



**HAL**  
open science

# Distributed estimation of the inertial parameters of an unknown load via multi-robot manipulation

Antonio Franchi, Antonio Petitti, Alessandro Rizzo

► **To cite this version:**

Antonio Franchi, Antonio Petitti, Alessandro Rizzo. Distributed estimation of the inertial parameters of an unknown load via multi-robot manipulation. 2014 53rd IEEE Conference on Decision and Control, Dec 2014, Los Angeles, United States. 10.1109/CDC.2014.7040346 . hal-01137826

**HAL Id: hal-01137826**

**<https://hal.science/hal-01137826>**

Submitted on 31 Mar 2015

**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 the Inertial Parameters of an Unknown Load via Multi-Robot Manipulation

Antonio Franchi, Antonio Petitti, and Alessandro Rizzo

**Abstract**—In this paper, we propose a distributed strategy for the estimation of the kinematic and inertial parameters of an unknown body manipulated by a team of mobile robots. We assume that each robot can measure its own velocity, as well as the contact forces exerted during the body manipulation, but neither the accelerations nor the positions of the contact points are directly accessible. Through kinematics and dynamics arguments, the relative positions of the contact points are estimated in a distributed fashion, and an observability condition is defined. Then, the inertial parameters (*i.e.*, mass, relative position of the center of mass and moment of inertia) are estimated using distributed estimation filters and a nonlinear observer in cooperation with suitable control actions that ensure the observability of the parameters. Finally, we provide numerical simulations that corroborate our theoretical analysis.

## I. INTRODUCTION

Last decades have seen raising attention to cooperative manipulation by teams of robotic agents, due to its relevant applications in several fields, such as search and rescue and disaster recovering, cooperative transportation, and service robotics. In [1], for example, a path planner for cooperative manipulation is proposed. It consists of two components; a global path planner, responsible of obstacle avoidance, and a local manipulation planner, responsible of the object's manipulation via the control of its position. In [2], the control of robotic teams with the aim of combining optimal goal regulation and relaxation of the formation rigidity constraint is carried out. Cooperative manipulation has also been dealt with in aerial applications [3], [4], using cables, or other interaction tools. However, the majority of the works does not focus on an accurate dynamic modeling of the payload, which is either modeled as a point mass or as a rigid body with unknown dynamical parameters. In [5], the problem of cooperatively manipulating an object on a plane by a team of non-holonomic wheeled robots is dealt with. The problem is solved by defining the dynamics of the whole system (*i.e.*, the robot team and the object) and through a decomposition technique into a task and a null space. The dynamic model of the object is accounted for, but its inertial parameters are assumed to be known, as well as the positions of the contact points between the robots and the object.

An accurate and on-line estimation of the inertial parameters of the payload is very useful for at least two

reasons: first, it allows to manipulate unknown objects with effective techniques once the inertial parameters are estimated, *e.g.*, by exploiting force control and pose estimation algorithms [6], [7]. Second, it is very important when time-varying inertial parameters are present, such as, *e.g.*, when part of the load on a cart is dropped or an additional load is added by an external agent. In the case of time-varying parameters, their on-line estimation will allow to implement effective adaptive control techniques, as well as to carry out fault detection (*e.g.*, raise an alarm when part of the load is dropped during manipulation), or event-based control (*e.g.*, change the controller's behavior when the load changes one or more of its characteristics). Moreover, since cooperative tasks exhibit great advantages when carried out via decentralized strategies [8], decentralized inertial parameter's estimation algorithms are highly desirable, to maintain the features of robustness, flexibility, and fault tolerance of the cooperative manipulation system as a whole.

Some effort has been made in the estimation of inertial parameters. In [6], an algorithm to estimate the inertial parameters of an unknown planar object is proposed. The object is pushed by robot fingers and the estimate is carried out on the basis of the fingertip velocities and accelerations, and of the exchanged forces. In [7], an approach to estimate the inertial parameters of rigid loads attached to a manipulator, is proposed. The proposed algorithm estimates the inertial parameters on-line by a total least-squares approach. The cited papers rely on acceleration measurements and are based on centralized approaches.

In this paper, we consider a team of Autonomous Ground Vehicles (AGVs) that manipulate an unknown rigid body by applying forces to contact points whose relative (and, therefore, also absolute) position is not known a priori. We propose a distributed algorithm for the estimation of the relative positions of the contact points and of the inertial parameters of the rigid body based on velocity measurements only (neither absolute/relative positions nor the accelerations of the contact points are needed). This is the first step in the design and implementation of advanced cooperative control strategies, based on the knowledge of the position of the contact points and on the estimate of the dynamical model of the payload.

The paper is structured as follows. In Sec. II we formally define the model and the estimation problem. A general overview of the estimation algorithm is given in Sec. III. In Sec. IV, we deal with the distributed estimation of the relative positions of the contact points, which is instrumental to estimate the payload's inertial parameters. In Sec. V, we propose a distributed algorithm for the estimation of the

A. Franchi is with LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse CEDEX 4, France. antonio.franchi@laas.fr

A. Petitti is with ISSIA-CNR, via Giovanni Amendola, 122/D-I 70126 Bari, Italy, petitti@ba.issia.cnr.it

A. Rizzo is with the MAE Dept, NYU Polytechnic School of Engineering, Brooklyn NY, 11201, USA, alessandro.rizzo@nyu.edu

A. Petitti and A. Rizzo are also with DEI, Politecnico di Bari, 70126 Bari, Italy [antonio.petitti, alessandro.rizzo]@poliba.it.

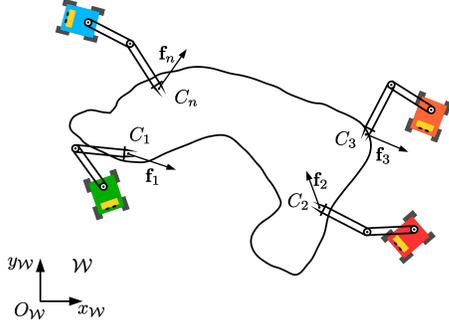


Fig. 1: Top view of the problem setting. A manipulation task has to be performed by a team of  $n = 4$  AGVs endowed with a planar manipulator each. The unknown inertial parameters of the load need to be estimated using local action, sensing and communication.

inertial parameters whose main part consists of the design of a nonlinear observer to track the relative position of the center of mass. Finally, in Sec. VI we validate the proposed approach by means of numerical simulations, and draw our conclusions in Sec. VII.

## II. PROBLEM STATEMENT

We consider a load modeled as a planar rigid body  $B$ , whose center of mass is denoted with  $C$ . The load is manipulated by a group of  $n$  AGVs, where each AGV can exert a force on a contact point  $C_i$ ,  $i = 1, \dots, n$ , of  $B$ . We assume that the number  $n$  is constant and known to all robots<sup>1</sup>. Consider a planar reference inertial frame  $\mathcal{W} = \{O_{\mathcal{W}} - x_{\mathcal{W}} y_{\mathcal{W}}\}$  and denote with  $\mathbf{p}_C \in \mathbb{R}^2$  the position of  $C$  in  $\mathcal{W}$ , with  $\mathbf{p}_{C_i}$  the position of  $C_i$  in  $\mathcal{W}$  and with  $\mathbf{f}_i$  the force applied by the  $i$ -th robot at  $C_i$  and expressed in  $\mathcal{W}$ . Figure 1 schematizes the problem setting. We also assume that friction on the load is negligible with respect to the total force exerted by the robots<sup>2</sup>.

The dynamical model of the load  $B$  is therefore that of a rigid body subjected to the  $n$  forces  $\mathbf{f}_1, \dots, \mathbf{f}_n$

$$\ddot{\mathbf{p}}_C = \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i \quad (1)$$

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_C)^\perp T \mathbf{f}_i, \quad (2)$$

where  $m > 0$  is the mass of  $B$ ,  $\omega \in \mathbb{R}$  is its angular velocity,  $J > 0$  is its moment of inertia, and the operator  $(\cdot)^\perp$  is the linear operator that, given a vector  $\mathbf{v} \in \mathbb{R}^2$ ,  $\mathbf{v} = (v^x \ v^y)^T$ , provides the perpendicular vector to  $\mathbf{v}$

$$\mathbf{v}^\perp = Q\mathbf{v} = \underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{=Q} \begin{pmatrix} v^x \\ v^y \end{pmatrix} = \begin{pmatrix} -v^y \\ v^x \end{pmatrix}. \quad (3)$$

From an actuation and sensing point of view, we assume that the  $i$ -th robot can control the exerted force  $\mathbf{f}_i$  and can measure the velocity of its contact point  $\dot{\mathbf{p}}_{C_i}$ . Note that we

<sup>1</sup>This assumption, that is the only source of centralized information needed, can be easily relaxed by implementing one of the several algorithms for the distributed estimation of a graph size before the algorithm starts [9].

<sup>2</sup>This can be ensured, *e.g.*, by endowing the load with wheels.

do not assume that the robot can measure the position of the contact point, nor its acceleration.<sup>3</sup> Furthermore, we do not assume kinematically controlled robots, *i.e.*, robots cannot set the speed of the contact point at will.

From a communication point of view, we assume that the robotic agents can transmit data through a network modeled by an undirected graph  $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ , where  $\mathcal{I} = \{1, \dots, n\}$ ,  $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$  is a set of unordered pairs of nodes representing the one-hop communication channels between agents. The set  $\mathcal{N}_i = \{j \in \mathcal{I} : (i, j) \in \mathcal{E}\}$  indicates the communication neighborhood of agent  $i$ . We also assume that the communication graph is connected, and that the link set  $\mathcal{E}$  does not change in time.

**Problem II.1** (Inertial Parameters' Distributed Estimation). *Define a distributed algorithm such that each robot  $i$  is able to compute an estimate of*

- 1) the constant parameters  $m$  and  $J$ , and
- 2) the time-varying vector  $\mathbf{p}_{C_i} - \mathbf{p}_C \in \mathbb{R}^2$ ,

*only controlling the local force  $\mathbf{f}_i$  (to be computed by the algorithm), and only resorting to local measurement  $\dot{\mathbf{p}}_{C_i}$  and to local information coming from the communication neighbors.*

In this work, we are interested in the fundamental problem of letting a multi-robot system estimate the inertial parameters in a distributed way. Thus, we consider a deterministic setting, *i.e.*, the noise affecting measurements, control inputs, and communication inputs is negligible. The noisy case will be the subject of a future work.

## III. ESTIMATION ALGORITHM

Define  $\mathbf{p}_G = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_{C_i}$  as the geometric center of all the contact points and define also  $\mathbf{z}_{ij} = \mathbf{p}_{C_i} - \mathbf{p}_{C_j}$ ,  $\mathbf{z}_i = \mathbf{p}_{C_i} - \mathbf{p}_G$ , and  $\mathbf{z}_C = \mathbf{p}_G - \mathbf{p}_C$ . Note that  $\mathbf{p}_{C_i} - \mathbf{p}_C = \mathbf{z}_i + \mathbf{z}_C$ . We first recall a simple fact.

**Fact III.1.** *Denote with  $\mathbf{z}_1$  and  $\mathbf{z}_2$  the relative positions between two pairs of points of  $B$  expressed in  $\mathcal{W}$ . Then, consider two time instants  $t'$  and  $t''$ . The following relation, based on the rigid-body constraint, provides a straightforward way to compute  $\mathbf{z}_2(t'')$  from  $\mathbf{z}_1(t')$ ,  $\mathbf{z}_2(t')$ , and  $\mathbf{z}_1(t'')$ :*

$$\begin{aligned} \mathbf{z}_2(t'') &= \Gamma(\mathbf{z}_1(t'), \mathbf{z}_2(t'), \mathbf{z}_1(t'')) = \\ &= \frac{(\mathbf{z}_2(t')^T \mathbf{z}_1(t')) \mathbf{z}_1(t'') + (\mathbf{z}_2(t')^T \mathbf{z}_1^\perp(t')) \mathbf{z}_1^\perp(t'')}{\|\mathbf{z}_1(t')\|}. \end{aligned} \quad (4)$$

The proposed distributed estimation algorithm follows a multi-step approach whose main parts are described in the next sections. Before going into the details, in order to facilitate the comprehension of the whole algorithm, we sketch all the steps in a chronological order (refer to Fig. 2 for a block diagram representation of the interconnections among the several steps).

The algorithm starts at time  $t = t_0$  and passes through four key time instants  $t_0 \leq t_1 \leq t_2 \leq t_3 \leq t_4$  that define

<sup>3</sup>Measuring the absolute position would require an additional sensing system, *e.g.*, a GPS. Measuring the acceleration, on the other hand, is typically noisy. Measuring the instantaneous velocity can be instead reliably done on board by odometry, optical flow, etc...

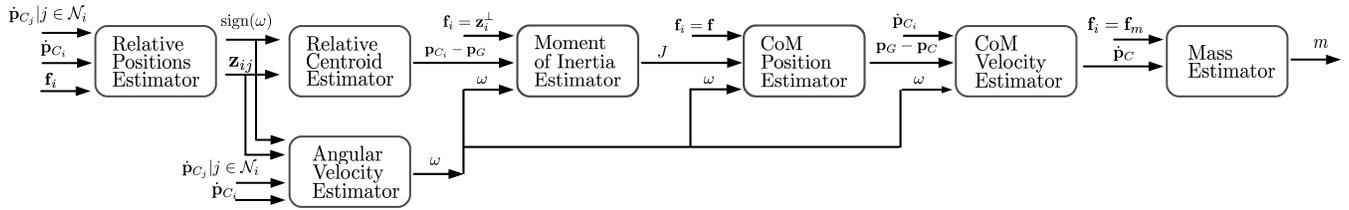


Fig. 2: A block diagram showing the sequence of estimators applied in order to solve Problem II.1.

when some crucial quantities become available due to the convergence time of some of the estimation steps. After  $t_4$ , the algorithm has identified the two constant parameters  $m$  and  $J$  and can observe the time-varying parameter  $\mathbf{p}_{C_i} - \mathbf{p}_C$ .

**Step 1:**  $\mathbf{z}_{ij}(t)$  becomes available after  $t_1$ . Robot  $i$  employs the velocity measurement  $\dot{\mathbf{p}}_{C_i}$  and the velocities of its neighbors  $\dot{\mathbf{p}}_{C_j}$ , with  $j \in \mathcal{N}_i$ , in order to obtain, in a distributed fashion, an estimate of the relative positions  $\mathbf{z}_{ij}(t)$ ,  $j \in \mathcal{N}_i$  and  $\text{sign}(\omega)$ . This estimation process needs a short convergence time to retrieve the inter-distance between contact points. Therefore,  $\mathbf{z}_{ij}(t)$  will be available for any time  $t \geq t_1$ , where  $t_1 \geq t_0$ . This step is detailed in Sec. IV;

**Step 2:**  $\mathbf{z}_i(t)$  becomes available after  $t_1$ . For any time  $t \geq t_1$ , robot  $i$  uses the relative position measurements  $\mathbf{z}_{ij}(t)$  (available thanks to Step 1) to compute, using the algorithm in [10], the vector  $\mathbf{z}_i(t)$ ;

**Step 3:**  $\omega(t)$  becomes available after  $t_1$ . For any time  $t \geq t_1$ , robot  $i$  uses the following formula to compute, locally, the angular velocity of  $B$ :

$$\omega(t) = \text{sign}(\omega) \left\| \frac{\dot{\mathbf{z}}_{ij}^T \mathbf{z}_{ij}^\perp(t)}{\|\mathbf{z}_{ij}\|^2} \right\|, \quad (5)$$

where  $j$  can be any possible neighbor in  $\mathcal{N}_i$ ;

**Step 4:**  $J$  becomes available after  $t_2$ . For all  $t \in (t_1, t_2)$ , each robot  $i$  applies a force  $\mathbf{f}_i(t) = \mathbf{z}_i^\perp(t)$  (available thanks to Step 2) and observes  $\omega(t)$  (available thanks to (5)). This allows robot  $i$  to reach, at  $t = t_2$ , an estimate of the constant parameter  $J$ . Therefore, for any  $t \geq t_2$ ,  $J$  is known by all the robots. This step is detailed in Sec. V-A;

**Step 5:**  $\mathbf{p}_{C_i} - \mathbf{p}_C$  becomes available after  $t_3$ . For any  $t \in (t_2, t_3)$ , each robot  $i$  applies the same constant nonzero force  $\mathbf{f}_i(t) = \mathbf{f}$  and measures  $\omega(t)$ . Using the nonlinear observer detailed in Sec. V-B together with the estimated value of  $J$  (known thanks to Step 4), each robot is able to obtain an estimate that eventually converges to the time-varying vector  $\mathbf{z}_C$ . For any  $t \geq t_3$ , each robot is then able to compute the sought vector  $\mathbf{p}_{C_i} - \mathbf{p}_C$  using

$$\begin{aligned} \mathbf{p}_{C_i}(t) - \mathbf{p}_C(t) &= \mathbf{z}_i(t) + \mathbf{z}_C(t) = \\ &= \mathbf{z}_i(t) + \Gamma(\mathbf{z}_C(t_3), \mathbf{z}_{ij}(t_3), \mathbf{z}_{ij}(t)), \end{aligned} \quad (6)$$

where  $\Gamma$  is defined in (4) and  $j$  is any neighbor in  $\mathcal{N}_i$ ;

**Step 6:**  $\dot{\mathbf{p}}_C(t)$  becomes available after  $t_3$ . For any  $t \geq t_3$ , the velocity of the center of mass is computed locally by any robot  $i$  using

$$\dot{\mathbf{p}}_C(t) = \dot{\mathbf{p}}_{C_i}(t) - \omega(t)(\mathbf{p}_{C_i}(t) - \mathbf{p}_C(t))^\perp, \quad (7)$$

where all the quantities in the right hand side of (7) are known at any  $t \geq t_3$  thanks to all the previous steps;

**Step 7:**  $m$  becomes available after  $t_4$ . For all  $t \in$

$(t_3, t_4)$ , each robot  $i$  applies the same constant nonzero force  $\mathbf{f}_i(t) = \mathbf{f}_m$  and measures  $\dot{\mathbf{p}}_C(t)$  thanks to (7). This allows robot  $i$  to reach, at  $t = t_4$ , an estimate of the constant parameter  $m$ , as detailed in Sec. V-C. Thus, at any  $t \geq t_4$  the mass  $m$  is known by all the robots.

For  $t \geq t_4$ , the constant parameters  $m$  and  $J$  are known by every robot (Steps 7 and 4). Furthermore, each robot  $i$  can instantaneously compute  $\mathbf{p}_{C_i}(t) - \mathbf{p}_C(t)$  (Step 5). Therefore, Problem II.1 is solved.

In the following sections we detail the algorithms used in Steps 1, 4, 5, and 7.

#### IV. DISTRIBUTED ESTIMATION OF RELATIVE POSITIONS OF THE CONTACT POINTS

In this section, we propose an algorithm to compute the relative positions  $\mathbf{z}_{ij}$ , for each  $j \in \mathcal{N}_i$ , only resorting to local sensing and 1-hop communication.

The fact that  $C_i$  and  $C_j$  belong to the same rigid body constrains their inter-distance to be constant over time, i.e.,

$$\dot{\mathbf{z}}_{ij}^T \mathbf{z}_{ij} = \text{const}. \quad (8)$$

Taking the time derivative of both sides of (8), we obtain a constraint on the difference between the velocities of  $C_i$  and  $C_j$  which, in fact, has to be perpendicular to  $\mathbf{z}_{ij}$ , i.e.,

$$\dot{\mathbf{z}}_{ij}^T \mathbf{z}_{ij} = 0. \quad (9)$$

Noting that  $\|\dot{\mathbf{z}}_{ij}\| > 0$  and  $\|\mathbf{z}_{ij}\| > 0$ , constraint (9) can be rewritten as

$$\frac{\mathbf{z}_{ij}}{\|\mathbf{z}_{ij}\|} = \text{sign}(\omega) \frac{\dot{\mathbf{z}}_{ij}^\perp}{\|\dot{\mathbf{z}}_{ij}^\perp\|} \quad \text{i.e.,} \quad \mathbf{z}_{ij} = \text{sign}(\omega) \|\mathbf{z}_{ij}\| \mathbf{y}_{ij}, \quad (10)$$

where we compactly recast  $\frac{\dot{\mathbf{z}}_{ij}^\perp}{\|\dot{\mathbf{z}}_{ij}^\perp\|}$  as  $\mathbf{y}_{ij}$  to emphasize that quantity  $\mathbf{y}_{ij}$  is available to robot  $i$  resorting to sensing and one-hop communication. Thus, the constant distance between the two points  $\|\mathbf{z}_{ij}\|$  represents the only unknown toward the computation of the time-varying vector  $\mathbf{z}_{ij}$ .

Differentiating both sides of (10), we obtain

$$\dot{\mathbf{z}}_{ij} = \text{sign}(\omega) \|\mathbf{z}_{ij}\| \dot{\mathbf{y}}_{ij}, \quad (11)$$

which cannot be directly used to compute  $\|\mathbf{z}_{ij}\|$ , since only  $\dot{\mathbf{z}}_{ij}$  and  $\mathbf{y}_{ij}$  are measured, but  $\dot{\mathbf{y}}_{ij}$  is not. Therefore, we apply the technique described in the Appendix using  $\dot{\mathbf{z}}_{ij}^f$  and  $\mathbf{y}_{ij}^f$  instead of  $\dot{\mathbf{z}}_{ij}$  and  $\mathbf{y}_{ij}$ , respectively. We observe that the sign of (11) depends on  $\text{sign}(\omega)$ . Therefore, in order to estimate a positive quantity, we use the squared norm of the filtered quantities. We can then summarize the previous derivations in the following result:

**Proposition IV.1** (Relative position estimation from velocity measures). *For each  $(i, j) \in \mathcal{E}$ , if  $\|\mathbf{z}_{ij}\| > 0$  and  $\|\dot{\mathbf{z}}_{ij}\| > 0$ , then  $\mathbf{z}_{ij}$  can be computed using only the velocities of  $C_i$  and  $C_j$  by means of the following estimator*

$$\mathbf{z}_{ij} = \sqrt{\frac{\|\dot{\mathbf{z}}_{ij}^f\|^2}{k_f^2 \|\mathbf{y}_{ij}^f - \mathbf{y}_{ij}^f\|^2}} \mathbf{y}_{ij} \quad (12)$$

where  $(\cdot)^f$  is a first-order low-pass filter and  $k_f$  is its gain.

Finally, we obtain  $\text{sign}(\omega)$  as  $\text{sign}\left[\left(\dot{\mathbf{z}}_{ij}^f\right)^T (\mathbf{y}_{ij}^f - \mathbf{y}_{ij}^f)\right]$ .

## V. ESTIMATION OF THE INERTIAL PARAMETERS

In this section we describe in detail the estimation algorithms for the constant quantities  $J$  and  $m$ , and a nonlinear observer for  $\mathbf{z}_C$ , which can be used in (6) in order to have an estimate of the sought parameter  $\mathbf{p}_{C_i} - \mathbf{p}_C$ .

Since  $\mathbf{p}_{C_i} - \mathbf{p}_C = \mathbf{z}_i + \mathbf{z}_C$ , we rewrite (2) as

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n \mathbf{z}_i^{\perp T} \mathbf{f}_i + \frac{1}{J} \mathbf{z}_C^{\perp T} \sum_{i=1}^n \mathbf{f}_i. \quad (13)$$

Each vector  $\mathbf{z}_i$  can be computed in a distributed way by the  $i$ -th robot resorting to the distributed algorithm presented in [10], which can be applied since the communication graph is connected by assumption.

### A. Estimation of the moment of inertia $J$

It is easy to verify that  $\sum_{i=1}^n \mathbf{z}_i^{\perp} = 0$ . Thus, if

$$\mathbf{f}_i = \mathbf{z}_i^{\perp}, \quad \forall i = 1 \dots n, \forall t \geq 0 \quad (14)$$

then (13) simplifies in

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n \mathbf{z}_i^{\perp T} \mathbf{z}_i^{\perp} = \frac{1}{J} \sum_{i=1}^n \|\mathbf{z}_i\|^2. \quad (15)$$

Thus,  $J$  can be computed by applying the following distributed algorithm:

- 1) Distributively compute the value  $w = \sum_{i=1}^n \|\mathbf{z}_i\|^2$  using an average consensus algorithm [11], *i.e.*, each robot sets a local state variable  $\xi_i|_{t=t_0} = \|\mathbf{z}_i\|^2$  and applies the local update rule

$$\dot{\xi}_i = \sum_{j \in \mathcal{N}_i} (\xi_j - \xi_i), \quad (16)$$

which, since the communication graph is connected by assumption, leads  $\xi_i$  to be asymptotically equal to  $w/n$ . Thus, after consensus is reached, each robot can compute  $w = n\xi_i$ ;

- 2) Each robot applies a constant force  $\mathbf{f}_i = \mathbf{z}_i^{\perp}$  ;
- 3) Locally estimate  $J$  by using the filtering approach described in the Appendix, substituting  $u$  with  $w$ ,  $y$  with  $\omega$ , which is known locally thanks to (5), and  $\theta$  with  $\frac{1}{J}$ .

**Remark V.1.** *No time synchronization is needed for the starting time of applications of forces in (14). The only requirement is that, eventually, all robots are applying the required forces.*

**Remark V.2.** *When applying the constant forces detailed in the previous algorithm, two cases can occur. The first case is that the body moves of pure rotation, therefore, the center of mass does not moves. This is, in any case, adequate to our purposes. The second case is that the body moves with constant angular acceleration. In this case, the forces can be safely applied only for a limited time, after which the movement of the body must be stopped, e.g., with a pure damping force based on velocity feedback. However, should the time be not enough for estimation purposes, the process can be repeated several times after each stop, to ensure the acquisition of the necessary measurements.*

### B. Observer for relative position of the Center-of-Mass $\mathbf{z}_C$

Denoting with  $\mathbf{f} \in \mathbb{R}^2$  any nonzero constant vector, if

$$\mathbf{f}_i = \mathbf{f}, \quad \forall i = 1, \dots, n, \forall t \geq 0, \quad (17)$$

then (13) becomes

$$\dot{\omega} = \frac{1}{J} \left( \sum_{i=1}^n \mathbf{z}_i^{\perp T} \right) \mathbf{f} + \frac{n}{J} \mathbf{z}_C^{\perp T} \mathbf{f} = \frac{n}{J} \mathbf{z}_C^{\perp T} \mathbf{f}, \quad (18)$$

where we rely, as before, on the fact that  $\sum_{i=1}^n \mathbf{z}_i^{\perp T} = 0$ . Letting  $\bar{\mathbf{f}} = (n/J)\mathbf{f}$  we have

$$\dot{\omega} = \mathbf{z}_C^{\perp T} \bar{\mathbf{f}}. \quad (19)$$

It is clear that, to keep the distributed character of the algorithm the value of  $\mathbf{f}$  must be agreed on in a distributed way. This can be easily achieved by means of standard consensus algorithms [12], [13]. Without loss of generality, up to a suitable coordinate change, we assume  $\bar{\mathbf{f}} = (0 \ f_y)^T$ . Considering that the vector  $\mathbf{z}_C$  has constant length and is rigidly attached to the object, we also have

$$\dot{\mathbf{z}}_C^{\perp} = -\mathbf{z}_C \omega. \quad (20)$$

We can write (19) and (20) as the following autonomous nonlinear system

$$\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_1 \bar{f}_y \\ y = x_3, \end{cases} \quad (21)$$

where we set  $x_C^x = x_1$ ,  $x_C^y = x_2$ ,  $\omega = x_3$ . We observe that the only available output is  $y = x_3$ , since the angular velocity is locally measurable by each robot using (5). Thus, the problem of estimating  $\mathbf{z}_C$  is tackled as the following nonlinear observability problem:

**Problem V.1.** *Distributively estimate  $\mathbf{z}_C$  is equivalent to observe the state of the autonomous nonlinear system (21), with output  $y = x_3 = \omega$ , where  $\bar{f}_y \neq 0$ , is a given constant value.*

In order to be able to design a suitable nonlinear observer we have first to study the observability of system (21).

**Proposition V.1.** *If  $x_3 \neq 0$  and  $\bar{f}_y \neq 0$ , then system (21) is locally observable in the sense of [14].*

*Proof.* The proof, omitted due to space constraints, follows the line of arguments in [14] and is based on the computation of the nonlinear observability matrix.  $\square$

Thus,  $\mathbf{z}_C$  is observable if and only if the angular velocity of the object and the constant force applied by each robot are both nonzero.

Since observability depends on the angular velocity  $\omega = x_3$ , it is important to analyze the trajectories of system (21). In particular, trajectories for which the angular velocity is constantly zero or asymptotically converges to zero would lead to an unobservable system. On the other hand, trajectories for which the angular velocity grows unbounded are practically unfeasible.<sup>4</sup>

The following result shows that, apart from a zero measure set, the trajectories of system (21) yield angular velocities that are suitable regarding both the observability and the practical feasibility issues.

**Proposition V.2.** *The following facts hold for system (21):*

- 1) *the origin is a stable equilibrium point;*
- 2) *the angular velocity  $x_3$  is bounded, and in particular:*

$$|x_3| \leq \sqrt{x_3^2(0) + 4\bar{f}_y \|\mathbf{z}_C\|} \quad (22)$$

- 3)  $\exists T \geq 0$  *such that  $x_3(t) = 0 \forall t \geq T$  if and only if the initial angular velocity  $x_3(0)$  is such that:*

$$x_3^2(0) = 2\bar{f}_y(x_2(0) \pm \|\mathbf{z}_C\|). \quad (23)$$

*Proof.* Omitted due to space constraints. It is based on Lyapunov arguments.  $\square$

Proposition V.2 guarantees that the angular velocity  $x_3$  does not vanish, and therefore the system remains observable for all time, except for at most four initial condition points, which constitute a zero measure set that has probability zero to happen in real-world robotics applications. Moreover, the Proposition shows that the angular velocity  $x_3$  and its derivatives remain bounded and gives an upper bound that can be decreased by acting on the input. This last feature is very important for the real applicability of the proposed method.

**Theorem V.1.** *The dynamical system*

$$\begin{aligned} \dot{\hat{x}}_1 &= -\hat{x}_2 x_3 + \bar{f}_y(x_3 - \hat{x}_3) \\ \dot{\hat{x}}_2 &= \hat{x}_1 x_3 \\ \dot{\hat{x}}_3 &= \hat{x}_1 \bar{f}_y + k_e(x_3 - \hat{x}_3), \end{aligned} \quad (24)$$

where  $k_e > 0$ , is an asymptotic observer for system (21), i.e., defining  $\hat{\mathbf{x}} = (\hat{x}_1 \hat{x}_2 \hat{x}_3)^T$  and  $\mathbf{x} = (x_1 x_2 x_3)^T$ ,  $\hat{\mathbf{x}} \rightarrow \mathbf{x}$  asymptotically, except for the zero measure initial condition set defined in (23).

*Proof.* Omitted due to space constraints. It is based on Lyapunov-LaSalle arguments.  $\square$

Note that observer (24) can be implemented in a distributed fashion by resorting only to local information.

<sup>4</sup>One cannot, in this case, stop the motion and restart it, as suggested in Remark V.2, since we are estimating a time-varying quantity and therefore we cannot restart the process each time with an improved initial estimate.

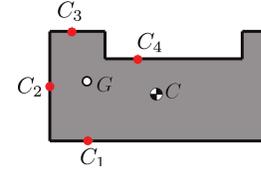


Fig. 3: The simulated payload is an eight-sided polygon, obtained from a rectangular plate of sides  $2\text{ m} \times 4\text{ m}$ , where a smaller rectangular portion of  $0.5\text{ m} \times 2.5\text{ m}$  sides has been cut off from the longer side. The  $n = 4$  contact points,  $C_1, C_2, C_3, C_4$ , the position of the center of mass,  $C$ , and the position of the geometric centroid,  $G$ , are illustrated.

### C. Estimation of the mass $m$

After having estimated  $J$  and observed  $\mathbf{z}_C$ , the quantity  $\mathbf{p}_{iC}$  is known by each robot  $i$  using (6), and therefore each robot can estimate  $\dot{\mathbf{p}}_C$  using (7). Thus, if each robot applies the same force vector as in Sec. V-B, Eq. (1) of the dynamical model of the payload becomes

$$\ddot{\mathbf{p}}_C = \frac{n}{m} \mathbf{f}, \quad (25)$$

i.e., a linear system with measured output  $\dot{\mathbf{p}}_C$ . Thus, an approach similar to the one used to estimate  $J$ , relying on the technique recalled in the Appendix, can be applied again to estimate  $m$ . We omit here the details for brevity. We remark that this solution is again distributed, since each robot needs only to receive velocity vectors from its neighbors.

## VI. NUMERICAL SIMULATIONS

A network of  $n = 4$  robots manipulates a  $C$ -shaped unknown planar object  $B$ , see Fig. 3. The object has a uniformly distributed mass  $m = 5\text{ kg}$ , and its inertia moment is  $J = 12.3795\text{ kg m}^2$ . The robot's communication network is a line topology, i.e.,  $\mathcal{E} = \{(1, 2), (2, 3), (3, 4)\}$ .

First, each robot executes Step 1 to estimate the relative positions of its neighbors by setting a random force  $\mathbf{f}_i$ . The results of such estimates are illustrated in Figs. 4(a)–4(f). Then, once Steps 2–3 are performed, Step 4 is executed, in which each robot  $i$  sets its input force as  $\mathbf{f}_i = \mathbf{z}_i^\perp$  in order to estimate the inertia moment  $J$ . The initial guess of  $J$  is set to  $1\text{ kg m}^2$ . The evolution of the estimate  $\hat{J}$  is illustrated in Fig. 5(a). In Step 5, the input force is set as  $\mathbf{f}_i = \mathbf{f} = (0 \ f_y)^T$ , for all  $i \in \mathcal{I}$ , in order to estimate the vector  $\mathbf{z}_C$ . The convergence of the observer to  $\mathbf{z}_C$  is illustrated in Fig. 6(a), where the value  $f_x = J/n$  is used by each agent so that  $\bar{f}_y = 1$ . Figures 6(b) and 6(c) show the trend of the error  $\mathbf{e}$  and of the Lyapunov function  $V(\mathbf{e})$  used in the proof of the observer, respectively. Both converge quickly to zero, as predicted by our theory. At this point, each robot is able to measure the velocity of the center of mass,  $\dot{\mathbf{p}}_C$ , in order to estimate the mass  $m$ , whose initial guess is set at  $\hat{m} = 1\text{ kg}$ . Convergence of the estimated parameter is illustrated in Fig. 5(b).

## VII. CONCLUSION

In this paper, we have proposed a distributed strategy for the estimation of the inertial parameters of an unknown body manipulated by a team of networked mobile robots.

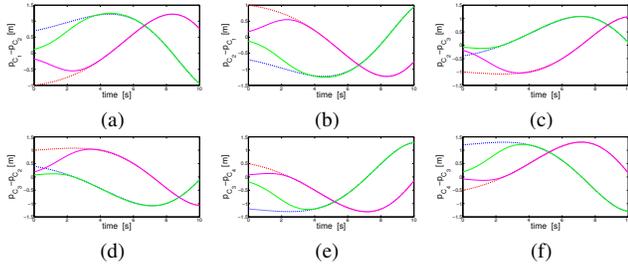


Fig. 4: Estimated coordinates of the displacements between the contact points (continuous lines) vs. ones (dashed lines) for (a)  $\mathbf{p}_{C_1} - \mathbf{p}_{C_2}$  (b)  $\mathbf{p}_{C_2} - \mathbf{p}_{C_1}$  (c)  $\mathbf{p}_{C_2} - \mathbf{p}_{C_3}$  (d)  $\mathbf{p}_{C_3} - \mathbf{p}_{C_2}$  (e)  $\mathbf{p}_{C_3} - \mathbf{p}_{C_4}$  (f)  $\mathbf{p}_{C_4} - \mathbf{p}_{C_3}$ .

The estimation is performed through a series of steps, that eventually yields a complete estimation. All the assumptions made are realistic. In particular, we do not assume that the robots' velocity can be controlled, yet applied forces have to be measured and controlled. Moreover, only the velocity of contact points should be measured, whereas their positions and accelerations are not needed. The proposed strategy involves low computational burden, simulation results are very satisfactory and confirm the effectiveness of our approach.

#### APPENDIX

We briefly recall a well known parameter estimation procedure based on filtering linear dynamics [15], which is used in our algorithm during the estimation of constant parameters such as the inter-distance between contact points, the moment of inertia  $J$  and the mass  $m$ . Notably, given the linear system  $\dot{\mathbf{y}} = \theta \mathbf{u}$ , the problem is to estimate the unknown constant parameter  $\theta \in \mathbb{R}$  on the basis of measurements of the quantities  $\mathbf{u} \in \mathbb{R}^l$  and  $\mathbf{y} \in \mathbb{R}^l$  only, where  $l \geq 1$ . To this aim, a low-pass filtered versions of the measured signals  $\mathbf{u}$  and  $\mathbf{y}$ , namely  $\mathbf{u}^f$  and  $\mathbf{y}^f$ , is computed as

$$\dot{\mathbf{u}}^f = k_f(\mathbf{u} - \mathbf{u}^f), \quad \dot{\mathbf{y}}^f = k_f(\mathbf{y} - \mathbf{y}^f),$$

with  $k_f > 0$ . Due to the linearity,  $\dot{\mathbf{y}}^f = \theta \mathbf{u}^f$  holds, i.e.,

$$k_f(\mathbf{y} - \mathbf{y}^f) = \theta \mathbf{u}^f, \quad (26)$$

which relates known quantities, except for  $\theta$ . We remark that (26) holds after a transient phase due to the filter initial conditions.

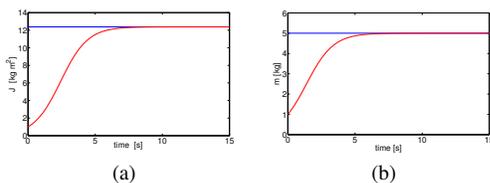


Fig. 5: Estimation of the inertia moment  $J$  (a) and of the mass  $m$  (b) of the unknown payload  $B$  made by the 4 robotic agents.

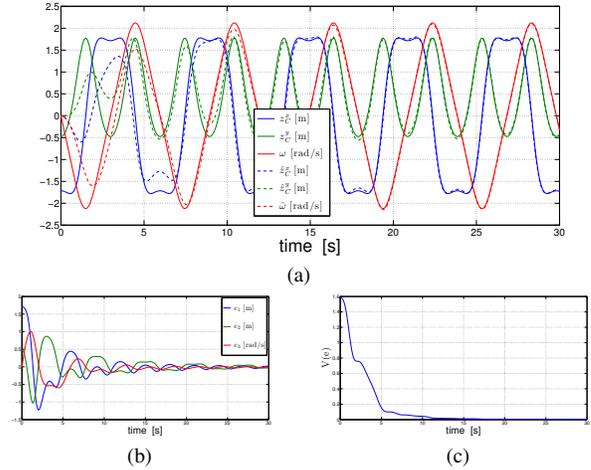


Fig. 6: Observation of the center-of-mass position: (a) Cartesian coordinates of the displacement  $\mathbf{z}_C = \mathbf{p}_G - \mathbf{p}_C$  and of the angular velocity  $\omega$  (dashed lines) versus their real values (continuous lines), (b) estimation errors, and (c) plot of the Lyapunov function  $V(\mathbf{e})$ .

#### REFERENCES

- [1] A. Yamashita, T. Arai, J. Ota, and H. Asama. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Trans. on Robotics*, 19(2):223–237, 2003.
- [2] 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*, pages 5227–5233, Tokyo, Japan, Nov. 2013.
- [3] 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*, pages 4335–4341, Hong Kong, China, May. 2014.
- [4] T. Lee, K. Sreenath, and V. Kumar. Geometric control of cooperating multiple quadrotor UAVs with a suspended payload. In *52nd IEEE Conf. on Decision and Control*, pages 5510–5515, Florence, Italy, Dec. 2013.
- [5] A. C. Satici and M. W. Spong. Nonholonomic cooperative manipulation of polygonal objects in the plane. In *51st IEEE Conf. on Decision and Control*, pages 2439–2446, Maui, HI, Dec. 2012.
- [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*, pages 356–361, Hong Kong and Macau, China, June 2005.
- [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*, pages 3845–3852, Nice, France, Sep. 2008.
- [8] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997.
- [9] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 2000.
- [10] R. Aragues, L. Carlone, C. Sagues, and G. Calafiore. Distributed centroid estimation from noisy relative measurements. *Systems & Control Letters*, 61(7):773–779, 2012.
- [11] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [12] S. Giannini, D. Di Paola, A. Petitti, and A. Rizzo. On the convergence of the max-consensus protocol with asynchronous updates. In *52nd IEEE Conf. on Decision and Control*, pages 2605–2610, Florence, Italy, Dec. 2013.
- [13] A. Fagiolini and A. Bicchi. On the robust synthesis of logical consensus algorithms for distributed intrusion detection. *Automatica*, 49(8):2339–2350, 2013.
- [14] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *iac*, 22(5):728–740, 1977.
- [15] J. J. E. Slotine and W. Li. *Applied nonlinear control*. Prentice Hall, 1991.