HAL
open science

# Decentralized Motion Control for Cooperative Manipulation with a Team of Networked Mobile Manipulators

Antonio Petitti, Antonio Franchi, Donato Di Paola, Alessandro Rizzo

**To cite this version:**

Antonio Petitti, Antonio Franchi, Donato Di Paola, Alessandro Rizzo. Decentralized Motion Control for Cooperative Manipulation with a Team of Networked Mobile Manipulators. 2016 IEEE International Conference on Robotics & Automation (ICRA), May 2016, Stockholm, Sweden. pp. 441-446. hal-01285043

**HAL Id: hal-01285043**

**https://hal.science/hal-01285043**

Submitted on 8 Mar 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Preprint version, final version at http://ieeexplore.ieee.org/ 2016 IEEE Int. Conf. on Robotics & Automation, Stockholm, Sweden

# Decentralized Motion Control for Cooperative Manipulation with a Team of Networked Mobile Manipulators

Antonio Petitti[1], Antonio Franchi[2,3], Donato Di Paola[1], and Alessandro Rizzo[4]

***Abstract*— In this paper we consider the cooperative control of the manipulation of a load on a plane by a team of mobile robots. We propose two different novel solutions. The first is a controller which ensures exact tracking of the load twist. This controller is partially decentralized since, locally, it does not rely on the state of all the robots but needs only to know the system parameters and load twist. Then we propose a fully decentralized controller that differs from the first one for the use of *i)* a decentralized estimation of the parameters and twist of the load based only on local measurements of the velocity of the contact points and *ii)* a discontinuous robustification term in the control law. The second controller ensures a practical stabilization of the twist in presence of estimation errors. The theoretical results are finally corroborated with a simulation campaign evaluating different manipulation settings.**

## I. Introduction

In the last few years, cooperative manipulation has become one of the most important research topics on multi-robot systems [1]. The realization of networked teams of robots able to manipulate and transport large or heavy loads is nearly being achieved, thanks to progresses in research and to the increased power of mobile manipulators [2]. The spectrum of applications includes search and rescue missions [3], complex tasks in unstructured environments [4], and human-robot interaction [5]. However, a number of research challenges are still open. Starting from the estimation of the physical characteristic of the manipulated body to the fully decentralized and robust control of the motion of the load, several issues must be addressed. On one hand, the on-line estimation of the inertial parameters of the load allows to manipulate unknown objects and to cope with time-varying inertial parameters [6]–[8]. Thus, decentralized inertial parameter estimation algorithms are highly desirable when dealing with decentralized motion control algorithms. On the other hand, the state of the art in cooperative manipulation regards the motion control of a load under the limiting assumption that its inertial parameters are known by all the robots in the team.

A common approach to the control problem is grounded on the distributed impedance control [9], in which path planning and control tasks are decoupled, i.e., each manipulator follows individually planned trajectories, taking into account the load geometry and the grasp points. In [10], the authors propose an LQR-optimal control scheme for cooperative manipulation subject to rigidity constraints, which regulates the configuration of the multi-robot team and maintains its formation. The approach only considers the maintenance of the rigidity constraint and is not suitable for establishing synchronized movements. Discrete and periodical individual trajectories to achieve coordinated manipulation can be generated by Dynamic Movement Primitives (DMPs) [11]. Decentralized control approaches to cooperative manipulation have also been presented. In [12], [13], a swarm of non-articulated mobile robots cooperate to control the velocity of a large rigid body, by means of local artificial potential fields. In [14], a local information-based control law is designed to solve the planar manipulation problem of rotating a rigid object to a desired orientation. The distributed control of the manipulators is realized at a kinematic level in continuous time, by means of inverse kinematics relations.

In this paper, we consider the problem of controlling a load whose motion is constrained on a plane and which can be manipulated by a team of networked mobile robots. In particular, we present two novel control strategies to solve the cooperative manipulation problem. The first control strategy is *partially decentralized*, in the sense that each robot computes the control law locally and without the knowledge of the state of all the other robots, yet the local control law is based on the *a priori* knowledge of the inertial/kinematic parameters of the manipulated load and its current twist. The second control strategy is instead *fully decentralized*. This strategy is similar to the first one apart from two crucial aspects. First, the load parameters and state are distributedly estimated leveraging recent results by some of the authors of this paper [6]–[8]. Second, the control law provides a robust control of the motion of the load also in presence of estimation errors. The price to pay for the full decentralization of the control strategy is a small bounded error in the tracking of the desired linear/angular velocity profile. This is unavoidable, due to the inherent delays in the multi-hop communication network established among the multi-robot team. Besides theoretical proofs, the feasibility of the fully decentralized approach is also assessed via numerical simulations in different operational scenarios and in presence of sensor noise.

## II. Problem Statement

We consider a team of $n$ mobile robots, with index set $\mathcal{I} = \{1, \ldots, n\}$ and $n \geq 2$, where each robot $i \in \mathcal{I}$ is equipped with a manipulator. A load, whose motion is constrained on a plane, has to be manipulated by the multi-robot team. We

[1]Institute of Intelligent Systems for Automation, National Research Council (ISSIA-CNR), 70126 Bari, Italy {petitti,dipaola}@ba.issia.cnr.it

[2]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France afranchi@laas.fr

[3]Univ de Toulouse, LAAS, F-31400 Toulouse, France

[4]Dipartimento di Automatica e Informatica (DAUIN), Politecnico di Torino, 10129 Torino, Italy alessandro.rizzo@polito.it

Work partially supported by the Short Term Mobility program of CNR Italy and partially funded by the Italian PRIN-MARIS project.

model the load as a rigid body $B$ whose center of mass (CoM) is denoted with $C$. Each robot $i \in \mathcal{I}$ exerts a torque and a force in a contact point, indicated as $C_i$, on the body $B$. We denote with $\mathbf{p}_C \in \mathbb{R}^2$ the position of $C$ in a reference inertial frame $\mathcal{W} = \{O_{\mathcal{W}} - \vec{\mathbf{x}}, \vec{\mathbf{y}}\}$ and with $\mathbf{p}_{C_i}$ the position of $C_i$ in $\mathcal{W}$. Furthermore, we indicate with $\mathbf{f}_i \in \mathbb{R}^2$ and $\tau_i \in \mathbb{R}$ the force and torque, respectively, applied by the $i$-th robot in $C_i$ and expressed in $\mathcal{W}$. Thus, we define the dynamical model of the load $B$ as

$$\begin{bmatrix} \dot{\mathbf{v}}_C \\ \dot{\omega} \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} m^{-1}\mathbf{I}_{2\times 2} & \begin{matrix} 0 \\ 0 \end{matrix} \\ J^{-1}\left[(\mathbf{p}_{C_i} - \mathbf{p}_C)^{\perp}\right]^T & J^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \tau_i \end{bmatrix}, \tag{1}$$

where $\mathbf{v}_C \in \mathbb{R}^2$ is the velocity of $C$, $m \in \mathbb{R}_{>0}$ is the mass of $B$, $J \in \mathbb{R}_{>0}$ is its moment of inertia, and $\omega \in \mathbb{R}$ is its rotational rate (i.e., the magnitude of the angular velocity vector, which is always directed as $\vec{\mathbf{x}} \times \vec{\mathbf{y}}$). We denote the twist of $B$ with $\mathbf{x} = \left[\mathbf{v}_C^T \ \omega\right]^T$. A schematic representation of the described setting is depicted in Fig. 1.

Finally, we assume that each robot $i \in \mathcal{I}$ is equipped with a wireless communication device with limited communication radius. We indicate with $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ the static undirected graph representing the communication network established among the robots, where $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ is the set of the available bidirectional communication links. The generic robot $i \in \mathcal{I}$ exchanges information only with its communication neighbors $\mathcal{N}_i = \{l \in \mathcal{I} \mid (i,l) \in \mathcal{E}\}$. Moreover, we assume that the communication graph $\mathcal{G}$ is and remains connected over time. Given the aforementioned scenario, in this paper we solve the following two problems.

**Problem II.1** (Partially Decentralized Motion Control of a Load)**.** *Consider a team of $n$ robots and a load $B$, where each robot knows the kinematic/inertial parameters of the load and its current twist, but does not know the states of the other robots. Design a partially decentralized control law such that the team is able to control the twist along a known desired trajectory. Specifically, given any twist reference $\mathbf{x}^d(t)$, continuously differentiable w.r.t. time, the control law must guarantee that*

$$\lim_{t\to\infty} \|\mathbf{x}^d(t) - \mathbf{x}\| = 0.$$

In order to define the second problem, let us consider the following constraint:

**Constraint II.1** (Local measurements and communication)**.** *Each robot $i \in \mathcal{I}$, which interacts with the load $B$ at the contact point $C_i$, can only locally measure the velocity $\mathbf{v}_{C_i}$ of the contact point and can only control the applied wrench $\left[\mathbf{f}_i^T \tau_i\right]^T$. These and possibly other internal quantities that are locally available can be transmitted via communication only to the robots in the neighbor set $\mathcal{N}_i$.*

**Problem II.2** (Fully Decentralized Motion Control of a Load)**.** *Given a team of $n$ robots, communicating through a network $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, and a load $B$, design a fully decentralized control law (i.e., which respects Constraint II.1) such that the team is able to practically control the twist of the load $B$. Specifically, given any velocity reference $\mathbf{x}^d(t)$, continuously differentiable w.r.t. time, the control law must guarantee that*

$$\lim_{t\to\infty} \|\mathbf{x}^d(t) - \mathbf{x}\| < \epsilon,$$

*where $\epsilon$ is a known, small bound.*

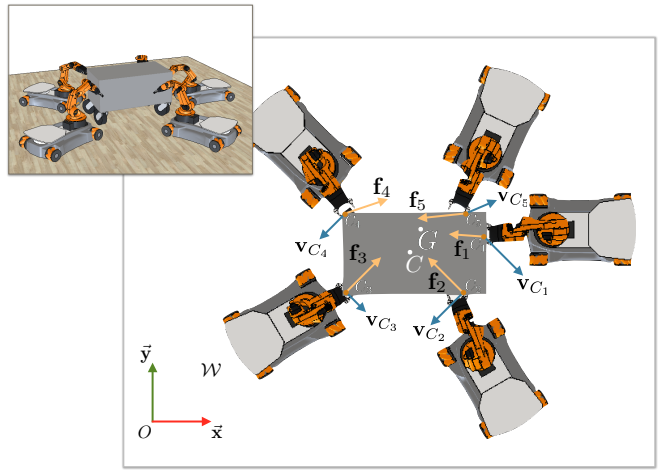


Fig. 1: A team of five KUKA youBot performing a cooperative manipulation task with the relevant symbols used in the paper.

**Remark II.1.** *We remark that this work focuses on the control of the twist of the manipulated object. The control of the position followed by the mobile base is out of the scope of this paper. Furthermore, we assume that each robot is able to adjust the position of its mobile base in order to avoid collisions (with other robots and the load), while guaranteeing the exertion of the desired force and torque.*

## III. Partially Decentralized Control

In this section, we propose a novel control algorithm for cooperative manipulation that solves Problem II.1. The proposed control law is based on the knowledge of the parameters of the dynamical model of the manipulated load $B$ expressed in (1) and of the number of robots $n$. In the following, and limited to this section, we assume that all these parameters, the velocity $\dot{\mathbf{v}}_C$, and the rotational rate $\omega$ are known (or measurable). We indicate with $\mathbf{p}_G \in \mathbb{R}^2$ the position in $\mathcal{W}$ of the geometric center $G$ of the contact points, i.e., $\mathbf{p}_G = n^{-1}\sum_{i=1}^{n} \mathbf{p}_{C_i}$. Moreover, we consider the following compact notation for vector differences: $\mathbf{z}_{ij} = \mathbf{p}_{C_i} - \mathbf{p}_{C_j}$, $\mathbf{z}_i = \mathbf{p}_{C_i} - \mathbf{p}_G$, and $\mathbf{z}_C = \mathbf{p}_G - \mathbf{p}_C$. Thus, substituting $\mathbf{p}_{C_i} - \mathbf{p}_C = \mathbf{z}_i + \mathbf{z}_C$ in (1), we obtain

$$\begin{bmatrix} \dot{\mathbf{v}}_C \\ \dot{\omega} \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} m^{-1}\mathbf{I}_{2\times 2} & \begin{matrix} 0 \\ 0 \end{matrix} \\ J^{-1}\mathbf{z}_i^{\perp T} & J^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \tau_i \end{bmatrix} + \begin{bmatrix} \begin{matrix} 0 \\ 0 \end{matrix} \\ J^{-1}\mathbf{z}_C^{\perp T} \end{bmatrix} \sum_{i=1}^{n} \mathbf{f}_i. \tag{2}$$

**Proposition III.1.** *Let be given a desired twist trajectory $\mathbf{x}^d(t) = \left[\mathbf{v}_C^{d\,T}(t) \ \omega^d(t)\right]^T$ for the load. Assume that the desired trajectory is continuously differentiable w.r.t. time. If each robot $i \in \mathcal{I}$ applies the following control law:*

$$\mathbf{f}_i = \frac{m}{n}\mathbf{u}_C - \frac{m\mathbf{z}_C^{\perp T}\mathbf{u}_C}{\sum_{i=1}^{n}\|\mathbf{z}_i\|^2}\mathbf{z}_i^{\perp} + \frac{Ju_\omega}{\sum_{i=1}^{n}\|\mathbf{z}_i\|^2}\mathbf{z}_i^{\perp}, \tag{3}$$

$$\tau_i = 0, \tag{4}$$

*where*

$$\mathbf{u}_C = \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C), \tag{5}$$
$$u_\omega = \dot{\omega}^d + k_\omega(\omega^d - \omega), \tag{6}$$

*then the velocity of the CoM and the rotational rate of the load converge globally and exponentially to the desired signals* $\mathbf{v}_C^d(t)$ *and* $\omega^d(t)$.

*Proof.* Plugging (3) and (4) in (2) we obtain

$$\dot{\mathbf{v}}_C = \sum_{i=1}^n \frac{\mathbf{u}_C}{n} + \left(\frac{m^{-1}Ju_\omega}{\sum_{i=1}^n \|\mathbf{z}_i\|^2} - \frac{\mathbf{z}_C^{\perp T}\mathbf{u}_C}{\sum_{i=1}^n \|\mathbf{z}_i\|^2}\right)\underbrace{\sum_{i=1}^n \mathbf{z}_i^\perp}_{=\mathbf{0}} = \mathbf{u}_C,$$

$$\dot{\omega} = \frac{m}{Jn}\mathbf{u}_C \underbrace{\sum_{i=1}^n \mathbf{z}_i^{\perp T}}_{=\mathbf{0}} - \frac{m\mathbf{z}_C^{\perp T}\mathbf{u}_C}{J\sum_{i=1}^n \|\mathbf{z}_i\|^2}\sum_{i=1}^n \mathbf{z}_i^{\perp T}\mathbf{z}_i^\perp +$$
$$+\frac{u_\omega}{\sum_{i=1}^n \|\mathbf{z}_i\|^2}\sum_{i=1}^n \mathbf{z}_i^\perp \mathbf{z}_i^{\perp T} + \frac{m}{Jn}\mathbf{z}_C^{\perp T}\sum_{i=1}^n \mathbf{u}_C -$$
$$-\frac{m\mathbf{z}_C^{\perp T}\mathbf{u}_C}{J\sum_{i=1}^n \|\mathbf{z}_i\|^2}\mathbf{z}_C^{\perp T}\underbrace{\sum_{i=1}^n \mathbf{z}_i^\perp}_{=\mathbf{0}} + \frac{u_\omega}{\sum_{i=1}^n \|\mathbf{z}_i\|^2}\mathbf{z}_C^{\perp T}\underbrace{\sum_{i=1}^n \mathbf{z}_i^\perp}_{=\mathbf{0}} =$$
$$= -\frac{m\mathbf{z}_C^{\perp T}\mathbf{u}_C}{J} + u_\omega + \frac{m\mathbf{z}_C^{\perp T}\mathbf{u}_C}{J} = u_\omega.$$

Hence, we obtain

$$\dot{\mathbf{v}}_C = \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C), \quad \dot{\omega} = \dot{\omega}^d + k_\omega(\omega^d - \omega), \tag{7}$$

which, in turn, implies $\mathbf{v}_C \to \mathbf{v}_C^d(t)$ and $\omega \to \omega^d(t)$ globally and exponentially. □

We observe from (3)–(4) that the proposed control law does not make use of the torque exerted by the robots of the network. This property can turn very useful when the load is large w.r.t. the characteristic dimensions of the robots and thus using local torques becomes impracticable.

**Remark III.1.** *We remark that* (3) *is not defined for the special case in which* $\mathbf{z}_i = \mathbf{0}$*, for all* $i$*. In fact, in this case it results* $\sum_{i=1}^n \|\mathbf{z}_i\|^2 = 0$*. However,* $\mathbf{z}_i = \mathbf{0}$*, for all* $i$*, (i.e., having all the contact points coinciding) is a meaningless, if not impossible, practical case in real applications.*

In order to implement the control law (3)–(6), the $i$-th robot needs to know the following seven quantities: i) $m$ and $J$, ii) $\mathbf{z}_C(t)$, $\mathbf{z}_i(t)$, $\sum_{i=1}^n \|\mathbf{z}_i\|^2$ (constant), and iii) $\mathbf{v}_C(t)$ and $\omega(t)$. We remark that the number and size of these quantities is independent from the number of robots.

## IV. Fully Decentralized Control

In this section, we propose a fully decentralized controller that solves Problem II.2, satisfying the limitations imposed by Constraint II.1. This achievement is made possible by

*i)* distributedly estimating $m$, $J$, $\mathbf{z}_C(t)$, $\mathbf{z}_i(t)$, $\sum_{i=1}^n \|\mathbf{z}_i\|^2$, $\mathbf{v}_C(t)$, and $\omega(t)$ using the algorithm in [7];

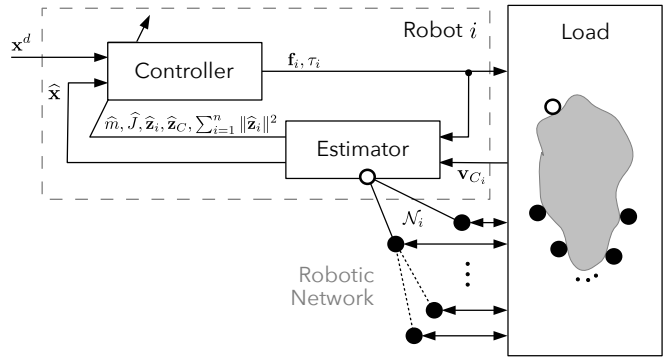


Fig. 2: A block diagram representing the closed loop system for the fully decentralized control. The estimator produces the inertial parameters needed to the decentralized controller. The dashed rectangle groups the modules running locally on the $i$-th robot.

*ii)* using the estimates of $m$, $J$, $\mathbf{z}_C(t)$, $\mathbf{z}_i(t)$, $\sum_{i=1}^n \|\mathbf{z}_i\|^2$ in place of the corresponding (unknown) actual quantities in (3); and

*iii)* modifying (5) and (6) in the controller algorithm to achieve robustness to estimation errors.

A schematic representation of the proposed approach is given in Fig. 2.

### A. Overview of the Decentralized Estimation Algorithm

To obtain a decentralized estimate of $m$, $J$, $\mathbf{z}_C(t)$, $\mathbf{z}_i(t)$, $\mathbf{v}_C(t)$, and $\omega(t)$ we use a recent decentralized estimation algorithm [6], [7], [8] proposed by part of the authors of this paper. Thus, once the estimated parameters are available to each robot, we can apply a robust and fully decentralized version of the partially decentralized control law, which will be described in Sec. IV-B. In this subsection, we present a brief summary of the decentralized parameter estimation algorithm. For the details about the working principles, including the elucidation of the measurement noise influence on the accuracy of the estimation, we refer the reader to [6], [7], and [8]. First, considering (2), the following holds:

**Fact IV.1.** *Assuming that* $\mathbf{p}_1$ *and* $\mathbf{p}_2$ *are the positions of any two points of* $B$ *expressed in* $\mathcal{W}$*, and considering two time instants* $t'$ *and* $t''$*, the rigid body constraint can be used to compute* $\mathbf{p}_2(t'')$ *from* $\mathbf{p}_1(t')$*,* $\mathbf{p}_2(t')$*, and* $\mathbf{p}_1(t'')$*:*

$$\mathbf{p}_2(t'') = \Gamma(\mathbf{p}_1(t'), \mathbf{p}_2(t'), \mathbf{p}_1(t'')) = \tag{8}$$
$$= \frac{\left(\mathbf{p}_2(t')^T\mathbf{p}_1(t')\right)\mathbf{p}_1(t'') + \left(\mathbf{p}_2(t')^T\mathbf{p}_1^\perp(t')\right)\mathbf{p}_1^\perp(t'')}{\|\mathbf{p}_1(t')\|}.$$

The decentralized estimation algorithm consists of four steps. The steps are sequentially executed, i.e., each step, of finite duration, needs only the information of the previous ones. We indicate with $t_0$ the start time of the algorithm and with $t_k$, $k = 1, \ldots, 4$, the end time of each step. Moreover, we indicate with the superimposed symbol $\widehat{\cdot}$ the estimates of a given inertial parameter.

**Step 1:** *Estimation of* $\mathbf{z}_{ij}$. Each robot $i$ applies an arbitrary force $\mathbf{f}_i(t)$ and uses $\widetilde{\mathbf{v}}_{C_i}$ (a noisy measurement of $\mathbf{v}_{C_i}$), combined with the measurements obtained from its neighbors. The vector $\mathbf{z}_{ij}$ is locally estimated by the

separate computation of its two factors: (i) $\vec{\mathbf{y}}_{ij}$ (time-varying axis) computed directly from $\widetilde{\mathbf{v}}_{C_i} - \widetilde{\mathbf{v}}_{C_j}$ (ii) $d_{ij}$ (constant coordinate) computed from $\widetilde{\mathbf{v}}_{C_i} - \widetilde{\mathbf{v}}_{C_j}$ via filtering and online linear least squares that converges to the optimal estimate of $\widehat{\mathbf{z}}_{ij}$ at time $t_1$.

**Step 2:** *Estimation of $J$ (constant).* For $t \geq t_1$, four substeps are executed in parallel to produce an estimate $\widehat{J}$ of the moment of inertia $J$.

*Step 2.1.* Each robot $i$ computes $\widehat{\mathbf{z}}_i(t)$, using $\widehat{\mathbf{z}}_{ij}(t)$ and the algorithm in [15].

*Step 2.2.* Each robot $i$ applies a force $\mathbf{f}_i(t) = k_z \widehat{\mathbf{z}}_i^{\perp}(t)$. While the body moves, each robot $i$ locally estimates the rotational rate of $B$ as $\widehat{\omega}_i(t) = -\left(\widehat{\mathbf{z}}_{ij}^T \widetilde{\mathbf{z}}_{ij}^{\perp}\right)\left(\widehat{\mathbf{z}}_{ij}^T \widehat{\mathbf{z}}_{ij}\right)^{-1}$, where $j \in \mathcal{N}_i$, and runs a dynamic consensus algorithm [16] to converge to a common estimate $\widehat{\omega}$.

*Step 2.3.* Each robot performs a local estimate hat $\widehat{J}_i$ of the moment of inertia of the load $J$, using the knowledge of the applied force and the estimated angular speed. Then, an average consensus algorithm [17] is used to reach an agreement on a common estimate $\widehat{J}$. This estimate becomes available at $t_2$.

**Step 3:** *Estimation of $\mathbf{z}_C(t)$.* Each robot $i$ applies an arbitrary nonzero force $\mathbf{f}_i(t)$ and is able to estimate $\omega(t)$. Then, each robot computes $\widehat{\mathbf{z}}_C$ using a nonlinear observer, detailed in [6], and considering the estimate $\widehat{J}$, previously obtained. After the convergence time of the observer, indicated with $t_3$, $\mathbf{z}_C(t)$ is estimated using the formula $\widehat{\mathbf{z}}_C(t) = \widehat{\mathbf{z}}_i(t) + \Gamma(\widehat{\mathbf{z}}_C(t_3), \widehat{\mathbf{z}}_{ij}(t_3), \widehat{\mathbf{z}}_{ij}(t))$, where $\Gamma$ is defined in (8) and $j \in \mathcal{N}_i$.

**Step 4:** *Estimation of $\mathbf{v}_C(t)$ and $m$ (constant).* Each robot $i$ applies an arbitrary nonzero force $\mathbf{f}_i(t)$ and computes $\widehat{\mathbf{v}}_C = \widetilde{\mathbf{v}}_{C_i} - \widehat{\omega}(t)\widehat{\mathbf{z}}_i(t)^{\perp}$, where all the quantities in the rhs are computed during the previous steps and are locally available after $t \geq t_3$. Based on $\widehat{\mathbf{v}}_C$ and on the average of the applied forces, each robot computes a local estimate hat $\widehat{m}_i$ of the body mass $m$. An average consensus algorithm is then used to converge to a common estimate of the mass $\widehat{m}$, which, for $t \geq t_4$, will be known by all the robots.

### B. Decentralized Robust Control

In this section, we modify the partially decentralized controller (3)–(6), in order to make it decentralized and robust to estimation uncertainties. We observe that the presented approach completely matches the requirements of Constraint II.1, since it does not require an all-to-all communication scheme, nor a central processing unit in charge of computing the control force to be applied by each mobile manipulator.

As explained in Section III, plugging (3) and (4) in (2), we obtain the dynamics in (7). However, when estimates are used in place of the corresponding real values, this is not true anymore, due to the existing uncertainties and noise. Thus, a more robust approach is needed. Let us first rewrite the control law (3) and (4) as

$$\widehat{\mathbf{f}}_i = \frac{\widehat{m}}{n}\widehat{\mathbf{u}}_C - \frac{\widehat{m}(\widehat{\mathbf{z}}_C^{\perp})^T \widehat{\mathbf{u}}_C}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2}\widehat{\mathbf{z}}_i^{\perp} + \frac{\widehat{J}\widehat{u}_\omega}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2}\widehat{\mathbf{z}}_i^{\perp}, \quad (9)$$

$$\widehat{\tau}_i = 0, \quad (10)$$

where $\widehat{\mathbf{u}}_C$ and $\widehat{u}_\omega$ are additional robustifying inputs. Then, define the vector $\widehat{\mathbf{x}} = \begin{bmatrix}\widehat{\mathbf{v}}_C^T & \widehat{\omega}\end{bmatrix}^T$. Furthermore, consider the error vectors $\mathbf{e} = \mathbf{x}^d - \mathbf{x}$ and $\widehat{\mathbf{e}} = \mathbf{x}^d - \widehat{\mathbf{x}}$. We note that it is possible to write $\mathbf{x} = \widehat{\mathbf{x}} + \boldsymbol{\zeta}$, where $\boldsymbol{\zeta}\left(\widehat{m}, \widehat{J}, \widehat{\mathbf{z}}_C(t), \widehat{\mathbf{z}}_i(t)\right)$ is an uncertainty term due to the presence of estimation errors. Plugging (9)–(10) in (2) yields the following dynamics:

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{v}}_C \\ \dot{\omega}\end{bmatrix} = \widehat{\mathbf{G}}(t)\begin{bmatrix}\widehat{\mathbf{u}}_C \\ \widehat{u}_\omega\end{bmatrix} = \widehat{\mathbf{G}}(t)\widehat{\mathbf{u}}, \quad (11)$$

where $\widehat{\mathbf{G}}(t) = \begin{bmatrix}\widehat{\mathbf{G}}_{11} & \widehat{\mathbf{g}}_{12} \\ \widehat{\mathbf{g}}_{21}^T & \widehat{g}_{22}\end{bmatrix}$, and

$$\widehat{\mathbf{G}}_{11} = \frac{\widehat{m}}{m}\left(1 - \frac{\widehat{\mathbf{z}}_C^{\perp T}\sum_{i=1}^n \widehat{\mathbf{z}}_i^{\perp}}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2}\right)\mathbf{I}_{2\times 2}, \quad \widehat{\mathbf{g}}_{12} = \frac{\widehat{J}}{m}\frac{\sum_{i=1}^n \widehat{\mathbf{z}}_i^{\perp}}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2},$$

$$\widehat{\mathbf{g}}_{21} = \frac{\widehat{m}}{J}\left(\mathbf{z}_C^{\perp} - \frac{\widehat{\mathbf{z}}_C^{\perp}}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2}\sum_{i=1}^n \mathbf{z}_i^{\perp T}\widehat{\mathbf{z}}_i^{\perp} + \frac{\mathbf{z}_C^{\perp}}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2}\widehat{\mathbf{z}}_C^{\perp T}\sum_{i=1}^n \widehat{\mathbf{z}}_i^{\perp}\right), \quad \widehat{g}_{22} = \frac{\widehat{J}}{J}\left(1 + \frac{\mathbf{z}_C^{\perp T}\sum_{i=1}^n \widehat{\mathbf{z}}_i^{\perp}}{\sum_{i=1}^n \|\widehat{\mathbf{z}}_i\|^2}\right).$$

We observe that $\widehat{\mathbf{G}}(t)$ depends both on the estimated and on the true values of the inertial parameters. In the ideal case where no estimation errors are present (that is to say, $\widehat{m} = m$, $\widehat{J} = J$, $\widehat{\mathbf{z}}_i = \mathbf{z}_i$, and $\widehat{\mathbf{z}}_C = \mathbf{z}_C$), then $\widehat{\mathbf{G}}(t) = \mathbf{I}_{3\times 3}$.

**Proposition IV.1.** *Let be given a desired twist trajectory $\mathbf{x}^d(t) = \left[\mathbf{v}_C^{d\,T}(t)\ \omega^d(t)\right]^T$ for the load. Assume that the desired trajectory is continuously differentiable w.r.t. time. Let each robot $i \in \mathcal{I}$ apply the control laws in* (9) *and* (10)*, where*

$$\widehat{\mathbf{u}} = \begin{bmatrix}\widehat{\mathbf{u}}_C \\ \widehat{u}_\omega\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{v}}_C^d \\ \dot{\omega}^d\end{bmatrix} + \rho(t, \mathbf{x})\frac{\widehat{\mathbf{e}}}{\|\widehat{\mathbf{e}}\|}, \quad (12)$$

*and $\rho(t, \mathbf{x}) : \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}_{>0}$ is a control gain whose design will be specified in the following. Then, the error $\widehat{\mathbf{e}}$ converges globally and in finite time to $\mathbf{0}$, while the error $\mathbf{e}$ converges to $\boldsymbol{\zeta}$ globally and in finite time.*

*Proof.* Consider $V(\mathbf{x}) = \frac{1}{2}\widehat{\mathbf{e}}^T\widehat{\mathbf{e}}$ as a Lyapunov candidate function, which is positive definite in $\widehat{\mathbf{e}} = \mathbf{0}$, and consider its time derivative $\dot{V}(\mathbf{x}) = \dot{\widehat{\mathbf{e}}}^T\widehat{\mathbf{e}}$. Computing the time derivative of $\widehat{\mathbf{e}}$, we obtain $\dot{\widehat{\mathbf{e}}} = \dot{\mathbf{x}}^d - \dot{\widehat{\mathbf{x}}} = \dot{\mathbf{x}}^d - \dot{\mathbf{x}} + \dot{\boldsymbol{\zeta}} = \dot{\mathbf{x}}^d - \widehat{\mathbf{G}}\widehat{\mathbf{u}} + \dot{\boldsymbol{\zeta}}$. The time derivative of the Lyapunov candidate function is

$$\begin{aligned}\dot{V}(\mathbf{x}) &= \left(\dot{\mathbf{x}}^d - \widehat{\mathbf{G}}\widehat{\mathbf{u}} + \dot{\boldsymbol{\zeta}}\right)^T\widehat{\mathbf{e}} \\ &= \left[\dot{\mathbf{x}}^d - \widehat{\mathbf{G}}\left(\dot{\mathbf{x}}^d + \rho\frac{\widehat{\mathbf{e}}}{\|\widehat{\mathbf{e}}\|}\right) + \dot{\boldsymbol{\zeta}}\right]^T\widehat{\mathbf{e}} \\ &= \left[\left(\mathbf{I}_{3\times 3} - \widehat{\mathbf{G}}\right)\dot{\mathbf{x}}^d\right]^T\widehat{\mathbf{e}} - \rho\frac{(\widehat{\mathbf{e}}^T\widehat{\mathbf{G}}^T)\widehat{\mathbf{e}}}{\|\widehat{\mathbf{e}}\|} + \dot{\boldsymbol{\zeta}}^T\widehat{\mathbf{e}} \\ &\leq \left(1 - \left\|\widehat{\mathbf{G}}\right\|\right)\|\dot{\mathbf{x}}^d\|\,\|\widehat{\mathbf{e}}\| - \rho\left\|\widehat{\mathbf{G}}\right\|\|\widehat{\mathbf{e}}\| + \left\|\dot{\boldsymbol{\zeta}}\right\|\|\widehat{\mathbf{e}}\|.\end{aligned}$$

Thus, if we choose $\rho(t, \mathbf{x})$ such that

$$\rho(t, \mathbf{x}) > \frac{(1 - \|\widehat{\mathbf{G}}\|)\|\dot{\mathbf{x}}^d\|}{\|\widehat{\mathbf{G}}\|} + \frac{\|\dot{\boldsymbol{\zeta}}\|}{\|\widehat{\mathbf{G}}\|}, \quad \forall t, \forall \mathbf{x}, \quad (13)$$

then $\dot{V}(\mathbf{x})$ results to be negative definite in $\widehat{\mathbf{e}} = \mathbf{0}$. Thus, the thesis holds due to the Lyapunov theorem [18] . Hence, from the definition of $\widehat{\mathbf{e}}$, it follows that $\mathbf{e}$ globally converges

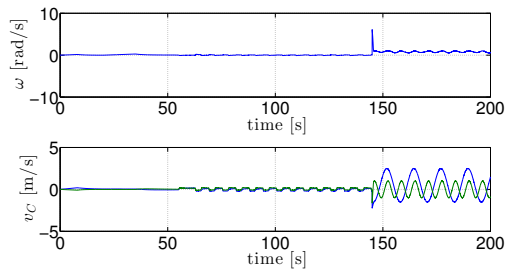


Fig. 3: The trajectory followed by the load twist $\mathbf{x}$ during the whole manipulation task. In the interval $t \in [0, 145[$ the first estimation phase takes place. The time interval $t \in [145, 200]$ is related to the *second estimation+control* phase that leads to the convergence of the state to the desired sinusoidal reference. In this simulation, we assume a team of $n = 10$ robots communicating over a line-topology network.

to $\boldsymbol{\zeta}$. Thus, the magnitude of the tracking error $\|\mathbf{e}\|$ is upper bounded by the magnitude of the estimation error $\boldsymbol{\zeta}$. □

Therefore, it follows from Proposition IV.1 that the control law in (9)–(10) and (12)–(13) solves Problem II.2 with $\epsilon = \|\boldsymbol{\zeta}\|$. However, the control law defined in (9)–(10) and (12)–(13) is discontinuous and may lead to control chattering, which can be notoriously avoided by smoothing the control discontinuity [18].

### C. Tuning of $\rho(t, \mathbf{x})$

As stated in Proposition IV.1, the convergence of the control depends on the choice of $\rho(t, \mathbf{x})$. If an approximate knowledge of the geometrical properties of $B$ is given, then it is possible to define an upper bound $\bar{g}$ of $\left\|\widehat{\mathbf{G}}\right\|$. Otherwise, after the estimation phase, the value of $\left\|\widehat{\mathbf{G}}\right\|$ can be approximated as the maximum value produced by the execution of Monte Carlo simulations based on the confidence intervals of the estimates. From specifications related to the accuracy of the estimation phase, it is possible to evaluate also an upper bound $\bar{\zeta}$ on the estimation error affecting $\widehat{\mathbf{v}}_C$ and $\widehat{\omega}$. Similarly, if $\dot{\mathbf{v}}_C^d$ and $\dot{\omega}^d$ are bounded, an upper bound $\bar{x}^d$ on $\left\|\dot{\mathbf{x}}^d\right\|$ exists. Therefore, in this case the value of $\rho(t, \mathbf{x})$ can be set as $\rho(t, \mathbf{x}) = \bar{x}^d/\bar{g} + \bar{x}^d + \bar{\zeta}/\bar{g} + \varepsilon$, where $\varepsilon$ is any non negative constant. A time-varying function that upper bounds $\left\|\dot{\mathbf{x}}^d(t)\right\|$ is another possible choice for $\rho(t, \mathbf{x})$, where, in this case, the actuation effort can be minimized, taking into account the norm of the velocity to be tracked. On the other hand, the time required to reach the convergence must be taken into account in the design of $\rho$, since it is bounded by (see, e.g., [18])

$$t_C = \frac{\|\mathbf{e}_0\|}{\rho(t, \mathbf{x})}, \tag{14}$$

where $\mathbf{e}_0$ is the error at the initial instant of the control phase.

As a final remark, it should be also taken into account that a large value of $\rho(t, \mathbf{x})$ can lead to instability when the required actuation effort is not physically admissible.

## V. Numerical Simulations

In this section, we assess the performance of the presented distributed control algorithm by means of numerical simulations. We assume a load of mass $m = 50$ kg and moment of inertia $J = 86.91$ kg m$^2$. We use two different teams of robots comprising $n = 4$ and $n = 10$ units, respectively, to execute a cooperative manipulation tasks. In both cases, the considered communication topology is assumed to be a line. The results of the simulation campaign are also reported in the accompanying video. The velocity measurement noise is Gaussian with zero mean and covariance matrix $\boldsymbol{\Sigma}_i = \sigma \mathbf{I}_{2\times 2}$, where $\sigma = 0.2$ m/s, $\forall i \in \mathcal{I}$. As in [7], the selected measurement noise yields uncertainties on the estimates of the mass and moment of inertia as $\widehat{m} = m \pm 0.0113$ kg and $\widehat{J} = J \pm 0.0004$ kg m$^2$, respectively. Moreover, the uncertainty of the observation of $\mathbf{z}_C$ can be approximated as $\sigma_{\mathbf{z}_C^x} = 0.075$ m along the $x$ coordinate and $\sigma_{\mathbf{z}_C^y} = 0.033$ m along the $y$ coordinate.

In order to assess the robustness of the proposed control algorithm, we run several independent trials using different velocity references. In Fig. 3, the trajectory followed by the components of the twist $\mathbf{x}$ during a complete cooperative manipulation task, performed by a team of $n = 10$ robots, is shown. The duration of the first estimation phase is 145 s, after which the control is activated, while the estimator still keeps estimating the time-varying quantities. We observe that in the latter phase, any time-varying reference trajectory $\mathbf{x}^d$ can be tracked, assuming that the constraints on the design of $\rho(t, \mathbf{x})$ are verified, as explained in Sec. IV-C.

In the following of this section, we focus on the control phase that occurs for $t > 145$ s, given that in the case under exam the estimation phase takes place in the interval $t = [0, 145]$ s (see the simulations in the accompanying video). In Fig. 4 the trend of the velocity of $C$, the angular rate of $B$, the force applied to $B$, and the tracking error, for a sinusoidal and a constant reference, respectively, are illustrated for a manipulation task performed by a team of $n = 10$ robots. We observe that the convergence time of the control algorithm is less than 1 s. Obviously, we achieve the same convergence time for the tracking of the desired angular rate. However, a small residual oscillating tracking error exists at steady state. This behavior is due to the estimation error because, as explained in Sec. IV-B, the control guarantees only the convergence of the error $\widehat{\mathbf{e}}$ to $\mathbf{0}$. If an estimation error, due to the measurement noise, is present in the observation of $\mathbf{v}_C$ and $\omega$, then it will cause a small oscillation of $\mathbf{e}$ around $\mathbf{0}$ at steady state. A similar oscillatory phenomenon is also present in the actuation signals. However, except for the beginning of the control phase, the magnitude of the forces exerted by the manipulators is always in the interval $\pm 3$ N. The same considerations can be made for a team of $n = 4$ robots dealing with the same cooperative manipulation task, as shown in Fig. 5. We observe that, for a team of 10 robots, the magnitude of the residual forces at steady state exerted by the manipulators is less than in the case where 4 units are involved, as it is reasonable to expect.

## VI. Conclusions

In this paper, we have presented a novel cooperative control approach for the manipulation of a load by means of a team of mobile robots. We have proposed two control

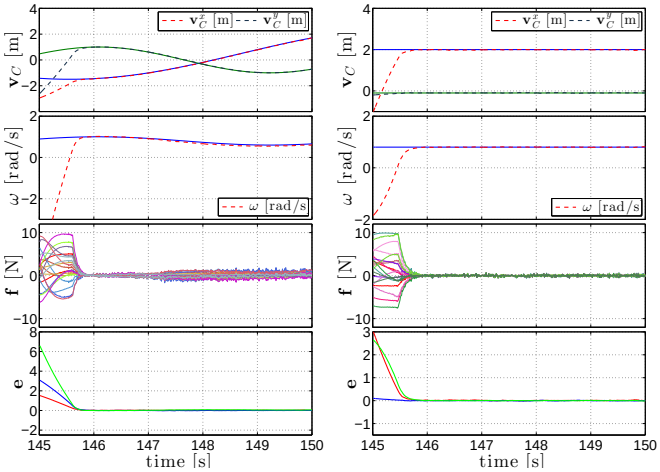


Fig. 4: Example of the tracking of a sinusoidal velocity reference (on the left) and a constant velocity reference (on the right) for the load $B$, performed by $n = 10$ robots. From the top, the trajectories followed by the velocity of the center of mass of $B$, $\mathbf{v}_C$, and by the angular rate, $\omega$, of $B$ are illustrated, as well as the exerted forces $\mathbf{f}^T = [\mathbf{f}_1^T, \ldots, \mathbf{f}_{10}^T]^T$ and the tracking error $\mathbf{e}$.

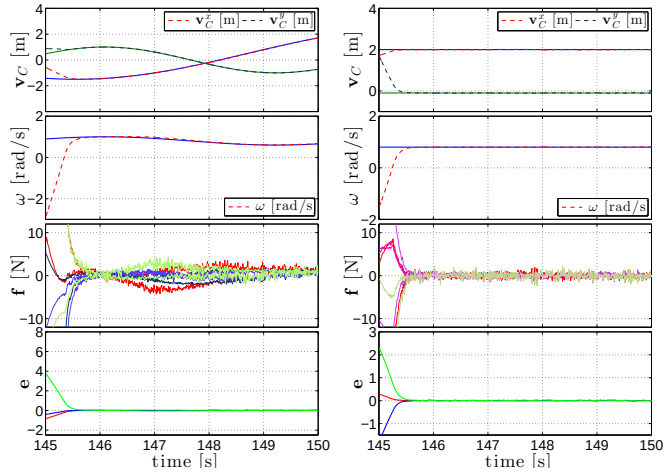


Fig. 5: Example of the tracking of a sinusoidal velocity reference (on the left) and a constant velocity reference (on the right) for the load $B$, performed by $n = 4$ robots. From the top, the trajectories followed by the velocity of the center of mass of $B$, $\mathbf{v}_C$, and by the angular rate, $\omega$, of $B$ are illustrated, as well as the exerted forces $\mathbf{f}^T = [\mathbf{f}_1^T, \ldots, \mathbf{f}_4^T]^T$ and the tracking error $\mathbf{e}$.

schemes: a partially decentralized controller, in which each robot computes the control law locally, but it has an a priori knowledge of the physical parameters of the load, and a fully decentralized approach in which each robot does not know any information about the load. In both cases, however, robots do not have access to any information about the state of the others robot in the team. We have proved that the partially decentralized control law converges toward a desired reference signal. Moreover, we have proved that the fully decentralized controller converges up to a small bounded error in the tracking of the desired linear/angular velocity profile. Future work will deal with the 3D case in which each aerial robot could be used as a force effector [19] thus allowing, e.g., distributed realization of the flying hand concept [20]. Nevertheless, the manipulation of an object in a 3D environment requires the estimation algorithm to be extended in a non trivial manner. Furthermore, we will aim to investigate the use of different control laws for the partially decentralized controller, inspired by different optimality criterions.

## References


[1] F. Caccavale and M. Uchiyama, "Cooperative manipulators," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 701–718.

[2] S. Erhart and S. Hirche, "Internal force analysis and load distribution for cooperative multi-robot manipulation," *IEEE Trans. on Robotics*, vol. 21, no. 3, pp. 1238–1243, 2015.

[3] G. Antonelli, K. Baizid, F. Caccavale, G. Giglio, and F. Pierri, "CAVIS: a control software architecture for cooperative multi-unmanned aerial vehIcle-manipulator systems," in *19th IFAC World Congress*, Cape Town, South Africa, Aug. 2014, pp. 24–29.

[4] N. Manerikar, G. Casalino, E. Simetti, S. Torelli, and A. Sperinde, "On autonomous cooperative underwater floating manipulation systems," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 523–528.

[5] I. Palunko, P. Donner, M. Buss, and S. Hirche, "Cooperative suspended object manipulation using reinforcement learning and energy-based control," in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Chicago, IL, Sep. 2014, pp. 885–891.

[6] A. Franchi, A. Petitti, and A. Rizzo, "Distributed estimation of the inertial parameters of an unknown load via multi-robot manipulation," in *53rd IEEE Conf. on Decision and Control*, Los Angeles, CA, Dec. 2014, pp. 6111–6116.

[7] A. Franchi, A. Petitti, and A. Rizzo, "Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 5517–5522.

[8] A. Franchi, A. Petitti, and A. Rizzo, "Distributed estimation for cooperative mobile manipulation," *arXiv*, vol. abs/1602.01891, Feb. 2016. [Online]. Available: http://arxiv.org/abs/1602.01891

[9] S. Erhart, D. Sieber, and S. Hirche, "An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov 2013, pp. 315–322.

[10] D. Sieber, F. Deroo, and S. Hirche, "Formation-based approach for multi-robot cooperative manipulation based on optimal control design," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 5227–5233.

[11] J. Umlauft, D. Sieber, and S. Hirche, "Dynamic movement primitives for cooperative manipulation and synchronized motions," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014, pp. 766–771.

[12] J. M. Esposito, "Decentralized cooperative manipulation with a swarm of mobile robots," in *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, St. Louis, MO, Oct 2009, pp. 5333–5338.

[13] ——, "Decentralized cooperative manipulation with a swarm of mobile robots: The approach problem," in *2010 American Control Conference*, Baltimore, MD, June 2010, pp. 4762–4767.

[14] J. Markdahl, Y. Karayiannidis, X. Hu, and D. Kragic, "Distributed cooperative object attitude manipulation," in *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 2012, pp. 2960–2965.

[15] R. Aragues, L. Carlone, C. Sagues, and G. Calafiore, "Distributed centroid estimation from noisy relative measurements," *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

[16] M. Zhu and S. Martinez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[17] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[18] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991.

[19] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bülthoff, and A. Franchi, "Turning a near-hovering controlled quadrotor into a 3D force effector," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014, pp. 6278–6284.

[20] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo, "The Flying Hand: a formation of uavs for cooperative aerial tele-manipulation," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014, pp. 4335–4341.