

Distributed Estimation of State and Parameters in Multi-Agent Cooperative Manipulation

Original

Distributed Estimation of State and Parameters in Multi-Agent Cooperative Manipulation / Franchi, A., Petitti, A., Rizzo, A.. - In: IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS. - ISSN 2325-5870. - ELETTRONICO. - 6:2(2019), pp. 690-701. [10.1109/TCNS.2018.2873153]

Availability:

This version is available at: 11583/2724981 since: 2019-09-03T17:06:57Z

Publisher:

IEEE

Published

DOI:10.1109/TCNS.2018.2873153

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

Distributed Estimation of State and Parameters in Multi-Agent Cooperative Load Manipulation

Antonio Franchi, *Senior Member, IEEE*, Antonio Petitti, Alessandro Rizzo, *Senior Member, IEEE*

Abstract—We present two distributed methods for the estimation of the kinematic parameters, the dynamic parameters, and the kinematic state of an unknown planar body manipulated by a decentralized multi-agent system. The proposed approaches rely on the rigid body kinematics and dynamics, on nonlinear observation theory, and on consensus algorithms. The only three requirements are that each agent can exert a 2D wrench on the load, it can measure the velocity of its contact point, and that the communication graph is connected. Both theoretical nonlinear observability analysis and convergence proofs are provided. The first method assumes constant parameters while the second one can deal with time-varying parameters and can be applied in parallel to any task-oriented control law. For the cases in which a control law is not provided, we propose a distributed and safe control strategy satisfying the observability condition. The effectiveness and robustness of the estimation strategy is showcased by means of realistic MonteCarlo simulations.

I. INTRODUCTION

In this paper, we propose what we believe is the first fully-distributed method for the estimation of all the quantities and parameters needed by a planar robotic multi-agent system to collectively manipulate an unknown load. In particular, the proposed algorithm provides the estimation of the kinematic parameters (equivalent to the grasp matrix), the dynamic parameters (relative position of the center of mass, mass, and rotational inertia) and the kinematic state of the load (velocity of the center of mass and rotational rate).

Most of the works on cooperative manipulation in the literature assume the *a priori* knowledge of the inertial parameters of the load, even though this assumption does not always hold in real-world scenarios [1]–[5]. Collective manipulation tasks would benefit from the implementation of on-line estimation strategies of inertial parameters of unknown loads for at least two reasons: first, existing control strategies, such as force control and pose estimation, could be effectively applied with satisfactory performance and a reduced control effort. Second, time-varying loads could be effectively manipulated, toward the implementation of adaptive or event-driven control strategies in uncertain environments. Furthermore, similarly to

other applications in multi-agent systems, a distributed and decentralized implementation of such estimation strategies would provide robustness and scalability. Research on the estimation of inertial parameters is at its early stage, and main limitations of the existing approaches reside in their centralization and in the use of absolute position and acceleration measurements, which are hard and costly to achieve, especially if accurate and noise-free information is needed. Moreover, centralized strategies are notoriously poorly scalable and not robust, due to the existence of single points of failure [6]–[9].

In this paper we propose two algorithms that have instead the following characteristics: (i) there is no central processing unit; (ii) each agent is only able to exchange information with its neighbors in the communication network; (iii) the network, modeled as an undirected graph, can have any topology, provided that it is connected; (iv) each agent is able only to perform local sensing and computation; and (v) the amount of memory and number of computations per step needed by each local instance of the algorithm do not depend on the number of agents but only on the number of communication neighbors. The only assumptions needed are that each agent is able to apply a wrench to the load at its contact point and to measure the velocity of such contact point. Any other measurement (such as, e.g., position, distance, acceleration, and gyro measurements) is not available to the agents. Finally, nothing is known about the manipulated load. The approaches are totally distributed, and rely on the geometry of the rigid body kinematics, the rigid body dynamics, on nonlinear observation theory, and on consensus strategies.

Related works: In [10], a decentralized motion control approach based on force consensus, which does not rely on explicit communication among the agents, is proposed. However, the result is achieved at the expense of assuming the presence of a leader agent that steers the load and on an even number of agents arranged in a particular shape called by the authors ‘centrosymmetric’. On the contrary, here we assume that agents can actually communicate, yet we do not assume the presence of a leader and we allow for any unknown spatial arrangement of the contact points. A communication-less adaptive motion control strategy is proposed in [11], under the assumption of a centralized measurement of the center of mass velocity and the load angular velocity. In the methods proposed here, no centralized measurement is needed. Furthermore, differently from [10], [11] our two methods let each agent estimate all the parameters of the problem, thus paving the way for the utilization of any control task (e.g., motion control, force control, etc.) on top of the proposed estimation algorithm. The authors in [12] show how a ‘communication-less’ parameter estimation can be achieved by

A. Franchi is with CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France and Univ de Toulouse, LAAS, F-31400 Toulouse, France afranchi@laas.fr

A. Petitti is with the Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council (STIIMA-CNR), 70126 Bari, Italy, antonio.petitti@stiima.cnr.it

A. Rizzo is with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy alessandro.rizzo@polito.it

This research was partially supported by the ANR, Project ANR-17-CE33-0007 MuRoPhen, by the Siebel Energy Institute, and by Compagnia di San Paolo.

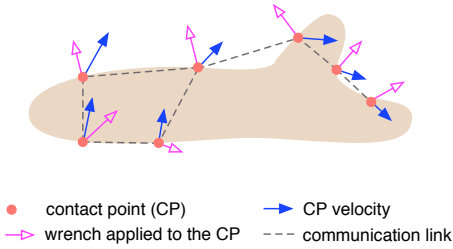


Fig. 1: Conceptual representation of a group of n agents manipulating an object on a plane. The orange dots represent the contact points of each agent, each blue arrow is the velocity of the contact point measured by each agent and each magenta hollow arrow is the wrench (force and torque) applied by each agent at the corresponding contact point. Dashed lines represent the communication links between agents, which all together constitute the communication graph.

adding some stronger assumptions. The method assumes initial parallel execution of synchronized control sequences by all the agents, which requires a form of centralization, prevents the simultaneous estimation of the parameters while performing the control task, and is not suited to estimate time-varying parameters. In our two methods, a-priori synchronization is not required, and the second method (see Sec. VI) can estimate time-varying parameters while performing any control task. The method in [12] assumes also that the robot are localized in position and orientation on a common frame and that they have enough strength to perform a hybrid position/force control and to lift the load from the ground, exploiting the gravity to estimate the mass. Such assumptions are absent in our setting. In fact, our main contribution is to demonstrate that if communication is available then it is possible to solve the estimation in a fully distributed way with minimal sensing.

II. MODEL AND PROBLEM STATEMENT

In this section, we formally define the problem of distributively estimating all the parameters and the time-varying quantities that are present in the mechanical model of a team of n planar robotic agents that cooperatively manipulate an unknown planar load, as illustrated in Fig. 1. We assume n to be constant and known to all robots. This assumption can be easily relaxed by implementing one of the several algorithms for the distributed estimation of a graph size [13].

We denote the inertial frame with $\mathcal{W} = O - xy$ and the load body frame with $\mathcal{B} = C - x_B y_B$, where C is the center of mass (CoM) of the unknown body, indicated with B . We indicate with $\mathbf{p}_C \in \mathbb{R}^2$ and $\mathbf{v}_C = \dot{\mathbf{p}}_C$ the position and velocity of C expressed in \mathcal{W} , respectively, and with $\omega \in \mathbb{R}$ the intensity of the load angular velocity, hereafter called simply *angular rate*. The dynamics of the manipulated load is described by the rigid body dynamical equation

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{g} = \mathbf{u}, \quad (1)$$

where $\boldsymbol{\nu} = (\mathbf{v}_C^\top \ \omega)^\top \in \mathbb{R}^3$ is the twist of B ; $\mathbf{M} = \text{diag}(m, m, J) \in \mathbb{R}^{3 \times 3}$ is the inertia matrix with $m > 0$ and $J > 0$ being the mass and the rotational inertia of the load, respectively; $\mathbf{g} \in \mathbb{R}^3$ is the wrench resulting from the environmental forces such as friction or gravitation. Without

loss of generality, in this work we set $\mathbf{g} = \mathbf{0}$. This is equivalent to assume a horizontal workspace and a wheeled load, for which the friction effects are negligible [14]. Finally, $\mathbf{u} \in \mathbb{R}^3$ denotes the external wrench applied by the agents to B , which will be characterized in the following. All the previous quantities are expressed with respect to the frame \mathcal{W} .

Each agent i contributes to the overall manipulation by exerting a wrench $\mathbf{u}_i = (\mathbf{f}_i^\top \ \tau_i)^\top \in \mathbb{R}^3$, expressed in \mathcal{W} , with $i = 1 \dots n$. The force $\mathbf{f}_i \in \mathbb{R}^2$ is applied by agent i to a contact point C_i of B , and $\tau_i \in \mathbb{R}$ is the intensity of the torque applied about the normal direction to the plane xy . We assume that contact points do not overlap, i.e., $C_i \neq C_j, \forall i, j = 1 \dots n$. The total external wrench applied to B is $\mathbf{u} = \sum_{i=1}^n \mathbf{G}_i \mathbf{u}_i = \mathbf{G} \bar{\mathbf{u}}$, where $\mathbf{G}_i \in \mathbb{R}^{3 \times 3}$ is the partial grasp matrix, $\mathbf{G} \in \mathbb{R}^{3 \times 3n}$ is the grasp matrix, and $\bar{\mathbf{u}} = (\mathbf{u}_1^\top, \dots, \mathbf{u}_n^\top)^\top$ is the stacked applied wrench vector that groups the generalized contact force components transmitted through the contact points [15]. The partial grasp matrix is defined as $\mathbf{G}_i = \mathbf{P}_i \bar{\mathbf{R}}_i$, where $\mathbf{P}_i = \begin{bmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} \\ [(\mathbf{p}_{C_i} - \mathbf{p}_C)^\perp]^\top & 1 \end{bmatrix}$ and $\bar{\mathbf{R}}_i = \mathbf{I}_{3 \times 3}$, in our setting, for all $i = 1 \dots n$. Here, $\mathbf{p}_{C_i} \in \mathbb{R}^2$ is the position of C_i in \mathcal{W} . The operator $(\cdot)^\perp$ rotates a vector by an angle of $\pi/2$, as is defined as $\mathbf{q}^\perp = Q\mathbf{q} = [-q^y \ q^x]^\top$ where $Q = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Hence, the dynamics (1) of the manipulated load is given by

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

Let $\mathbf{p}_G \in \mathbb{R}^2$ represent the position of the geometric center G of the contact points expressed in \mathcal{W} , i.e., $\mathbf{p}_G = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_{C_i}$. We compactly define $\mathbf{z}_i = \mathbf{p}_{C_i} - \mathbf{p}_G$ and $\mathbf{z}_C = \mathbf{p}_G - \mathbf{p}_C$. Hence, substituting $\mathbf{p}_{C_i} - \mathbf{p}_C = \mathbf{z}_i + \mathbf{z}_C$ in (2) we obtain

$$\begin{bmatrix} \dot{\mathbf{v}}_C \\ \dot{\omega} \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} m^{-1} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} \\ J^{-1} \mathbf{z}_i^{\perp \top} & J^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \tau_i \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ J^{-1} \mathbf{z}_C^{\perp \top} \end{bmatrix} \mathbf{f}_i. \quad (3)$$

Inspecting the dynamics (3), it is possible to see [16] that in order to effectively manipulate the load by controlling its velocity \mathbf{v}_C and angular rate ω , it is of fundamental importance that each agent i has an estimate of the constant parameters m and J , the time-varying vectors $\mathbf{z}_i(t)$ and $\mathbf{z}_C(t)$, and the quantities to be controlled, i.e., $\mathbf{v}_C(t)$ and $\omega(t)$.

Finally, the communication network among agents is modeled as a connected undirected graph \mathcal{G} , whose both node and link set are assumed to be time-invariant. The set \mathcal{N}_i denotes the set of one-hop neighbors of agent i in the communication graph, while \mathcal{A} denotes the graph adjacency matrix.

The problem under investigation is formally stated next.

Problem (Distributed Estimation in Multi-Agent Cooperative Manipulation). *Given n agents communicating through an ad-hoc network and manipulating an unknown load B ; assume that each agent i can only*

- 1) locally measure the velocity \mathbf{v}_{C_i} of the contact point C_i ,
- 2) locally know the applied wrench \mathbf{u}_i acting on C_i ,
- 3) communicate with its one-hop neighbors in the communication network.

Design a fully-distributed algorithm such that each agent i is able to estimate the following six quantities:

- 1) the (constant) mass m of the load,
- 2) the (constant) rotational inertia J of the load,
- 3) the (time-varying) relative position $\mathbf{z}_i(t)$ of C_i with respect to the geometric center G of the contact-points,
- 4) the (time-varying) relative position $\mathbf{z}_C(t)$ of the Center of Mass (CoM) C of B with respect to G ,
- 5) the (time-varying) velocity $\mathbf{v}_C(t)$ of C , and
- 6) the (time-varying) angular rate $\omega(t)$ of B .

In this work, we consider a strict definition of distributed algorithm, such that it is highly scalable with respect to the number of agents n . The main requisite of such an algorithm is that the complexity of the computations performed locally by each agent (in terms both of the number of elementary operations and of size of the input/output data) has to be constant with respect to the number of agents n [17].

In the next sections we shall constructively prove that the Problem is solvable as long as the communication network is connected, i.e., there exists a multi-hop communication path from any agent to any other agent in the network.

III. OVERVIEW OF THE TWO ALGORITHMS

An overview of the first proposed distributed estimation algorithm is given in the scheme of Fig. 2. Each rectangular box in the scheme corresponds to a computation performed locally by each agent i . Each circle, instead, corresponds to a consensus-like distributed algorithm that is used to compute the only five global quantities that we shall prove to be needed in the distributed estimation process. The number of these global quantities is independent from the number of agents, and they can be estimated using standard distributed algorithms. Therefore, the overall distributiveness of the approach is ensured. The convergence of the adopted distributed algorithms requires only that the overall communication graph is connected (no all-to-all communication is required). The same applies for our distributed estimation algorithm.

To better understand the overall functioning of the algorithm, it is convenient to logically decompose its structure in a *purely kinematic* phase, followed by a *dynamical* one. In the former, only the rigid body kinematics constraints and the velocity measurements are used. After this phase, each agent i is able to estimate the time-varying quantities $\mathbf{z}_i(t)$ and $\omega(t)$. In the latter, the applied wrench and the rigid body dynamics are also used. After this phase, each agent is able to estimate the remaining quantities, i.e., J , $\mathbf{z}_C(t)$, $\mathbf{v}_C(t)$, and m . The two phases are described in Sections IV and V, respectively.

All the estimation blocks are cascaded, hence convergence/inconsistency issues of feedback estimation structures do not affect our scheme. A clarification about the convergence of our strategy is in order. Some steps of the estimation procedure are achieved through averaging consensus algorithms, which are known to converge asymptotically. Although this aspect theoretically yields infinite convergence times, an ϵ -approximate global consensus [18], up to an arbitrary accuracy, can be achieved after a convenient finite time interval.

The first estimation algorithm assumes constant J and m and represents the best choice, in terms of noise filtering, in that case. The estimation of J requires a special wrench

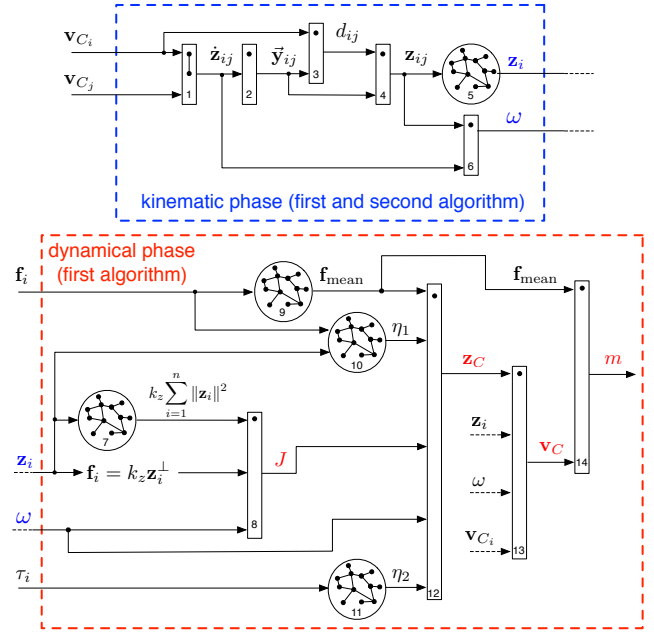


Fig. 2: Overview of the first proposed distributed estimation algorithm. **Top** (dashed blue box): *purely kinematic* phase, where only the velocity measurements and the rigid body kinematics are used. After this phase, the estimates of the time-varying quantities $\mathbf{z}_i(t)$ and $\omega(t)$ (in blue) become available to each agent i . **Bottom** (dashed red box): *dynamical* phase, where also the knowledge of the wrenches and the rigid body dynamics are used. After this phase, the quantities J , $\mathbf{z}_C(t)$, $\mathbf{v}_C(t)$, and m (in red) become available to each agent i .

input, which prevents the use of another control algorithm in that phase. To overcome such possible drawbacks, a second estimation algorithm is also proposed. The second algorithm is designed as a variant of the first one, and described in Sec. VI, see Fig. 3. It can deal with changing J and m and does not require any particular wrench in any of its phases.

IV. KINEMATIC PHASE

The objective of this phase is to distributively compute an estimate of the time-varying quantities $\mathbf{z}_i(t)$ and $\omega(t)$, based only on the locally measured velocities and the rigid body kinematic constraints. The basic idea of this phase is to split the estimations of $\mathbf{z}_i(t)$ and $\omega(t)$ in two parts. The former is common to both estimations and essentially consists of the estimation of $\mathbf{z}_{ij}(t) = \mathbf{p}_i(t) - \mathbf{p}_j(t)$. This part is described in Sec. IV-A. The latter comprises two separate estimators of $\mathbf{z}_i(t)$ and $\omega(t)$ and is described in Sec. IV-B

The reason for passing through the estimation of the quantities \mathbf{z}_{ij} is briefly explained in the following. In [19], a distributed algorithm is proposed that allows the estimation of the centroid of the positions of a network of agents by only measuring the relative positions between pairs of communicating agents. This algorithm can be used to distributively estimate $\mathbf{z}_i(t)$ if each pair of neighbors know the relative position of the contact points C_i and C_j , i.e., $\mathbf{z}_{ij}(t)$. Nevertheless, here each agent only measures the velocity of its contact point and not its position. Our first contribution is to show that, thanks to the rigid body constraint, it is possible to estimate $\mathbf{z}_{ij}(t)$ only resorting to the measures $\mathbf{v}_i(t)$ and $\mathbf{v}_j(t)$. Hereinafter,

to enhance readability, we shall drop the time dependence of variables where such a dependence is clear from the context.

A. Estimation of $\mathbf{z}_{ij}(t)$

The time-varying vector $\mathbf{z}_{ij}(t)$ that we want to estimate has to obey to the *nonlinear* rigid body constraint

$$\mathbf{z}_{ij}^\top \mathbf{z}_{ij} = \text{const.} \quad (4)$$

This implies that, even though the direction of $\mathbf{z}_{ij}(t)$ may vary in time, its norm $\|\mathbf{z}_{ij}\|$ is constant. Taking the time derivative of both sides of (4) yields $\dot{\mathbf{z}}_{ij}^\top \mathbf{z}_{ij} = 0$, which implies that the directions of \mathbf{z}_{ij} and $\dot{\mathbf{z}}_{ij}^\perp = Q\dot{\mathbf{z}}_{ij}$ coincide. We can then explicitly decompose \mathbf{z}_{ij} in two factors

$$\mathbf{z}_{ij} = d_{ij} \vec{\mathbf{y}}_{ij}, \quad (5)$$

where $\vec{\mathbf{y}}_{ij} = \dot{\mathbf{z}}_{ij}^\perp / \|\dot{\mathbf{z}}_{ij}^\perp\| \in \mathbb{S}^1$ is the unit vector denoting the time-varying oriented line (axis) along which \mathbf{z}_{ij} lies, and $d_{ij} \in \mathbb{R}$ is the coordinate of \mathbf{z}_{ij} on $\vec{\mathbf{y}}_{ij}$.

Let each agent i send to all $j \in \mathcal{N}_i$ the (measured) velocity of its contact point \mathbf{v}_{C_i} using the one-hop communication links. Then, each agent i can compute the velocity difference

$$\dot{\mathbf{z}}_{ij} = \mathbf{v}_{C_i} - \mathbf{v}_{C_j}, \quad (6)$$

and the corresponding orthogonal vector $\dot{\mathbf{z}}_{ij}^\perp$, for each $j \in \mathcal{N}_i$. As a consequence, $\vec{\mathbf{y}}_{ij}$ is locally available to each agent i , $\forall j \in \mathcal{N}_i$. This is the first milestone of our algorithm, which is formally stated in the following result.

Result 1. *The axis $\vec{\mathbf{y}}_{ij}$ along which \mathbf{z}_{ij} lies is directly computed from local measurements and one-hop communication as $\vec{\mathbf{y}}_{ij} = \dot{\mathbf{z}}_{ij}^\perp / \|\dot{\mathbf{z}}_{ij}^\perp\| = (\mathbf{v}_{C_i} - \mathbf{v}_{C_j})^\perp / \|\mathbf{v}_{C_i} - \mathbf{v}_{C_j}\|$, as long as $\|\dot{\mathbf{z}}_{ij}\| = \|\mathbf{v}_{C_i} - \mathbf{v}_{C_j}\| \neq 0$.*

To obtain the sought \mathbf{z}_{ij} , only the estimation of d_{ij} is left. Due to the rigid body constraint (4), $|d_{ij}| = \|\mathbf{z}_{ij}\| = \text{const}$ holds, i.e., d_{ij} is either equal to $\|\mathbf{z}_{ij}\|$ or to $-\|\mathbf{z}_{ij}\|$, depending on the fact that $\vec{\mathbf{y}}_{ij}$ and \mathbf{z}_{ij} have the same or the opposite direction. However, since in (5) both $\mathbf{z}_{ij}(t)$ and $\vec{\mathbf{y}}_{ij}(t)$ are continuous functions of time (for $\vec{\mathbf{y}}_{ij}$ this holds in any open interval in which $\|\dot{\mathbf{z}}_{ij}\| \neq 0$), we have that $\text{sign}(d_{ij}) = \text{const} \quad \forall t \in T$, as long as $\|\dot{\mathbf{z}}_{ij}\| \neq 0, \quad \forall t \in T$. Thus, in any time interval T in which $\|\dot{\mathbf{z}}_{ij}\| \neq 0$ and under the reasonable assumption that the input wrenches are continuous in t over T , we can differentiate both sides of (5), thus obtaining

$$\dot{\mathbf{z}}_{ij} = d_{ij} \dot{\vec{\mathbf{y}}}_{ij}, \quad (7)$$

which forms a linear estimation problem that agent i can locally solve to estimate the sought d_{ij} . In fact, among the quantities that appear in (7), agent i knows the quantity $\dot{\mathbf{z}}_{ij}$ and the time integral of $\frac{d}{dt} \vec{\mathbf{y}}_{ij}$, i.e., $\vec{\mathbf{y}}_{ij}$. Therefore, the estimate of d_{ij} can be carried out using a standard online linear estimation technique described, e.g., in [20], and summarized in the Appendix of [21]. This technique has also the property of averaging out the possible measurement noise. To this aim, the time interval T can be tuned on the basis of the noise level that has to be averaged out in the velocity measurements.

Note that after the first estimation of d_{ij} there is no need to further estimate $|d_{ij}|$, since this is a constant quantity. Thus,

the only signal to keep track of is $\text{sign}(d_{ij})$. This can be instantaneously achieved by implementing two linear observers of the dynamic system (7): one that assumes $\text{sign}(d_{ij}) = 1$ and the other assuming $\text{sign}(d_{ij}) = -1$. Then, it is sufficient to select, at each time-step, the observer that provides the best estimate in terms of, e.g., measurement residual.

To conclude the description of the algorithm, every time it happens to be $\|\dot{\mathbf{z}}_{ij}\| = 0$, the last estimate of \mathbf{z}_{ij} is kept frozen. In a real implementation the introduction of a suitable threshold to cope with the possible noise is recommended.

We summarize the results of this section in the following.

Result 2. *The vector \mathbf{z}_{ij} is estimated locally by agent i and j by the separate computation of two quantities*

- $\vec{\mathbf{y}}_{ij}$ (time-varying axis), computed directly from $\mathbf{v}_{C_i} - \mathbf{v}_{C_j}$ (see Result 1)
- d_{ij} (norm-constant coordinate along $\vec{\mathbf{y}}_{ij}$), computed from $\mathbf{v}_{C_i} - \mathbf{v}_{C_j}$ by solving (7) via filtering and applying online Linear Least Squares (LLS).

This part of the algorithm is referred with blocks 1, 2, 3, and 4 in the diagram of Fig. 2.

B. Estimation of $\mathbf{z}_i(t)$ and $\omega(t)$

The estimated quantities $\mathbf{z}_{ij}(t)$ provide a straightforward way to estimate \mathbf{z}_i . This estimation phase corresponds to block 5 in the diagram of Fig. 2.

Result 3. *Once the estimate of $\mathbf{z}_{ij}(t)$ is available to each agent i , $\forall j \in \mathcal{N}_i$, each agent i estimates \mathbf{z}_i by using the centroid estimation algorithm described in [19].*

In order to estimate the angular rate ω , we use the following relation from rigid body kinematics

$$\omega \mathbf{z}_{ij} = -\dot{\mathbf{z}}_{ij}^\perp, \quad (8)$$

where $\dot{\mathbf{z}}_{ij}^\perp$ is locally computed from (6), and \mathbf{z}_{ij} is locally estimated, as shown in Sec. IV-A. Multiplying both sides of (8) by \mathbf{z}_{ij}^\top , we obtain that, for each pair of communicating agents i and j , an estimate of ω is directly given by

$$\omega = -(\mathbf{z}_{ij}^\top \dot{\mathbf{z}}_{ij}^\perp) (\mathbf{z}_{ij}^\top \mathbf{z}_{ij})^{-1}. \quad (9)$$

Result 4. *ω is locally computed using (9), where $\dot{\mathbf{z}}_{ij}$ comes from direct measurement and one-hop communication and \mathbf{z}_{ij} from Result 2.*

This part of the algorithm corresponds to block 6 in the diagram of Fig. 2. The use of (9) provides agent i with as many estimates of ω as the number of its one-hop neighbors $|\mathcal{N}_i|$. In the ideal case of noiseless velocity measurements, all those estimates are identical. In the case of noisy velocity measurements, this redundancy can be exploited to average out the noise either at the local level (e.g., by averaging the different estimates corresponding to each neighbor) or at the global level (by, e.g., using some dynamic consensus strategy among all agents [22]). Clearly, the order of the dynamic consensus algorithm used is strictly related to the time variations of ω and, therefore, to the time variations of its estimates [22]. Moreover, such consensus will theoretically

converge asymptotically. However, dynamic consensus algorithms, able to track the average of their dynamic inputs up to a given bound, can be used [23].

Clearly, a measurement of the angular rate ω can also be obtained equipping each agent with a gyroscope placed at the contact point. Nevertheless, one of the goals, and contributions, of our work is to show that this additional sensor is not strictly needed to accomplish the estimation task.

Remark IV.1. This estimation approach relies on the agreement on a common reference frame. In fact, the measured velocities \mathbf{v}_{C_i} used to estimate \mathbf{z}_i , are referred to the same reference frame. Two possible approaches can be put forward in real-world applications: *i)* agents should communicate to agree on a common reference frame, or *ii)* agents use additional sensors (i.e., vision, compass, or infrared array) to perform conversions between quantities related to different frames.

V. DYNAMICAL PHASE

The objective of this phase (corresponding to the dashed red box in Fig. 2) is to estimate the remaining quantities, i.e., the (constant) rotational inertia J , the (time-varying) position $\mathbf{z}_C(t)$ of the CoM of B relative to the geometric center G of the contact points, the (time-varying) velocity of the CoM $\mathbf{v}_C(t)$, and the (constant) mass m . The order in which they are estimated follows a dependency hierarchy, since some phase needs information about the outcome of previous ones. Thus, the order of estimation cannot be altered without preventing the correct execution of the proposed strategy. This phase makes use of the velocity measurements \mathbf{v}_{C_i} , the applied wrench \mathbf{u}_i , as well as the rigid body kinematics and dynamics. The basic operations executed in this phase are summarized in the following:

- 1) (*estimation of J*) we exploit the knowledge of \mathbf{z}_i to apply a particular input wrench that cancels the effect of \mathbf{z}_C in (3), thus, obtaining a reduced dynamics in which J is the only unknown; then, we estimate J using LLS;
- 2) (*estimation of $\mathbf{z}_C(t)$*) we use all the previously estimated quantities, the rotational dynamics in (3), and the rigid body constraint to recast the estimation of \mathbf{z}_C to a nonlinear observation problem that can be locally solved by each agent with an observer designed by us;
- 3) (*estimation of $\mathbf{v}_C(t)$*) we use rigid body kinematics to compute $\mathbf{v}_C(t)$ from all the quantities estimated so far;
- 4) (*estimation of m*) we use a distributed estimation of the total force produced by the agents and $\mathbf{v}_C(t)$ to finally estimate the constant m using LLS.

A. Estimation of J

Assuming J as a constant quantity, our strategy is to impose a specific wrench for a short time interval in order to let its estimate converge close enough to the real value. After this finite time interval, any wrench can be applied again. This feature enables the concurrent execution of estimation and ordinary manipulation tasks.

Let us isolate the rotational dynamics from (3)

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n \mathbf{z}_i^{\perp \top} \mathbf{f}_i + \frac{1}{J} \mathbf{z}_C^{\perp \top} \sum_{i=1}^n \mathbf{f}_i + \frac{1}{J} \sum_{i=1}^n \tau_i, \quad (10)$$

where: *(i)* J is the constant to be estimated; *(ii)* $\omega(t)$ is locally known by each agent thanks to Result 4; *(iii)* $\mathbf{z}_i(t)$ is locally known by each agent thanks to Result 3; *(iv)* \mathbf{f}_i and τ_i are locally known by each agent, since they are applied by the agent itself; *(v)* \mathbf{z}_C is still unknown. If we were able to eliminate \mathbf{z}_C from (10), then J would become the only unknown in (10). It is easy to verify that the influence of \mathbf{z}_C in (10) is eliminated if each agent i applies a force \mathbf{f}_i such that $\sum_{i=1}^n \mathbf{f}_i = 0$. A possible choice is to set $\mathbf{f}_i = k_z \mathbf{z}_i^{\perp}$, where $k_z \neq 0$ is an arbitrary constant. In fact, this choice implies $\sum_{i=1}^n \mathbf{f}_i = k_z \sum_{i=1}^n \mathbf{z}_i^{\perp} = k_z \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^{\perp} = k_z Q \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G) = 0$. Note that this force can be computed by each agent in a distributed way, since \mathbf{z}_i is locally known thanks to Result 3.

By applying $\mathbf{f}_i = k_z \mathbf{z}_i^{\perp}$, the rotational dynamics (10) becomes $\dot{\omega} = k_z J^{-1} \sum_{i=1}^n \|\mathbf{z}_i\|^2 + \frac{1}{J} \sum_{i=1}^n \tau_i$. In order to further simplify the distributed computation, let us also impose $\tau_i = 0, \forall i = 1 \dots n$, limited to the time interval in which J is estimated. Hence, (10) is further simplified in

$$\dot{\omega} = J^{-1} k_z \sum_{i=1}^n \|\mathbf{z}_i\|^2. \quad (11)$$

Equation (11) expresses a linear relation where the only unknown is the proportionality factor J^{-1} . In fact, ω and $k_z \|\mathbf{z}_i\|^2$ are locally known to each agent i , which implies that the constant quantity $k_z \sum_{i=1}^n \|\mathbf{z}_i\|^2$ can be computed distributively through an average consensus [24] right after the moment in which each agent is able to estimate \mathbf{z}_i (block 7 in the diagram of Fig. 2). Therefore, the estimation of J is recast in (11) as a LLS estimation problem that can be solved resorting to the same strategy used to estimate d_{ij} in (7) (block 8 in the diagram of Fig. 2). A summary follows.

Result 5. *Each agent distributively computes the constant sum $k_z \sum_{i=1}^n \|\mathbf{z}_i\|^2$ using \mathbf{z}_i from Result 3 followed by average consensus. Then, each agent i applies a force $\mathbf{f}_i = k_z \mathbf{z}_i^{\perp}$ for a given time interval, the moment of inertia J is distributively computed by solving the LLS problem (11).*

Ideally, in a noise-free setting, every agent concludes this phase with the same estimate of J . In realistic settings, where noise is present, each agent may have a slightly different estimate of J . Hence, a standard average consensus algorithm can be executed to average out the noise and improve the estimate of J . Also, in this case such consensus will theoretically converge asymptotically, but the convergence to a bounded ball centered in the average can be achieved in finite time and detected by means of suitable distributed strategies [25].

B. Estimation of \mathbf{z}_C

The main idea behind the estimation of the time-varying quantity $\mathbf{z}_C(t)$ is to rewrite (10) in order to let only the following kinds of quantities appear (in addition to $\mathbf{z}_C(t)$):

- global quantities that can be distributively estimated;
- local quantities available from the problem setting (measurements or inputs) or from the previous results.

Next, we demonstrate that such a rewriting is possible and also that the estimation of $\mathbf{z}_C(t)$ boils down to a solvable nonlinear

observation problem. Let us first decompose the local force $\mathbf{f}_i(t)$ in two parts and recall two important identities

$$\mathbf{f}_i(t) = \frac{1}{n} \sum_{i=1}^n \mathbf{f}_i(t) + \Delta \mathbf{f}_i(t) = \mathbf{f}_{\text{mean}}(t) + \Delta \mathbf{f}_i(t), \quad (12)$$

$$\sum_{i=1}^n \mathbf{z}_i^{\perp \top} = 0 \quad \text{and} \quad \sum_{i=1}^n \Delta \mathbf{f}_i = 0. \quad (13)$$

We can then rewrite (10), using (12) and (13), as

$$\begin{aligned} \dot{\omega} &= J^{-1} \left(\sum_{i=1}^n \mathbf{z}_i^{\perp \top} \right) \mathbf{f}_{\text{mean}} + n J^{-1} \mathbf{z}_C^{\perp \top} \mathbf{f}_{\text{mean}}(t) + \\ &J^{-1} \sum_{i=1}^n \mathbf{z}_i^{\perp \top} \Delta \mathbf{f}_i + J^{-1} \mathbf{z}_C^{\perp \top} \sum_{i=1}^n \Delta \mathbf{f}_i + J^{-1} \sum_{i=1}^n \tau_i = \\ &\underbrace{n J^{-1} \mathbf{z}_C^{\perp \top} \mathbf{f}_{\text{mean}}}_{\mathbf{z}_C^{\perp \top} \bar{\mathbf{f}}} + \underbrace{J^{-1} \sum_{i=1}^n \mathbf{z}_i^{\perp \top} \Delta \mathbf{f}_i}_{\eta_1} + \underbrace{J^{-1} \sum_{i=1}^n \tau_i}_{\eta_2}, \text{ i.e.,} \\ \dot{\omega} &= \mathbf{z}_C^{\perp \top} \bar{\mathbf{f}} + \eta_1 + \eta_2. \end{aligned} \quad (14)$$

The global quantities¹

- $\bar{\mathbf{f}} = \frac{n}{J} \mathbf{f}_{\text{mean}}$,
- $\eta_1 = J^{-1} \sum_{i=1}^n \mathbf{z}_i^{\perp \top} \Delta \mathbf{f}_i = J^{-1} \sum_{i=1}^n \mathbf{z}_i^{\perp \top} \mathbf{f}_i$, and
- $\eta_2 = J^{-1} \sum_{i=1}^n \tau_i$

can be all distributively estimated in parallel using three instances of the dynamic consensus algorithm [22] (blocks 9, 10, and 11 in the diagram of Fig. 2). The choice of the specific dynamic consensus algorithm strongly depends on the nature of the tracked signals. A dynamic consensus algorithm like the one presented in [23] enables the estimate of its convergence time given the rate of convergence. The only mild assumption made is that the input signals are continuous and bounded (in [23], Theorem 5.1). The global quantity $\omega(t)$ is known thanks to Result 4. The only unknown in (14) is $\mathbf{z}_C(t)$. Define $\eta = \eta_1 + \eta_2$. The following result holds:

Result 6. *The rotational dynamics is given by*

$$\dot{\omega} = \mathbf{z}_C^{\perp \top} \bar{\mathbf{f}} + \eta, \quad (15)$$

where $\omega(t)$ is known thanks to Result 4 and $\bar{\mathbf{f}}(t)$ and $\eta(t)$ are locally known to each agent through distributed computation.

We use Eq. (15) to form a dynamical system where ω and $\mathbf{z}_C(t)$ are the state variables and $\bar{\mathbf{f}}$ and η are the inputs. Recalling that \mathbf{z}_C is a constant-norm vector, rigidly attached to the object, the following holds:

$$\dot{\mathbf{z}}_C = \omega \mathbf{z}_C^{\perp}. \quad (16)$$

Combining (15) and (16), we obtain the nonlinear system

$$\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_1 u_2 - x_2 u_1 + u_3 \end{cases}, \quad y = x_3, \quad (17)$$

where $\mathbf{z}_C = (z_C^x \ z_C^y)^{\top} = (x_1 \ x_2)^{\top}$ is the unknown part of the state vector, $\omega = x_3$ is the measured part of the state vector and, consequently, can be considered as the system output, and $\bar{\mathbf{f}} = (\bar{f}_x \ \bar{f}_y)^{\top} = (u_1 \ u_2)^{\top}$, $\eta = u_3$ are known inputs.

¹The same considerations made in Remark IV.1 are in order.

Result 7. *Estimating $\mathbf{z}_C(t)$ is equivalent to observe the state $(x_1 \ x_2)^{\top}$ of the nonlinear system (17) with known output $y = x_3 = \omega$ and known inputs $u_1 = \bar{f}_x$, $u_2 = \bar{f}_y$, and $u_3 = \eta$.*

In [26], we studied the observability of (17):

Proposition 1. *If $x_3 \neq 0$ and $(u_1 \ u_2)^{\top} \neq \mathbf{0}$, then system (17) is locally observable in the sense of [27].*

Proof. Given in [26]. \square

Note that the applied torques τ_i , for $i = 1 \dots n$ (which are included in u_3) have no influence on the observability of $\mathbf{z}_C(t)$. In [28], an observability condition that involves the angular velocity is also given. However, the condition expressed in [28] pertains the estimation of the kinematic parameters and requires that the direction of the angular velocity does not remain constant over the time, while in our setting the direction is constant and we only require that the norm is not constantly zero. In [26], we also proposed a nonlinear observer for system (17), which is summarized in the following result.

Proposition 2. *Consider the following dynamical system*

$$\begin{cases} \dot{\hat{x}}_1 = -\hat{x}_2 x_3 + u_2 (y - \hat{x}_3) \\ \dot{\hat{x}}_2 = \hat{x}_1 x_3 - u_1 (y - \hat{x}_3) \\ \dot{\hat{x}}_3 = \hat{x}_1 u_2 - \hat{x}_2 u_1 + k_e (y - \hat{x}_3) + u_3 \end{cases}, \quad (18)$$

where $k_e > 0$. If $y(t) \neq 0$ and $(u_1(t) \ u_2(t))^{\top} \neq \mathbf{0}$, then (18) is an asymptotic observer for (17). Hence, defining $\hat{\mathbf{x}} = (\hat{x}_1 \ \hat{x}_2 \ \hat{x}_3)^{\top}$ and $\mathbf{x} = (x_1 \ x_2 \ x_3)^{\top}$, one has that $\hat{\mathbf{x}}(t) \rightarrow \mathbf{x}(t)$ asymptotically.

Proof. Given in [26]. \square

Thanks to Proposition 2 we can state the following result:

Result 8. *The relative position of the CoM w.r.t. the center of the contact points, i.e., $\mathbf{z}_C(t)$, is distributively computed by using the observer (18) and thanks to the local knowledge of n , J , ω , \mathbf{f}_{mean} , and $\sum_{i=1}^n \mathbf{z}_i^{\perp \top} \Delta \mathbf{f}_i$ from the previous results.*

The estimation of $\mathbf{z}_C(t)$ described beforehand is schematized in blocks 9, 10, 11 (dynamic consensus algorithms) and 12 (observer introduced in (18)) in the diagram of Fig. 2.

The inputs $u_{1,2,3}$ of the observer arise from a dynamic consensus phase. As already stated, dynamic consensus algorithms converge asymptotically, thus, only the convergence to a ball centered in the average of the values of nodes can be guaranteed in finite time, in dependence of the convergence rate [23]. For this reason, it is important to analyze what happens to the observer's state when an additive input disturbance is present. To this aim, we define $\tilde{u}_i = u_i + \varepsilon_i$, with $i = 1, 2, 3$ and analyze the dynamics of $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$

$$\begin{cases} \dot{e}_1 = -e_2 x_3 + \tilde{u}_2 e_3 \\ \dot{e}_2 = e_1 x_3 - \tilde{u}_1 e_3 \\ \dot{e}_3 = e_1 \tilde{u}_2 - e_2 \tilde{u}_1 - k_e e_3 - \varepsilon_3 \end{cases}. \quad (19)$$

Defining the class \mathcal{K} function² $\gamma(r) = \frac{r}{k_e(\theta-1)}$, where $0 < \theta < 1$ and $k_e > 0$, the following result holds.

Proposition 3. *The error system of (18), i.e., (19), is Input-to-State Stable (ISS) with $\gamma(r) = \frac{r}{k_e(\theta-1)}$, where $0 < \theta < 1$ and $k_e > 0$, according to Definition 4.7, [29].*

Proof. Provided in the technical report associated with this paper, which can be downloaded at: <https://arxiv.org/abs/1602.01891> \square

C. Estimation of \mathbf{v}_C

The velocity of the center of mass $\mathbf{v}_C(t)$ is estimated locally by each agent i using the rigid body constraint $\frac{d}{dt}(\mathbf{p}_C - \mathbf{p}_{C_i}) = \omega(\mathbf{p}_C - \mathbf{p}_{C_i})^\perp$, which can be rewritten as

$$\mathbf{v}_C(t) = \mathbf{v}_{C_i}(t) - \omega(t)(\mathbf{z}_C(t) + \mathbf{z}_i(t))^\perp, \quad (20)$$

whose right-hand-side elements are all known since:

- $\mathbf{v}_{C_i}(t)$ is locally measured by agent i
- $\omega(t)$, $\mathbf{z}_C(t)$, and $\mathbf{z}_i(t)$ are known by each agent i thanks to Results 4, 8, and 3, respectively.

Result 9. *The CoM velocity $\mathbf{v}_C(t)$ is distributively computed using (20) and the knowledge of $\mathbf{v}_{C_i}(t)$, $\omega(t)$, $\mathbf{z}_C(t)$, $\mathbf{z}_i(t)$.*

Block 13 in the diagram of Fig. 2 represents this part.

D. Estimation of the mass m

The estimation of the mass m is a straightforward consequence of the estimation of the $\mathbf{v}_{C_i}(t)$ and average force. In fact, rewriting (2) as $\dot{\mathbf{v}}_C = \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i = \frac{n}{m} \mathbf{f}_{\text{mean}}$, we obtain

$$\dot{\mathbf{v}}_C = m^{-1} n \mathbf{f}_{\text{mean}}, \quad (21)$$

where *i*) n is known, *ii*) \mathbf{f}_{mean} is distributively estimated from \mathbf{f}_i using dynamic consensus (as in Result 6), and *iii*) \mathbf{v}_C is known locally by each agent i thanks to Result 9. Thus, the problem is recast as the linear least square estimation problem (21) that can be solved resorting to the same strategy used to estimate d_{ij} in (7) and J in (11).

Result 10. *The mass m is distributively computed from the knowledge of \mathbf{v} and n , and \mathbf{f}_{mean} by solving an online linear least square problem via filtering (21).*

Block 14 in the diagram of Fig. 2 represents this part.

VI. INERTIA AND MASS CHANGING DURING THE TASK

In some particular cases, it might happen that the values of J and m change during the manipulation because, e.g., an object which was part of the load is dropped or, viceversa, a new object is added to the load. Such changes cause discrete jumps at certain instants in the values of J and m , which could be estimated again using the methods proposed in Sections V-A and V-D. However, in order to do so the robots would need to detect that J and m have changed and to trigger again the estimation algorithms. Furthermore, the estimation of J proposed in V-A requires that each agent applies a

²According to Definition 4.2, [29], a continuous function $\alpha : [0, a) \rightarrow [0, \infty)$ is said to belong to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$.

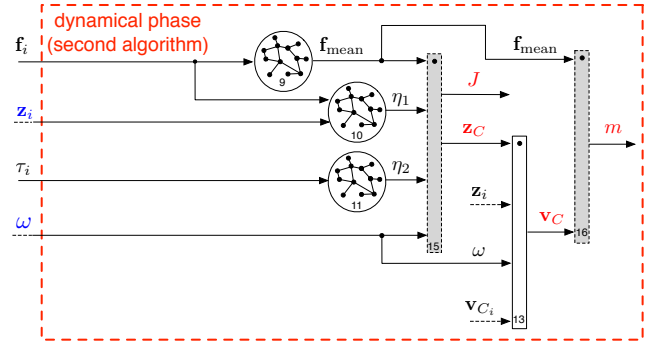


Fig. 3: Overview of the dynamical phase for the second algorithm. Compared to the dynamical part of the first algorithm (Fig. 2) the consensus 7 and the step estimator 8 blocks are removed; while block 12 is replaced by the new estimator (24) (new gray block 15); and block 14 is replaced by the new estimator (26) (new gray block 16).

specific force $\mathbf{f}_i = k_z \mathbf{z}_i^\perp$. However, in some cases it might be inconvenient to temporarily pause the manipulation and apply such forces for estimating again J .

In order to deal with such possibilities we propose here a variant of the first estimation algorithm which is based on two new estimators, see Fig. 3. The first is an alternative observer that extends (18) including J in the state vector. Such estimation method can run in the background during the manipulation, thus overcoming all the aforementioned possible pitfalls. The second is a new observer for m that can also run in the background and does not need any trigger or special coordination among agents. Both observers are derived next.

Assuming that J is not known, Result 6 can be recast as:

Result 11. *The rotational dynamics is given by*

$$\dot{\omega} = J^{-1} \mathbf{z}_C^\perp \tilde{\mathbf{f}} + J^{-1} \tilde{\eta}, \quad (22)$$

where $\omega(t)$ is known thanks to Result 4, and $\tilde{\mathbf{f}} = n \mathbf{f}_{\text{mean}}$ and $\tilde{\eta} = J \eta_1 + J \eta_2 = \sum_{i=1}^n \mathbf{z}_i^\perp \mathbf{f}_i + \sum_{i=1}^n \tau_i$ are locally known to each agent through distributed computation.

Combining (22) and (16), and knowing the fact that $\dot{J} = 0$ at any time except in the few isolated instants in which the unknown load inertia undergoes discrete, step-like changes, we obtain an extended version of (17)

$$\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_4 x_1 u_2 - x_4 x_2 u_1 + x_4 u_3 \\ \dot{x}_4 = 0 \end{cases}, \quad y = x_3, \quad (23)$$

where $(x_1 \ x_2)^\top = \mathbf{z}_C = (z_C^x \ z_C^y)^\top$ and $x_4 = J^{-1}$ are the unknown parts of the state vector, $x_3 = \omega$ is its measured part and, consequently, can be considered as the system output, and $(u_1 \ u_2)^\top = \mathbf{f}$, $u_3 = \tilde{\eta}$ are known inputs.

Proposition 4. *Consider the following dynamical system*

$$\begin{cases} \dot{\hat{x}}_1 = -\hat{x}_2 x_3 + u_2 (y - \hat{x}_3) \\ \dot{\hat{x}}_2 = \hat{x}_1 x_3 - u_1 (y - \hat{x}_3) \\ \dot{\hat{x}}_3 = \hat{x}_1 u_2 - \hat{x}_2 u_1 + \hat{x}_4 u_3 + k_e (y - \hat{x}_3) \\ \dot{\hat{x}}_4 = -u_3 (y - \hat{x}_3), \end{cases} \quad (24)$$

where $k_e > 0$. If $y(t) \not\equiv 0$, $u_3(t) \not\equiv 0$ and $(u_1(t) \ u_2(t))^\top \not\equiv \mathbf{0}$, then (24) is an asymptotic observer of a modified version of (23), in which the following change of variable is performed $x_1 \rightarrow x_1 x_4$, $x_2 \rightarrow x_2 x_4$. Thus, defining $\hat{\mathbf{x}} = (\hat{x}_1 \ \hat{x}_2 \ \hat{x}_3 \ \hat{x}_4)^\top$ and $\mathbf{x} = (x_1 x_4 \ x_2 x_4 \ x_3 \ x_4)^\top$, one has that $\hat{\mathbf{x}}(t) \rightarrow \mathbf{x}(t)$ asymptotically, which in turn implies that $(\hat{x}_1/\hat{x}_4 \ \hat{x}_2/\hat{x}_4) \rightarrow (x_1 \ x_2)$ asymptotically.

Proof. Provided in the technical report associated with this paper, which can be downloaded at: <https://arxiv.org/abs/1602.01891> \square

In the ideal case, the estimates carried out by all the robots are identical. In the case of noisy measurements, the noise can be averaged out by means of a dynamic consensus algorithm [23], as done for the estimation of ω .

In order to observe m , we design an observer for the following nonlinear dynamical system

$$\begin{cases} \dot{z}_1 = z_2 u \\ \dot{z}_2 = 0 \end{cases}, \quad y = z_1, \quad (25)$$

which is easily derived from (21) by defining $z_1 = v_{C,x} + v_{C,y}$, $z_2 = m^{-1}$, $u = f_{\text{mean},x} + f_{\text{mean},y}$, and by imposing $\dot{m} = 0$ for the same reasons for which it was previously assumed $\dot{J} = 0$.

Proposition 5. Consider the following dynamical system

$$\begin{cases} \dot{\hat{z}}_1 = \hat{z}_2 u + k_1(y - \hat{z}_1) \\ \dot{\hat{z}}_2 = k_2(y - \hat{z}_1), \end{cases} \quad (26)$$

where $k_1, k_2 \in \mathbb{R}$. If $k_1 > 0$, $k_2 u > 0$, and $k_1^2 > 4k_2 u$ hold, then (26) is an asymptotic observer of (25).

Proof. Provided in the technical report associated with this paper, which can be downloaded at: <https://arxiv.org/abs/1602.01891> \square

The same observation made for the estimation of a time-varying J is valid in this case: in the ideal case of noiseless measurements, the estimates carried out by all the robots are identical. In the case of noisy measurements, the noise can be averaged out by means of a dynamic consensus algorithm [23].

VII. DECENTRALIZED OBSERVABILITY-BASED CONTROL

In the previous sections, we have shown how to solve the Problem in Sec. II. Apart from the phase of the first-algorithm in which J is estimated (Sec. V-A), in all the other phases we did not suggest any control input to move the load and perform the estimation. The user of the algorithm is free to use any control input, as long as it ensures the observability conditions, i.e., non-zero angular rate ω and non-zero average force \mathbf{f}_{mean} . In each phase, either one or both of the two conditions are needed to ensure a convergent estimation.

In the following, we prove that an extremely basic control strategy satisfies, under very mild conditions, the aforementioned observability requirements. Furthermore, this control strategy: (i) can be implemented relying only on local perception and communication (it is, therefore, distributed); and (ii) does not require the knowledge of parameters and quantities that are the objectives of the distributed estimation

(it is estimation-‘agnostic’). Hence, it can be applied during the estimation process and independently from it.

Proposition 6. Assume that the following local control rule is used: $\mathbf{f}_i = \mathbf{f}^* = \text{const}$, $\tau_i = 0$, $\forall i = 1 \dots n$, and denote with ω_0 the rotational rate at $t = 0$, then

1) $\omega(t)$ remains bounded, in particular:

$$|\omega(t)| \leq \sqrt{\omega_0^2 + 4nJ^{-1}\|\mathbf{f}^*\|\|\mathbf{z}_C\|} \quad \forall t \geq 0 \quad (27)$$

2) $\exists \bar{t} \geq 0$ such that ω becomes $\omega \equiv 0 \ \forall t \geq \bar{t}$, if and only if the following condition hold

$$2n\mathbf{z}_C(0)^\top \mathbf{f}^* - J\omega^2(0) = 2n\|\mathbf{z}_C\|\|\mathbf{f}^*\|. \quad (28)$$

Thus, the proposed control law is suitable for the estimation process apart from a zero-measure case implied by (28).

Proof. Provided in the technical report associated with this paper, which can be downloaded at: <https://arxiv.org/abs/1602.01891> \square

The use of the local control action in Proposition 6 ensures the sought observability conditions under the very mild conditions specified in (28). However, it causes the load CoM velocity to grow linearly over time (see, e.g., (21)). Therefore, it is wise to modify the proposed control action by periodically changing the direction of the common force (i.e., switching between \mathbf{f}^* and $-\mathbf{f}^*$ on a periodical basis). In this way, the CoM velocity will boundedly oscillate around zero.

It is also important to define a control strategy that is able to stop the load motion if needed (like, e.g., at the end of all the estimation phases). This is provided in the following.

Proposition 7. Assume that the following local control rule is used: $\mathbf{f}_i = -b\mathbf{v}_{C_i}$, $\tau_i = 0$, $\forall i = 1 \dots n$, with $b > 0$. Then, both ω and \mathbf{v}_C converge asymptotically to zero with a convergence rate that is proportional to b .

Proof. Provided in the technical report associated with this paper, which can be downloaded at: <https://arxiv.org/abs/1602.01891> \square

VIII. NUMERICAL RESULTS

This section and the Appendix show numerical results that validate our approach. First, we demonstrate the working principles of the first algorithm simulating a network of $n = 10$ agents manipulating an unknown load on a plane. Then, the validity of the first algorithm is extensively assessed through a detailed simulation campaign in a wide range of operational conditions. Finally we show some simulations for the second algorithm using time-varying mass and inertia observers.

A. Manipulation of an unknown load by a team of 10 agents

We consider a planar load with $m = 50 \text{ kg}$ and $J = 86.89 \text{ kg m}^2$, manipulated by a team of $n = 10$ agents communicating over a line-topology network. Such a topology is the worst case for the algorithm convergence rate, which is an increasing function of the network diameter [30]. The velocity measurements are affected by an additive zero-mean Gaussian noise with covariance matrix $\Sigma_i = \sigma^2 \mathbf{I}_{2 \times 2}$, and

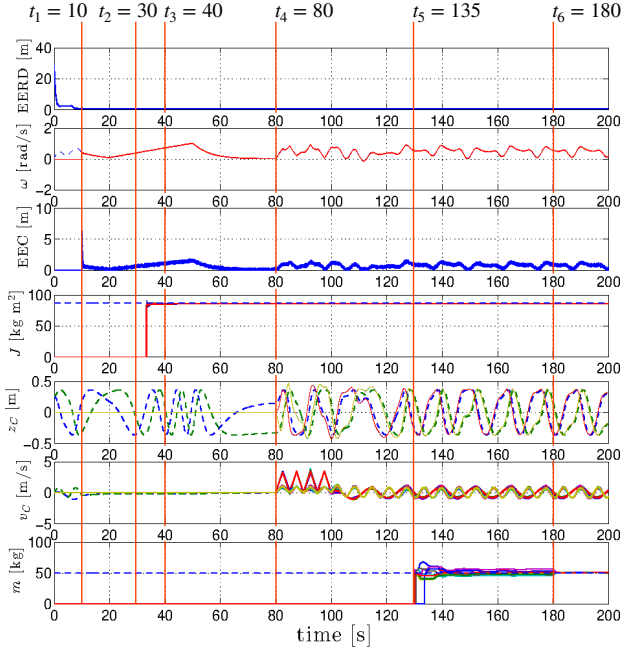


Fig. 4: An illustrative simulation of the whole estimation algorithm described in Secs. IV and V. From top to bottom, respectively: the trend of the EERD index, the estimates of ω , the trend of the EEC index, the estimates of J , the observations of \mathbf{z}_C , the estimates of \mathbf{v}_C , and the estimates of m . Dashed lines indicate true values, while solid lines indicate estimates.

$\sigma = 0.3$ m/s. The quantities involved in the execution and assessment of the first algorithm are illustrated in Fig. 4.

As a first step, starting from $t_0 = 0$, each agent applies an arbitrary force and executes the procedure described in Sec. IV-A to estimate the relative distances between contact points. We observe that the presence of noise in the velocity measurements can make the signal-to-noise ratio too small to make an effective use of acquired measurement. In this case, we opt for keeping the last valid measurement as constant, and not to update the measurement with quantities that are too noisy to be useful. The signal-to-noise ratio threshold in order to consider the measured velocity is set to $\|\dot{\mathbf{z}}_{ij}\| \leq 0.5$ m/s. The first plot of Fig. 4 reports the convergence to zero of the Estimation Error Relative Distance (EERD) index, defined as $\text{EERD}(t) = \sum_{i=1}^{n-1} \sum_{j=i}^n \mathcal{A}(i, j) \|(\mathbf{z}_{ij}(t) - \hat{\mathbf{z}}_{ij}(t))\|_2$, where variable $\hat{\mathbf{z}}_{ij}$ indicates the estimate of \mathbf{z}_{ij} . Consistently, here and henceforth, the estimate of a quantity \star is indicated with a superimposed hat, i.e., $\hat{\star}$. Dynamic consensus blocks have been implemented by means of the First Order Input Dynamic Consensus Filter [23] with parameter values set as $\epsilon = 0.01$ and $\beta = 10$. Static average consensus detection is detected by means of the method presented in [25]. Starting from $t_1 = 10$ s, each agent applies the control rules given in Propositions 6 and 7, which guarantee both the observability and boundedness of $[\mathbf{v}_C^T \ \omega]^T$. At $t_1 = 10$ s, ω and \mathbf{z}_i start to be estimated, as described in Sec. IV-B. The second and third plots of Fig. 4 illustrate, respectively, the trend of the angular velocity ω and its estimate $\hat{\omega}$, and the quadratic performance index on the estimation of the relative distance to the center of mass, i.e., $\text{EEC}(t) = \sum_{i=1}^n \|(\mathbf{z}_i(t) - \hat{\mathbf{z}}_i(t))\|_2$. At $t = 20$ s,

the first step of the dynamical phase is executed, as described in Sec. V-A. First, each agent runs an average consensus in order to locally estimate the constant value $k_z \sum_{i=1}^n \|\mathbf{z}_i\|^2$. Such consensus will theoretically converge asymptotically. However, we use the technique presented in [25] to assess a suitable stopping condition. Specifically, we distributively determine when the consensus has been reached within a given error bound, which in our experiment is set to 0.001 m. At $t_3 = 30$ s each agent i runs a least square estimation of J using also the knowledge of $\hat{\omega}$. Each agent checks the convergence of the least squares estimation evaluating the variance of the estimator [31]. From $t_3 = 40$ s, the local estimates are transmitted over the network and an average consensus is run to agree on a common estimate, which in our case is $\hat{J} = 85.67$ kg m² (fourth plot in Fig. 4). Also in this case, we use the technique presented in [25] with error bound set to 0.01 kg m². Then, the angular rate is brought to zero (Proposition 7). Afterwards, at $t_4 = 80$ s each agent starts the nonlinear observation of \mathbf{z}_C described in Sec. V-B. The observer errors reach zero at about $t_5 = 135$ s, as illustrated in the fifth plot of Fig. 4. The estimate $\hat{\mathbf{v}}_C$ is then computed using (20) (sixth plot of Fig. 4), which in turn allows to compute \hat{m} , as explained in Sec. V-D, by a preliminary collections of samples and local least squares estimations. An average consensus phase, starting at $t_6 = 180$ s, leads to an accurate estimate of m at $t_f = 200$ s (seventh and last plot of Fig. 4). The same techniques previously described to detect consensus convergence [25] is applied here using a bound of 0.01 kg. The duration of the entire algorithm is 200 s, of which a large portion is needed to collect samples to run the local least squares and the consensus algorithms for the constant parameters m , J and d_{ij} . The duration of these phases depends on the noise level. Ideally, in the absence of noise, a single sample would be sufficient to perform the estimation, while in the real, noisy case, a trade-off between robustness [20] and duration of the estimation phase is necessary. Finally, also the convergence time of $\hat{\mathbf{z}}_C$ can be shortened by acting on the value of k_e in (18), and the gains of the consensus algorithms can be tuned in order to speed up the agreement [24].

Additional extensive results are given in the Appendix.

IX. CONCLUSIONS

In this paper, we propose two fully-distributed methods for the estimation of the parameters needed by a planar multi-agent system to collectively manipulate an unknown load, i.e., the kinematic and dynamic parameters, as well as the estimate of the kinematic state of the load, i.e., the velocity of the center of mass and its rotational rate. The approaches are totally distributed and rely on the geometry of the rigid body kinematics, on the rigid body dynamics, on nonlinear observation theory, and on consensus strategies. They are based on a sequence of steps that leads to states in which all agent agrees on the estimated parameters, during such steps any motion control law can be used (apart from a single step in the first algorithm). The only requirements are related to the communication network, which is only required to be connected, and to the capability of each agent to control

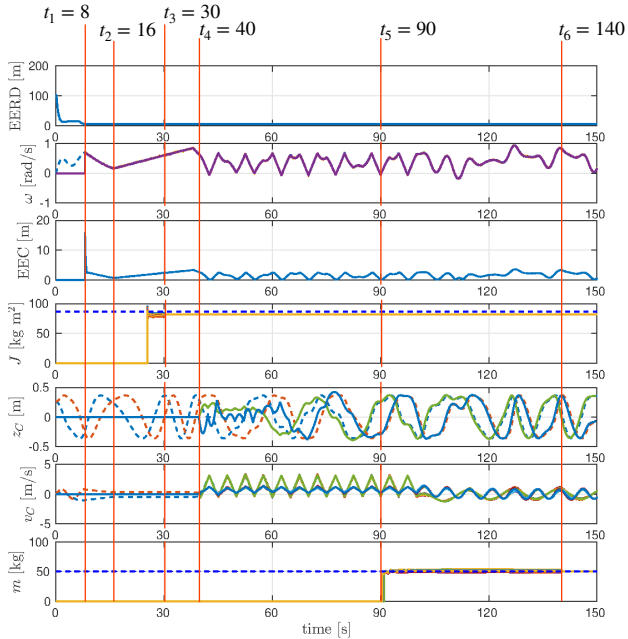


Fig. 5: Simulation with the same setup as in Fig. 4, but with a fully connected topology and noise $\sigma = 0.3$ m/s.

the local force applied to the load, while measuring the velocity of the contact point. Extensive numerical simulations confirm the effectiveness of our approach and its robustness to measurement noise and system size. Future works will deal with the manipulation of 3D objects in the aerial domain, extending the aerial manipulation methods in [32]–[34].

APPENDIX

A. Manipulation of an unknown load by a team of 10 agents

Figures. 5 and 6 illustrate the same simulation setup of Sec. VIII-A in the case of a fully connected topology and two different levels of noise. Comparing the time sequence $t_1 \dots t_6$, we observe that a more connected topology and a lower noise level are factors that sensibly reduce the execution time of each step.-

B. Performance Assessment and Uncertainty Propagation

We analyze the performance of the algorithm by running a Monte Carlo simulation campaign. Specifically, a wide range of operational conditions is considered, defined by the pair (n, σ) where n indicates the number of agents in the network and σ the standard deviation defining the covariance of a zero-mean Gaussian noise, added to velocity measurements, as $\Sigma_i = \sigma^2 \mathbf{I}_{2 \times 2}$, for each agent $i = 1, \dots, n$.

The two parameters assume values over a 2D grid formed with the values $n \in \{5, 8, 11, 14, 17, 20\}$ and $\sigma \in \{0.1, 0.2, 0.3, \dots, 1\}$ m/s. Fifty independent simulations are run for each parameter pair. To ensure consistency of comparison in each simulation the mass is set as $m = 5$ n kg (and the inertia J is computed accordingly). The agents communicate over the worst-case line-topology.

Figure 7 illustrates the results, where each box-plot corresponds to the 50 independent simulations executed for a

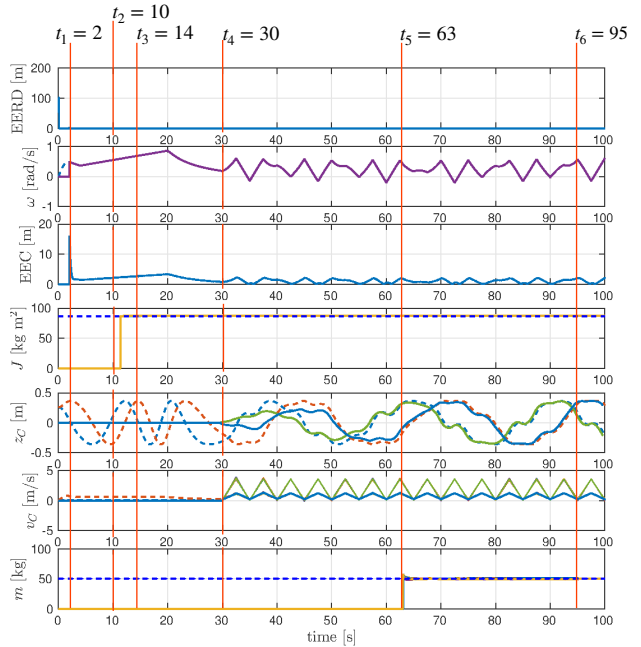


Fig. 6: Simulation with the same setup as in Fig. 4, but with a fully connected topology and noise $\sigma = 0.0001$ m/s.

given parameter pair. For consistency of comparison across the changing n , we use the Normalized Root Mean Square Error (NRMSE) and the Relative Error (RE), respectively for time-varying and constant parameters. The results show that the estimation accuracy decreases with an increasing noise level σ for the EERD, the EEC, the estimation of ω , the estimation of \mathbf{v}_C , and the estimation of J . On the contrary, the degradation in the estimation accuracy is of minor importance in the estimation of m . We observe that the error in the estimation of J is strongly dependent on n : fixing a value for σ , the estimation accuracy increases with an increase in the number of agents. This is related to the use of the consensus algorithm for averaging out the noise, as will be better shown later in this section. On the other hand, fixing the value for σ , the dispersion of the RE in the estimation of m increases with increasing n . This behavior is related to the greater number of noisy measurements used to estimate \mathbf{v}_C , used in turn to estimate m . The degree of dispersion of the estimation of ω is clearly influenced by the noise level, the dispersion is close to zero for the EERD, EEC, and for the error in the estimation of J . Finally, fixing the number of agents n , the degree of the dispersion is constant with respect to the noise variation for the estimation error of \mathbf{v}_C and m , confirming that the estimation of \mathbf{v}_C and m is strongly influenced by the level of noise.

As already mentioned in Sec. V-A, for the estimation of constant parameters, such as J and m , an average consensus algorithm is executed to average out the noise and improve the estimate. Figure 8 shows the box-plots of the RE of the estimation of J and m , before (on the left side) and after (on the right side) the average consensus run. We observe that the application of the average consensus algorithm decreases significantly the average estimation error and also the dispersion.

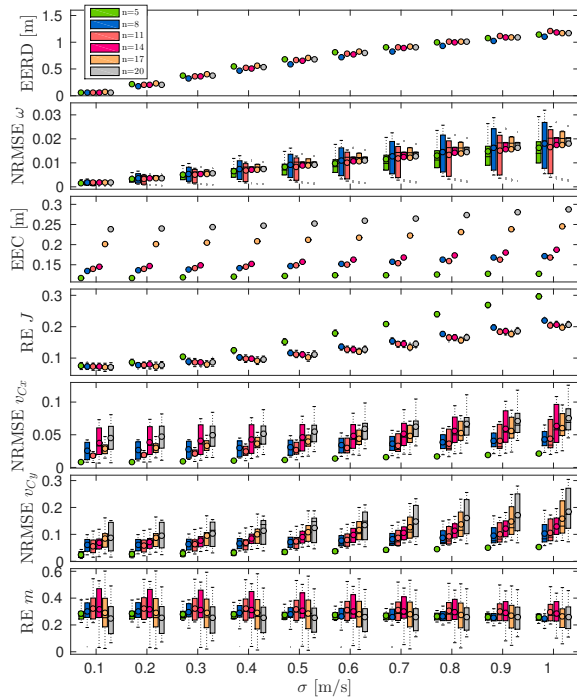


Fig. 7: Box-plots obtained by running the whole algorithm in different operational conditions, selected by setting the pair (n, σ) . The bottom of a box indicates the first quartile, while the top indicates the third quartile. The dot inside a box is the second quartile, i.e., the median. The whisker at the bottom/top of a box indicates the lowest/highest data within 1.5 of the Interquartile Range of the lower/upper quartile.

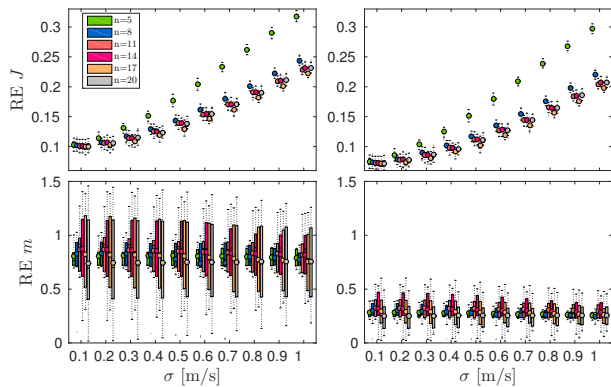


Fig. 8: Box-plots of the Relative Error (RE) of the estimation of m and J obtained by running the whole algorithm in different conditions described by the pairs (n, σ) . The left and right column show the RE's before and after running the average consensus, respectively.

C. Inertia Moment and Mass Changing along the Task

To show the ability of the second algorithm to deal with changing m and J we simulate a planar load with $m = 50$ kg and $J = 86.89$ kg m², manipulated by a team of $n = 10$ agents communicating over a line-topology network and implementing the observers introduced in Sec. VI. A zero-mean Gaussian noise with covariance matrix $\Sigma_i = \sigma^2 \mathbf{I}_{2 \times 2}$ and $\sigma = 0.3$ m/s is added to the velocity measurements. After 100s, we simulate a step-like decrease of mass m and, consequently, a decrease of the moment of inertia J . Simulation results are illustrated in Fig. 9. As theoretically proven, both the observers converge

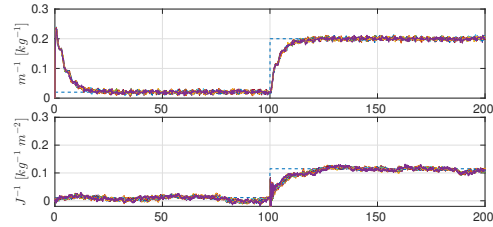


Fig. 9: Simulation of the performance of the observer for the time-varying inertial parameters described in Sec. VI. To: estimates of m^{-1} . Bottom: the estimates of J^{-1} . Dashed lines indicate true values, while solid lines indicate estimates.

to a bounded region around the new true values.

REFERENCES

- [1] K. I. Kim and Y. F. Zheng, "Two strategies of position and force control for two industrial robots handling a single object," *Robotics and Autonomous Systems*, vol. 5, no. 4, pp. 395–403, 1989.
- [2] S. A. Schneider and R. H. Cannon, "Object impedance control for cooperative manipulation: theory and experimental results," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.
- [3] I. D. Walker, R. A. Freeman, and S. I. Marcus, "Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators," *Robotics and Autonomous Systems*, vol. 10, no. 4, pp. 396–409, 1991.
- [4] J. Szewczyk, F. Plumet, and P. Bidaud, "Planning and controlling cooperating robots through distributed impedance," *Journal of Robotic Systems*, vol. 19, no. 6, pp. 283–297, 2002.
- [5] 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.
- [6] Y. Yong, T. Arima, and S. Tsujio, "Inertia parameter estimation of planar object in pushing operation," in *2005 IEEE Int. Conf. on Information Acquisition*, Hong Kong and Macau, China, June 2005, pp. 356–361.
- [7] D. Kubus, T. Kroger, and F. M. Wahl, "On-line estimation of inertial parameters using a recursive total least-squares approach," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, Sep. 2008, pp. 3845–3852.
- [8] S. Erhart and S. Hirche, "Adaptive force/velocity control for multi-robot cooperative manipulation under uncertain kinematic parameters," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 307–314.
- [9] 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.
- [10] Z. Wang and M. Schwager, "Multi-robot manipulation with no communication using only local measurements," in *54th IEEE Conf. on Decision and Control*, Osaka, Japan, Dec. 2015, pp. 380–385.
- [11] P. Culbertson and M. Schwager, "Decentralized adaptive control for collaborative manipulation," in *2018 IEEE Int. Conf. on Robotics and Automation*, Brisbane, Australia, May 2018, pp. 278–285.
- [12] A. Marino and F. Pierri, "A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots," *Robotics and Autonomous Systems*, vol. 103, pp. 122–133, 2018.
- [13] G. Tel, *Introduction to Distributed Algorithms*. Cambridge University Press, 2000.
- [14] B. Francis and M. Maggiore, "Models of mobile robots in the plane," in *Flocking and Rendezvous in Distributed Robotics*, B. Francis and M. Maggiore, Eds. Springer, 2017, pp. 7–23.
- [15] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 671–700.
- [16] A. Pettiti, A. Franchi, D. Di Paola, and A. Rizzo, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators," in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 441–446.
- [17] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bühlhoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

- [18] N. E. Manitara and C. N. Hadjicostis, "Distributed stopping for average consensus in digraphs," *IEEE Trans. on Control of Network Systems*, 2017.
- [19] 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.
- [20] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991.
- [21] 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.
- [22] M. Zhu and S. Martinez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.
- [23] S. S. Kia, J. Cortés, and S. Martínez, "Singularly perturbed algorithms for dynamic average consensus," in *2013 European Control Conference*, Zurich, Switzerland, Dec. 2013, pp. 1758–1763.
- [24] 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.
- [25] V. Yadav and M. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *2007 45th Allerton Conf. on Communications, Control and Computing*, Monticello, IL, Sep. 2007, p. 715720.
- [26] 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.
- [27] R. Hermann and A. J. Krener, "Nonlinear controllability and observability," *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.
- [28] F. Aghili, "Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model," *IEEE/ASME Trans. on Mechatronics*, vol. 18, no. 5, pp. 1544–1554, 2013.
- [29] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2001.
- [30] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Review*, vol. 53, no. 4, pp. 747–772, 2011.
- [31] S. Van de Geer, "Least squares estimators," ser. Encyclopedia Statistics in The Behavioral Sciences, B. Everitt and D. Howell, Eds. John Wiley, 2005.
- [32] 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.
- [33] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bühlhoff, 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.
- [34] B. Yüksel, C. Secchi, H. H. Bühlhoff, and A. Franchi, "Reshaping the physical properties of a quadrotor through IDA-PBC and its application to aerial physical interaction," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014, pp. 6258–6265.



Antonio Franchi (S'07-M'11-SM'16) received the Ph.D. degree in system engineering from Sapienza University of Rome, Rome, Italy, in 2010 and the Habilitation to Direct Research (HDR) in Sciences from National Polytechnic Institute of Toulouse, Toulouse, France, in 2016. In 2009, he was a Visiting Scholar with University of California at Santa Barbara, Santa Barbara, CA, USA. From 2010 to 2014, he was Research Scientist, Senior Research Scientist, and the Project Leader of the Autonomous Robotics and Human Machine Systems Group, Max Planck

Institute for Biological Cybernetics in Tübingen, Germany. Since 2014, he has been a Tenured CNRS Researcher with the RIS team, LAAS-CNRS, Toulouse, France. He published more than 120 papers in peer-reviewed international journals and conferences. His main research interests include robotic systems, with a special regard to control of for aerial robots and multiple-robot systems. Dr. Franchi received the IEEE RAS ICYA Best Paper Award in 2010. He is an Associate Editor for IEEE TRANSACTIONS ON ROBOTICS. He is the co-founder of the IEEE RAS Technical Committee on Multiple Robot Systems and of the International Symposium on Multi-Robot and Multi-Agent Systems.



Antonio Petitti received the B.S. and M.S. (Laurea Specialistica) degrees (summa cum laude) in Automation Engineering from Politecnico di Bari, Italy, in 2008 and 2010, respectively. From June 2011 to May 2018, he was Research Assistant at the Institute of Intelligent Systems for Automation (ISSIA) of the National Research Council (CNR), Italy. In 2015, he received the Ph.D. degree in Electrical and Information Engineering at Politecnico di Bari, Italy, and the joint Ph.D. degree of high qualification Scuola Interpolitecnica di Dottorato in Information and Communication Technologies. In 2013 and 2014 he was Visiting Research Fellow at ARHMS group, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, and at RIS group LAAS-CNRS, Toulouse, France, respectively. Since June 2018, he has been Researcher at the Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA) of the CNR, Bari, Italy. His scientific interests are focused on consensus theory and applications, distributed estimation, modeling and control of robotic networks.



Alessandro Rizzo (SM'08) received the Laurea degree (summa cum laude) in computer engineering and the Ph.D. degree in automation and electronics engineering from the University of Catania, Italy, in 1996, and 2000, respectively. He is an Associate Professor at Politecnico di Torino, Italy, where he is engaged in conducting and supervising research on cooperative robotics, complex networks and systems, modeling and control of nonlinear systems. Since 2012, he has also been a Visiting Professor at the New York University Tandon School of Engineering,

Brooklyn, NY, USA. He is the author of two books, two international patents, and more than 130 papers on international journals and conference proceedings. Prof. Rizzo has been the recipient of the award for the best application paper at the IFAC world triennial conference in 2002 and of the award for the most read papers in Mathematics and Computers in Simulation (Elsevier) in 2009. Prof. Rizzo is also a Distinguished Lecture of the IEEE Nuclear and Plasma Science Society. More details can be found at the website staff.polito.it/alessandro.rizzo.