiple robotic systems, mainly related to design of the tasks and the hierarchies. Another contribution is the implementation and evaluation of the approach on two largely different robotic systems, which cover two separate domains (wheeled and aerial), so as to demonstrate the generality of the proposed

concept. Moreover, the implemented task hierarchies are quite rich, so as to demonstrate the effectiveness of the approach in realistic and non-trivial use cases.

2. Set-based Task-Priority Inverse Kinematics

In this Section, the set-based task-priority approach to kinematic control of robotic systems is reviewed. For a detailed description of the approach, the reader is referred to Moe et al. (2016).

A task for a robotic system is defined as a function of the system's state, $\mathbf{x} \in \mathbb{R}^n$, where n is the number of Degrees of Freedom (DOFs), as

$$\sigma = \mathbf{f}_s(\mathbf{x}) \in \mathbb{R}^m, \quad (1)$$

where m is the number of task variables. The mapping between the task velocity $\dot{\sigma}$ and the system velocity ζ can be expressed via the $(m \times n)$ task Jacobian matrix, \mathbf{J}_σ , as

$$\dot{\sigma} = \frac{\partial \mathbf{f}_s(\mathbf{x})}{\partial \mathbf{x}} \zeta = \mathbf{J}_\sigma \zeta. \quad (2)$$

Given a desired task trajectory $\sigma_d(t)$, the corresponding reference system's velocity, ζ_r , can be computed by resorting to a Closed-Loop Inverse Kinematics (CLIK) algorithm (Chilverini, 1997):

$$\zeta_r = \mathbf{J}_\sigma^\dagger (\dot{\sigma}_d + \mathbf{A} \tilde{\sigma}), \quad (3)$$

where $\mathbf{J}_\sigma^\dagger = \mathbf{J}_\sigma^T (\mathbf{J}_\sigma \mathbf{J}_\sigma^T)^{-1}$ is the Moore-Penrose pseudoinverse of the task Jacobian, $\mathbf{A} \in \mathbb{R}^{m \times m}$ is a positive-definite matrix of gains, and $\tilde{\sigma} = \mathbf{f}_e(\sigma_d, \sigma)$ is the task error.

In the presence of under-actuated systems, the system velocity can be partitioned into controlled and non-controlled velocities as

$$\zeta = \begin{bmatrix} \zeta_c \\ \zeta_u \end{bmatrix}. \quad (4)$$

The non-controlled variables can be taken into account in (3) as in Baizid et al. (2017)

$$\zeta_{c,r} = \mathbf{J}_{\sigma_c}^\dagger (\sigma_d + \mathbf{A} \tilde{\sigma} - \mathbf{J}_{\sigma_u} \zeta_u), \quad (5)$$

where \mathbf{J}_{σ_c} and \mathbf{J}_{σ_u} are the task Jacobians referred to controlled and uncontrolled variables, respectively.

When the number of system DOFs, n , is larger than the task dimension m , the system is redundant with respect to the task and multiple tasks can be commanded simultaneously. In order to solve the potential conflicts that could arise among the tasks, a common solution is to define a priority among them and compute the reference system velocity that fulfills the task hierarchy *at best*, removing from the solution the velocity components of the lower-priority tasks that conflict with the higher priority ones. In this way, the optimal execution of the primary task is ensured, while all the lower-priority ones are executed only for those components that do not affect the higher-priority tasks.

In order to combine a hierarchy of h tasks, it is necessary to assign h levels of priority (hereafter indexed by the integer i), from the primary one ($i = 1$) to the lowest priority one ($i = h$). Then, the reference system velocity can be computed recursively as in Siciliano and Slotine (1991)

$$\zeta_{r,h} = \sum_{i=1}^h (\mathbf{J}_i \mathbf{N}_{i-1}^A)^\dagger (\dot{\boldsymbol{\sigma}}_{i,d} + \mathbf{A}_i \tilde{\boldsymbol{\sigma}}_i - \mathbf{J}_i \zeta_{i-1}), \quad (6)$$

where \mathbf{N}_i^A projects a vector onto the null space of the augmented Jacobian matrix, obtained by stacking all the task Jacobian matrices from task 1 to i , i.e.,

$$\mathbf{J}_i^A = [\mathbf{J}_1^T \quad \mathbf{J}_2^T \quad \dots \quad \mathbf{J}_i^T]^T, \quad (7)$$

and is defined as

$$\mathbf{N}_i^A = \mathbf{I} - (\mathbf{J}_i^A)^\dagger \mathbf{J}_i^A, \quad (8)$$

with $\mathbf{N}_0^A = \mathbf{I}$ (the identity matrix of suitable dimensions), $\zeta_0 = \mathbf{0}_n$ (the $(n \times 1)$ null vector), and ζ_i is the velocity contribution of the i -th task in the hierarchy.

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*. Recently, in (Moe et al., 2016), this framework has been extended to handle also *set-based tasks*, in which the task value has to be kept in a certain set of values delimited by two *activation thresholds*: a lower threshold ($\sigma_{a,l}$) and an upper threshold ($\sigma_{a,u}$). The key idea is to consider a set-based task as an equality-based one that can be inserted or removed from the hierarchy depending on the operational conditions. In fact, a set-based task has to be activated when its value is near one of the activation thresholds, while it has to be deactivated when the solution (6) that does not contain it pushes its value toward the valid set. Once activated, a set-based task is inserted in the task hierarchy as an equality-based task with desired value equal to a predefined *safety threshold* ($\sigma_{s,u}$ or $\sigma_{s,l}$), as shown in Fig. 1.

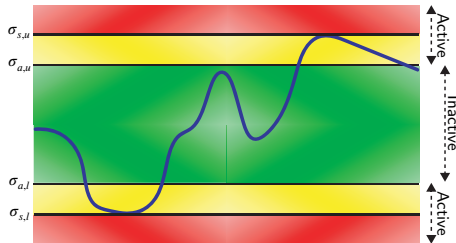


Figure 1: Evolution of a generic set-based task, highlighting the *activation thresholds* ($\sigma_{a,u}$ and $\sigma_{a,l}$) and the *safety thresholds* ($\sigma_{s,u}$ and $\sigma_{s,l}$)

Since a set-based task can be either active or inactive, a system with q set-based tasks is characterized by 2^q possible combinations of set-based tasks being active/inactive, referred to as *modes* of the system. The algorithm developed in Moe et al. (2016) 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^q -th mode all set-based tasks are active.

More details about the activation/deactivation algorithm can be found in Di Lillo et al. (2020).

3. Framework for cooperative multi-robot systems

When a robotic system is composed by multiple subsystems (or agents) that are required to cooperate to achieve common mission goals (e.g. multi-arm systems for cooperative manipulation/transportation of objects, fleets of coordinated mobile robots), the set-based kinematic control approach has to be suitably enriched to take into account the different ways in which the robotic agents composing the system interact.

The developments reported in the following extend some basic concepts related to coordinated control of cooperative manipulators, that can be found, e.g., in (Caccavale and Uchiyama, 2016), to general multi-robot systems.

Let us consider a robotic system composed by N subsystems, each having n_i ($i = 1 \dots N$) DOFs. The entire system's state is represented by the vector stacking the state variables of all the subsystems

$$\mathbf{x} = [\mathbf{x}_1^T \quad \mathbf{x}_2^T \quad \dots \quad \mathbf{x}_N^T]^T \in \mathbb{R}^n, \quad (9)$$

where $n = \sum_{i=1}^N n_i$. A generic m -dimensional task function $\boldsymbol{\sigma}_x$ depends, in general, on the state of the agents composing the multi-robot system

$$\boldsymbol{\sigma}_x = \mathbf{f}_s(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N), \quad (10)$$

or only on the state of a subset of agents. The assigned target value for the task $\boldsymbol{\sigma}_{x,d} = \mathbf{f}_d(t, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ can depend on the state of (a subset of) the other agents as well.

The task Jacobian matrix is composed as:

$$\mathbf{J}_x = \begin{bmatrix} \frac{\partial \mathbf{f}_s(\mathbf{x})}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_s(\mathbf{x})}{\partial \mathbf{x}_2} & \dots & \frac{\partial \mathbf{f}_s(\mathbf{x})}{\partial \mathbf{x}_N} \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad (11)$$

in which the i -th block represents the derivative of the task function with respect to the state variables of the i -th subsystem.

In order to devise a general framework for multi-robot systems, three possible levels of cooperation between subsystems required for the accomplishment of a task $\boldsymbol{\sigma}_x$ are defined:

- non-cooperative tasks
- tasks involving loose cooperation
- tasks involving tight cooperation.

In the following, the three levels are defined with reference to two agents, labeled by the subscripts i and j , respectively, for the sake of clarity. The extension to the case of three or more agents is straightforward.

A task does not require cooperation among subsystems (*non-cooperative task*) if its actual value depends on the state of only one subsystem

$$\sigma_x = f_s(\mathbf{x}_i), \quad (12)$$

and its target value does not depend on the state of the other agents. Thus, the corresponding velocity commands computed via (6) do not affect the state of the other agents. Examples of tasks belonging to this class are joint limits or obstacle avoidance for a single agent.

A task requires *loose cooperation* between two subsystems i and j if its actual value depends on the state of i , but its target value depends on the state of another agent j , i.e.,

$$\sigma_x = f_s(\mathbf{x}_i) \wedge \sigma_{x,d} = f_d(t, \mathbf{x}_j). \quad (13)$$

In other words, while the task is determined by the state of the corresponding agent (e.g., the end-effector pose of a robotic arm is determined by its joint variables), the target value depends on the state of other agents (e.g., when one robot is equipped with a camera required to keep in its field of view a reference point on another robotic system). Thus, coordination between the agents is achieved only via the target value in a master-slave fashion. In case of non-cooperative tasks and tasks involving loose cooperation, the Jacobian matrix has non-zero elements only in the columns corresponding to the i th subsystem

$$\mathbf{J}_x = \begin{bmatrix} \mathbf{O} & \dots & \frac{\partial f_s(\mathbf{x})}{\partial \mathbf{x}_i} & \dots & \mathbf{O} \end{bmatrix}. \quad (14)$$

The task requires a *tight cooperation* between the subsystems i and j if its actual value depends on the state of the two subsystems

$$\sigma_x = f_s(\mathbf{x}_i, \mathbf{x}_j). \quad (15)$$

Thus, the task requires intrinsic coordination between the agents. In this case, the Jacobian matrix has non-zero elements in the blocks corresponding to the involved subsystems

$$\mathbf{J}_x = \begin{bmatrix} \mathbf{O} & \dots & \frac{\partial f_s(\mathbf{x})}{\partial \mathbf{x}_i} & \dots & \frac{\partial f_s(\mathbf{x})}{\partial \mathbf{x}_j} & \dots & \mathbf{O} \end{bmatrix}. \quad (16)$$

This is the case in which multiple robots have to operate cooperatively to reach a common goal, e.g. the cooperative transportation of an object.

It is possible to additionally categorize each level of cooperation depending on the functional role of the task, obtaining three groups: *safety*, *operational* and *optimization tasks*. The level of priority of each task in the overall mission hierarchy can be related to both the functional role and the level of cooperation required for the task. The priority level among the functional roles is, then, usually set as

Table 1

Priority of tasks depending on the functional role and the required level of cooperation.

Functional role	Level of cooperation	Priority
Safety	Tight	1
	Loose	2
	None	3
Operational	Tight	4
	Loose	5
	None	6
Optimization	Tight	7
	Loose	8
	None	9

1. Safety tasks, which include those tasks aimed at avoiding damages of the system and/or of objects and humans in the surrounding environment. They are all defined as set-based tasks and activated only when the corresponding value exceeds the pre-defined activation thresholds, with higher priority with respect to the other tasks.
2. Operational tasks, i.e. those tasks strictly needed for the accomplishment of the assigned mission.
3. Optimization tasks, whose goal is to adjust the configuration of the system so as to keep the system far from singularity as well as unsafe configurations. They strongly depend on the system peculiarities.

Within the same functional role, it is necessary to set further priorities among the different levels of cooperation, obtaining the overall hierarchy summarized in Table 1.

In the following, the framework is used to design the kinematic control of two mobile robotic systems: a dual-arm aerial manipulator and a dual-arm wheeled manipulator. In both cases, the cooperative agents are manipulators mounted on a moving base, but the challenges to be tackled are different, in terms of mission goals and requirements.

4. Kinematics of dual-arm mobile robot

The framework described in Section 3 can be re-arranged to the case of a robotic system composed by a mobile base characterized by $n_V = 6$ DOFs and two manipulators, each characterized by n_* DOFs ($\star = L, R$).

Therefore, the system's state is described by the vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_V^T & \boldsymbol{\phi}_V^T & \mathbf{q}_L^T & \mathbf{q}_R^T \end{bmatrix}^T \in \mathbb{R}^\mu, \quad (17)$$

where $\mu = n_V + n_L + n_R$, \mathbf{p}_V is the position of the mobile base in inertial frame, $\boldsymbol{\phi}_V$ is the platform orientation, expressed,

e.g., in Euler angles, q_L and q_R are the joint positions of the left and right manipulator, respectively. The system velocity vector is

$$\zeta = \begin{bmatrix} v_V^T & \dot{\phi}_V^T & \dot{q}_L^T & \dot{q}_R^T \end{bmatrix}^T \in \mathbb{R}^\mu, \quad (18)$$

where v_V is the linear velocity of the platform, $\dot{\phi}_V$ is the derivative of the Euler angles, related to the angular velocity, \dot{q}_L and \dot{q}_R are the left and right arm joint velocities, respectively. The Jacobian matrix of a generic task σ_x can be expressed as

$$J_x = \begin{bmatrix} \underbrace{\begin{bmatrix} \frac{\partial f_s(x)}{\partial p_V} & \frac{\partial f_s(x)}{\partial \phi_V} \end{bmatrix}}_{m \times n_V} & \underbrace{\begin{bmatrix} \frac{\partial f_s(x)}{\partial q_L} \end{bmatrix}}_{m \times n_L} & \underbrace{\begin{bmatrix} \frac{\partial f_s(x)}{\partial q_R} \end{bmatrix}}_{m \times n_R} \end{bmatrix} \in \mathbb{R}^{m \times \mu}, \quad (19)$$

where the contributions of the mobile base, the left and the right arms to the task velocity with their dimensions are highlighted.

Let us define six frames of interest for the tasks Jacobian matrices definition conducted in the following Section

- the fixed inertial frame \mathcal{F} ;
- the mobile base frame \mathcal{F}_V , attached to the center of mass of the mobile base;
- the left arm base frame $\mathcal{F}_{L,B}$;
- the right arm base frame $\mathcal{F}_{R,B}$;
- the left end-effector frame, $\mathcal{F}_{L,E}$;
- the right end-effector frame, $\mathcal{F}_{R,E}$.

The transformation matrix that expresses the homogeneous transformation between frame \mathcal{F}_X and \mathcal{F}_Y is $T_X^Y \in \mathbb{R}^{4 \times 4}$ and it is composed by the rotation matrix $R_X^Y \in \mathbb{R}^{3 \times 3}$ and the position vector $p_X^Y \in \mathbb{R}^3$. In the following, the superscript will be omitted when a quantity is expressed in the inertial frame.

5. Definition of tasks

5.1. Safety tasks

Joint limits ($m = 1$): upper and lower limits have to be placed on each manipulators' joint subject to mechanical limits. The task function is the value of the i -th joint variable:

$$\sigma_{L,joint,i} = q_{L,i}, \quad \sigma_{R,joint,i} = q_{R,i}, \quad (20)$$

while the task Jacobians are given by

$$J_{L,joint,i} = \begin{bmatrix} \underbrace{\begin{bmatrix} \mathbf{0}_{n_V} \end{bmatrix}}_{\text{base}} & \underbrace{\begin{bmatrix} 0 & \dots & 1 & \dots & 0 \end{bmatrix}}_{\text{left arm}} & \underbrace{\begin{bmatrix} \mathbf{0}_{n_R} \end{bmatrix}}_{\text{right arm}} \end{bmatrix}$$

(21)

$$J_{R,joint,i} = \begin{bmatrix} \underbrace{\begin{bmatrix} \mathbf{0}_{n_V} \end{bmatrix}}_{\text{base}} & \underbrace{\begin{bmatrix} \mathbf{0}_{n_L} \end{bmatrix}}_{\text{left arm}} & \underbrace{\begin{bmatrix} 0 & \dots & 1 & \dots & 0 \end{bmatrix}}_{\text{right arm}} \end{bmatrix} \quad (22)$$

the task is non-cooperative, since the task function and its desired value do not depend on the state of the mobile base or the other arm.

Virtual wall ($m = 1$): a virtual wall could be placed in order to limit the arms' workspace in such a way to avoid collisions and/or singular configurations. The task function is defined as ($*$ = L,R)

$$\sigma_{\text{wall}} = \hat{n}^T (p_{*E}^{*,B} - p_1), \quad (23)$$

where $p_{*E}^{*,B}$ is the position of the considered end effector with respect to its base frame, and

$$\hat{n} = \frac{(p_2 - p_1) \times (p_3 - p_1)}{\|(p_2 - p_1) \times (p_3 - p_1)\|}, \quad (24)$$

is the outer normal unit vector of the plane determined by the three points p_1 , p_2 and p_3 . The task Jacobians for the left and right arm are given by

$$J_{L,\text{wall}} = \begin{bmatrix} \underbrace{\begin{bmatrix} \mathbf{0}_{n_V} \end{bmatrix}}_{\text{base}} & \underbrace{\begin{bmatrix} -\hat{n}^T J_{L,\text{pos}}^{L,B} \end{bmatrix}}_{\text{left arm}} & \underbrace{\begin{bmatrix} \mathbf{0}_{n_R} \end{bmatrix}}_{\text{right arm}} \end{bmatrix} \quad (25)$$

$$J_{R,\text{wall}} = \begin{bmatrix} \underbrace{\begin{bmatrix} \mathbf{0}_{n_V} \end{bmatrix}}_{\text{base}} & \underbrace{\begin{bmatrix} \mathbf{0}_{n_L} \end{bmatrix}}_{\text{left arm}} & \underbrace{\begin{bmatrix} -\hat{n}^T J_{R,\text{pos}}^{R,B} \end{bmatrix}}_{\text{right arm}} \end{bmatrix}, \quad (26)$$

where $J_{L,\text{pos}}^{L,B}$ and $J_{R,\text{pos}}^{R,B}$ are the position Jacobian matrices of the left and right arms, respectively, expressed in their base frames. The task is non-cooperative, since the task and its desired value do not depend on the state of the other agents.

5.2. Operational tasks

In this paper, it has been assumed that a typical task involving a mobile dual arm manipulator requires to assign:

- the pose of the end effector of one of the two arms (in the following, for the sake of simplicity, it has been assumed the left one) in the inertial frame (*global pose*), expressed as a function of the mobile base pose and the arm joint variables;
- the pose of the second end effector (in the following, for the sake of simplicity, it has been assumed the right one) constrained, e.g., by assigning the relative pose of the two end-effectors expressed in the arm base frame (*Relative Pose*) or by equipping it with a directional sensor whose orientation with respect to the other end effector has to be kept fixed (e.g. by keeping the first end effector in the *Field of View* of a camera mounted on the second end effector).

Of course, such a selection is not to be considered exhaustive, since new tasks could be defined in order to face different ways of assigning a coordinated motion (Caccavale and Uchiyama, 2016).

Global pose ($m = 6$): the task is the vector stacking the position, $\mathbf{p}_{L,E}$, and the unit quaternion, $\mathbf{Q}_{L,E}$, (Roberson and Schwertassek, 1988) of the left end effector with respect to the inertial frame

$$\boldsymbol{\sigma}_{\text{global}} = \begin{bmatrix} \mathbf{p}_{L,E} \\ \mathbf{Q}_{L,E} \end{bmatrix}. \quad (27)$$

The position vector $\mathbf{p}_{L,E}$ can be written as

$$\begin{cases} \mathbf{p}_{L,E} = \mathbf{p}_V + \mathbf{R}_V \mathbf{p}_{L,E}^V \\ \mathbf{R}_{L,E} = \mathbf{R}_V \mathbf{R}_{L,E}^V, \end{cases} \quad (28)$$

where $\mathbf{p}_{L,E}^V$ and $\mathbf{R}_{L,E}^V = \mathbf{R}_{L,B}^V \mathbf{R}_{L,E}^{L,B}$ are the relative position and orientation of the frame $\mathcal{F}_{L,E}$ with respect to the frame \mathcal{F}_V , expressed in frame \mathcal{F}_V . The time derivative of (28) is then given by

$$\begin{cases} \dot{\mathbf{p}}_{L,E} = \dot{\mathbf{p}}_V - \mathcal{S}(\mathbf{R}_V \mathbf{p}_{L,E}^V) \boldsymbol{\omega}_V + \mathbf{R}_V \dot{\mathbf{p}}_{L,E}^V \\ \dot{\boldsymbol{\omega}}_{L,E} = \boldsymbol{\omega}_V + \mathbf{R}_V \dot{\boldsymbol{\omega}}_{L,E}^V \end{cases} \quad (29)$$

where $\mathcal{S}(\cdot)$ is the skew symmetric operator performing the cross-product (Siciliano et al., 2009) and $\dot{\mathbf{p}}_{L,E}^V$ and $\dot{\boldsymbol{\omega}}_{L,E}^V$ are the relative velocities of $\mathcal{F}_{L,E}$ with respect to the frame \mathcal{F}_V expressed in frame \mathcal{F}_V , which can be rearranged as the standard manipulator differential kinematics. Taking into account (29) and the relationship between the mobile base angular velocity and the derivative of its Euler angles, $\boldsymbol{\omega}_V = \mathbf{T}(\boldsymbol{\phi}_V) \dot{\boldsymbol{\phi}}_V$ (Siciliano et al., 2009), the global pose task Jacobian can be expressed as:

$$\mathbf{J}_{\text{global}} = \begin{bmatrix} \overbrace{\mathbf{J}_{\text{inertial,L}}^{\text{base \& left arm}}} & \overbrace{\mathbf{O}_{6 \times n_R}^{\text{right arm}}} \end{bmatrix} \quad (30)$$

where $\mathbf{O}_{\alpha \times \beta}$ is the $(\alpha \times \beta)$ null matrix and

$$\mathbf{J}_{\text{inertial,L}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathcal{S}(\mathbf{R}_V \mathbf{p}_{L,E}^V) \mathbf{T}(\boldsymbol{\phi}_V) & \mathbf{R}_{L,B} \mathbf{J}_{L,\text{pos}}^{L,B} \\ \mathbf{O}_{3 \times 3} & \mathbf{T}(\boldsymbol{\phi}_V) & \mathbf{R}_{L,B} \mathbf{J}_{L,\text{ori}}^{L,B} \end{bmatrix}, \quad (31)$$

where $\mathbf{I}_{\alpha \times \beta}$ is the $(\alpha \times \beta)$ identity matrix while $\mathbf{J}_{L,\text{pos}}^{L,B}$ and $\mathbf{J}_{L,\text{ori}}^{L,B}$ are, respectively, the position and orientation Jacobian matrices of the left manipulator expressed in its base frame (Siciliano et al., 2009). This task requires tight cooperation between the mobile base and the left arm, as $\boldsymbol{\sigma}_{\text{global}}$ depends on the state of the two agents and $\mathbf{J}_{\text{global}}$ has non-zero elements on both the blocks of the two subsystems. It is worth noticing that this task has also a non-cooperative formulation, in which the reference is expressed in one of the arm base frames and the inverse kinematics solution involves only that arm. In such case the jacobian matrix is composed as:

$$\mathbf{J}_{\text{pose,L}} = \begin{bmatrix} \mathbf{J}_{L,\text{pos}}^{L,B} \\ \mathbf{J}_{L,\text{ori}}^{L,B} \end{bmatrix}, \quad \mathbf{J}_{\text{pose,R}} = \begin{bmatrix} \mathbf{J}_{R,\text{pos}}^{R,B} \\ \mathbf{J}_{R,\text{ori}}^{R,B} \end{bmatrix}. \quad (32)$$

When dealing with mobile manipulators, it is often useful to weight differently the motion contribution of the mobile base and of the arms. This is usually achieved by using the weighted pseudoinverse of the Jacobian matrix, defined as

$$\mathbf{J}_{\text{global}}^{\dagger W} = \mathbf{W}^{-1} \mathbf{J}_{\text{global}}^T \left(\mathbf{J}_{\text{global}} \mathbf{W}^{-1} \mathbf{J}_{\text{global}}^T \right)^{-1}, \quad (33)$$

where \mathbf{W} is a positive definite diagonal matrix of weights that can be split into three diagonal matrices

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{\text{base}} & \mathbf{O}_{n_V \times n_L} & \mathbf{O}_{n_V \times n_R} \\ \mathbf{O}_{n_L \times n_V} & \mathbf{W}_L & \mathbf{O}_{n_L \times n_R} \\ \mathbf{O}_{n_R \times n_V} & \mathbf{O}_{n_R \times n_L} & \mathbf{W}_R \end{bmatrix} \in \mathbb{R}^{\mu \times \mu}, \quad (34)$$

where \mathbf{W}_{base} gathers the weights of the DOFs of the vehicle, \mathbf{W}_L and \mathbf{W}_R the ones of the two arms. It is worth noticing that \mathbf{W}_R does not actually play any role in the computation of $\mathbf{J}_{\text{global}}^{\dagger W}$ as the columns of $\mathbf{J}_{\text{global}}$ corresponding to the right arm joints are null.

In this work, the user-defined weights in \mathbf{W}_{base} and \mathbf{W}_L are heuristically selected in such way to always penalize the base movement. The behavior that we want to have is that the robot makes use mostly of the arms to reach a target location, until the end-effector reaches a virtual wall that surrounds the arm base frame. When this happens, the corresponding high-priority task gets active, stopping the arm motion and the robot is forced to use the base to move toward the desired location, despite of the relative weighting between the base and the arm joints.

Relative pose ($m = 6$): the task function is the vector stacking the position and orientation of the left end effector with respect to the right one

$$\boldsymbol{\sigma}_{\text{rel}} = \begin{bmatrix} \mathbf{p}_{R,E}^{L,B} - \mathbf{p}_{L,E}^{L,B} \\ \mathbf{Q}_{L,E}^{R,E} \end{bmatrix}, \quad (35)$$

where $\mathbf{p}_{L,E}^{L,B}$ and $\mathbf{p}_{R,E}^{L,B}$ are the position of the left and right end-effectors, respectively, expressed in a common reference frame (the left arm base frame), and $\mathbf{Q}_{L,E}^{R,E}$ is the unit quaternion expressing the orientation of the left end-effector with respect to the right one. The expression of the relative Jacobian without considering the mobile base is (Jamisola and Roberts, 2015)

$$\mathbf{J}_{\text{rel, fixed}} = \begin{bmatrix} -\boldsymbol{\Psi}_{R,E}^{L,E} \boldsymbol{\Omega}_{L,E}^{L,E} \mathbf{J}_L^{L,B} & \boldsymbol{\Omega}_{L,E}^{R,B} \mathbf{J}_R^{R,B} \end{bmatrix} \in \mathbb{R}^{6 \times n_L + n_R}, \quad (36)$$

where:

$$\boldsymbol{\Psi}_X^Y = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathcal{S}(\mathbf{p}_X^Y) \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad \boldsymbol{\Omega}_X^Y = \begin{bmatrix} \mathbf{R}_X^Y & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{R}_X^Y \end{bmatrix} \quad (37)$$

and $\mathbf{J}_L^{L,B}$, $\mathbf{J}_R^{R,B}$ are the position and orientation Jacobian matrices of the left and right arms expressed in their base frames. The Jacobian of the relative pose task is then composed as

$$\mathbf{J}_{\text{rel}} = \left[\overbrace{\mathbf{O}_{6 \times n_V}}^{\text{base}} \quad \overbrace{\mathbf{J}_{\text{rel, fixed}}}_{\text{left \& right arms}} \right]. \quad (38)$$

Clearly, this task requires tight cooperation between the two arms.

Field of View ($m = 3$): Directional devices or sensors mounted on the end effector of the right arm, such as a laser or a camera, do not need to constrain the whole end-effector orientation, but only the outgoing unit vector, which is required to point toward a target location. By defining the task function as the outgoing unit vector of the right end effector expressed in the inertial frame

$$\boldsymbol{\sigma}_{\text{FOV}} = \mathbf{R}_{R,E} \mathbf{e}_3, \quad (39)$$

where $\mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, the task Jacobian, without considering the mobile base and the left arm, can be computed as

$$\mathbf{J}_{R,\text{FOV}} = -\mathcal{S}(\boldsymbol{\sigma}_{\text{FOV}}) \mathbf{J}_{R,\text{ori}}^{R,B}. \quad (40)$$

In the case of a dual-arm manipulator, the desired value for the task function could be the unit vector

$$\boldsymbol{\sigma}_{\text{FOV,d}} = \frac{\mathbf{p}_{L,E} - \mathbf{p}_{R,E}}{\|\mathbf{p}_{L,E} - \mathbf{p}_{R,E}\|}, \quad (41)$$

representing the direction from the right end effector to the left one. The Jacobian of the field of view task is then composed as

$$\mathbf{J}_{\text{FOV}} = \left[\overbrace{\mathbf{O}_{3 \times n_V}}^{\text{base}} \quad \overbrace{\mathbf{O}_{3 \times n_L}}^{\text{left arm}} \quad \overbrace{\mathbf{J}_{R,\text{FOV}}}_{\text{right arm}} \right] \in \mathbb{R}^{3 \times \mu}. \quad (42)$$

Clearly, this task requires loose cooperation between the two arms, as $\boldsymbol{\sigma}_{\text{FOV,d}}$ depends on the state of the other arm and \mathbf{J}_{FOV} has only one non-zero block.

5.3. Optimization tasks

Manipulability ($m = 1$): This task is aimed at ensuring that the system's configuration is far from kinematic singularities. To define the task function, the usual manipulability measure for a robot manipulator defined in Yoshikawa (1985) as $w(\mathbf{x}) = \sqrt{\det(\mathbf{J}_m \mathbf{J}_m^T)}$ could be adopted, where \mathbf{J}_m is the following Jacobian

$$\mathbf{J}_m = \begin{bmatrix} \mathbf{J}_{\text{inertial,L}} \\ \mathbf{J}_{\text{inertial,R}} \end{bmatrix}. \quad (43)$$

The Jacobian matrices in \mathbf{J}_m have been computed by resorting to the International Standard of Units to avoid numerical issues (Lin and Burdick, 2000), in particular lengths in meters and angles in radians, and then normalized in order to

make the measure of manipulability independent from units. Given this assumption, the measure of manipulability is considered unitless in the following. The task function is

$$\sigma_{\text{man}} = w(\mathbf{x}) \geq 0, \quad (44)$$

and the corresponding Jacobian is

$$\mathbf{J}_{\text{man}} = \left[\overbrace{\frac{\partial w(\mathbf{x})}{\partial \mathbf{p}_V}}^{\text{base}} \quad \overbrace{\frac{\partial w(\mathbf{x})}{\partial \boldsymbol{\phi}_V}}^{\text{left arm}} \quad \overbrace{\frac{\partial w(\mathbf{q}_L)}{\partial \mathbf{q}_L}}^{\text{right arm}} \quad \overbrace{\frac{\partial w(\mathbf{x})}{\partial \mathbf{q}_R}} \right] \in \mathbb{R}^{1 \times \mu}; \quad (45)$$

This task requires tight cooperation between the two arms and the base. It is worth noticing that this kind of task can be considered both at a low priority level as an equality-based task and at a high priority level as a set-based task, as done, e.g., in (Di Lillo et al., 2019). In the experiments described in the following sections we consider it as a low-priority optimization task, aiming at maximizing its value as much as possible given the execution of all the higher-priority ones.

Mobile base configuration optimization: This task is aimed at assigning a particular configuration of the mobile base with respect to an auxiliary frame, \mathcal{F}_A , suitably defined. The mobile base configuration optimization task value can be defined as

$$\boldsymbol{\sigma}_{\text{opt}} = \begin{bmatrix} \mathbf{L}_p \mathbf{p}_V^A \\ \mathbf{L}_o \boldsymbol{\phi}_V^A \end{bmatrix} \in \mathbb{R}^{(m_p+m_o) \times 1}, \quad (46)$$

where \mathbf{p}_V^A and $\boldsymbol{\phi}_V^A$ are the position and orientation of the mobile base frame with respect to the auxiliary frame, $\mathbf{L}_p \in \mathbb{R}^{m_p \times 3}$ and $\mathbf{L}_o \in \mathbb{R}^{m_o \times 3}$ are matrices that suitably select the components to constraint.

The Jacobian of the task, without considering the two arms, is

$$\mathbf{J}_{\text{opt, base}} = \begin{bmatrix} \mathbf{L}_p \mathbf{R}_A^T & \mathbf{O}_{m_p \times 3} \\ \mathbf{O}_{m_o \times 3} & \mathbf{L}_o \mathbf{R}_A^T \mathbf{T}(\boldsymbol{\phi}_V) \end{bmatrix} \in \mathbb{R}^{(m_p+m_o) \times n_V}. \quad (47)$$

The optimization task Jacobian is then composed as

$$\mathbf{J}_{\text{opt}} = \left[\overbrace{\mathbf{J}_{\text{opt, base}}}^{\text{base}} \quad \overbrace{\mathbf{O}_{(m_p+m_o) \times n_L + n_R}}^{\text{left \& right arms}} \right]. \quad (48)$$

Center of mass ($m = 2$) This task is aimed at ensuring that the Center of Mass (CoM) of the dual-arm system is as much as possible aligned with that of the mobile base, in such a way to avoid to destabilize the motion. The task function is

$$\sigma_{\text{CoM}} = \left\| \begin{bmatrix} c_x^V & c_y^V \end{bmatrix} \right\|, \quad (49)$$

where c_{\star}^V ($\star = x, y$) is the component of the position of the CoM of the system composed by the two arms along the \star axis of frame \mathcal{F}_V

$$c_{\star}^V = \frac{\sum_{i=1}^{n_L} m_{L,i} \rho_{L,i,\star}^V + \sum_{j=1}^{n_R} m_{R,j} \rho_{R,j,\star}^V}{m_L + m_R}, \quad (50)$$

where $m_{L,i}$ and $m_{R,j}$ are the masses of the links of the left and right manipulators, respectively, $\rho_{L,i,\star}^V$ and $\rho_{R,j,\star}^V$ are the \star component of the CoM of the i -th link of the left manipulator and the j -th link of the right manipulator, respectively, expressed in the frame \mathcal{F}_V . Finally, m_L and m_R are the masses of the two manipulators. The task Jacobian is given by

$$\mathbf{J}_{\text{CoM}} = \begin{bmatrix} \underbrace{\mathbf{0}_{n_V}}_{\text{base}} & \underbrace{\mathbf{J}_{\text{CoM}}^L}_{\text{left arm}} & \underbrace{\mathbf{J}_{\text{CoM}}^R}_{\text{right arm}} \end{bmatrix} \in \mathbb{R}^{1 \times \mu}. \quad (51)$$

where $\mathbf{J}_{*,\text{CoM}}$ ($*=L,R$) is given by

$$\mathbf{J}_{*,\text{CoM}} = \frac{c_x^V \sum_{i=1}^{n_*} j_{*,1}^{(\ell_i)} + c_y^V \sum_{i=1}^{n_*} j_{*,2}^{(\ell_i)}}{(m_L + m_R) \sqrt{c_x^{V2} + c_y^{V2}}} \quad (52)$$

with $j_{*,1}^{(\ell_i)}$ and $j_{*,2}^{(\ell_i)}$ are the contributions of the first and second row, respectively, of the Jacobian \mathbf{J}_{*}^V , relative to the joint velocities up to the link i . This task requires tight cooperation between the two arms.

6. Experiments on an aerial dual-arm manipulator

The task-priority control scheme described in Section 2 has been applied to a mobile manipulator composed by an under-actuated multirotor aerial platform, characterized by 4 DOFs, and two anthropomorphic arms, each with 4 DOFs. The system and the coordinate frames described in Section 3, is shown in Figure 6.1. In the following, two different experimental case studies are presented: the first one, firstly presented in Cataldi et al. (2019), requires a loose cooperation of the arms while no cooperation is required in the second case study. A video showing the execution of the operation is provided as supplementary material for the present paper.

6.1. Anthropomorphic Dual Arm

The manipulator integrated in the aerial platform is the anthropomorphic, compliant and lightweight dual arm proposed in (Suarez et al., 2018a), whose main parameters are summarized in Table 2. 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. Each arm, equipped with three joints at the shoulder and one at the elbow, provides 4 DOFs for end effector positioning in a human-like kinematic configuration.

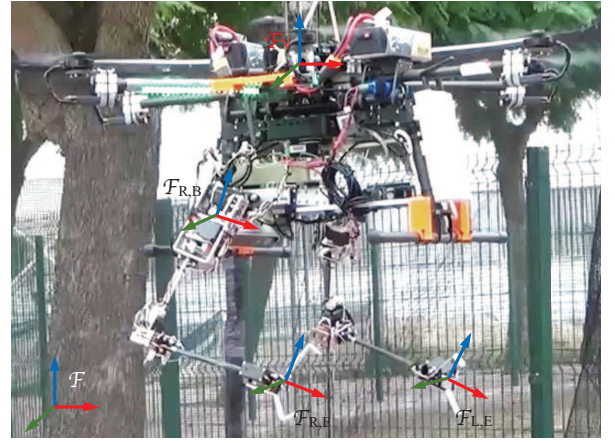


Figure 2: Dual-arm aerial manipulator and considered coordinate frames

Table 2

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	± 90 deg
Shoulder roll range	$[-30, 120]$ deg
Shoulder yaw range	± 90 deg
Elbow pitch range	± 135 deg

6.2. Aerial Platform

The aerial platform is an hexarotor manufactured by Drone Tools¹, whose main specifications are summarized in Table 3. 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 to 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 (Suarez et al., 2018a). The configuration in Figure 6.1 corresponds to the nominal operational position, while in the take-off and landing the arms rotate ± 90 deg around the shoulder roll joint (Suarez et al., 2017).

6.3. Hardware/Software control architecture

The low level control of the aerial platform is based on an integral backstepping algorithm including the dynamic model of the system, the compensation of both the arms' movements and the interaction wrenches with the environment (Suarez et al., 2018b). It has been 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

¹<https://www.dronetools.es/>

Table 3

Main specifications of the aerial platform.

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

executes three main software modules: the UAV Abstraction Layer (UAL), the dual-arm control program and the aerial manipulator control module, which implements the inverse kinematics methods detailed in Section 2. 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 in such a way to command the joints to track the references received from the aerial manipulator controller.

6.4. First case study

In the experiments the actual position of the vehicle has been obtained via a motion capture system (VICON), while its orientation is measured by the Inertial Measurement Unit and the joint positions are provided by the servos. In the following, the implemented tasks are detailed and classified into the categories proposed in Section 3.

According to the assigned mission, the left arm is commanded to track a desired trajectory, while the right arm is in charge of moving a camera, attached to its end-effector, in such a way that its outgoing vector is aligned with the outgoing vector of the left arm's end-effector. At lower priority two optimization tasks are implemented, i.e., the center of mass, aimed at ensuring that the center of mass of the dual-arm system is, as much as possible, aligned with that of the aerial platform, in such a way to avoid to destabilize the flight and reduce the power consumption, and the manipulability of the arms. For preserving the system integrity, safety tasks are implemented as set-based tasks: the joint limits for all the arms' joints, virtual walls between the two arms and between the arms and the vehicle.

The overall implemented task hierarchy is composed as follows:

1. Joint limits on the four joints of both arms (safety, non-cooperative, 8 tasks, $m = 1$ each);
2. Virtual walls between the arms (safety, non-cooperative, 2 tasks, $m = 1$ each) and between the arms and the aerial platform (safety, non-cooperative, 2 tasks, $m = 1$ each);
3. Global pose of the left end-effector (operational, tight cooperation, $m = 6$);
4. Field of view for the right end-effector (operational, loose cooperation, $m = 2$);
5. Center of mass (optimization, tight cooperation, $m = 1$);

Table 4

Task gains for the experiments on an aerial dual-arm manipulator

First case study		Second case study	
Task	Gain	Task	Gain
Joint limits	2	Joint limits	2
Virtual walls	2.5	Virtual walls	2.5
Glob. position	$35 \mathbf{I}_3$	Glob. position	$35 \mathbf{I}_3$
Glob. orientation	$7 \mathbf{I}_3$	Glob. orientation	$7 \mathbf{I}_3$
Field of view	0.6	Position (right ee)	$35 \mathbf{I}_3$
Center of mass	1	Center of mass	1
Manipulability	20	Manipulability	20

6. Manipulability of the system (optimization, tight cooperation, $m = 1$).

All the chosen task gains are reported in Table 4.

It is worth noticing that, since the aerial platform is under-actuated, among the triplet of Euler angles representing its orientation, only the yaw angle, ψ_V , is a controllable variable, while roll and pitch angles, φ_V and ϑ_V , are considered as intermediate control inputs for position control. Thus, by rewriting the vector \mathbf{x} in terms of controllable and uncontrollable variables as follows

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_u \end{bmatrix}, \quad \mathbf{x}_c = \begin{bmatrix} p_V \\ \psi_V \\ q_L \\ q_R \end{bmatrix}, \quad \mathbf{x}_u = \begin{bmatrix} \varphi_V \\ \vartheta_V \end{bmatrix},$$

the kinematic inversion has been performed as in (5), where the controlled and uncontrolled Jacobian matrices have been obtained as detailed in Section 5, by suitably selecting the columns relative to controlled and uncontrollable variables, respectively. More details on the kinematics of the dual-arm aerial manipulator can be found in Cataldi et al. (2019).

Figure 3 reports the experimental results. It is clear that the left end-effector successfully follows the desired path. The results of the field of view task of the right end-effector is reported in the plot as the misalignment between the outgoing vectors of the two end-effectors. It is clear that it converges to zero, meaning that the left end-effector is eventually kept inside the field of view of the camera mounted on the right one. During the experiment two joints of the right arm reach the imposed limits, and the corresponding safety tasks get active and stop their motion in order to respect the constraints. Regarding the two optimization tasks placed at a lower priority in the hierarchy, the misalignment between the center of mass of the dual-arm system and the vehicle slowly decreases, while the system manipulability measure increases as much as possible given the fulfillment of the

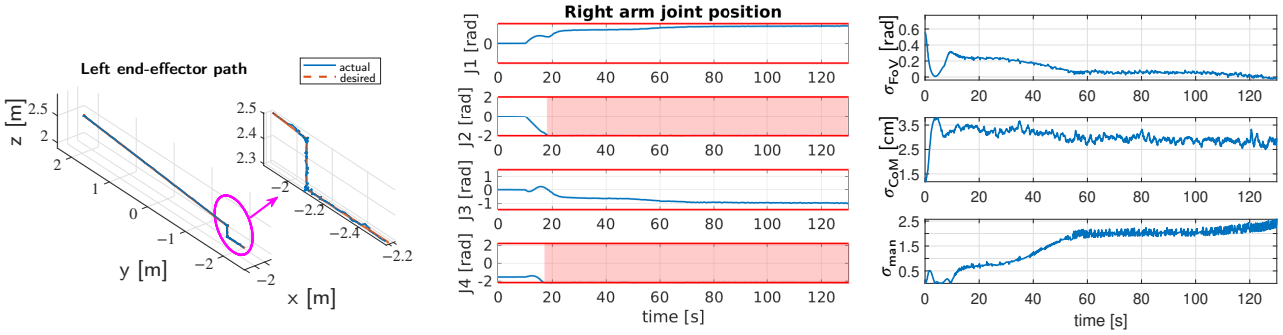


Figure 3: Fist case study. Left: left end-effector actual (solid blue) and desired (dashed-red) path; Center: right arm joint positions with the corresponding limits (red lines); Right: field of view of the right end-effector, center of mass of the system, manipulability measure of the left arm. The red background highlights the activation of a set-based task.

higher-priority tasks. The virtual wall tasks and the joint limits of the left arm are not shown for sake of brevity, as they never activate during this experiment.

6.5. Second case study

In this second case study, a desired position and orientation trajectory is planned for the left end-effector, while a desired trajectory only for the end-effector position is specified for the right arm.

The same optimization and safety tasks of the previous case study have been implemented, obtaining the following hierarchy:

1. Joint limits on the four joints of both arms (safety, non-cooperative, 8 tasks, $m = 1$ each);
2. Virtual walls between the arms (safety, non-cooperative, 2 tasks, $m = 1$ each) and between the arms and the aerial platform (safety, non-cooperative, 2 tasks, $m = 1$ each);
3. Global pose of the left end-effector (operational, tight cooperation, $m = 6$);
4. Position of the right end-effector (operational, non-cooperative, $m = 2$);
5. Center of mass (optimization, tight cooperation, $m = 1$);
6. Manipulability of the system (optimization, tight cooperation, $m = 1$).

Figure 4 show the experimental results, and it can be seen that both the end-effectors successfully follow the desired path. Regarding the safety tasks, the third joint of both the right and the left arms get active during the motion fulfilling the imposed constraints. This time, given the specific trajectory chosen for the experiment, the tasks regarding the virtual walls between the two arms get active for both the end-effector, keeping the distance between them at the imposed minimum (red horizontal line in the plot). It is worth noticing that the two virtual wall tasks refer to the same virtual wall that is placed between the arms. The center of mass task value correctly decreases until $t = 50$ s, then it starts increasing due to the activation of the virtual wall tasks. The

same happens for the manipulability task, that increases until the activation of the higher-priority task forces it to decrease and eventually converge to $4 \cdot 10^{-3}$.

7. Experiments on a wheeled mobile manipulator

The chosen robotic platform is the Kinova Movo, a mobile robot manufactured by Kinova² equipped with two 7 DOF Kinova Ultra lightweight robotic Jaco² arms. The mobile base is equipped with four Swedish wheels thus making the robot holonomic, and linked to it there is a prismatic joint that allows to change the height of the torso and, thus, the vertical position of the two-arm system.

Based on (17), the state of the system can be customized as:

$$s = \left[p_V^T \quad \theta \quad q_L^T \quad q_R^T \right]^T \in \mathbb{R}^{18}, \quad (53)$$

where $p_V = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the vector containing the x , y position of the mobile base in inertial frame and the z position of the torso joint, θ is the platform orientation in the horizontal plane, q_L and q_R are the joint positions of the left and right manipulator, respectively.

The frames of interest, defined in Section 6.1, are shown in Figure 5. For the considered system, the matrix T_V can be computed via measurements provided by the localization system, that makes use of odometry and a compass for position and orientation estimation. In the matrices $T_{L,B}^V$, $T_{R,B}^V$, the rotation matrices $R_{L,B}^V$, $R_{R,B}^V$ and the first two elements of the position vectors, namely $p_{L,B}^V$ and $p_{R,B}^V$, are constant, while the third components depend on the position of the prismatic joint at the torso. Finally, the matrices $T_{L,E}^{L,B}$, $T_{R,E}^{R,B}$ can be obtained by expressing the kinematic chain of the manipulators as Denavit-Hartenberg parameters and then composing the transformation matrices of all the frames from the base to the arm end-effector.

²www.kinovarobotics.com/en/products/mobile-manipulators

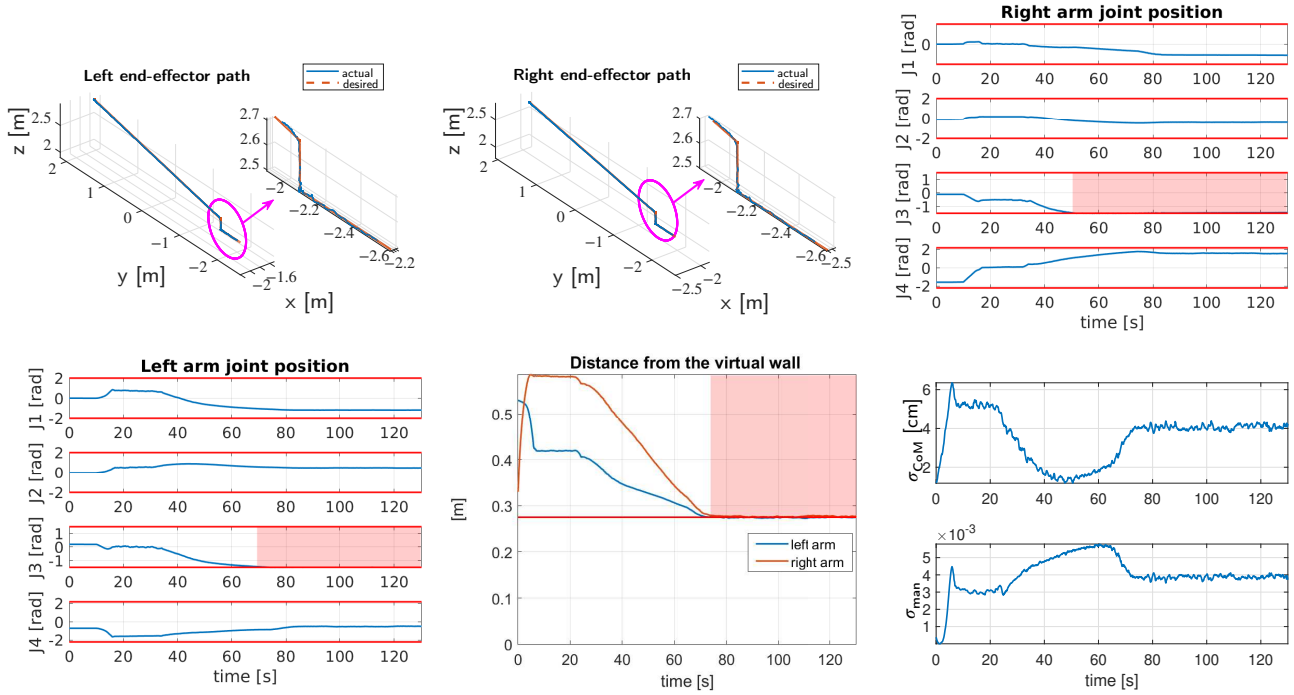


Figure 4: Second case study. Top left: left end-effector actual (solid blue) and desired (red-dashed) path; Top center: right end-effector actual (solid blue) and desired (red-dashed) path; Top right: right arm joint positions and corresponding limits (red lines); Bottom left: Left arm joint positions and corresponding limits (red lines); Bottom center: distance between the two end-effectors and the virtual wall placed between the arms, the red line expresses the minimum distance imposed by the corresponding set-based tasks; Bottom right: center of mass position and manipulability measure of the left arm. The red background highlights the activation of a set-based task.

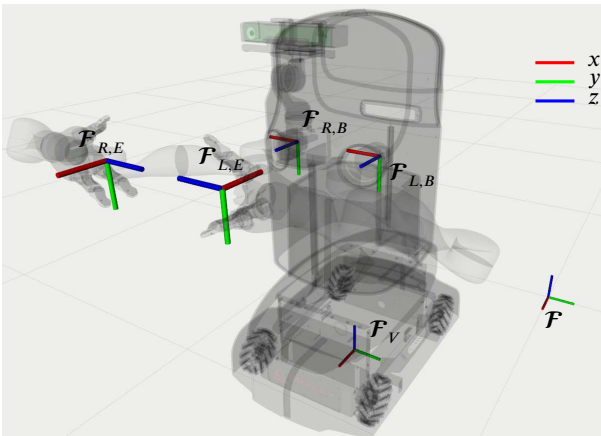


Figure 5: Frames of interest for the wheeled dual-arm mobile robot

A typical domestic use case scenario has been considered, in which the robot has to bring a tray in a desired location holding it with two hands.

The platform frame of the robot is initially centered in

$$p_{V,ini} = \begin{bmatrix} 0 & 0 & 0.2 \end{bmatrix}^T,$$

while holding a 40 cm-long tray with both hands. The left

end-effector initial position is

$$\sigma_{\text{global, pos}} = \begin{bmatrix} 0.66 & 0.15 & 0.85 \end{bmatrix}^T m$$

with an orientation

$$\sigma_{\text{global, ori}} = \begin{bmatrix} 0 & -0.7071 & 0.7071 & 0 \end{bmatrix}^T.$$

The desired global position and orientation is given as a constant reference to the left end effector and it is chosen to be behind the robot, with an orientation of -90 deg around the y -axis with respect to the initial orientation (see Fig. 6).

The reference for the relative pose task is set equal to the initial one, allowing it to hold horizontally the tray.

In the following, the results obtained by applying two different task hierarchies are discussed, aiming at highlighting the contribution of the task categories in the overall motion of the system.

7.1. Safety + Operational tasks

In this subsection, the proposed control scheme is implemented by considering only safety and operational tasks (global and relative poses) in the hierarchy composed as follows:

1. Limits on three joints of both arms (safety, non-cooperative, 6 tasks, $m = 1$ each)

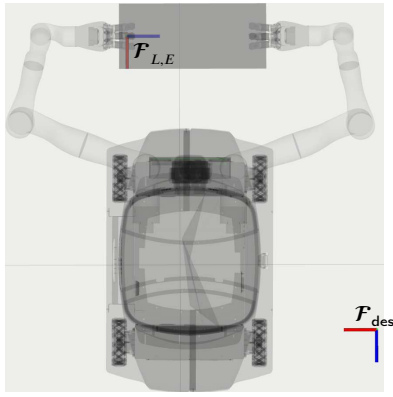


Figure 6: Initial configuration and desired left end-effector frame.

Table 5

Task gains for the experiments on a wheeled mobile manipulator

First case study		Second case study	
Task	Gain	Task	Gain
Joint limits	1.5	Joint limits	1.5
Virtual walls	1.2	Virtual walls	1.2
Rel. position	$3 \mathbf{I}_3$	Rel. position	$3 \mathbf{I}_3$
Rel. orientation	$4 \mathbf{I}_3$	Rel. orientation	$4 \mathbf{I}_3$
Glob. position	$0.3 \mathbf{I}_3$	Glob. position	$0.3 \mathbf{I}_3$
Glob. orientation	$0.7 \mathbf{I}_3$	Glob. orientation	$0.7 \mathbf{I}_3$
		Base optimization	0.8

2. Virtual walls around the base frames of both arms (safety, non-cooperative, 12 tasks, $m = 1$ each)
3. Relative pose (operational, tight cooperation, $m = 6$)
4. Global pose (operational, tight cooperation, $m = 6$)

The chosen task gains are reported in Table 5.

The experimental results in Fig. 7 show that both the global and relative position and quaternion errors reach a null steady-state value, meaning that the left end-effector reaches the desired location, while keeping the relative pose with respect to the right one at a constant value. Regarding the global pose error, the initial high error is due to the fact that the reference is not a smooth trajectory, but a constant point expressed in inertial frame. The limits imposed by the safety tasks are not violated during the entire experiment. However, it can be noticed that several safety tasks become active during the motion and that they get stuck at the desired safety thresholds (red horizontal lines in the plots): the second and fourth joints of the right arm, the wall on the z direction of the right arm base frame and the wall on the x direction of the left arm base frame.

7.2. Safety + operational + optimization tasks

It is worth noticing that the steady-state configuration reached in the previous case study, shown in Fig 8, has three active set-based tasks and it is quite *unnatural*, with the tray on the right side of the robot body and not aligned with it. To avoid such unnatural configuration, an optimization task is designed aimed at adjusting the position and the orientation of the mobile base, in order to keep the absolute position of the dual-arm system (Caccavale and Uchiyama, 2016) (e.g., the mid-point between the two end-effectors) aligned as much as possible with the x axis of the mobile base frame during the motion, resulting in a more *natural* steady-state configuration.

Based on the task presented in Section 5.3, the auxiliary frame \mathcal{F}_A has been centered in the mid-point between the two end-effectors, as shown in Fig. 9, while the task function has been customized by setting the following selection matrices:

$$\mathbf{L}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{L}_o = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \quad (54)$$

The desired mobile frame position has been set as $\mathbf{L}_p \mathbf{p}_{V, \text{des}}^A = \begin{bmatrix} -0.7 & 0 \end{bmatrix}^T$ m, while the desired orientation is set as $\theta_{\text{des}}^* = 0$, making the mobile base frame centered and aligned with the auxiliary one, as shown in Fig. 9.

In the Jacobian (48), the contribution of the velocities of the arm joints in the motion of \mathcal{F}_A has been intentionally ignored, because the arms are used to fulfill the operational tasks at a higher-priority in the hierarchy.

Therefore, the following hierarchy has been implemented:

1. Limits on three joints of both arms (safety, non-cooperative, 6 tasks, $m = 1$ each)
2. Virtual walls around the base frames of both arms (safety, non-cooperative, 12 tasks, $m = 1$ each)
3. Relative pose (operational, tight cooperation, $m = 6$)
4. Global pose (operational, tight cooperation, $m = 6$)
5. Base configuration optimization (optimization, loose cooperation, $m = 3$).

Figure 10 shows the time histories relative to the operational tasks errors, the safety task values and the optimization task error. It can be noticed that the optimization task error reaches a null-steady-state value, while all the safety tasks are kept inside the assigned boundaries and the operational tasks are effectively accomplished. Noticeably, in this case, due to the effect of the optimization task that moves the mobile base in order to center and align the tray with the mobile base frame, the steady-state configuration (shown in Fig. 8.b) has all the safety tasks deactivated, resulting in a less constrained system and in a more *natural* posture.

8. Conclusions

In this paper a novel task-priority kinematic control framework for coordinated motion for kinematically redundant multi-robot systems has been proposed. The framework has been

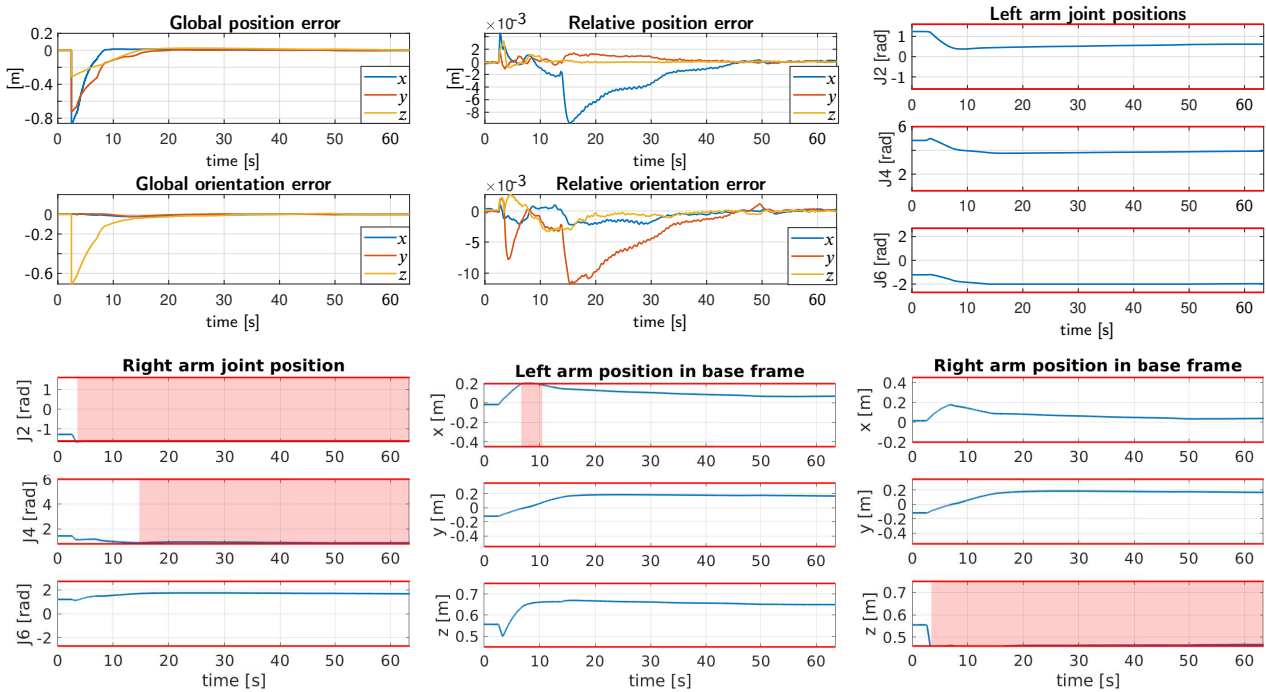


Figure 7: Third case study, safety + operational tasks. From top-left to bottom-right: global pose error; relative pose error; left and right arm joint positions together with the imposed safety thresholds; left and right arm pose expressed in their base frames together with the safety thresholds imposed by the virtual wall tasks. All the safety tasks are respected, the red background highlights the activation of a set-based task.

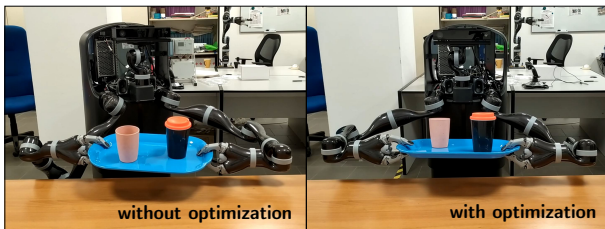


Figure 8: Left: steady-state configuration for the hierarchy containing safety and operational tasks. Right: steady-state configuration for the hierarchy containing also the optimization task. The tray is centered with respect to the robot body, resulting in a more *natural* posture.

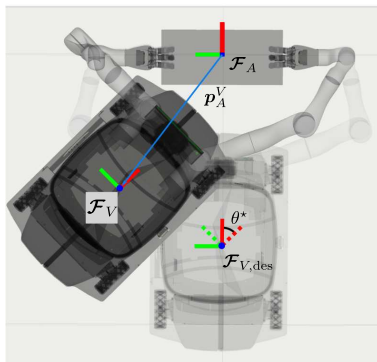


Figure 9: Mobile base configuration optimization task frames and variables

conceived to include both tasks expressed as equality constraints and set-based tasks, i.e., tasks expressed via inequality constraints. The effectiveness of the approach has been experimentally demonstrated on two multi-arm systems aimed at operating in different scenario. Future work will be devoted to the inclusion of tasks involving physical interaction with the external environment, e.g., to execute assembly operations and/or human-robot interactions, as well as to a fair and deep comparison between the proposed approach and the QP-based strategies.

Acknowledgement

This research has been supported partly by the EC Horizon 2020 Program under grant agreement No. 644271 (IP project AEROARMS) and partly by MIUR PON R&I 2014-2020 Program (project ICOSAF, ARS01_00861).

References

- Antonelli, G., Arrichiello, F., Chiaverini, S., 2008. The Null-Space-based Behavioral control for autonomous robotic systems. *Journal of Intelligent Service Robotics* 1, 27–39. URL: <http://dx.doi.org/10.1007/s11370-007-0002-3>, doi:10.1007/s11370-007-0002-3.
- Antonelli, G., Arrichiello, F., Chiaverini, S., 2010. The NSB control: a behavior-based approach for multi-robot systems. *Paladyn Journal of Behavioral Robotics* 1, 48–56.
- Arbo, M.H., Gravdahl, J.T., 2018. Stability of the tracking problem with task-priority inverse kinematics. *IFAC-PapersOnLine* 51, 121–125.

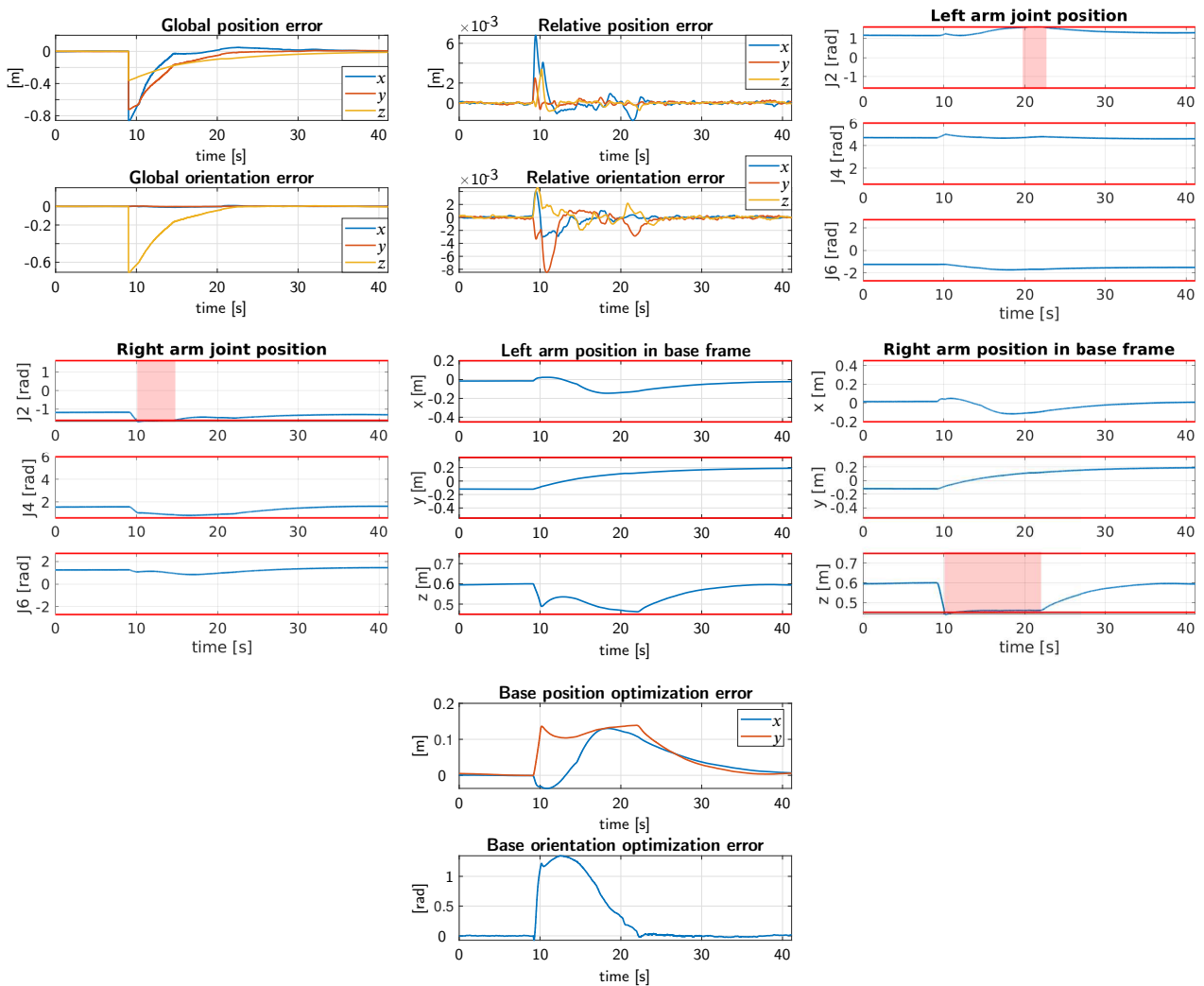


Figure 10: Third case study, safety + operational + optimization tasks. From top-left to bottom-right: global pose error; relative pose error; left and right arm joint positions together with the imposed safety thresholds; left and right arm pose expressed in their base frames together with the safety thresholds imposed by the virtual wall tasks; optimization task error. The red background highlights the activation of a set-based task. All the safety task are respected and the optimization function aligns the center of the tray to the robot body.

Azimian, H., Looi, T., Drake, J., 2014. 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, pp. 498–503.

Baizid, K., Giglio, G., Pierri, F., Trujillo, M.A., Antonelli, G., Caccavale, F., Viguria, A., Chiaverini, S., Ollero, A., 2017. Behavioral control of unmanned aerial vehicle manipulator systems. *Autonomous Robots* 41, 1203–1220.

Basso, E.A., Pettersen, K.Y., 2020. Task-priority control of redundant robotic systems using control Lyapunov and control barrier function based quadratic programs. *arXiv preprint arXiv:2001.07547*.

Caccavale, F., Uchiyama, M., 2016. Springer Handbook of Robotics. B. Siciliano, O. Khatib, (Eds.), Springer-Verlag, Heidelberg, D. chapter Cooperative Manipulation. pp. 989–1006.

Cataldi, E., Real, F., Suarez, A., Di Lillo, P., Pierri, F., Antonelli, G., Caccavale, F., Heredia, G., Ollero, A., 2019. Set-based inverse kinematics control of an anthropomorphic dual arm aerial manipulator, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, pp. 2960–2966.

Chiaverini, S., 1997. Singularity-robust task-priority redundancy resolu-

tion for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation* 13, 398–410.

Di Lillo, P., Arrichiello, F., Antonelli, G., Chiaverini, S., 2018. Safety-related tasks within the set-based task-priority inverse kinematics framework, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6130–6135. doi:10.1109/IROS.2018.8593884.

Di Lillo, P., Arrichiello, F., Di Vito, D., Antonelli, G., 2020. BCI-controlled assistive manipulator: developed architecture and experimental results. *IEEE Transactions on Cognitive and Developmental Systems*, 1–1.

Di Lillo, P., Chiaverini, S., Antonelli, G., 2019. Handling robot constraints within a set-based multi-task priority inverse kinematics framework, in: IEEE International Conference on Robotics and Automation (ICRA), pp. 7477–7483.

Escande, A., Mansard, N., Wieber, P.B., 2014. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *International Journal of Robotics Research* 33, 1006–1028.

Hollerbach, J., Ki Suh, 1987. Redundancy resolution of manipulators through torque optimization. *IEEE Journal on Robotics and Automation* 3, 308–316.

Jamisola, R.S., Roberts, R.G., 2015. A more compact expression of rela-

- tive jacobian based on individual manipulator jacobians. *Robotics and Autonomous Systems* 63, 158–164.
- Khatib, O., 1987. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation* 3, 43–53.
- Lin, Q., Burdick, J.W., 2000. Objective and frame-invariant kinematic metric functions for rigid bodies. *The International Journal of Robotics Research* 19, 612–625.
- Moe, S., Antonelli, G., Teel, A., Pettersen, K., Schrimpf, J., 2016. 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* 3, 16. doi:10.3389/frobt.2016.00016.
- Muscio, G., Pierri, F., Trujillo, M., Cataldi, E., Antonelli, G., Caccavale, F., Viguria, A., Chiaverini, S., Ollero, A., 2018. Coordinated control of aerial robotic manipulators: Theory and experiments. *IEEE Transactions on Control Systems Technology* 26, 1406–1413. doi:10.1109/TCST.2017.2716905.
- Nakamura, Y., Hanafusa, H., Yoshikawa, T., 1987. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research* 6, 3–15. URL: <https://doi.org/10.1177/027836498700600201>, doi:10.1177/027836498700600201, arXiv:<https://doi.org/10.1177/027836498700600201>.
- Nakanishi, J., Cory, R., Mistry, M., Peters, J., Schaal, S., 2008. Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research* 27, 737–757.
- Notomista, G., Mayya, S., Selvaggio, M., Santos, M., Secchi, C., 2020. A set-theoretic approach to multi-task execution and prioritization. arXiv preprint arXiv:2003.02968.
- Roberson, R.E., Schwertassek, R., 1988. *P Dynamics of Multibody Systems*. Springer-Verlag, Berlin, D.
- Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G., 2009. *Robotics: modelling, planning and control*. Springer Verlag.
- Siciliano, B., Slotine, J.J., 1991. A general framework for managing multiple tasks in highly redundant robotic systems, in: *Proceedings 5th International Conference on Advanced Robotics*, Pisa, I. pp. 1211–1216.
- Simetti, E., Casalino, G., Torelli, S., Sperindé, A., Turetta, A., 2013. Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project. *Journal of Field Robotics* 31(3), 364–385.
- Suarez, A., Heredia, G., Ollero, A., 2018a. Design of an anthropomorphic, compliant and lightweight dual arm for aerial manipulation. *IEEE Access* 6, 29173–29189.
- Suarez, A., Heredia, G., Ollero, A., 2018b. Physical-virtual impedance control in ultra-lightweight and compliant dual arm aerial manipulators. *IEEE Robotics and Automation Letters* 3, 2553–2560.
- Suarez, A., Jimenez-Cano, A., Vega, V., Heredia, G., Rodriguez-Castaño, A., Ollero, A., 2017. Lightweight and human-size dual arm aerial manipulator, in: *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, IEEE. pp. 1778–1784.
- Yoshikawa, T., 1985. Manipulability of robotic mechanisms. *The international journal of Robotics Research* 4, 3–9.