Distributed Estimation of State and Parameters in Multi-Agent Cooperative Load Manipulation
Antonio Franchi, Antonio Petitti, Alessandro Rizzo

To cite this version:
Antonio Franchi, Antonio Petitti, Alessandro Rizzo. Distributed Estimation of State and Parameters in Multi-Agent Cooperative Load Manipulation. IEEE Transactions on Control of Network Systems, 2018, 10.1109/TCNS.2018.2873153. hal-01879384

HAL Id: hal-01879384
https://hal.laas.fr/hal-01879384
Submitted on 23 Sep 2018

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.
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
II. MODEL AND PROBLEM STATEMENT

In this section, we formally define the problem of distributively estimating 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 \( W = O - xy \) and the load body frame with \( B = C - xyB \), where \( C \) is the center of mass (CoM) of the unknown body, indicated with \( B \). We indicate with \( p_C \in \mathbb{R}^2 \) and \( v_C = \dot{p}_C \) the position and velocity of \( C \) expressed in \( W \), 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

\[
\mathbf{M}\ddot{\mathbf{v}} + \mathbf{g} = \mathbf{u},
\]

where \( \mathbf{v} = (v_C^T, \omega)^T \in \mathbb{R}^3 \) 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 \( \mathbf{g} = 0 \). 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 \( W \).

Each agent \( i \) contributes to the overall manipulation by exerting a wrench \( \mathbf{u}_i = (f_i^T, \tau_i)^T \in \mathbb{R}^3 \), expressed in \( W \), with \( i = 1 \ldots n \). The force \( f_i \in \mathbb{R}^2 \) 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., \( C_i \neq C_j \), \( \forall i, j = 1 \ldots n \). The total external wrench applied to \( B \) is \( \mathbf{u} = \sum_{i=1}^n \mathbf{G}_i \mathbf{u}_i = \mathbf{G}\mathbf{u} \), where \( \mathbf{G}_i \in \mathbb{R}^{3 \times 3} \) is the partial grasp matrix, \( \mathbf{G} \in \mathbb{R}^{3 \times 3} \) is the grasp matrix, and \( \mathbf{u} = (u_1^T, \ldots, u_n^T)^T \) 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 \( \mathbf{G}_i = \mathbf{P}_i \mathbf{R}_i \), where \( \mathbf{P}_i = \left[ \begin{array}{c} \mathbf{I}_{2 \times 2} \ 0_{2 \times 1} \\ \mathbf{p}_C - \mathbf{p}_C \end{array} \right] \in \mathbb{R}^{3 \times 3} \) and \( \mathbf{R}_i = \mathbf{I}_{3 \times 3} \), in our setting, for all \( i = 1 \ldots n \). Here, \( \mathbf{p}_C \in \mathbb{R}^3 \) is the position of \( C_i \) in \( W \). The operator \( \cdot^\perp \) rotates a vector by an angle of \( \pi/2 \), as is defined as \( \mathbf{q}^\perp = Q\mathbf{q} = [-q'' \ q''']^T \) where \( Q = \begin{bmatrix} 0_{1 \times 1} & 0_{1 \times 1} \\ 0_{1 \times 1} & 1_{1 \times 1} \end{bmatrix} \). Hence, the dynamics (1) of the manipulated load is given by

\[
\begin{align*}
\dot{\mathbf{v}}_C &= \sum_{i=1}^n \left[ m^{-1}\mathbf{I}_{2 \times 2} 0_{2 \times 1} \mathbf{0}_{2 \times 1} \mathbf{I}_{2 \times 2} \right] \left[ J^{-1} \mathbf{Q}_{i} \right] \mathbf{f}_i, \\
\omega &= \sum_{i=1}^n \left[ m^{-1}\mathbf{I}_{2 \times 2} 0_{2 \times 1} \mathbf{0}_{2 \times 1} \mathbf{I}_{2 \times 2} \right] \left[ J^{-1} \mathbf{Q}_{i} \right] \mathbf{f}_i. 
\end{align*}
\]

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

\[
\begin{align*}
\dot{\mathbf{v}}_C &= \sum_{i=1}^n \left[ m^{-1}\mathbf{I}_{2 \times 2} 0_{2 \times 1} \mathbf{0}_{2 \times 1} \mathbf{I}_{2 \times 2} \right] \left[ J^{-1} \mathbf{z}_i^\perp \right] \mathbf{f}_i + \left[ 0_{2 \times 1} \mathbf{0}_{2 \times 1} \mathbf{0}_{2 \times 1} \mathbf{I}_{2 \times 2} \right] \mathbf{f}_i, \\
\omega &= \sum_{i=1}^n \left[ m^{-1}\mathbf{I}_{2 \times 2} 0_{2 \times 1} \mathbf{0}_{2 \times 1} \mathbf{I}_{2 \times 2} \right] \left[ J^{-1} \mathbf{z}_i^\perp \right] \mathbf{f}_i.
\end{align*}
\]

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 \( \omega \), 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 \( G \), whose both node and link set are assumed to be time-invariant. The set \( \mathcal{N} \) 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 \( \mathbf{v}_C \) 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:

---

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.
1) the (constant) mass $m$ of the load, 
2) the (constant) rotational inertia $J$ of the load, 
3) the (time-varying) relative position $z_i(t)$ of $C_i$ with respect to the geometric center $G$ of the contact-points, 
4) the (time-varying) relative position $z_C(t)$ of the Center of Mass (CoM) $C$ of $B$ with respect to $G$, 
5) the (time-varying) velocity $v_C(t)$ 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 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 $z_i(t)$ 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., $J$, $z_C(t)$, $v_C(t)$, 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 $\epsilon$-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 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 $z_i(t)$ 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 $z_i(t)$ and $\omega(t)$ in two parts. The former is common to both estimations and essentially consists of the estimation of $z_{ij}(t) = p_i(t) - p_j(t)$. This part is described in Sec. IV-A. The latter comprises two separate estimators of $z_i(t)$ and $\omega(t)$ and is described in Sec. IV-B.

The reason for passing through the estimation of the quantities $z_{ij}(t)$ 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 $z_i(t)$ if each pair of neighbors know the relative position of the contact points $C_i$ and $C_j$, i.e., $z_{ij}(t)$. 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 $z_{ij}(t)$ only resorting to the measures $v_i(t)$ and $v_j(t)$.
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

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

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 $\dot{\mathbf{z}}^T_{ij} \mathbf{z}_{ij} = 0$, which implies that the directions of $\mathbf{z}_{ij}$ and $\dot{\mathbf{z}}^T_{ij} = Q \dot{\mathbf{z}}_{ij}$ coincide. We can then explicitly decompose $\mathbf{z}_{ij}$ in two factors

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

where $\mathbf{y}_{ij} = \mathbf{z}^T_{ij}/||\mathbf{z}^T_{ij}|| \in S^1$ 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 $\mathbf{y}_{ij}$.

Let each agent send to all $j \in N_i$ the (measured) velocity of its contact point $\mathbf{v}_{C_j}$, 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 $\mathbf{h}_{ij}$, for each $j \in N_i$. As a consequence, $\mathbf{y}_{ij}$ is locally available to each agent $i$, $\forall j \in N_i$. This is the first milestone of our algorithm, which is formally stated in the following result.

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

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

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

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 $\mathbf{z}_{ij}$ and the time integral of $\frac{d}{dt} \mathbf{y}_{ij}$, i.e., $\mathbf{y}_{ij}$. 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 $||\mathbf{z}_{ij}|| = 0$, 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

- $\mathbf{y}_{ij}$ (time-varying axis), computed directly from $\mathbf{v}_{C_i} - \mathbf{v}_{C_j}$ (see Result 1)
- $d_{ij}$ (norm-constant coordinate along $\mathbf{y}_{ij}$), 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 $i$, $\forall j \in N_i$, each agent $i$ estimates $\mathbf{z}_i$ by using the centroid estimation algorithm described in [19].

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

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

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

$$\omega = -\left(\mathbf{z}^T_{ij} \dot{\mathbf{z}}^T_{ij}\right) \left(\mathbf{z}^T_{ij} \mathbf{z}_{ij}\right)^{-1}. \quad (9)$$

Result 4. $\omega$ is locally computed using (9), where $\mathbf{z}_{ij}$ 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 $\omega$ as the number of its one-hop neighbors $|N_i|$. In the ideal case of noiseless velocity measurements, all these 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 $\omega$ 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 $\omega$ 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 $v_i$, used to estimate $z_i$, 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 $z_C(t)$ of the CoM of $B$ relative to the geometric center $G$ of the contact points, the (time-varying) velocity of the CoM $v_C(t)$, 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 $v_C$, the applied wrench $u_i$, 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 $z_i$ to apply a particular input wrench that cancels the effect of $z_C$ in (3), thus, obtaining a reduced dynamics in which $J$ is the only unknown; then, we estimate $J$ using LLS;
2) *(estimation of $z_C(t)$)* we use all the previously estimated quantities, the rotational dynamics in (3), and the rigid body constraint to recast the estimation of $z_C$ to a nonlinear observation problem that can be locally solved by each agent with an observer designed by us;
3) *(estimation of $v_C(t)$)* we use rigid body kinematics to compute $v_C(t)$ 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 $v_C(t)$ 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)

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^{n} z_i^T f_i + \frac{1}{J} z_C^T \sum_{i=1}^{n} f_i + \frac{1}{J} \sum_{i=1}^{n} \tau_i, \tag{10}$$

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

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

$$\dot{\omega} = J^{-1} k_i \sum_{i=1}^{n} ||z_i||^2. \tag{11}$$

Equation (11) expresses a linear relation where the only unknown is the proportionality factor $J^{-1}$. In fact, $\omega$ and $k_i ||z_i||^2$ are locally known to each agent $i$, which implies that the constant quantity $k_i \sum_{i=1}^{n} ||z_i||^2$ can be computed distributively through an average consensus [24] right after the moment in which each agent is able to estimate $z_i$ (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 $\delta_{ij}$ in (7) (block 8 in the diagram of Fig. 2). A summary follows.

**Result 5.** Each agent distributively computes the constant sum $k_i \sum_{i=1}^{n} ||z_i||^2$ using $z_i$ from Result 3 followed by average consensus. Then, each agent $i$ applies a force $f_i = k_i z_i^+$ 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 $z_C$

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

- 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 $z_C(t)$ boils down to a solvable nonlinear
observation problem. Let us first decompose the local force \( f_i(t) \) in two parts and recall two important identities

\[
f_i(t) = \frac{1}{n} \sum_{i=1}^{n} f_i(t) + \Delta f_i(t) = f_{\text{mean}}(t) + \Delta f_i(t),
\]

where \( \Delta f_i(t) = \sum_{i=1}^{n} z_i^T \Delta f_i = 0 \) and \( \sum_{i=1}^{n} \Delta f_i = 0 \).

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

\[
\omega = J^{-1} \left( \sum_{i=1}^{n} z_i^T \right) f_{\text{mean}} + nJ^{-1} z_C^T f_{\text{mean}} + J^{-1} \sum_{i=1}^{n} z_i^T \Delta f_i = J^{-1} \sum_{i=1}^{n} z_i^T \Delta f_i + J^{-1} \sum_{i=1}^{n} \tau_i, \quad \text{i.e.,}
\]

\[
\omega = z_C^T \tilde{f} + \eta_1 + \eta_2.
\]

The global quantities

1. \( \tilde{f} = \frac{1}{n} f_{\text{mean}} \),
2. \( \eta_1 = J^{-1} \sum_{i=1}^{n} z_i^T \Delta f_i = J^{-1} \sum_{i=1}^{n} z_i^T f_i \),
3. \( \eta_2 = J^{-1} \sum_{i=1}^{n} \tau_i \),

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 \( z_C(t) \).

Define \( \eta = \eta_1 + \eta_2 \). The following result holds:

**Result 6.** The rotational dynamics is given by

\[
\dot{\omega} = z_C^T \dot{\tilde{f}} + \eta,
\]

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

We use Eq. (15) to form a dynamical system where \( \omega \) and \( z_C(t) \) are the state variables and \( \tilde{f} \) and \( \eta \) are the inputs. Recalling that \( z_C \) is a constant-norm vector, rigidly attached to the object, the following holds:

\[
z_C = \omega z_C^T.
\]

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

\[
\begin{align*}
\dot{x}_1 &= -x_2 x_3 + u_2 (y - \hat{x}_3) \\
\dot{x}_2 &= x_1 x_3 - u_1 (y - \hat{x}_3) \\
\dot{x}_3 &= x_1 u_2 - x_2 u_1 + u_3
\end{align*}
\]

where \( z_C = (x_1, x_2, y - \hat{x}_3)^T \) 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 \( \tilde{f} = (\tilde{f}_x \tilde{f}_y)^T = (u_1 u_2)^T \), \( \eta = u_3 \) are known inputs.

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

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

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

**Proof.** Given in [26].

Note that the applied torques \( \tau_i \), for \( i = 1 \ldots n \) (which are included in \( u_3 \)) have no influence on the observability of \( z_C(t) \). 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

\[
\begin{align*}
\dot{x}_1 &= -x_2 x_3 + u_2 (y - \hat{x}_3) \\
\dot{x}_2 &= x_1 x_3 - u_1 (y - \hat{x}_3) \\
\dot{x}_3 &= x_1 u_2 - x_2 u_1 + k_0 (y - \hat{x}_3) + u_3
\end{align*}
\]

where \( k_0 > 0 \). If \( y(t) \neq 0 \) and \( (u_1 u_2)^T \neq 0 \), then (18) is an asymptotic observer for (17). Hence, defining \( \hat{x} = (\hat{x}_1 \hat{x}_2 \hat{x}_3)^T \) and \( x = (x_1 x_2 x_3)^T \), one has that \( \hat{x}(t) \rightarrow x(t) \) asymptotically.

**Proof.** Given in [26].

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., \( z_C(t) \), is distributively computed by using the observer (18) and thanks to the local knowledge of \( n, J, \omega, f_{\text{mean}}, \) and \( \sum_{i=1}^{n} z_i^T \Delta f_i \) from the previous results.

The estimation of \( z_C(t) \) 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 \( \bar{u}_i = u_i + \varepsilon_i \), with \( i = 1, 2, 3 \) and analyze the dynamics of \( e = x - \hat{x} \)

\[
\begin{align*}
\dot{e}_1 &= -e_2 x_3 + \bar{u}_2 e_3 \\
\dot{e}_2 &= e_1 x_3 - \bar{u}_1 e_3 \\
\dot{e}_3 &= e_1 \bar{u}_2 - e_2 \bar{u}_1 - k_0 e_3 - \varepsilon_3
\end{align*}
\]

1. The same considerations made in Remark IV.1 are in order.
Defining the class $K$ function $\gamma(r) = \frac{\theta r}{e^{(\theta-1)r} - 1}$, 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 $\gamma(r) = \frac{r}{e^{(\theta-1)r} - 1}$, 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

**C. Estimation of $v_C$**

The velocity of the center of mass $v_C(t)$ is estimated locally by each agent $i$ using the rigid body constraint $\frac{d}{dt}(p_C - p_{C_i}) = \omega(p_C - p_{C_i})$, which can be rewritten as

$$v_C(t) = v_{C_i}(t) - \omega(t)(z_{C_i}(t) + z_i(t))^\perp,$$

(20)

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

- $v_{C_i}(t)$ is locally measured by agent $i$
- $\omega(t)$, $z_{C_i}(t)$, and $z_i(t)$ are known by each agent $i$ thanks to Results 4, 8, and 3, respectively.

**Result 9.** The CoM velocity $v_C(t)$ is distributively computed using (20) and the knowledge of $v_{C_i}(t)$, $\omega(t)$, $z_{C_i}(t)$, $z_i(t)$.

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 $v_{C_i}(t)$ and average force. In fact, rewriting (2) as $v_C = \frac{1}{m} \sum_{i=1}^n f_i = \frac{1}{m} f_{\text{mean}}$, we obtain

$$\dot{\hat{v}}_C = m^{-1} \hat{f}_{\text{mean}},$$

(21)

where i) $n$ is known, ii) $f_{\text{mean}}$ is distributively estimated from $f_i$ using dynamic consensus (as in Result 6), and iii) $v_C$ 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 $\hat{d}_I$ in (7) and $J$ in (11).

**Result 10.** The mass $m$ is distributively computed from the knowledge of $v$ and $n$, and $f_{\text{mean}}$ 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, vice versa, 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 specific force $f_i = k_e z_i^\perp$. 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

$$\dot{\omega} = J^{-1} u_C^\perp \hat{f} + J^{-1} \hat{\eta},$$

(22)

where $\omega(t)$ is known thanks to Result 4, and $\hat{f} = nf_{\text{mean}}$ and $\hat{\eta} = J\eta_1 + J\eta_2 = \sum_{i=1}^n z_i^\perp f_i + \sum_{i=1}^n \tau_i$ 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)

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

(23)

where $(x_1, x_2)^\perp = z_C = (z_C^T, z^T_{\eta})$ 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 $(u_1, u_2)^\perp = \hat{f}, u_3 = \hat{\eta}$ are known inputs.

**Proposition 4.** Consider the following dynamical system

$$\begin{cases}
\dot{x}_1 = -x_2 x_3 + u_2 (y - \hat{x}_3) \\
\dot{x}_2 = \hat{x}_1 x_3 - u_1 (y - \hat{x}_3) \\
\dot{x}_3 = \hat{x}_1 u_2 - \hat{x}_2 u_1 + \hat{x}_4 u_3 + k_e (y - \hat{x}_3) \\
\dot{x}_4 = -u_3 (y - \hat{x}_3),
\end{cases}$$

(24)
where $k_e > 0$. If $y(t) \neq 0$, $u_3(t) \neq 0$ and $(v_1(t), v_2(t))^\top \neq 0$, then (24) is an asymptotic observer of a modified version of (23), in which the following change of variable is performed $x_1 \to x_1 x_4$, $x_2 \to x_2 x_4$. Thus, defining $\mathbf{x} = (x_1 \ x_2 \ x_3 \ x_4)^\top$ and $\mathbf{z} = (x_1 x_4 \ x_2 x_4 \ x_3 \ x_4)^\top$, one has that $\dot{\mathbf{x}}(t) \to \mathbf{x}(t)$ asymptotically, which in turn implies that $(x_1 \ x_2 \ x_3 \ x_4) \to (x_1 \ x_2)$ asymptotically.

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

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 $\omega$.

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

$$
\begin{align*}
\dot{z}_1 &= z_2 u, \\
\dot{z}_2 &= 0,
\end{align*}
$$

(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 $J = 0$.

**Proposition 5.** Consider the following dynamical system

$$
\begin{align*}
\dot{z}_1 &= \dot{z}_2 u + k_1 (y - z_1) \\
\dot{z}_2 &= k_2(y - \dot{z}_1),
\end{align*}
$$

(26)

where $k_1, k_2 \in \mathbb{R}$. If $k_1 > 0$, $k_2 u > 0$, and $k_1^2 > 4 k_2 u$ 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

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 $\omega$ and non-zero average force $f_{\text{mean}}$. 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: $\mathbf{f}_i = \mathbf{f}^* = \text{const}$, $\tau_i = 0$, $\forall i = 1 \ldots n$, and denote with $\omega_0$ the rotational rate at $t = 0$, then

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

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

(27)

2. $\exists \bar{\xi} > 0$ such that $\omega \equiv 0 \ \forall t \geq \bar{\xi}$, if and only if the following condition hold

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

(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

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: $\mathbf{f}_i = -b \mathbf{v}_C$, $\tau_i = 0$, $\forall i = 1 \ldots n$, with $b > 0$. Then, both $\omega$ 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

**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 $m = 50$ kg and $J = 86.89$ 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
$\sigma = 0.3 \text{ m/s}$. 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, because the measurements can make the signal-to-noise ratio too small to reach the accuracy of the system. In this case, we use the technique presented in [25] to assess 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 $\hat{z}_C$ 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
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 $t_1 \ldots t_6$, 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 $(n, \sigma)$ where $n$ indicates the number of agents in the network and $\sigma$ the standard deviation defining the covariance of a zero-mean Gaussian noise, added to velocity measurements, as $\Sigma_i = \sigma^2 I_{2 \times 2}$, for each agent $i = 1, \ldots, n$.

The two parameters assume values over a 2D grid formed with the values $n \in \{5, 8, 11, 14, 17, 20\}$ and $\sigma \in \{0.1, 0.2, 0.3, \ldots, 1\}$ 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 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 $\sigma$ for the EERD, the EEC, the estimation of $\omega$, the estimation of $v_C$, 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 $\sigma$, 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 $\sigma$, 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 $v_C$, used in turn to estimate $m$. The degree of dispersion of the estimation of $\omega$ 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 $v_C$ and $m$, confirming that the estimation of $v_C$ 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-plot 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. 5: Simulation with the same setup as in Fig. 4, but with a fully connected topology and noise $\sigma = 0.3$ m/s.

Fig. 6: Simulation with the same setup as in Fig. 4, but with a fully connected topology and noise $\sigma = 0.0001$ m/s.
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 m$^2$, 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 $\Sigma_i = \sigma^2 I_{2 \times 2}$ 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 to a bounded region around the new true values.

Fig. 7: Box-plots obtained by running the whole algorithm in different operational conditions, selected by setting the pair $(n, \sigma)$. 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 $(n, \sigma)$. The left and right column show the RE's before and after running the average consensus, respectively.

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

REFERENCES

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 (STIMA) 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 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.
Proofs of Propositions

Technical report associated with the paper:
“Distributed Estimation of State and Parameters in Multi-Agent Cooperative Load Manipulation”
IEEE Transactions on Control of Network Systems

Antonio Franchi, Antonio Petitti, Alessandro Rizzo

Abstract—This document is a technical attachment to [1]. Here, we present the proofs of the Propositions.

I. HOW TO CITE THIS WORK

This technical report is accompanying the IEEE Transactions on Control of Network Systems paper [1].

@Article
(author = \{A. Franchi and A. Petitti and A. Rizzo\},
title = \{Distributed Estimation of State and Parameters in Multi-Agent Cooperative Load Manipulation\},
journal = \{IEEE Transactions on Control of Network Systems\},
year = \{2021\},
doi = \{TBD\},
)

Proof of Proposition 3 in [1]. In the ideal case with $e_{1,2,3} = 0$, the origin of the system in (19) is asymptotically stable (Theorem V.1, [2]). Consider the following Lyapunov function

$$V(e) = \frac{1}{2} e^T e,$$

then

$$V = -e_1 e_2 x_3 + e_1 u_2 e_3 + e_2 e_1 x_3 - e_2 u_1 e_3 + \ldots$$

$$- e_3 u_2 - e_3 u_1 - k_e e_3^2 - e_3 e_3$$

$$= -k_e e_3^2 + e_3 e_3 \leq -k_e e_3^2 + |e_3| |e_3|.$$ (TR.1)

We observe that

$$-k_e e_3^2 + |e_3| |e_3| =$$

$$= -k_e e_3^2 + |e_3| |e_3| + k_e e_3^2 - k_e e_3^2 - k_e e_3^2 - k_e e_3^2.$$ (TR.2)

Considering that $|e| \geq |e_3|$ and $|e| \geq |e_3|$, we have that

$$V \leq -k_e |e|^2 + |e_3| |e_3| + k_e e_3^2 + k_e e_3^2$$

$$\leq -k_e |e|^2 + |e| |e| + k_e e_3^2 + k_e e_3^2.$$ (TR.3)

Moreover, considering that $k_e |e|^2 \geq k_e e_3^2 + k_e e_3^2$, we can write

$$V \leq -k_e |e|^2 + |e| |e| + k_e e_3^2 + k_e e_3^2$$

$$\leq -k_e |e|^2 + |e| |e| + k_e e_3^2.$$ (TR.4)

Thus, we rewrite the foregoing inequality as

$$\dot{V} \leq -k_e (1 - \theta) |e|^2 - k_e \theta |e|^2 + |e| |e| + k_e |e|^2,$$ (TR.5)

where $0 < \theta < 1$. The inequality $-k_e \theta |e|^2 + |e| |e| + k_e |e|^2 \leq 0$ holds if $|e| \geq \frac{|e|}{k_e \theta}$. Thus,

$$\dot{V} \leq -k_e (1 - \theta) |e|^2, \forall |e| \geq \frac{|e|}{k_e \theta}.$$ (TR.6)

Hence, the system is ISS (Theorem 4.9, [3]).

Proof of Proposition 4 in [1]. Define the error vector as

$$e = [e_1 e_2 e_3]^T = [(x_1 - x_3) (x_2 - x_3) (x_3 - x_3) (x_3 - x_3)].$$

After some algebra, the error dynamics is given by

$$\dot{e} = \begin{bmatrix} 0 & -x_3 & -u_2 & 0 \\ x_3 & 0 & u_1 & 0 \\ u_2 & -u_1 & -k_e & -u_3 \\ 0 & 0 & u_3 & 0 \end{bmatrix} e = [U + \text{diag}(0, 0, -k_e, 0)] e,$$ (TR.7)

where $U$ is skew symmetric, i.e., $U + U^T = 0$. Define the following candidate Lyapunov function: $V(e) = \frac{1}{2} e^T e$, whose time derivative along the system trajectories is

$$\dot{V} = e^T \dot{e} = e^T (U e - k_e e_3^2) = -k_e e_3^2,$$ (TR.8)

which is negative semidefinite. Now in order to study the invariant set that ensures that $V = 0$ we impose that $e_3 \equiv 0$, which implies, in particular, that $x_3 = 0$, $\dot{x}_3 = 0$, and $\ddot{x}_3 = 0$. Considering, for simplicity, the case in which inputs are stepwise constant, the last three equations (24)–(TR.8) result in the following system of linear equations:

$$\begin{bmatrix} u_2 & -u_1 & u_3 \\ -x_3 u_1 & -x_3 u_2 & 0 \\ -x_3^2 u_2 & -x_3^2 u_3 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = E \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$ (TR.9)

The determinant of $E$ is $-u_3 x_3^2 (u_1^2 + u_2^2)$. If the assumptions of the theorem hold, then $E$ is nonsingular and therefore the only trajectory of the system that ensures $V = 0$ is $e = 0$. □

Proof of Proposition 5 in [1]. Define the error vector

$$e = (e_1 e_2)^T = (x_3 - \dot{x}_3 z_2 - \dot{z}_3)^T.$$ The error dynamics is given by

$$\dot{e} = \begin{bmatrix} 0 \\ -k_1 u \\ -k_2 \end{bmatrix} e = A e.$$. (TR.10)
Consider the Lyapunov candidate \( V = \frac{1}{2} \mathbf{e}^\top \mathbf{e} \), its time derivative is \( \dot{V} = \mathbf{e}^\top \mathbf{A} \mathbf{e} \). If the eigenvalues of \( \mathbf{A} \) are real and non-greater than \( -\varepsilon < 0 \), then asymptotic stability is guaranteed. After some algebra, the eigenvalues of \( \mathbf{A} \) are

\[
\lambda_{1,2} = \frac{1}{2} \left( -k_1 \pm \sqrt{k_1^2 - 4k_2} \right).
\]

Therefore, \( \lambda_{1,2} \) are real if \( k_1^2 > 4k_2 \). Moreover, if \( k_1 > \varepsilon_1 > 0 \) and \( k_2 > \varepsilon_2 > 0 \), then \( \exists \varepsilon > 0 \) s.t. \( \lambda_{1,2} < -\varepsilon < 0 \). \( \square \)

Proof of Proposition 6 in [1]. In order to prove (27), let us consider the quantity \( \alpha = \omega^2 - 2nJ^{-1} \mathbf{z}_C \mathbf{f}^\top \). Let us now take the derivative of \( \alpha \) w.r.t. time. Using (15) and (16), we obtain \( \alpha = 0 \), i.e., \( \alpha \) is an invariant along the system trajectories when \( f_i = f^\top = \text{const} \), \( \forall i = 1 \ldots n \). In particular, \( \alpha(t) = \alpha(0) \), which implies

\[
\dot{\alpha}(t) = 0 \Rightarrow \alpha(t) = \alpha(0) = 0.
\]

Proof of Proposition 7 in [1]. From (20) and using the identities on the left-hand side in (13), it is straightforward to derive the following two identities:

\[
\sum_{i=1}^{n} \mathbf{v}_i = n \mathbf{v}_C + n \omega \mathbf{z}_C^\perp, \quad \mathbf{v}_C = \mathbf{v}_C + \omega (\mathbf{z}_C + \mathbf{z}_i)^\perp,
\]

which can be used to obtain, respectively,

\[
\eta \mathbf{f}_{\text{mean}} = -b \sum_{i=1}^{n} \mathbf{v}_i = -bn \mathbf{v}_C - nb \omega \mathbf{z}_C^\perp \quad \text{(TR.16)}
\]

and

\[
\eta = -b \sum_{i=1}^{n} \mathbf{z}_i \mathbf{v}_i = -b \left( \mathbf{v}_C + \omega \mathbf{z}_C^\perp \right) \sum_{i=1}^{n} \mathbf{z}_i^\perp + \omega \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^\top \mathbf{z}_i^\perp = -b \omega \sum_{i=1}^{n} \| \mathbf{z}_i \|^2 \quad \text{(TR.17)}
\]

Plugging (TR.16) and (TR.18) in (15) and (21) we obtain

\[
J \dot{\omega} = -bn \left( \mathbf{z}_C^\top \mathbf{v}_C + \omega \mathbf{z}_C^\top \mathbf{z}_C \right) - b \omega \sum_{i=1}^{n} \| \mathbf{z}_i \|^2
\]

\[
m \dot{\mathbf{v}}_C = -bn \left( \mathbf{v}_C + \omega \mathbf{z}_C^\top \right).
\]

Let us consider \( V = J \omega^2 + m \| \mathbf{v}_C \|^2 \) as a Lyapunov candidate function. We obtain

\[
\dot{V} = -bn \left( \mathbf{v}_C \mathbf{v}_C^\top + 2 \omega \mathbf{z}_C \mathbf{z}_C^\top + \omega \mathbf{z}_C \mathbf{z}_C^\top + \omega \mathbf{z}_C \mathbf{z}_C^\top \right) - b \omega^2 \sum_{i=1}^{n} \| \mathbf{z}_i \|^2 =
\]

\[
- \left( n \| \mathbf{v}_C + \omega \mathbf{z}_C^\top \|^2 + \omega^2 \sum_{i=1}^{n} \| \mathbf{z}_i \|^2 \right) < 0 \quad \forall [\mathbf{v}_C \omega] \neq 0^\top,
\]

which proves the thesis of the proposition. \( \square \)

References

