Citation: Rafael Herguedas, Gonzalo López-Nicolás, and Carlos Sagüés. Multirobot transport of deformable objects with collision avoidance. IEEE Systems Journal, 17(2):3224–3234, 2023.

Multirobot transport of deformable objects with collision avoidance

Rafael Herguedas, *Student Member, IEEE,* Gonzalo López-Nicolás, *Senior Member, IEEE,*
and Carlos Sagüés, *Senior Member, IEEE*

***Abstract*—In the problem of autonomous transport of deformable objects, we propose a multirobot approach for steering a large object to a target configuration (object dimensions, orientation and position). Firstly, we present a deformation model based on the evolution over time of the dimensions and rotation of the object's bounding box. We consider the object is grasped by a set of mobile robots with double-integrator dynamics. Then, we propose a set of nominal controllers that allows reaching a desired configuration of the bounding box that models the deformable object. In order to prevent collisions of the object with static or dynamic obstacles, we formulate a control barrier function (CBF) that exploits our deformation model. Finally, we integrate the nominal controllers and the CBF into a quadratic-programming controller, which includes overstretching avoidance and velocity constraints. We report simulation results to show the performance of this approach in different test scenarios.**

***Index Terms*—Deformable models, multirobot systems, transportation, collision avoidance, optimization.**

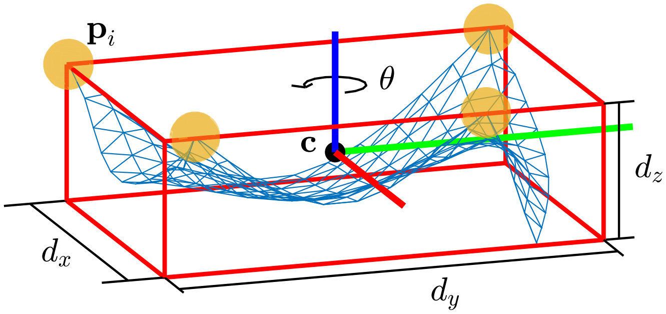

Fig. 1. Elastic sheet (blue mesh), held by four grippers (large spheres), and its bounding box, formed by the red lines that intersect between them. The main parameters of the bounding box (BB) are: the centroid $\mathbf{c}$, the dimensions d_x, d_y and d_z, and the orientation around the vertical axis θ. A grasping point $\mathbf{p}_i$ and the BB x-y-z reference frame in $\mathbf{c}$ are also depicted.

I. Introduction

TRANSPORT and manipulation of deformable objects involve tasks that are complex even for human operators. Automation of these tasks requires to deal with different challenges, which range from measuring deformation of the object to coordinating different robots for achieving the manipulation goals [1]. Despite their complexity, autonomous systems for transport and manipulation of deformable objects can improve the safety of workers and the efficiency of processes in industrial, medical and domestic contexts, among others [2]. In particular, we consider transport tasks of large deformable objects that must be steered to a target configuration, while avoiding collisions. This represents a paradigmatic case of industries such as the textile or the footwear manufacturing ones, where large fabric parts are transported and progressively transformed between different working stations. Other sectors that could also obtain potential benefits from this approach include construction and logistics. Modern construction techniques incorporate composite materials that are reinforced with layers of fabric, which could be transported and applied by robotic teams. As for the logistics sector, the ability to handle deformable goods with autonomous systems could increase the storage capacity and would reduce the cost of rigid packaging.

When manipulating deformable objects that are large, fragile, heavy or difficult to grasp, multiple manipulators are usually required [3]. Multirobot systems extend the robustness and resilience of the single agent methods, which are more sensitive to perturbations, but the control actions must be tightly coordinated due to the nature of the tasks [4]. We refer to robustness as the capability of a system to work under disturbances that do not cause structural changes on the system, and resilience as the ability of a system to autonomously recover its original function from external and/or internal disturbances, which cause an interruption in the normal operation of the system [5]. Works dealing with multirobot transport of deformable objects include a pioneer approach for nonholonomic manipulators based on nonsmooth Lyapunov functions, with guaranteed collision avoidance [6]. In that proposal, deforming the object is the means to avoid obstacles, but it is not a control goal by itself and any feasible deformation state is expected. Another study describes the design of a table with multiple actuators, with caterpillar locomotion-based mechanisms, for manipulation and transport of delicate objects [7]. Simultaneous control of the shape, scale, rotation and position of a formation of robots carrying a deformable object, is achieved with a formation control method [8]. If some specific shape and scale are required, deformation of the object in 2D can be explicitly controlled in that work. A different approach for simultaneous transport and shape control was developed for robots with non-holonomic constraints [9]. Those methods do not tackle the problem of

The authors are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain. {rherguedas, gonlopez, csagues} @unizar.es

This work was supported by projects COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF), TED2021-130224B-I00 and via grants PGC2018-098719-B-I00 and PID2021-124137OB-I00, funded by MCIN/AEI/10.13039/501100011033. The first author was partially supported by the EU through the European Social Fund (ESF) "Construyendo Europa desde Aragón".

explicitly controlling the deformation of the object in 3D, and they do not consider collision avoidance in their formulation. We can also find approaches based on Dynamic Movement Primitives (DMP) [10] or human-robot collaborative systems for transport of highly deformable planar materials [11], which do not test collision avoidance against dynamic obstacles.

Despite being challenging, modeling deformation is a fundamental step in manipulation of deformable objects. There are many available approaches for this purpose, ranging from the well-known finite element method to the more recent deep learning-based developments [12]. One of the main differences between our proposal and the deep learning methods, which are very popular nowadays, is the required amount of information. Compared to our approach, they require more data for being trained to accomplish the desired task, and it is more difficult to predict the behavior of neural networks. In our work, we choose a geometric model based on a deformable bounding box of the object. Different variants of this concept have been exploited in the past. A 2D version, called *outlined rectangle*, has been considered to obtain a convenient shape of the multirobot transportation system in previous studies [13], [14]. Similarly, the 2D deformable box is adopted in a motion planning strategy for approximating the shape of a team of manipulators and the transported rigid payload [15]. Another study about probabilistic roadmap motion planning for deformable robots obtains the deformation of the robot from a deformed bounding box [16]. A deformable bounding box with variable width is also utilized to describe the shape of an hexapod robot for navigation through confined spaces [17]. None of these previous works obtains a deformable bounding box model in 3D as the one we propose. Our approach allows predicting the evolution of the three dimensions of the bounding box under a specific set of control actions.

Guaranteeing safety under uncertainties and unexpected events constitutes one of the greatest concerns in robotics. Collision avoidance is a classical safety problem, which has been solved with different optimization techniques, such as sequential convex programming [18], [19] and model predictive control (MPC) [20], [15], to name a few. Control barrier functions (CBFs) represent a powerful tool for keeping the state of a system in the safe region. They can be applied to restrict the outputs from the nominal controller of a system, which potentially can induce unsafe behaviors (e.g. collisions). Collision avoidance systems based on this technique have been developed for systems of multiple mobile robots [21] or UAVs [22], [23].

The method we propose builds upon our previous proposal for multirobot transport of deformable objects with minimal motions [24]. In contrast to this former study, the 3D deformable bounding box model is now built in terms of accelerations. Even though the new model has the same structure, the dynamics of the grippers are significantly different with respect to the single integrator alternative. Also in [24] the model was not exploited to have direct control over the shape and orientation around the vertical axis of the object's BB. Another difference is that in the former study the motion of the transported object was constrained to a given route, but in the present work the route is computed online by the Q-P controller. Additionally, the new collision avoidance system based on the CBF is more robust and effective than the previous system, in which a nonlinear optimization was developed. The new system allows transporting a deformable object in 3D environments to a desired position, without collisions, while simultaneously deforming and rotating it to the desired dimensions and orientation, without overstretching the object during the process. Our main contributions in the present study are the deformable bounding box model for mobile manipulators with double-integrator dynamics, the CBF formulation for collision avoidance between the deformable object and static or dynamic obstacles, and the quadratic-programming (Q-P) controller, which integrates the nominal controllers with the CBF and additional safety constraints.

II. Problem statement

Let us consider a deformable object to be transported in a 3D environment by $N \geq 3$ mobile robots, which move on the floor following double-integrator dynamics. Each robot mounts a rotational-free gripper that grasps a set of points $\mathbf{p}_i \in \mathbb{R}^3$, $i = 1, ..., N$ in the object boundary. We model the object shape by its *ground-parallel bounding box*, defined as follows [24]:

Definition 1: **Ground-parallel bounding box.** The ground-parallel bounding box (BB) is a box that contains the object and whose top and bottom faces are parallel to the ground plane. It is defined by its centroid $\mathbf{c} \in \mathbb{R}^3$, the box dimensions $\mathbf{d} = [d_x, d_y, d_z]^\intercal$ and the box orientation around the vertical axis $\boldsymbol{\theta} = [0, 0, \theta]^\intercal$. $\mathbf{d}$ and $\boldsymbol{\theta}$ are expressed in the BB reference frame, with origin in $\mathbf{c}$ and whose axes are perpendicular to the BB faces.

This structure is oriented around the main plane of motion in our problem, and therefore considers the rotations with greater impact in the transport task. Note that the rotations around the remaining axes, with much lower impact, are not included as a design criterion to reduce the dimensionality of the problem. However, we will show that 4 degrees of freedom (x, y and z dimensions and rotation around z) add flexibility to avoid obstacles in 3D, while offering a wide range of possible control configurations. We obtain the BB of the object from a set of points $\mathbf{v}_m$, $m = 1, ..., M$ measured on its surface, by means of range sensors or fiducial markers. We do not need these points to be accurate, as long as they represent an approximate shape of the transported object.

We consider that there are obstacles in the environment, described by a set of points $\mathbf{q}_j \in \mathbb{R}^3$, $j = 1, ..., J$ and detected by the system with onboard range sensors, up to a limit distance R.

The task we envision consists in steering the BB of the grasped object without collisions to a target configuration, defined as the combination of position, dimensions, and orientation. This can be understood as a conventional transport process (i.e. translating the object from one place to another) with additional control over the shape and orientation of the object along translation. We propose the following errors to assess the completion of this task:

$$\mathbf{e}_c = \mathbf{c} - \mathbf{c}_t \ , \tag{1}$$

$$\mathbf{e}_d = \mathbf{d} - \mathbf{d}_t \ , \tag{2}$$

$$e_\theta = \theta - \theta_t \ , \tag{3}$$

where $\mathbf{c}_t \in \mathbb{R}^3$ is the target position of the BB centroid, which is set in the 2D plane (the z coordinate is ignored), $\mathbf{d}_t \in \mathbb{R}^3$ are the target dimensions of the BB and θ_t is the target BB orientation around the vertical axis.

III. Deformable bounding box model

A. Model description

In this section we define the object deformation model considered in our study. The model is based on the *deformable bounding box* paradigm, which extends the concept of BB (Definition 1) with new properties.

Definition 2: **Deformable bounding box.** The deformable bounding box (DBB) is the BB whose dimensions and rotation change over time when certain actions are applied to the object. These actions are traction/compression accelerations in the horizontal plane $\ddot{\mathbf{G}} = [\ddot{G}_x, \ddot{G}_y, 0]^\intercal$ and vertical rotation acceleration $\ddot{\boldsymbol{\phi}} = [0, 0, \ddot{\phi}]^\intercal$. As the object deforms, the DBB evolves smoothly between consecutive time instants.

The process for obtaining this kind of BB starts by computing the ground-parallel minimal bounding box of the object. The ground-parallel minimal bounding box is equivalent to the extrusion of the 2D minimal bounding box of the object along the object's height. Then, we obtain the BB reference frame with origin in $\mathbf{c}$ whose axes are perpendicular to the BB faces. Note that the bounding box this model requires does not correspond to the ground-parallel minimal bounding box, whose dimensions and orientation may present sharp changes over time. Therefore, the next step involves applying Principal Component Analysis (PCA) to the collection of vectors $\mathbf{v}_{bm} = \mathbf{v}_m - \mathbf{c}, \ \forall m$. These vectors are stacked in the matrix $\mathbf{V}_b \in \mathbb{R}^{M\times 3}$, which after applying the PCA yields $\mathbf{V}_b = \mathbf{U}\mathbf{S}\mathbf{V}^\intercal$. We can obtain the principal direction in the horizontal plane of the points in the global reference system as $\varphi = \text{atan2}(V_{2,1}, V_{1,1})$, where the subscripts denote the corresponding elements of the matrix. In this case, the principal direction of the points in 2D constitutes a line that minimizes the distance to them. In the following deformation instants the angles between the principal direction and the horizontal axes of the BB reference will be preserved, which means that the orientation of the BB around the vertical axis will follow the evolution of the principal direction of the object in the horizontal plane. The principal direction represents a convenient reference for orienting the BB, due to the fact that it varies as smoothly as the object points do. Then, the orientation of the BB can be computed as $\theta_k = \theta_0 - \varphi_0 + \varphi_k$. Finally, the dimensions $\mathbf{d}$ of the BB are computed from the maximum and minimum coordinates of the points $\mathbf{v}_m$ in the new BB reference frame.

We build the object deformation model upon Definition 2 in the following manner:

$$\left[\ddot{\mathbf{d}}^\intercal, \ddot{\theta}\right]^\intercal = \mathbf{J}\left[\ddot{G}_x, \ddot{G}_y, \ddot{\phi}\right]^\intercal = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \\ k_{41} & k_{42} & k_{43} \end{bmatrix} \begin{bmatrix} \ddot{G}_x \\ \ddot{G}_y \\ \ddot{\phi} \end{bmatrix} , \tag{4}$$

where

$$\ddot{\mathbf{d}}_k = \left(\dot{\mathbf{d}}_k - \dot{\mathbf{d}}_{(k-1)}\right) / \ \Delta t \ , \tag{5}$$

$$\dot{\mathbf{d}}_k = \left(\mathbf{d}_k - \mathbf{d}_{(k-1)}\right) / \ \Delta t \ , \tag{6}$$

$$\ddot{\theta}_k = \left(\dot{\theta}_k - \dot{\theta}_{(k-1)}\right) / \ \Delta t \ , \tag{7}$$

$$\dot{\theta}_k = \left(\theta_k - \theta_{(k-1)}\right) / \ \Delta t \ . \tag{8}$$

$\Delta t = t_k - t_{(k-1)}$ is the time interval between instants k and $k-1$. The interaction matrix $\mathbf{J}$ in (4) maps the input accelerations to the deformation $\ddot{\mathbf{d}}$ and rotation $\ddot{\theta}$ accelerations of the DBB by means of 12 parameters. Nine of them vary depending on the object's mechanical properties and the grippers' configuration. We consider that the rotation accelerations are small, and therefore centrifugal effects can be neglected. This makes DBB deformations independent from DBB rotations, i.e. $k_{13} = k_{23} = k_{33} = 0$. The intuition behind this proposal is that the dynamics of deformation can be approximated with sufficient accuracy by this linear model. Given that the dimensions of the BB and, therefore, the dimensions of the object are constrained, as we describe in Section IV-C the set of deformation states is bounded.

Once we have described the main aspects of the model, we need to connect it with the actuators that produce the required inputs. Next equations show how the grippers accelerations in the global reference are computed from $\ddot{G}_x$, $\ddot{G}_y$ and $\ddot{\phi}$:

$$\begin{aligned} \ddot{\mathbf{p}}_i = \mathbf{R}\ (&\ddot{\mathbf{c}}_t^{BB} + \ddot{\boldsymbol{\phi}} \times \mathbf{p}_i^{BB} + \dot{\boldsymbol{\phi}} \times (\dot{\boldsymbol{\phi}} \times \mathbf{p}_i^{BB}) \\ &+ 2\dot{\boldsymbol{\phi}} \times \dot{\mathbf{p}}_i^{BB} + \frac{\ddot{\mathbf{G}} \circ \text{sgn}(\mathbf{p}_i^{BB})}{2}) \ , \end{aligned} \tag{9}$$

$$\mathbf{p}_i^{BB} = \mathbf{R}^\intercal \left(\mathbf{p}_i - \mathbf{c}\right) \ , \tag{10}$$

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{11}$$

where $\ddot{\mathbf{c}}_t^{BB} \in \mathbb{R}^3$ is the target horizontal acceleration of the BB centroid in the BB reference ($\ddot{c}_{tz}^{BB} = 0$, by convention), $\mathbf{R} \in SO(3)$ is the rotation matrix from the BB to the global coordinate systems, '$\times$' is the cross product operator, '$\circ$' is the element-wise product operator and $\mathbf{p}_i^{BB}$ is the position of gripper i in the BB reference. Equation (9) represents the acceleration of the grippers as if they were connected to a rigid BB (see [25, eq. (6.10)], about the linear acceleration of a manipulator according to the rigid body dynamics). However, the last term corrects the resulting acceleration with the deformation inputs to the DBB [24]. The main purpose of this correction term is to preserve the *action-counteraction balance* of the system (see Remark 1). This balance allows preventing global displacements of the BB, when deforming it, by applying the half of the traction/compression inputs from opposite directions, at each of the two horizontal axes of the BB reference frame.

One of the most prominent advantages of this model, in contrast with the 2D counterpart, is that it does not only allow to predict horizontal deformations and rotation around z from the inputs $\ddot{\mathbf{G}}$ and $\ddot{\phi}$, but also the vertical deformation of the object's BB. Besides allowing to achieve specific configurations of the BB in 3D, this kind of deformation control can be useful for collision avoidance. For instance, if an obstacle in height is encountered and the imposed constraints are satisfied, the system can surmount the obstacle instead of going around it, and reduce the traveled distance. However, in case a more conservative behavior is sought, in which the bounding box never goes over the incoming obstacles, the user can increase the safety distance or constrain the minimum vertical dimension as we will explain later (see Sections IV-B and IV-C).

Remark 1: Note that at least one positive and one negative component of the element-wise multiplication $\ddot{\mathbf{G}} \circ \operatorname{sgn}(\mathbf{p}_i^{BB})$ must exist for the two non-zero components of $\ddot{\mathbf{G}}$. If this condition is satisfied, the action-counteraction balance is preserved for each BB horizontal axis.

Next proposition provides the general rule for guaranteeing the action-counteraction balance as described in Remark 1.

Proposition 1: Consider a set of $N \geq 3$ grippers carrying a deformable object. These grippers are positioned so that they are separated less than π radians from each of the neighboring ones, around the object's BB centroid $\mathbf{c}$. With this setup, the action-counteraction requirement is met at each BB horizontal axis for any possible orientation of the BB reference frame and the rotated inputs.

Proof 1: According to Remark 1, the positive and negative contributions of the non-zero components of $\ddot{\mathbf{G}}$ must be performed by, at least, one gripper each. If the grippers are separated less than π radians from each of the neighboring ones, three quadrants of every possible BB reference frame will always include a gripper. Therefore, the action-counteraction balance will always hold.

B. Model identification

In this section, we describe the model identification problem in terms of accelerations, with additional details and further analysis with respect to our previous work [24]. As we mentioned in the previous section, 9 parameters of the DBB model vary according to the particular object that is transported, the number of grippers and the relative position of the grippers with respect to the object. This implies that they must be obtained for each case. The most basic identification strategy consists in taking 3 different measurements of $\ddot{\mathbf{d}}$ and $\ddot{\theta}$ from randomly chosen input actions $\ddot{\mathbf{G}}$ and $\ddot{\phi}$. Then, an estimate of the model parameters would be obtained by solving the system of equations. However, this method is highly sensitive to errors coming from the sensing devices and the specificity of the measurements set, and in general produces low quality results. An ordinary least-squares approach improves the quality of the previous solution by linearly adjusting the parameters from a set of $S > 3$ measurements:

$$[k_{11}, k_{12}, ..., k_{43}]^\intercal = (\mathbf{A}^\intercal\, \mathbf{A})^{-1} \mathbf{A}^\intercal\, \mathbf{b} \,, \tag{12}$$

$$\mathbf{A} = \text{blkdiag}(\begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} \\ \vdots & \\ \ddot{G}_{xS} & \ddot{G}_{yS} \end{bmatrix}, \begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} \\ \vdots & \\ \ddot{G}_{xS} & \ddot{G}_{yS} \end{bmatrix}, \begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} \\ \vdots & \\ \ddot{G}_{xS} & \ddot{G}_{yS} \end{bmatrix}, \begin{bmatrix} \ddot{G}_{x1} & \ddot{G}_{y1} & \ddot{\phi}_1 \\ \vdots & & \\ \ddot{G}_{xS} & \ddot{G}_{yS} & \ddot{\phi}_S \end{bmatrix}) \,, \tag{13}$$

$$\mathbf{b} = [\ddot{d}_{x1}, \ddot{d}_{x2}, ..., \ddot{d}_{xS}, \ddot{d}_{y1}, ..., \ddot{d}_{yS}, \ddot{d}_{z1}, ..., \ddot{d}_{zS}, \ddot{\theta}_1, ..., \ddot{\theta}_S]^\intercal \,, \tag{14}$$

with $\mathbf{A} \in \mathbb{R}^{4S \times 9}$ and $\mathbf{b} \in \mathbb{R}^{4S}$.

Although this technique is able to diminish the errors coming from sensor noise and local effects, created by some measurements, the particular choice of measurements and its quantity still affect the accuracy of the resulting DBB model. We obtain a convenient selection of measurements by applying the *observability index maximization* technique. Observability indexes derive from the alphabet optimalities, and they provide statistical information about the numerical conditioning of the model and its variance [26]. The field of robot calibration covers most of the approaches that have exploited the observability index maximization, for purposes like improving the robustness of calibration to sensor noise [27] or for selecting the index that yields the most accurate robot calibrations [28], among other applications.

Let us denote $\sigma_r \leq \sigma_{r-1} \leq ... \leq \sigma_1$ the r singular values of the regressor matrix $\mathbf{A}$. The five observability indexes we study are computed as [28]: $O_1 = (\sigma_r\, \sigma_{r-1} ... \sigma_1)^{1/r}/S^{1/2}$, $O_2 = \sigma_r/\sigma_1$, $O_3 = \sigma_r$, $O_4 = \sigma_r^2/\sigma_1$ and $O_5 = (\sum_{i=1}^{r} 1/\sigma_i)^{-1}$. O_1 represents the volume of a hyperellipsoid in which the singular values correspond to the length of the axes. It is an indicator of the data scatter, i.e. the exploration of the parametric space. Instead of measuring the size of the hyperellipsoid, O_2 quantifies the ratio between the shortest and the largest axes. In other words, it shows how well explored some of the parameters are in comparison with the rest. O_3 indicates the least explored parameter (worst case). O_4 is the combination of O_2 and O_3, and O_5 is the harmonic mean of the singular values divided by the number of model parameters.

In order to maximize these indexes we apply Algorithm 1, which is based in the DETMAX algorithm [29]. First, we generate a set of candidate inputs η with different combinations of $\ddot{G}_x$, $\ddot{G}_y$ and $\ddot{\phi}$. Starting with an initial experiment design of S random input combinations from η, the algorithm maximizes an observability index by continuously exchanging inputs (rows of $\mathbf{A}$) between the experiment design and η. The process stops when the index converges to a maximum value, up to a user-defined tolerance. Then, we obtain an optimal matrix $\mathbf{A}$ and, by applying the input combinations it contains to the system, the matrix of measurements $\mathbf{b}$. Finally, we compute from (12) the parameters of the DBB model. In contrast to Mitchell's proposal, our algorithm considers a constant experiment size, and instead of obtaining D-optimal experiment designs we optimize the design by maximizing an observability index.

Algorithm 1 Maximize an observability index O.

```
$O_{\text{diff}} \leftarrow \infty$
$O_{-1} \leftarrow 0$
optimStop $\leftarrow$ $tolerance$
**exp** $\leftarrow$ $S$ random inputs from $\eta$
**while** $O_{\text{diff}} >$ optimStop **do**
    $O_{\text{prev}} \leftarrow 0$
    **for** $i = 1, ..., \text{size}(\eta)$ **do**
        **newExp** $\leftarrow$ [**exp**, $\eta_i$]
        $O \leftarrow computeIndex$ (**newExp**)
        **if** $O_{\text{prev}} < O$ **then**
            $\kappa \leftarrow i$ # *auxiliary variable*
            $O_{\text{prev}} \leftarrow O$
        **end if**
    **end for**
    **exp** $\leftarrow$ [**exp**, $\eta_\kappa$] # *add input that maximizes O*
    **for** $j = 1, ..., S+1$ **do**
        **newExp** $\leftarrow \left[\text{exp}_1, ..., \text{exp}_{j-1}, \text{exp}_{j+1}, ..., \text{exp}_{S+1}\right]$
        $O \leftarrow computeIndex$ (**newExp**)
        **if** $O_{\text{prev}} > O$ **then**
            $\kappa \leftarrow j$
            $O_{\text{prev}} \leftarrow O$
        **end if**
    **end for**
    **exp** $\leftarrow \left[\text{exp}_1, ..., \text{exp}_{\kappa-1}, \text{exp}_{\kappa+1}, ..., \text{exp}_{S+1}\right]$ # *remove input that less diminishes O*
    $O \leftarrow computeIndex$ (**exp**)
    $O_{\text{diff}} \leftarrow (O - O_{-1})$
    $O_{-1} \leftarrow O$
**end while**
**return** $O$, **exp**
```

This technique allows obtaining suitable DBB models from a reduced set of measurements, which we select in advance to the model identification and the task execution. A comparative analysis of the different observability indexes, in terms of utility and performance, is shown in Section V-A. In that section we also study how the experiment size S affects the DBB model accuracy, and we justify the selection of O_1 for obtaining the most appropriate DBB models. In the studied test cases, we have realized that a DBB model with constant parameters, identified before performing the task, performs well the transport objectives. Even if not considered here, it could also be interesting to study the effects of updating the model along the manipulation.

IV. Controller for deformable object transport

A. Nominal control system

The control actions must steer the system to the desired configuration, reducing to zero the errors $\mathbf{e}_c$ (1), $\mathbf{e}_d$ (2) and e_θ (3). We propose the following set of nominal controllers for mobile manipulators with double-integrator dynamics:

$$\ddot{\mathbf{G}} = -\,k_{1d}\,\mathbf{e}_d - k_{2d}\,\dot{\mathbf{d}}\ , \tag{15}$$

$$\ddot{\phi} = -\,k_{1r}\,e_\theta - k_{2r}\,\dot{\theta}\ , \tag{16}$$

$$\ddot{\mathbf{c}}_t^{BB} = \mathbf{R}^\intercal(-k_{1t}\,\mathbf{e}_c - k_{2t}\,\dot{\mathbf{c}})\ , \tag{17}$$

where k_{1d}, k_{2d}, k_{1r}, k_{2r}, k_{1t} and k_{2t} are positive control gains. We aggregate the nominal control inputs in a single column vector

$$\mathbf{u}_n = [\ddot{G}_x, \ddot{G}_y, \ddot{\phi}, \ddot{c}_{tx}^{BB}, \ddot{c}_{ty}^{BB}]^\intercal\ . \tag{18}$$

Note that these controllers are centralized in order to drive the system with tightly coordinated actions, required in manipulation tasks as the ones we tackle.

B. Obstacle avoidance with control barrier funcion

Since collision avoidance is not explicitly accounted by the nominal controllers, we introduce a flexible, minimally invasive and robust solution to prevent collisions with obstacles in the environment. In particular, we consider an optimization method based on a CBF, which regulates the nominal control inputs so that the system is always in a safe state. We adopt a modified version of the centralized approach proposed by Wang *et al.* [21], where multiple robots with intersecting trajectories are able to reach target positions while avoiding robot-to-robot and robot-to-obstacle collisions. In our case, the transported deformable object makes collision avoidance more challenging to be guaranteed, as we must also consider object-to-obstacle collisions. Besides, we do not deal explicitly with robot-to-robot collisions, since they are implicitly avoided by constraining the minimum BB dimensions, as we will describe in the next section.

Firstly, we define a set of virtual points $\mathbf{p}_{vl} \in \mathbb{R}^3$, $l = 1, ..., L$ uniformly distributed over the faces of the BB. The dynamics of these points are linked to the DBB as

$$\begin{aligned}\ddot{\mathbf{p}}_{vl} = {} & \ddot{\mathbf{c}}_t^{BB} + \ddot{\boldsymbol{\theta}} \times \mathbf{p}_{vl} + \dot{\boldsymbol{\theta}} \times (\dot{\boldsymbol{\theta}} \times \mathbf{p}_{vl}) + 2\dot{\boldsymbol{\theta}} \times \dot{\mathbf{p}}_{vl} \\ & + (\mathbf{p}_{vl} - [0, 0, d_z/2]^\intercal) \circ \ddot{\mathbf{d}} \oslash \mathbf{d}\ .\end{aligned} \tag{19}$$

where '$\oslash$' is the element-wise division. Note that the top face of the BB is constant in height, due to the fact that the grippers move in the 2D plane. This is the reason why we include the $-[0, 0, d_z/2]^\intercal$ term, that shifts the zero vertical movement plane to the top face of the BB. Then, we can write an affine system

$$\dot{\mathbf{x}}_l = f(\mathbf{x}_l) + g(\mathbf{x}_l)\mathbf{u}\ , \tag{20}$$

where $\mathbf{x}_l = [\mathbf{p}_{vl}, \dot{\mathbf{p}}_{vl}]^\intercal$ and $f(\mathbf{x}_l)$ and $g(\mathbf{x}_l)$ are locally Lipschitz continuous functions. We propose the following criterion for collision avoidance of the system (20):

$$\|\mathbf{p}_{vlj}\| + \int_{t_0}^{t_f} \dot{p}_{vlj}^{\perp}(t)dt \geq D_{lj}^{min}\ , \tag{21}$$

where $\mathbf{p}_{vlj} = \mathbf{p}_{vl} - \mathbf{q}_j^{BB}$, $\mathbf{q}_j^{BB} = \mathbf{R}^\intercal(\mathbf{q}_j - \mathbf{c})$,

$$\dot{p}_{vlj}^{\perp} = \frac{\mathbf{p}_{vlj}^\intercal}{\|\mathbf{p}_{vlj}\|}\dot{\mathbf{p}}_{vlj} \tag{22}$$

is the relative velocity between $\mathbf{p}_{vl}$ and $\mathbf{q}_j^{BB}$ projected in the normal direction, $\dot{\mathbf{p}}_{vlj} = \dot{\mathbf{p}}_{vl} - \dot{\mathbf{q}}_j^{BB}$, $\dot{\mathbf{q}}_j^{BB} = \mathbf{R}^{\intercal}(\dot{\mathbf{q}}_j - \dot{\mathbf{c}})$ and

$$t_f = \frac{\dot{p}^{\perp}_{vlj}(t_f) - \dot{p}^{\perp}_{vlj}(t_0)}{\alpha_l} + t_0 \ . \tag{23}$$

At time t_f the points completely stop ($\dot{p}^{\perp}_{vlj}(t_f) = 0$) when applying the maximum deceleration $\alpha_l = \|\ddot{\mathbf{p}}_{vl}\|_{\infty}$, and D_{lj}^{min} is the minimum allowed distance between the virtual point $\mathbf{p}_{vl}$ and the obstacle $\mathbf{q}_j^{BB}$. Note that we take equations (20), (21), (22) and (23) as proposed by Wang *et al.* [21]. By substituting (23) and (22) in (21) and solving the definite integral, we obtain the candidate CBF

$$h_{lj} = 2\,\alpha_l\,(\|\mathbf{p}_{vlj}\| - D_{lj}^{min}) - \dot{p}^{\perp 2}_{vlj} \ . \tag{24}$$

According to its definition [30], h_{lj} is a suitable CBF if the following expression is satisfied:

$$\sup_{\mathbf{u}}[\mathcal{L}_f h_{lj}(\mathbf{x}_l) + \mathcal{L}_g h_{lj}(\mathbf{x}_l)\mathbf{u}] \geq -\varepsilon(h_{lj}(\mathbf{x}_l)) \ , \tag{25}$$

where $\mathcal{L}$ stands for the Lie derivative, $\mathbf{u} \in \mathbb{R}^5$ similarly to (18) is the control input and ε is an extended class K_{∞} function. Then, we apply this definition to our candidate CBF to obtain the set of linear constraints

$$\begin{aligned}\mathbf{p}_{vlj}^{\intercal}\dot{\mathbf{p}}_{vlj}\frac{\mathbf{p}_{vlj}^{\intercal}}{\|\mathbf{p}_{vlj}\|}\mathbf{d}_M\mathbf{J}_{ext}\mathbf{u} \leq \frac{\|\mathbf{p}_{vlj}\|}{2}\varepsilon(h_{lj}(\mathbf{x}_l)) + \alpha_l\mathbf{p}_{vlj}^{\intercal}\dot{\mathbf{p}}_{vl}\\ - \mathbf{p}_{vlj}^{\intercal}\dot{\mathbf{p}}_{vlj}\frac{\mathbf{p}_{vlj}^{\intercal}}{\|\mathbf{p}_{vlj}\|}(\boldsymbol{\theta}_M\mathbf{p}_{vl} + \boldsymbol{\theta}_{MV}\dot{\mathbf{p}}_{vl}) \ ,\end{aligned} \tag{26}$$

where

$$\mathbf{d}_M = \begin{bmatrix} \frac{p_{vlx}}{d_x} & 0 & 0 & -p_{vly} & 1 & 0 \\ 0 & \frac{p_{vly}}{d_y} & 0 & p_{vlx} & 0 & 1 \\ 0 & 0 & \frac{p_{vlz}}{d_z} - 0.5 & 0 & 0 & 0 \end{bmatrix} , \tag{27}$$

$$\mathbf{J}_{ext} = \begin{bmatrix} \mathbf{J} & \mathbf{0}_{4\times 2} \\ \mathbf{0}_{2\times 3} & \mathbf{I}_{2\times 2} \end{bmatrix} , \tag{28}$$

$$\boldsymbol{\theta}_M = \left[-\dot{\theta}^2, -\dot{\theta}^2, 0\right] \mathbf{I}_{3\times 3} \ , \tag{29}$$

$$\boldsymbol{\theta}_{MV} = \begin{bmatrix} 0 & -2\dot{\theta} & 0 \\ 2\dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \tag{30}$$

These equations constraint the system to maintain a safety distance D_{lj}^{min} with every obstacle point $\mathbf{q}_j$. Therefore, obstacles are treated as spheres of radius D_{lj}^{min}, but only for collision avoidance. The function ε determines how close the system will remain to the safety boundary when avoiding collisions.

C. Quadratic-programming controller

The nominal control inputs $\mathbf{u}_n$ must be modified to avoid collisions when obstacles intersect with the desired route. We achieve the collision-free configurations by means of a quadratic-programming (Q-P) based controller, which includes the CBF:

$$\begin{aligned} &\text{Given} \qquad \mathbf{u}_n, \mathbf{p}_{vl}, \dot{\mathbf{p}}_{vl}, \mathbf{q}_j^{BB}, \mathbf{J}, \mathbf{d}, \dot{\boldsymbol{\theta}}, D_{lj}^{min}, \alpha_l, \forall l, j \\ &\underset{\mathbf{u}}{\text{minimize}} \quad \xi = \sum_{i=1}^{5} k_{wi}(u_i - u_{ni})^2 \\ &\text{subject to:} \\ &\qquad \mathbf{A}_{lj}\mathbf{u} \leq b_{lj} \ , \forall l, j, \ l = 1, ..., L, \ j = 1, ..., J \ , \\ &\qquad \|\mathbf{u}\|_{\infty} \leq \alpha_l, i = 1, ..., N \end{aligned} \tag{31}$$

where $k_{wi} > 0, \forall i$ are control weights that satisfy $\sum_{i=1}^{5} k_{wi} = 1$, L is the number of virtual points on the BB, J is the number of points describing the obstacles,

$$\mathbf{A}_{lj} = \mathbf{p}_{vlj}^{\intercal}\dot{\mathbf{p}}_{vlj}\frac{\mathbf{p}_{vlj}^{\intercal}}{\|\mathbf{p}_{vlj}\|}\mathbf{d}_M\mathbf{J}_{ext} \ , \tag{32}$$

$$\begin{aligned} b_{lj} =& \frac{\|\mathbf{p}_{vlj}\|}{2}\varepsilon(h_{lj}(\mathbf{x}_l)) + \alpha_l\,\mathbf{p}_{vlj}^{\intercal}\dot{\mathbf{p}}_{vl} \\ & - \mathbf{p}_{vlj}^{\intercal}\dot{\mathbf{p}}_{vlj}\frac{\mathbf{p}_{vlj}^{\intercal}}{\|\mathbf{p}_{vlj}\|}(\boldsymbol{\theta}_M\mathbf{p}_{vl} + \boldsymbol{\theta}_{MV}\dot{\mathbf{p}}_{vl}) \ . \end{aligned} \tag{33}$$

The k_{wi} constants allow regulating the effects of the Q-P controller over the nominal controller. For instance, if we take higher values of k_{w1} and k_{w2} than those of k_{w3}, k_{w4} and k_{w5}, nominal rotations and global displacements will have preference over nominal deformations for being modified.

We can also implement additional constraints in this structure to avoid overstretching the object and also to limit the rotation and translation velocities. If we assume uniform accelerations between instants k and $k+1$, the linear inequalities

$$\begin{bmatrix} 2\left((\mathbf{d}^{max} - \mathbf{d}_k)/\Delta t^2 - \dot{\mathbf{d}}_k/\Delta t\right) \\ (\dot{\theta}^{max} - \dot{\theta}_k)/\Delta t \\ (\dot{\mathbf{c}}^{max} - [\dot{c}_{kx}, \dot{c}_{ky}]^{\intercal})/\Delta t \end{bmatrix} \geq \mathbf{J}_{ext}\ \mathbf{u} \ , \tag{34}$$

$$\begin{bmatrix} 2\left((\mathbf{d}^{min} - \mathbf{d}_k)/\Delta t^2 - \dot{\mathbf{d}}_k/\Delta t\right) \\ (\dot{\theta}^{min} - \dot{\theta}_k)/\Delta t \\ (\dot{\mathbf{c}}^{min} - [\dot{c}_{kx}, \dot{c}_{ky}]^{\intercal})/\Delta t \end{bmatrix} \leq \mathbf{J}_{ext}\ \mathbf{u} \ , \tag{35}$$

guarantee that $\mathbf{d}$, $\dot{\theta}$ and $\dot{\mathbf{c}}$ will be bounded between a set of maximum ($\mathbf{d}^{max}$, $\dot{\theta}^{max}$, $\dot{\mathbf{c}}^{max}$) and minimum ($\mathbf{d}^{min}$, $\dot{\theta}^{min}$, $\dot{\mathbf{c}}^{min}$) defined limits. Note that the constraint over the BB vertical dimension can also prevent the object from touching the ground, since the grippers travel at constant height.

By means of this controller, we obtain the closest control input to the nominal one that satisfies the constraints from the CBF and the linear inequalities (34) and (35). The proposed centralized formulation induces tightly coordinated motions of the robots that aim at preserving the deformable object without damages. In addition, we consider a small number of mobile manipulators in the studied practical cases. Therefore, the centralized version of the CBF and the Q-P controller seems appropriate for the current state of the method.

V. Results

A. Observability indexes analysis

In this section, we will identify the observability index that provides the most appropriate DBB model from a reduced set of measurements, for different grippers configurations and materials. We evaluate the quality of a model from its mean squared error, which is computed as

$$MSE = \frac{1}{S}\sum_{i=1}^{S}([\ddot{\mathbf{d}}_i, \ddot{\theta}_i] - [\tilde{\ddot{\mathbf{d}}}_i, \tilde{\ddot{\theta}}_i])^2 \,, \tag{36}$$

where $[\tilde{\ddot{\mathbf{d}}}_i, \tilde{\ddot{\theta}}_i]$ are the outputs predicted by the model.

Then, we start by creating a pool of 10^4 random combinations of $\ddot{G}_x$, $\ddot{G}_y$ and $\ddot{\phi}_z$. We divide this pool in four sets of 2500 input combinations, and we apply each set to a different configuration of the system:

1) The first configuration consists in a rectangular (2 × 3 [m]) mesh of mass–spring–damper elements (similar to the one in Fig. 1), with 77 nodes and stiffness, damping and nodal mass of 20 [N/m], 0.5 [N·s/m] and 0.025 [kg] respectively. This object is held by four grippers at the corners of the rectangle.
2) The second configuration includes the same object than the first one, but it is held by three grippers located as follows: two of them grasp the corners of one of the 3 [m] side, while the remaining one grasps the node in the middle of the opposing 3 [m] side of the mesh.
3) The third configuration includes the same object, but it is held by five grippers: four of them grasp the corners of the mesh, while the remaining one grasps the node in the middle of one of the 3 [m] side of the mesh.
4) The fourth configuration is the same than the first one except for the object: the stiffness of the mesh elements is set to 200 [N/m].

After that, the outputs $\ddot{d}_x$, $\ddot{d}_y$, $\ddot{d}_z$ and $\ddot{\theta}$ are obtained. The system configurations included in the pool of inputs and outputs provide sufficient depth for the analysis, since we consider similar cases in the next sections. The next step consists in creating an initial set of $S \geq 3$ inputs, by taking random samples from the pool. Algorithm 1 is then executed to maximize each observability index, by iteratively adding and removing inputs from the experiment design. Once the algorithm converges ($tolerance = 10^{-3}$ is chosen), the model is obtained from the selected inputs and the resulting outputs with least-squares.

Figure 2 shows the results of experiment designs in the range $3 \leq S \leq 50$. Due to the fact that the index optimization algorithm is affected by local maxima, the presented values are the mean of 20 maximization executions with different initial designs. Note that O_2, O_3, O_4 and O_5 present monotonically increasing values, while O_1 first increases up to 0.81, for 7 measurements, and then shows a slight continuous decay, hardly noticeable due to the scale. The mean squared errors of the resulting models are also depicted. It can be seen that the errors obtained with the O_1 maximization are lower than the ones obtained with O_2, O_3 and O_4 in the interval $3 \leq S \leq 20$ (which is, in practice, our range of interest in the number of measurements), and very similar to the ones of O_5 at some points. In addition, the error of the model with O_1 follows the opposite direction to the evolution of the index values. This is the behavior we seek: the O_1 value evolution indicates when to stop taking measurements for getting a representative model with minimal error.

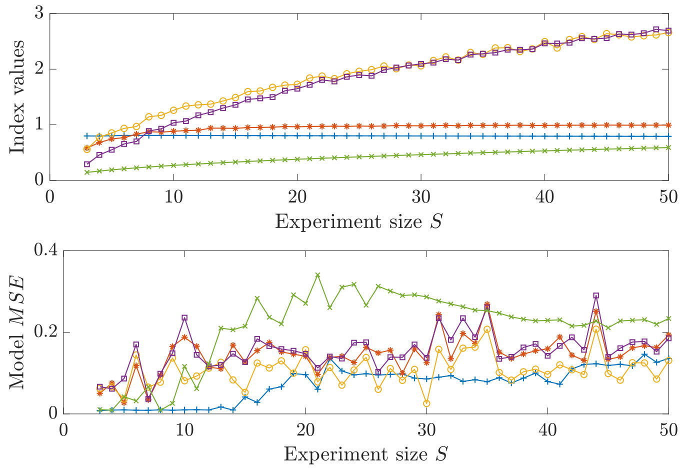

Fig. 2. Comparison between the maximization of indexes O_1 (blue '+'), O_2 (orange '∗'), O_3 (yellow '○'), O_4 (purple '□') and O_5 (green '×'), for experiment designs of increasing size S. In the upper plot, the mean observability index values of 20 simulations with different initial designs are depicted. In the lower plot, the mean values of the resulting models MSE, from the previous simulations, are shown.

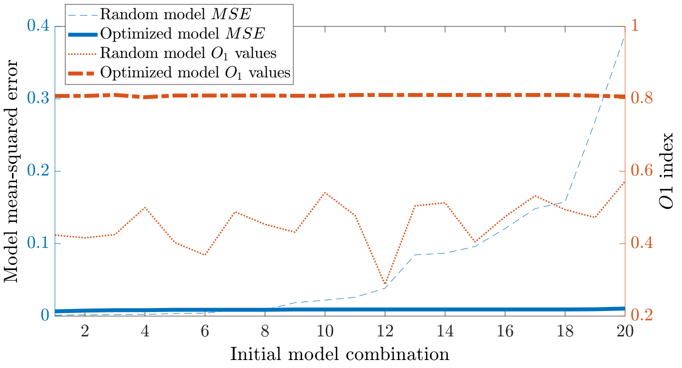

Fig. 3. Comparison of 20 different experiments of size $S = 7$, created by maximizing the observability index O_1 (thick lines) and selecting random inputs (thin lines). The index values (orange) and the mean squared error of the models (blue) are shown, sorted into increasing values of the error.

An additional test is carried out for verifying the effectiveness of O_1 and the maximization algorithm. Figure 3 represents the value of O_1 (thick dashed line) and the mean squared error of the resulting DBB model (thick blue line) after 20 index maximizations, which are executed with different initial experiment designs of size $S = 7$. For comparison, the values of O_1 (thin dotted line) and the mean squared error (thin dashed line) of 20 different random experiment designs with $S = 7$ are shown. The index-error pairs are sorted into increasing values of the error. We can see that the models obtained through index maximization are much more homogeneous and more accurate. In addition, the system is explored in higher depth and, therefore, better characterized, as the observability index values indicate.

B. Test with static obstacles

We evaluate our proposal in three different simulation scenarios in Matlab®. The first scenario includes three static

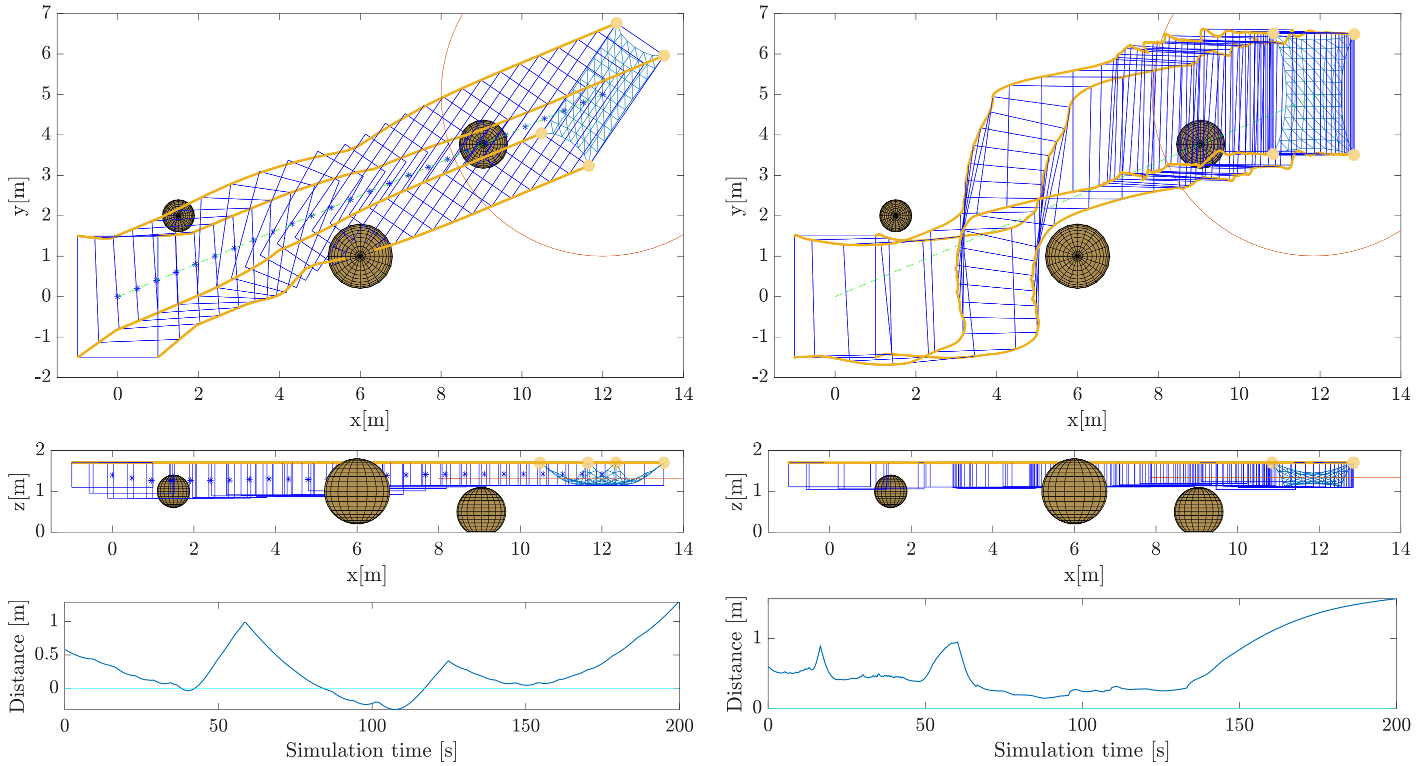

Fig. 4. The goal is to steer, without collisions, the object (blue mesh) from the initial position at [0, 0] meters to the target position at [12, 5] meters. Images at the left side correspond to our previous method [24], while at the right side we show the new method with the Q-P controller. The top and middle images show the top and side views of the test circuit, with the grippers at the corners of the sheet and their trajectories in orange. The bounding boxes of the object over time (blue boxes), the obstacles (brown spheres), the sensors range (orange circumference) and the ideal transport route (green dashed line) are also depicted. The bottom images show the absolute minimum distances between the mesh nodes and the nearest obstacle center. We subtract the radius of the nearest obstacle to check whether the node is inside (negative distance) or outside the obstacle (positive distance). We can see that our previous method is unable to avoid collisions at some instants, and we have no control over the shape and the orientation of the object. In contrast, the proposed method prevents collisions with the obstacles and preserves the initial dimensions and orientation of the box.

obstacles, and allows us to compare the Q-P controller with our previous method for agents with single integrator dynamics [24]. We deploy a team of $N = 4$ robots grasping a 2×3 [m] rectangular sheet by its four corners, at 1.7 [m] height. The sheet is modeled with a mesh of mass-spring-damper elements of 20 [N/m] stiffness, 0.5 [N·s/m] damping and 77 nodes with a 0.025 [kg] nodal mass. Note that these properties are not known by the system. Instead, we obtain the DBB model by maximizing the O_1 index from a set of $S = 7$ measurements. The control parameters are set as follows: $k_{1d} = 0.4$, $k_{2d} = 1$, $k_{1r} = 1$, $k_{2r} = 2.5$, $k_{1t} = 0.01$, $k_{2t} = 0.3$, $\varepsilon = 0.5\,h_{lj}$, $\alpha_l = 10$ [m/s^2], $D_{l1}^{min} = 0.55$ [m], $D_{l2}^{min} = 0.95$ [m], $D_{l3}^{min} = 0.65$ [m], $k_{w1} = k_{w2} = 0.2$, $k_{w3} = 0.1$, $k_{w4} = k_{w5} = 0.25$ and $R = 4$ [m]. The limit values to avoid overstretching and excessive velocities are set as $\mathbf{d}^{max} = [4, 6, 1.5]$ [m], $\mathbf{d}^{min} = [0.5, 1, 0.1]$ [m], $\dot{\theta}^{max} = -\dot{\theta}^{min} = \pi$ [rad/s] and $\dot{c}_x^{max} = \dot{c}_y^{max} = -\dot{c}_x^{min} = -\dot{c}_y^{min} = 0.5$ [m/s]. It is worth noting that fine tuning of the control parameters is not needed to successfully complete the task. In addition, we define $L = 488$ evenly distributed virtual points over the BB faces. The goal of the task is to transport the sheet without collisions to a target position in [12,5] meters, while getting the initial dimensions and orientation of the BB at the end of the test. Figure 4 shows the performance of our method in the referred task compared with the previous work. We can see that the Q-P controller is able to achieve the desired configuration of the object without collisions, while our previous approach fails in the latter purpose and it does not allow controlling the shape and orientation of the object. The bottom plots, which display the minimum distance between any mesh node and the nearest detected obstacle, illustrate the failure of the previous method and the success of the new one. When the value is negative, it means the node has penetrated that distance inside the volume of the obstacle. From this test we see that the impact of the imposed route in our previous method is drastically reduced in the current proposal, with a more flexible and robust formulation.

C. Test with dynamic obstacles

In the second test scenario we evaluate the system's performance under dynamic obstacles. Again, a team of $N = 4$ robots grasps a deformable sheet with the same mechanical properties than in the previous test, but in a different configuration. The DBB model is obtained following the same procedure, and the control parameters are equal except $k_{1d} = 1.0$, $k_{2d} = 2.0$, $k_{1r} = 1.5$, $k_{2r} = 3.0$, $\alpha_l = 15$ [m/s^2]. With these changes we get a faster behavior of the nominal controller, and the system reacts better to the incoming obstacles. The goal is to transport the sheet without collisions to the target position in [12,5] meters, while simultaneously reducing a 20% the

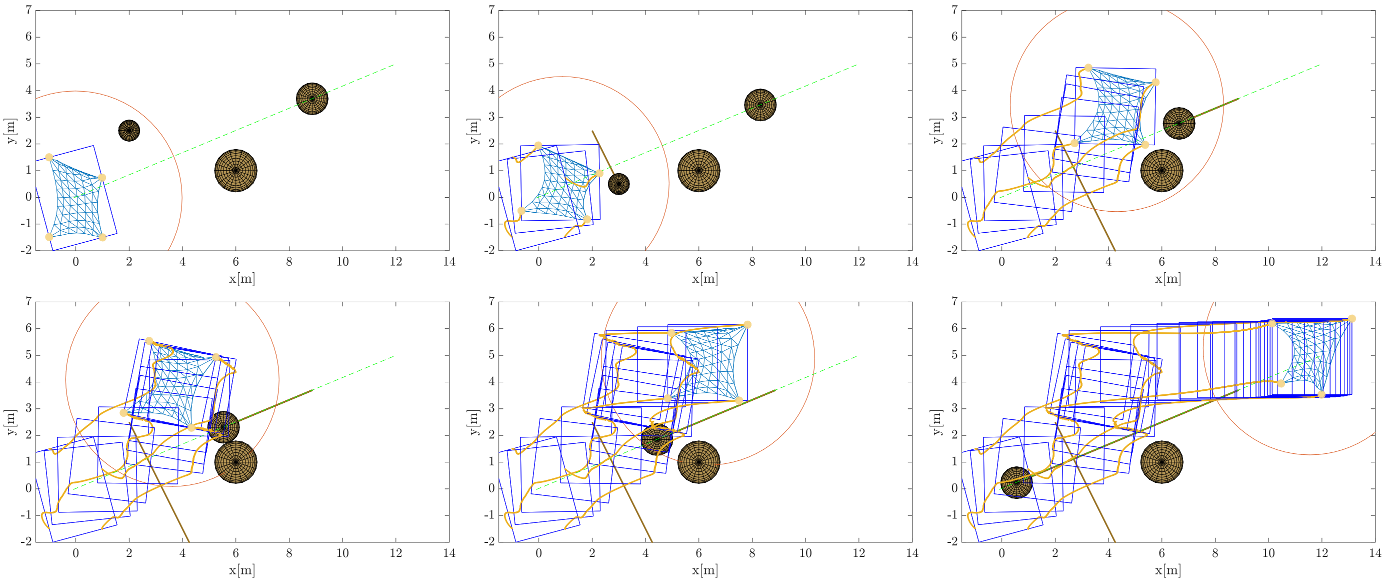

Fig. 5. From top to bottom and left to right, we show six instants (t = 0,10,40,60,80,150 [s]) of the second test. The system reaches the position [12,5] meters while avoiding collision with two dynamic and one static obstacles (see the video for full sequence).

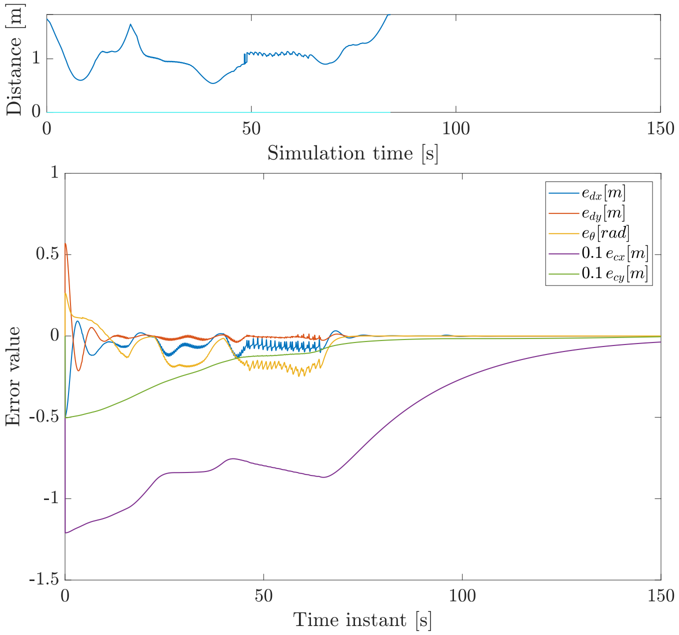

Fig. 6. At the top, absolute minimum distances between the mesh nodes and the nearest obstacle, at different time instants in the second test case. At the bottom, values of the deformation, orientation and position control errors. Note that the position error $\mathbf{e}_c$ is scaled by 0.1.

horizontal y-dimension of the BB, increasing a 20% the x-dimension and getting a 0 degrees orientation of the BB. In this case, we consider two dynamic and one static obstacles. Figure 5 shows six different snapshots, while Fig. 6 shows the absolute minimum distances between the mesh nodes and the nearest obstacle as well as the control errors. Despite the small bounces starting at 50 [s] and stopping around 70 [s], when the second obstacle approaches the system, we can see that the object reaches the desired configuration without collisions. Note that the robots stop detecting obstacles at t = 84 [s], which is the reason why the minimum distances plot shows no values from there on. The average computational time of the algorithm is 0.016 [s], which seems fast enough for a wide range of practical cases with real time response. This fact is verified with the results we show in the next section.

D. *Test with realistic conditions*

Next we tackle a transport task in a scenario with realistic conditions. The goal of the task consists in transporting a 0.40×0.55 [m] rectangular cloth-like object by its four corners to a specific position, preserving the initial orientation of the BB and expanding a 20% its x-y dimensions. We add random noise in the position of the robots, with 0.01 [m] amplitude, and the robots' linear velocities are saturated to 0.05 [m/s]. The algorithm also runs with real-time measurements. It is worth to mention that the robots follow unicycle kinematics in this case, instead of double-integrator ones. Therefore, we integrate the acceleration outputs and then transform the resulting velocities to the unicycle model by means of a diffeomorphism [31]. An agent acting as a static obstacle is placed between the initial and the goal positions. The control parameters are set in the following manner: $k_{1d} = 0.02$, $k_{2d} = 0.5$, $k_{1r} = 0.01$, $k_{2r} = 0.3$, $k_{1t} = 0.0005$, $k_{2t} = 0.06$, $\varepsilon = 5000\,h_{lj}^3$, $\alpha_l = 0.05$ [m/s^2], $D_{l1}^{min} = 0.14$ [m], $k_{w1} = k_{w2} = k_{w3} = k_{w4} = k_{w5} = 0.2$ and $R = 5$ [m]. The limit values to avoid overstretching and excessive velocities are set as $\mathbf{d}^{max} = [0.6, 0.9, 1]$ [m], $\mathbf{d}^{min} = [0.15, 0.15, 0.01]$ [m], $\dot{\theta}^{max} = -\dot{\theta}^{min} = 1.8$ [rad/s] and $\dot{c}_x^{max} = \dot{c}_y^{max} = -\dot{c}_x^{min} = -\dot{c}_y^{min} = 0.05$ [m/s]. In addition, we define $L = 152$ evenly distributed virtual points over the BB faces.

Figure 7 shows four different time instants of the simulation. We can see that the robots are able to drive the virtual deformable object from the initial position, at the bottom left side, to the final position at the top right side of the arena, avoiding the brown area (with 0.14 [m] radius) around the

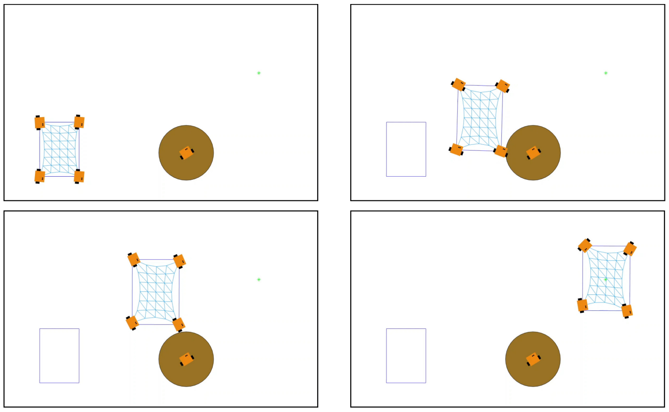

Fig. 7. From top to bottom and left to right, we show four instants (t = 0,100,150,450 [s]) of the simulation with realistic conditions. We can see that the virtual object is driven to the desired configuration while avoiding collisions with the static robot, placed between the initial and the final position.

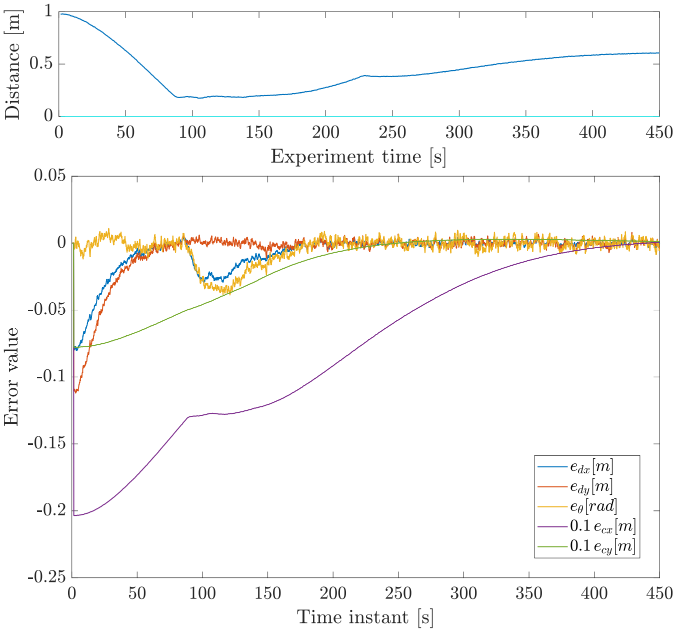

Fig. 8. At the top, absolute minimum distances between the nodes representing the object and the obstacle, at different time instants in the realistic case. At the bottom, components of the deformation, orientation and position errors. The position error $\mathbf{e}_c$ is again scaled by a 0.1 factor.

obstacle robot in the process. Figure 8 shows the minimum node-obstacles distances, always above cero, and the control errors, which tend to zero over time. These results confirm that the proposed DBB model is valid and useful for computing the dimensions, orientation and position of the BB under the control actions. Also the Q-P controller we present exploits the model successfully to achieve the control goal in all the tested cases.

A video with additional material and results can be found as supplementary material.

VI. Conclusion

We have presented a method for achieving a desired configuration of a large deformable object in terms of position, dimensions and orientation of its bounding box, without collisions. Our approach is based on a 3D deformable bounding box model for mobile robots that manipulate the object with double-integrator dynamics. It allows predicting the object evolution in shape and orientation under specific control inputs. We exploit this model to formulate a set of nominal controllers, which steer the object to the target configuration. Then, we propose a control barrier function that guarantees collision avoidance by means of an integrated quadratic-programming controller. This controller includes additional linear constraints that limit the control inputs, so that the bounding box dimensions and the translation and rotation velocities remain in the admissible ranges. Simulation results we report show that method successfully completes different kinds of transport tasks, in which the objects are steered to specific configurations in environments with static and dynamic obstacles.

With respect to the future research lines, we think that a deep learning-based strategy, trained with a complete set of cases and combined with CBFs, could be comparable to our system in terms of performance. Alternative control strategies to investigate are distributed or hybrid approaches for increasing the generality and resilience of the method in practical cases with a high number of manipulators. In particular, related works to be considered are the distributed controllers Wang *et al.* propose [21] or the distributed CBFs by Tan and Dimarogonas [32]. Concurrent design could be another interesting line of research [33]. Following the Design For Control (DFC) technique, the number of mobile manipulators, their parameters (maximum velocities, accelerations, load capacity, etc.), the grasping points and the model could be optimized in parallel to the design of the control architecture.

References

[1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics and Automation Magazine*, pp. 2–12, 2022.

[2] J. Sanchez, J. A. Corrales, B. C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[3] R. Herguedas, G. Lopez-Nicolas, R. Aragues, and C. Sagues, "Survey on multi-robot manipulation of deformable objects," in *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 977–984.

[4] J. T. Shepard and C. A. Kitts, "A multirobot control architecture for collaborative missions comprised of tightly coupled, interconnected tasks," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1435–1446, 2018.

[5] F. Wang, Z. Qian, Z. Yan, C. Yuan, and W. Zhang, "A novel resilient robot: Kinematic analysis and experimentation," *IEEE Access*, vol. 8, pp. 2885–2892, 2020.

[6] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.

[7] Z. Deng, M. Stommel, and W. Xu, "A novel soft machine table for manipulation of delicate objects inspired by caterpillar locomotion," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1702–1710, 2016.

[8] M. Aranda, J. Sanchez, J. A. Corrales Ramon, and Y. Mezouar, "Robotic Motion Coordination Based on a Geometric Deformation Measure," *IEEE Systems Journal*, vol. 16, no. 3, pp. 3689–3699, 2022.

[9] G. Lopez-Nicolas, R. Herguedas, M. Aranda, and Y. Mezouar, "Simultaneous shape control and transport with multiple robots," in *IEEE International Conference on Robotic Computing (IRC)*, 2020, pp. 218–225.

[10] Z. Cui, W. Ma, J. Lai, H. K. Chu, and Y. Guo, "Coupled multiple dynamic movement primitives generalization for deformable object manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5381–5388, 2022.

[11] D. Kruse, R. J. Radke, and J. T. Wen, "Human-robot collaborative handling of highly deformable materials," in *American Control Conference (ACC)*, 2017, pp. 1511–1516.

[12] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[13] J. Jiao, Z. Cao, N. Gu, S. Nahavandi, Y. Yang, and M. Tan, "Transportation by multiple mobile manipulators in unknown environments with obstacles," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2894–2904, 2017.

[14] Z. Cao, N. Gu, J. Jiao, S. Nahavandi, C. Zhou, and M. Tan, "A novel geometric transportation approach for multiple mobile manipulators in unknown environments," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1447–1455, 2018.

[15] R. Tallamraju, D. H. Salunkhe, S. Rajappa, A. Ahmad, K. Karlapalem, and S. V. Shah, "Motion planning for multi-mobile-manipulator payload transport systems," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1469–1474.

[16] O. Burchan Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 2126–2133.

[17] R. Buchanan, T. Bandyopadhyay, M. Bjelonic, L. Wellhausen, M. Hutter, and N. Kottege, "Walking posture adaptation for legged robot navigation in confined spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2148–2155, 2019.

[18] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5356–5363.

[19] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[20] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[21] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[22] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 1656–1661.

[23] A. Hegde and D. Ghose, "Multi-UAV collaborative transportation of payloads with obstacle avoidance," *IEEE Control Systems Letters*, vol. 6, pp. 926–931, 2022.

[24] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Collision-free transport of 2D deformable objects," in *International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 430–435.

[25] J. J. Craig, *Introduction to robotics: Mechanics and Control*, 3rd ed. Pearson Prentice Hall, 2005.

[26] J. Hollerbach, W. Khalil, and M. Gautier, "Model Identification," in *Springer Handbook of Robotics*, 1st ed., B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, ch. 14, pp. 321–344.

[27] D. Daney, Y. Papegay, and B. Madeline, "Choosing measurement poses for robot calibration with the local convergence method and tabu search," *The International Journal of Robotics Research*, vol. 24, no. 6, pp. 501–518, 2005.

[28] A. Joubair, A. Tahan, and I. Bonev, "Performances of observability indices for industrial robot calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2477–2484.

[29] T. J. Mitchell, "An algorithm for the construction of "d-optimal" experimental designs," *Technometrics*, vol. 16, no. 2, pp. 203–210, 1974.

[30] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference (ECC)*, 2019, pp. 3420–3431.

[31] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[32] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2022.

[33] Q. Li, W. Zhang, and L. Chen, "Design for control-a concurrent engineering approach for mechatronic systems design," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 2, pp. 161–169, 2001.