2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Local Motion Planning for Collaborative Multi-Robot Manipulation of Deformable Objects

Javier Alonso-Mora[1], Ross Knepper[2], Roland Siegwart[3] and Daniela Rus[1]

Abstract— **This paper presents a formalism that exploits deformability during manipulation of soft objects by robot teams. A hybrid centralized/distributed approach restricts centralized planning to high-level global guidance of the object for consensus. Low-level control is thus delegated to the individual manipulator robots, which retain manipulation and collision avoidance guarantees by passing forces to one another through the object. A distributed receding horizon planner provides local control, formulated as a convex optimization problem in velocity space and incorporating constraints for both collision avoidance and shape maintenance. We demonstrate teams of mobile manipulators autonomously carrying various deformable objects.**

I. Introduction

Conventional automated assembly robots operate affixed to the factory floor in an environment where uncertainty is managed and engineered away by careful human design. This approach has been very successful for production lines, where the majority of parts behave as rigid bodies. In the near future, agile assembly systems will incorporate mobile manipulator robots to make assembly systems adaptable to changing circumstances, including the routine manipulation of soft objects that require many hands to control.

A. Contribution

The focus of this paper is a method for collaborative manipulation of deformable objects, objects of variable shape. In this work, a geometric formulation is exploited that links the problem with that of pattern formation.

We contribute a scalable framework for collaborative carrying of deformable objects and introduce two variants of a convex optimization for local motion planning and control (one centralized, one distributed), that account for static and dynamic obstacles. In the distributed case, we rely on a system of explicit force exchange as a means of coordination.

We experimentally demonstrate up to three KUKA YouBot [1] mobile manipulators carrying various deformable objects (a video is available online).

B. Assumptions and limitations

To achieve efficient centralized and distributed optimizations, the following main assumptions are made in this work:

- a purely kinematic model of the robots and object,
- piecewise constant robot speeds,

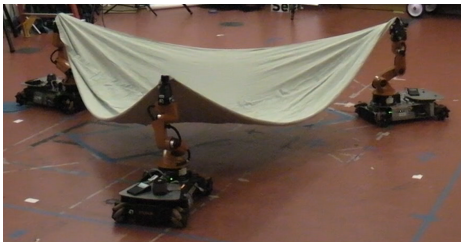


Fig. 1. Three mobile manipulators carrying a deformable bed sheet.

- treatment of the object as a reconfigurable polygon, the configuration of which is given by the gripper positions.

These assumptions are reasonable thanks to the low inertia of the carried objects and the holonomicity of the mobile platform. Furthermore, they enable a very efficient local planner that allows for high-frequency updates, in the order of 10Hz, which in turn, renders the assumptions reasonable.

II. Related works

Currently many fabrication processes, such as car manufacturing, are mostly automated with fixed manipulators, but mobile robots are steadily increasing their role in automated processes. The factory of the future will consist of mobile robots working hand in hand with humans, providing higher flexibly and making it possible to automate large scale processes, like those of the aeronautics sector [2].

One task of mobile robots is to collaboratively manipulate an object. A set of pushing protocols to cooperatively move rigid structures was first proposed by [3]. In contrast, we exploit a geometric formulation of the problem and coin it as a constraint optimization. Regarding manipulation of *rigid objects* by multiple robots, several approaches have been proposed. Either centralized: using feedback control [4] or potential fields [5]; or distributed using caging [6], force sensing [7], or leader following [8] combined with force sensing. We borrow the idea of force sensing, but consider it only when the object reaches a limit in deformation.

Centralized methods for manipulation of *flexible objects* by multiple robots include: elastodynamic equations [9], which do no consider collision avoidance, optimal control [10], where the number of variables quickly grows with the number of robots, and navigation functions [11], which are computed with respect to a fix set of obstacles. Recently, centralized methods for collaborative handling (in obstacle-free environments) of both rigid objects suspended by cables [12] and a flexible net [13] have been demonstrated with aerial vehicles. In contrast. our method can be

[1] J. Alonso-Mora and D. Rus are at CSAIL-MIT, 32 Vassar St, 02139 Cambridge MA, USA jalonsom,rus@mit.edu
[2]R. Knepper is at Cornell University, USA rak@cs.cornell.edu
[3]R. Siegwart is at ETH Zurich, Switzerland rsiegwart@ethz.ch
*This work was supported in part by ONR MURI Antidote N00014-09-1-1031, SMARTS N00014-09-1051, the Boeing Company and Disney.

distributed, accounts for static and dynamic obstacles and does not require any pre-computation with respect to the environment; albeit it applies to holonomic platforms.

Besides multi-robot navigation, manipulation of flexible objects by multiple fixed robots has also been explored [14], as well as motion planning for elastic objects within the context of medical applications [15]. Deformable objects have been extensively studied in computer graphics, where several complex methods for simulation were recently developed [16]. We retain the idea of using a representation of the object given by a triangulation, although in a much simplified form and adapted to the mobile manipulation task.

The receding horizon local planner presented in this paper builds on the concept of Velocity Obstacles (VO) [17], where the velocities leading to a collision are characterized in velocity space for omnidirectional agents, and on their recent extensions by [18] and [19]. [18] proposed the idea of approximating the VO by a linear constraint, as well as how to take into account collaboration in the avoidance. [19] presented a centralized optimization based on a similar concept and extended to generalized dynamics.

The remainder of this paper is structured as follows. Section III presents the system architecture. In Section IV the local planning approach is described. In Section V experimental results are discussed, and Section VI concludes the paper, outlining an extension to human-robot collaboration.

III. System Architecture

Throughout this paper x denotes scalars, $\mathbf{x}$ vectors, $||\mathbf{x}||$ its Euclidean norm, $\hat{\mathbf{x}}$ its direction, X matrices, and $\mathcal{X}$ sets.

We architected a system to perform distributed manipulation of deformable objects. Specifically, the task is to transport an object $\mathcal{B} \subset \mathbb{R}^2$ from its initial configuration $\mathcal{B}_0$ to a final configuration $\mathcal{B}_F$ while respecting its deformation limits and avoiding collisions with static obstacles $\mathcal{O} \subset \mathbb{R}^2$ and other agents of radii r_i, for $i \in \mathcal{A} = \{1, \dots, n\}$. In this section, we provide a high-level overview of agent and object representations as well as the hierarchy of motion planners that constitutes our distributed planning framework.

A. Agents

Two types of agents interact in the course of the task:

1) *Manipulator* ($\mathcal{M}$) robots transport the object.
2) *Reacting agents* ($\mathcal{R}$) are dynamic obstacles that do not assist in object transport.

Let $\mathcal{A} = \mathcal{M} \cup \mathcal{R}$. Typical manipulators, like the KUKA YouBot [1] are omnidirectional platforms. We consider m omnidirectional manipulators, with radii r_i, position $\mathbf{p}_i \in \mathbb{R}^2$ and velocity $\mathbf{v}_i = \dot{\mathbf{p}}_i$, for $i \in \mathcal{M} = \{1, \dots, m\}$, collaboratively carrying the deformable object $\mathcal{B}$ and with input velocity $\mathbf{u}_i$. Limits $||\mathbf{u}_i|| < u_{max}$ are enforced. Each manipulator is modeled by a mobile platform and an extendable arm, denoting by $\breve{\mathbf{p}}$, $\breve{\mathbf{v}}$ and $\breve{\mathbf{u}}$ the position, velocity and input velocity for the gripper. *The upper "gripper-like" symbol $\breve{}$ denotes variables for the gripper.* Fig. 2 shows two possible models. For simplicity, we represent the manipulators and agents with a circular footprint, but the results readily extend to robots of arbitrary shape with the assumption of constant orientation within the short time horizon of the local planner. The disk of radius r_i centered at $\mathbf{c}$ is denoted by $D_{\mathbf{c},r_i} \subset \mathbb{R}^2$. In practice r_i is equal to the maximum arm length $\bar{a}$.

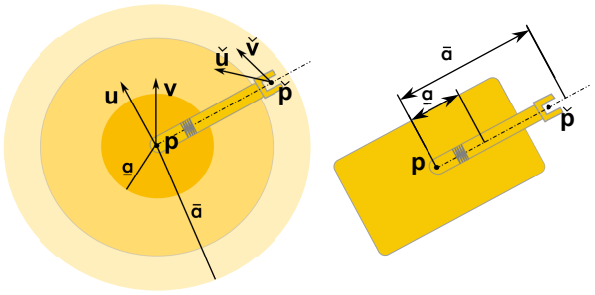


Fig. 2. Equivalent schemas of the mobile manipulator with extensible arm (minimum $\underline{a}$ and maximum $\bar{a}$ length) and its circular model. The orientation of the robot equals that of the arm.

B. Object Parametrization

Each manipulator computes the configuration of the object based on observations of other manipulators' gripper locations, such that $\kappa(\breve{\mathbf{p}}_1, \dots, \breve{\mathbf{p}}_m) \in \mathbb{R}^{2m} \mapsto \mathcal{B} \subset \mathbb{R}^2$. In its simplest version the vertexes equal the grippers' positions, but the definition is general and the object may have an arbitrary shape. See Fig. 3 for a representative schema.

The valid configurations of the object are described by the inter-manipulator distances to neighbors. In particular, the total number of neighbors needed to fully define the shape of the object is given by the edges of the Delaunay triangulation of the manipulator positions. The number of edges is thereby reduced from $O(n^2)$ to $O(n)$.

***Definition** 1 (Object shape):* The shape of object $\mathcal{B}$ is defined by a Delaunay triangulation of the m grippers' positions $\breve{\mathbf{p}}_l$, $l \in \mathcal{M}$. Edges in the triangulation define a manipulator's *neighbors*. Constraints on the minimum $\underline{d}_i^j$ and maximum $\bar{d}_i^j$ edge distance between manipulators i and j define allowable shape deformations. For $m = 2$ the valid object configurations are given by $\underline{d}_1^2 \leq ||\breve{\mathbf{p}}_1 - \breve{\mathbf{p}}_2|| \leq \bar{d}_1^2$.

The shape of a triangle is uniquely defined by its edge distances and therefore a shape is defined by its triangulation. The Delaunay triangulation is chosen as it maximizes the minimum angle of the triangles, minimizing sensitivity to robot motion. Fig. 3 shows a schema with maximum and minimum configurations.

In the case of a rigid object, its position $\mathbf{p}_B$, orientation θ_B, linear $\mathbf{v}_B$ and angular ω_B velocities are uniquely defined. For the case of a deformable object we define them as the mean of grippers' position $\breve{\mathbf{p}}_i$ and velocity $\breve{\mathbf{v}}_i$,

$$\begin{aligned} \mathbf{p}_B &= \textstyle\sum_{i\in\mathcal{M}} \frac{\breve{\mathbf{p}}_i}{m}, & \theta_B &= \theta_B^0 + \textstyle\sum_{i\in\mathcal{M}} \frac{\widehat{\breve{\mathbf{p}}_i - \mathbf{p}_B} - \theta_{iB}^0}{m}, \\ \mathbf{v}_B &= \textstyle\sum_{i\in\mathcal{M}} \frac{\breve{\mathbf{v}}_i}{m}, & \omega_B &= \textstyle\sum_{i\in\mathcal{M}} \frac{(\breve{\mathbf{v}}_i - \mathbf{v}_B)\cdot\mathbf{n}_{iB}}{m||\breve{\mathbf{p}}_i - \mathbf{p}_B||}, \end{aligned} \tag{1}$$

where θ_B^0 is the reference orientation of the object, θ_{iB}^0 the reference angles $\widehat{\breve{\mathbf{p}}_i^0 - \mathbf{p}_B^0}$, $[\mathbf{n}_{iB}, 0] = [0, 0, 1] \times [\mathbf{t}_{iB}, 0]$ and $\mathbf{t}_{iB} = \breve{\mathbf{p}}_i - \mathbf{p}_B$. The mean of angles is given by the angle of the mean of unit vectors.

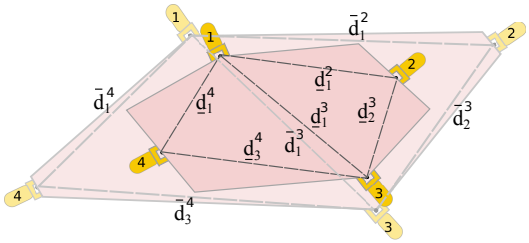


Fig. 3. Schema of a deformable object with minimum and maximum size triangulations with respect to the grasping points. Edges have upper ($\bar{d}_i^j$) and lower bounds ($\underline{d}_i^j$).

We define the *expansion factor* of the object by $e_B = \sum_{i \in \mathcal{M}} \frac{(\check{\mathbf{v}}_i - \mathbf{v}_B) \cdot \mathbf{t}_{iB}}{m ||\check{\mathbf{p}}_i - \mathbf{p}_B||}$, the average of the variation in distance between each manipulator and the instantaneous center of the object. Alternatively, it might be beneficial in some cases to use the expansion of the neighbors' edges.

C. Global Motion Planner

Recognizing that the individual manipulators do not directly communicate, a guarantee on coordination among manipulators requires that a common approximate global plan for the object is available to all manipulators. The global motion planner provides guidance towards the ultimate goal in the form of a trajectory. The need for consensus arises primarily from a desire to ensure that all manipulators select trajectories from the same homotopy class. We implemented the global planner using the off-the-shelf OMPL [20] library and configured it to replan periodically.

D. Local Motion Planner

An instance of the local motion planner runs on each manipulator and receives goals from the global motion planner. The local planner (Section IV) commands the linear and angular velocities of the manipulators and guarantees shape maintenance and collision avoidance.

E. Communication

The system architecture is designed for scalability, in order to support carrying large, complex objects with many robots. Therefore, low-throughput message passing is restricted to be linear in the number of agents, whereas high-rate observations are at constant scale per manipulator on average. Messages come in a variety of forms, detailed below.

1) Global-to-Local Communication: The centralized global planner issues global guidance to all the manipulator robots at a rate of 0.1 Hz. The message comprises a trajectory for a simplified representation of the object, comprising a series of waypoints. Each waypoint contains the following state information for the object: position, orientation, and a scalar *expansion factor*. The manipulators store the trajectory and move towards each goal in sequence.

2) Inter-Agent Communication: Manipulators do not directly exchange messages. Instead, they observe the position and velocity of their neighbors.

In addition, manipulators indirectly pass forces among each other in the form of a resultant force exerted upon the deformable object. Since the object deforms in response to these forces, a non-zero resultant occurs only among robots that are at a limit defined by one manipulation constraint.

Agents are ignorant of one another's planned trajectories A manipulator observes each agent's position and velocity. As future trajectories are unknown, the manipulator applies the constant velocity assumption to the agent for a given time horizon, and then predicts its behavior according to the agent's class membership.

3) Force communication: For distributed manipulation, forces transmitted through the object are required for showing intent and for coordination. Manipulators exert forces on the object when it reaches its limits[1]. Manipulator i may exert force $\bar{\mathbf{f}}_i$ applied at its gripping position $\check{\mathbf{p}}_i$.

Manipulators are controlled in velocity. For a free-moving vehicle we assume that its velocity $\mathbf{v}_i$ equals its control input $\mathbf{u}_i$ (and $\check{\mathbf{v}}_i = \check{\mathbf{u}}_i$). For a manipulator grasping an object carried jointly with other manipulators, its velocity may not equal its control input due to external constraints. In this case a force is transmitted to the object related to the difference between the commanded velocity and the feasible velocity[2]

$$\bar{\mathbf{f}}_i = f(\check{\mathbf{u}}_i - \check{\mathbf{v}}_i). \tag{2}$$

We assume that each manipulator knows its own exerted force $\bar{\mathbf{f}}_i$ and can sense (at the gripping position $\check{\mathbf{p}}_i$) the resultant force $\underline{\mathbf{f}}_i$ and moment $\underline{\mathbf{m}}_i$ of the forces exerted by all other agents.

To illustrate the concept, consider the ordinary case of two humans jointly carrying a rigid object. While moving the object, a change in the direction by person A – such as to avoid an obstacle – will create a tension on the object that can be sensed by person B. Person B would then be able to react accordingly by adopting a velocity compatible with the sensed force. On the other hand, when carrying a deformable object such as a towel, forces are only sensed when the object reaches its upper limit in shape.

IV. Receding Horizon Local Motion Planner

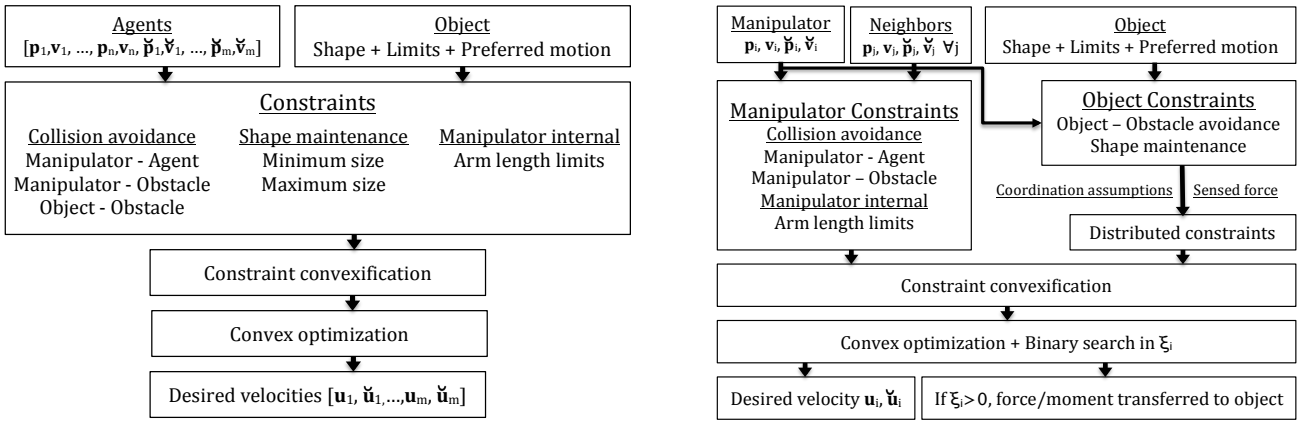
Given an approximated global plan for the manipulated object, shared by all the manipulators, the preferred linear velocity $\bar{\mathbf{v}}_B$, angular velocity $\bar{\omega}_B$ and preferred expansion $\bar{e}_B$ of the object are computed with a proportional controller to guide the object to the state at the next waypoint.

In each time step, given the current position and velocity of all manipulators and agents, a new velocity is computed by solving an optimization problem in velocity space. In this section we formulate it as a convex optimization with quadratic cost and linear and quadratic constraints. See Figure 4 for an schema.

We distinguish between two types of constraints, for collision avoidance and for shape maintenance. In the distributed case, the shape maintenance constraints are treated

[1]Forces arising from a spring model can also be added.

[2]The velocity of the robot wheels is typically controlled via current. If the wheel does not turn at the expected rate, a torque is created in the motor, which is transmitted as a force to the grabbed object. If a PI controller is used, this force is modeled by a first order system with saturation.



(a) Centralized (b) Distributed

Fig. 4. Schemas of the centralized and distributed algorithms, including inputs, constraints and outputs..

as soft constraints that might be relaxed in order to avoid collisions. To maintain shape, the resulting velocity would be partially transformed into a force, transferred to the object and sensed by the other manipulators (see Section III-E.3), thus indicating the unfeasibility. In this section, we detail the optimization costs, constraints, algorithms, and guarantees.

Throughout this section recall that the upper "gripper-like" symbol $\breve{}$ denotes variables for the gripper and the lower $_B$ identifies variables for the center of the object.

A. Optimization cost

The optimization cost is given by several terms, with user-defined weights K_*. The first and second terms penalize changes in manipulator and gripper velocity, the third penalizes relative velocities between the gripper and the center of the manipulator and the remaining terms penalize the deviation to the preferred motion parameters of the object.

For the *distributed* case, the cost $\mathcal{C}(\mathbf{u}_i, \breve{\mathbf{u}}_i)$ is given by

$$K_0||\mathbf{u}_i - \mathbf{v}_i||^2 + K_1||\breve{\mathbf{u}}_i - \breve{\mathbf{v}}_i||^2 + K_2||\breve{\mathbf{u}}_i - \mathbf{u}_i||^2 + ||\breve{\mathbf{u}}_i - \bar{\mathbf{v}}_i||^2,$$

where

$$\bar{\mathbf{v}}_i = \bar{\mathbf{v}}_B + \bar{\omega}_B||\breve{\mathbf{p}}_i + \mathbf{p}_B||\mathbf{n}_{iB} + \bar{e}_B||\breve{\mathbf{p}}_i - \mathbf{p}_B||\mathbf{t}_{iB} + \mathbf{v}_{i,rep}$$

is the preferred velocity for the gripper given the desired linear $\bar{\mathbf{v}}_B$ and angular $\bar{\omega}_B$ velocities and expansion $\bar{e}_B$ of the object. The term $\mathbf{v}_{i,rep}$ is the total repulsive velocity for manipulator i with respect to neighboring agents in $\mathcal{A}$, computed as in [19] and similar to a repulsive potential field.

For the *centralized* case, with $\mathbf{u}_{1:m} = [\mathbf{u}_1, ..., \mathbf{u}_m]$ and recalling Equation 1 that relates the gripper velocities with the motion of the object, the cost $\mathcal{C}(\mathbf{u}_{1:m}, \breve{\mathbf{u}}_{1:m})$ is given by

$$K_0||\mathbf{u}_{1:m} - \mathbf{v}_{1:m}||^2 + K_1||\breve{\mathbf{u}}_{1:m} - \breve{\mathbf{v}}_{1:m}||^2 + K_2||\breve{\mathbf{u}}_{1:m} - \mathbf{u}_{1:m}||^2 + \left\|\bar{\mathbf{v}}_B - \sum_{i\in\mathcal{M}} \frac{\breve{\mathbf{u}}_i}{m}\right\|^2 + \left|\bar{\omega}_B - \sum_{i\in\mathcal{M}} \frac{(\breve{\mathbf{u}}_i - \mathbf{v}_B)\cdot\mathbf{n}_{iB}}{m||\breve{\mathbf{p}}_i - \mathbf{p}_B||}\right|^2 + \left|\bar{e}_B - \sum_{i\in\mathcal{M}} \frac{(\breve{\mathbf{u}}_i - \mathbf{v}_B)\cdot\mathbf{t}_{iB}}{m||\breve{\mathbf{p}}_i - \mathbf{p}_B||}\right|^2,$$

B. Shape maintenance constraints

Given a Delaunay triangulation of the manipulated object and its limits (see Section III-B), the necessary constraints for shape maintenance are defined in this section. Following the receding horizon approach, manipulators are considered to move at constant speed for a given time horizon τ_s for which the constraints must be satisfied (the shape of the object must remain within its limits). These constraints are formulated with respect to the gripper (tip of the manipulator's arm).

The change in velocity for the gripper of manipulator $i \in \mathcal{M}$ is denoted by $\Delta\breve{\mathbf{v}}_i = \breve{\mathbf{u}}_i - \breve{\mathbf{v}}_i$. The force $\underline{\mathbf{f}}_i$ and moment $\underline{\mathbf{m}}_i$ sensed by manipulator i provide an indication of the resultant change in velocity that the other manipulators require. Recalling Eq. (2), the required change in velocity is given by[3] $\Delta\mathbf{v}^F_{i|j} = f^{-1}(\underline{\mathbf{f}}_i + \underline{\mathbf{m}}_i(\breve{\mathbf{p}}_j - \mathbf{p}_B))$.

In the *distributed* case, each manipulator maintains a convention about the change in velocity of the other manipulators. To globally maintain the constraint satisfaction, for every pair of neighbors i, j, the conservative convention is that the change in relative velocity equals twice the change of velocity of each manipulator[4] minus the term $\Delta\mathbf{v}^F_{i|j}$, which is expected to be executed by both agents. This leads to

$$\Delta\breve{\mathbf{v}}_{ij} = 2(\Delta\breve{\mathbf{v}}_i - \Delta\mathbf{v}^F_{i|j}). \tag{3}$$

When distributed, the optimization can become overconstrained. In that case collision avoidance constraints are treated as hard constraints (that must always be satisfied) and shape constraints as soft constraints. The later are relaxed by $\xi_i \geq 0$, representing the amount by which the minimum/maximum inter-agent distances are reduced/increased.

***Constraint** 1 (Minimum neighbor distances):* For every pair of neighbor manipulators $i, j \in \mathcal{M}$ connected by an edge of the object triangulation, the gripper velocities must satisfy $\|(\breve{\mathbf{p}}_i + \breve{\mathbf{u}}_i t) - (\breve{\mathbf{p}}_j + \breve{\mathbf{u}}_j t)\| \geq \underline{\mathrm{d}}^j_i - \xi_i$, for all $t \in [0, \tau_s]$.

For each $t \in [0, \tau_s]$, the set of infeasible relative velocities $\breve{\mathbf{u}}_{ij} = \breve{\mathbf{u}}_i - \breve{\mathbf{u}}_j$ is given by a circle of radius $(\underline{\mathrm{d}}^j_i - \xi_i)/t$

[3]Our experiments use a simplistic model. We assume $f(\Delta\mathbf{v}) = \mathbf{f}$ and $f^{-1}(\mathbf{f}) = \Delta\mathbf{v}/2$. The factor is added to avoid over reaction and oscillations.

[4]Other conventions are possible, such as constant neighbor's velocity.

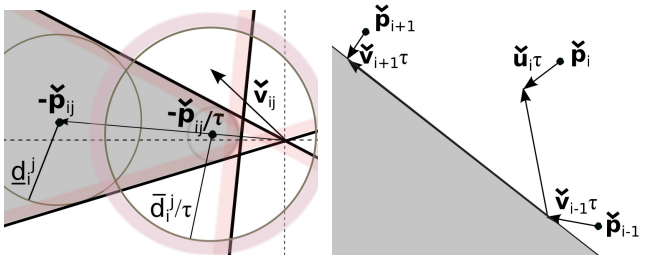


Fig. 5. Manipulation constraints for two manipulators. *Left:* In relative velocity space: quadratic constraint (circle, pink - infeasible side) for maximum inter-manipulator distance (Constraint 2) and non-convex constraint (grey - infeasible) for minimum distance (Constraint 1), approximated by three linear constraints. *Right:* In grey, convex shape constraint (Constraint 3) of infeasible positions $\breve{\mathbf{p}}_i + \breve{\mathbf{u}}_i \tau_s$.

centered at $-\breve{\mathbf{p}}_{ij}/t = -(\breve{\mathbf{p}}_i - \breve{\mathbf{p}}_j)/t$. Their union defines a cone, in grey in Fig. 5-left. Its complement is non-convex. We first approximate it by three excluding linear constraints of the form $\mathbf{n} \cdot \breve{\mathbf{u}}_{ij} \leq b$ and given by (see Fig. 5-left)

$$\begin{bmatrix} \cos(\alpha \pm \beta) \\ \sin(\alpha \pm \beta) \end{bmatrix} \cdot \breve{\mathbf{u}}_{ij} \leq 0, \quad -\frac{\breve{\mathbf{p}}_{ij}}{\breve{p}_{ij}} \cdot \breve{\mathbf{u}}_{ij} \leq \frac{\breve{p}_{ij} - \underline{d}_i^j + \xi_i}{\tau_s}, \quad (4)$$

where $\breve{p}_{ij} = ||\breve{\mathbf{p}}_{ij}||$, $\alpha = \text{atan2}(-\breve{\mathbf{p}}_{ij})$ and $\beta = \text{acos}((\underline{d}_i^j - \xi_i)/\breve{p}_{ij})$. And second, by selecting the linear constraint with maximum constraint satisfaction for the current relative velocity, given by $\underset{\mathbf{n},b}{\text{argmin}}(\mathbf{n} \cdot \breve{\mathbf{v}}_{ij} - b)$.

For the *distributed* optimization, by substitution of $\breve{\mathbf{u}}_i = \breve{\mathbf{v}}_i + \Delta\breve{\mathbf{v}}_i$ and recalling Eq. (3), the constraint is given by

$$\begin{aligned} &\mathbf{n} \cdot \breve{\mathbf{u}}_{ij} - b = \mathbf{n} \cdot (\breve{\mathbf{v}}_i - \breve{\mathbf{v}}_j + \Delta\breve{\mathbf{v}}_{ij}) - b = \\ &\mathbf{n} \cdot (\breve{\mathbf{v}}_i - \breve{\mathbf{v}}_j + 2\Delta\breve{\mathbf{v}}_i - 2\Delta\mathbf{v}^F_{i|j}) - b = \\ &\mathbf{n} \cdot \breve{\mathbf{u}}_i + (\mathbf{n} \cdot (-\breve{\mathbf{v}}_i - \breve{\mathbf{v}}_j - 2\Delta\mathbf{v}^F_{i|j}) - b) \leq 0, \end{aligned} \quad (5)$$

***Constraint** 2 (Maximum neighbor distances):* For every pair of neighbor manipulators $i, j \in \mathcal{M}$ connected by an edge of the object triangulation, the gripper velocities must satisfy $\|(\breve{\mathbf{p}}_i + \breve{\mathbf{u}}_i t) - (\breve{\mathbf{p}}_j + \breve{\mathbf{u}}_j t)\|^2 \leq (\bar{d}_i^j + \xi_i)^2$, for all $t \in [0, \tau_s]$.

This constraint can be seen to be quadratic and convex in its *centralized* formulation. For the *distributed* case, by substitution of $\breve{\mathbf{u}}_j = \breve{\mathbf{v}}_j + \Delta\breve{\mathbf{v}}_j$ and recalling Eq. (3) the constraint is

$$\breve{\mathbf{u}}_i^T I \breve{\mathbf{u}}_i + \mathbf{l}^T \breve{\mathbf{u}}_i \leq ((\bar{d}_i^j + \xi_i)/\tau_s)^2 - \mathbf{l}^T\mathbf{l})/4, \quad (6)$$

where $\mathbf{l} = (\breve{\mathbf{p}}_i - \breve{\mathbf{p}}_j)/\tau_s - \breve{\mathbf{v}}_i - \breve{\mathbf{v}}_j - 2\Delta\mathbf{v}^F_{i|j}$ and I is the 2x2 identity matrix.

***Constraint** 3 (Convex polygonal shape):* If a convex shape is required, the future position of manipulator $i \in \mathcal{M}$ must guarantee $[0, 0, 1] \cdot ((\breve{\mathbf{p}}_{i(i-1)} + \breve{\mathbf{u}}_{i(i-1)}\tau_s) \times (\breve{\mathbf{p}}_{(i+1)(i-1)} + \breve{\mathbf{u}}_{(i+1)(i-1)}\tau_s)) \geq 0$, as shown in Fig. 5-right..

The constraint is rewritten as $(\breve{\mathbf{p}}_{i(i-1)} + \breve{\mathbf{u}}_{i(i-1)}\tau_s)^T(\breve{\mathbf{p}}_{(i+1)(i-1)} + \breve{\mathbf{u}}_{(i+1)(i-1)}\tau_s)^\perp \geq 0$, where $\mathbf{x}^\perp$ denotes the perpendicular vector of $\mathbf{x}$ given by $[\mathbf{x}^\perp, 0] = [0, 0, 1] \times [\mathbf{x}, 0]$ and linearized with the approximations $\breve{\mathbf{u}}_{i+1} = \breve{\mathbf{v}}_{i+1}$ and $\breve{\mathbf{u}}_{i-1} = \breve{\mathbf{v}}_{i-1}$,

$$\mathbf{n}_c \cdot \breve{\mathbf{u}}_i \leq \mathbf{n}_c \cdot (\breve{\mathbf{p}}_{i(i-1)}/\tau_s - \breve{\mathbf{v}}_{i-1}), \quad (7)$$

where $\mathbf{n}_c = -(\breve{\mathbf{p}}_{(i+1)(i-1)} + \breve{\mathbf{v}}_{(i+1)(i-1)}\tau_s)^\perp$. This constraint is optional.

C. Collision avoidance (CA) constraints

Each manipulator must avoid colliding with static obstacles, other manipulators and dynamic obstacles. Furthermore, the motion of each manipulator shall be such that collisions between the carried object and static and dynamic obstacles are avoided. Collision-free motion shall be guaranteed for a maximum time horizon τ_c [5].

Non-convex collision avoidance constraints are approximated by three linear constraints, where the first and last represent avoidance to the right and to the left of the obstacle and the middle one represents a head-on maneuver which remains collision-free up to $t = \tau_c$. Only one of the linear constraints is selected and added to the convex optimization as will be discussed in Remark 1.

***Constraint** 4 (CA manipulator - static obstacle):* For every manipulator $i \in \mathcal{M}$ and neighboring obstacle $\mathcal{O}$, the constraint is given by the future positions in collision with the manipulator enclosing disk, $\mathbf{p}_i + \mathbf{u}_i t \notin \mathcal{O} \oplus D_{\mathbf{0},r_i}$, $\forall t \in [0, \tau_c]$. Here, $\oplus$ designates the Minkowski sum. The method permits arbitrary compact obstacles, but for simplicity we consider $\mathcal{O}$ convex. We approximate the non-convex constraint by three linear constraints as

$$\begin{aligned} \left[\cos(\beta_- + \tfrac{\pi}{2}), \quad \sin(\beta_- + \tfrac{\pi}{2})\right] \cdot \mathbf{u}_i &\leq 0, \\ \left[\cos(\tfrac{\beta_- + \beta_+}{2}), \quad \sin(\tfrac{\beta_- + \beta_+}{2})\right] \cdot \mathbf{u}_i &\leq \tilde{d}/\tau_c, \\ \left[\cos(\beta_+ - \tfrac{\pi}{2}), \quad \sin(\beta_+ - \tfrac{\pi}{2})\right] \cdot \mathbf{u}_i &\leq 0, \end{aligned} \quad (8)$$

where β_- and β_+ are the angles of the tangents to both sides of the cone (see Fig. 6-left) and $\tilde{d}$ the projected distance from $\mathbf{p}_i$ to the enlarged obstacle $\mathcal{O} \oplus D_{\mathbf{0},r_i}$ onto the axis of the velocity cone. One of the three linear constraints is selected as will be discussed in Remark 1 and introduced in the convex optimization.

Boundary wall constraints can be directly added, given by $\mathbf{n}_{wall} \cdot \mathbf{u}_i \leq d(\text{wall}, \mathbf{p}_i)/\tau_c$, with $\mathbf{n}_{wall}$ the normal vector to the wall and $d(\text{wall}, \mathbf{p}_i)$ the distance to it.

***Constraint** 5 (CA manipulator - manipulator/agent):* For manipulator $i \in \mathcal{M}$ and agent $j \in \mathcal{A}$, the constraint is given by $\|(\mathbf{p}_i + \mathbf{u}_i t) - (\mathbf{p}_j + \mathbf{u}_j t)\| \geq r_i + r_j$, for all $t \in [0, \tau_c]$.

This constraint is equivalent in form to Constraint 1 and can be approximated by the three linear constraints equivalent to those of Eq. (4), where $r_i + r_j$ is substituted in place of $\underline{d}_i^j - \xi_i$.

For manipulator $j \in \mathcal{M}$ the linear constraint $\{\mathbf{n} \cdot \mathbf{u}_{ij} \leq b\}$ with maximum constraint satisfaction for the current relative velocity $\mathbf{v}_{ij}$ is selected. For the *distributed* optimization, both manipulators take equal effort in avoiding the collision, $\Delta\mathbf{v}_i = -\Delta\mathbf{v}_j = \Delta\mathbf{v}_{ij}/2$, leading to

$$\mathbf{n} \cdot \mathbf{u}_{ij} \leq b \Rightarrow \mathbf{n} \cdot \mathbf{u}_i \leq b/2 + \mathbf{n} \cdot (\mathbf{v}_i + \mathbf{v}_j)/2. \quad (9)$$

For agent $j \in \mathcal{R}$ different options exist to select one of the three linear constraints, as will be discussed in Remark 1.

[5] Since the object shape can be modified quickly (10Hz control loop), we select in our experiments 4 s $= \tau_c > \tau_s = 0.5$ s.

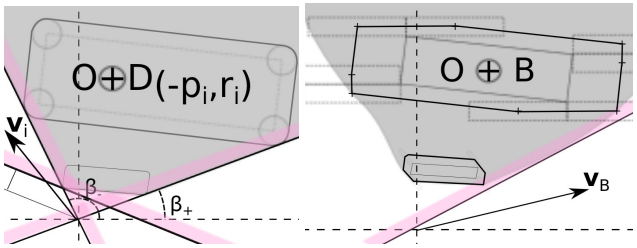


Fig. 6. Collision avoidance constraints. *Left:* In manipulator i's velocity space, constraint 4 (grey - infeasible) for avoidance of a rectangular obstacle and its approximation by three linear constraints (pink - infeasible). *Right:* In object's velocity space, constraint 6 (grey - infeasible) for avoidance of a rectangular obstacle for a rectangular object with non-zero angular velocity and its linearization with respect to the current velocity of the object.

For collaborative agents the constraint reduces to Eq. (9) and for non-collaborative ones to $\mathbf{n} \cdot \mathbf{u}_i \leq b + \mathbf{n} \cdot \mathbf{v}_j$.

***Constraint** 6 (CA manipulated object):* Let $\mathbf{u}_B = \sum_{i \in \mathcal{M}} \breve{\mathbf{u}}_i / m$ and $\mathcal{B}_{\mathbf{x},\theta} \subset \mathbb{R}^2$ the object $\mathcal{B}$ at $\mathbf{p}_B = \mathbf{x}$ and with orientation $\theta_B = \theta$. For each proximal static obstacle $\mathcal{O}$, the constraint is given by $\{\mathbf{u}_B | \mathcal{O} \cap \mathcal{B}_{\mathbf{p}_B + \mathbf{u}_B t, \theta_B + \omega_B t} = \emptyset, \forall t \in [0, \tau_c]\}$, and for each nearby agent $j \in \mathcal{A}$ by $\{\mathbf{u}_B | D_{\mathbf{p}_j + \mathbf{v}_j t, r_j} \cap \mathcal{B}_{\mathbf{p}_B + \mathbf{u}_B t, \theta_B + \omega_B t} = \emptyset, \forall t \in [0, \tau_c]\}$.

An example of this constraint for the case of a static rectangular obstacle and a rotating rectangular object is shown in Figure 6-right. The constraint is linearized with respect to the current velocity $\mathbf{v}_B$ of the object. If $C \subset \mathbb{R}^2$ denotes the infeasible velocities $\mathbf{u}_B$, the idea is as follows: points of C are obtained by sampling $t \in [0, \tau_c]$, the convex hull of C is then computed, followed by the half-plane $H = \{\mathbf{u}_B \,|\, \mathbf{n} \cdot \mathbf{u}_B \leq b\}$ that satisfies $C \cap H = \emptyset$ and minimizes $\mathbf{n} \cdot \mathbf{v}_B - b$.

This linear constraint can be added in the *centralized* optimization. In the *distributed* case the hypothesis is made that the change in velocity of the object equals the change in velocity of each manipulator[6], $\Delta \mathbf{v}_i = \Delta \mathbf{v}_B$, leading to

$$\mathbf{n} \cdot \mathbf{u}_B \leq b \;\Rightarrow\; \mathbf{n} \cdot \mathbf{u}_i \leq b - \mathbf{n} \cdot (\mathbf{v}_B - \mathbf{v}_i). \tag{10}$$

***Remark** 1 (Linearization of manipulators' CA constraints):* Constraints 4 and 5 are approximated by three linear constraints, where only one must be selected. Sensible choices include: predefined side for avoidance (left or right), maximize constraint satisfaction with respect to the current relative velocity $[\min(\mathbf{n} \cdot (\mathbf{v}_i - \mathbf{v}_j) - b)]$ or with respect to the preferred velocity $[\min(\mathbf{n} \cdot (\bar{\mathbf{u}}_i - \mathbf{v}_j) - b)]$, where $\mathbf{v}_j$ is the current velocity of the obstacle. As different linearization options will lead to different local minima, we opt for the last option which takes into account the preferred motion of the manipulated object (recall Sec. IV-A), common to all manipulators.

D. Arm length constraints

***Constraint** 7 (Minimum arm length):* For each manipulator it is given by $\|\breve{\mathbf{p}}_i - \mathbf{p}_i + (\breve{\mathbf{u}}_i - \mathbf{u}_i)t\| \geq \underline{a}$, for all $t \in [0, \tau_s]$.

Non-convex constraint, linearized like Constraint 1.

[6]To account for differences in velocity between manipulators, for example due to a non-zero angular velocity of the object, we do not assume $\mathbf{u}_B = \mathbf{u}_i$.

***Constraint** 8 (Maximum arm legth):* For each manipulator the constraint is given by $\|\breve{\mathbf{p}}_i - \mathbf{p}_i + (\breve{\mathbf{u}}_i - \mathbf{u}_i)t\| \leq \bar{a}$.

Quadratic constraint, derived like Constraint 2.

E. Algortihm

***Algorithm** 1 (Centralized optimization):* A single convex optimization, Fig. 4(a), is solved where the new velocity commands of all manipulators $\mathbf{u}_{1:m} = [\mathbf{u}_1, ..., \mathbf{u}_m]$ and grippers $\breve{\mathbf{u}}_{1:m} = [\breve{\mathbf{u}}_1, ..., \breve{\mathbf{u}}_m]$ are jointly computed. Force exchange is not required.

$$\begin{aligned} \underset{\mathbf{u}_{1:m}, \breve{\mathbf{u}}_{1:m}}{\arg\min} \quad & \mathcal{C}(\mathbf{u}_{1:m}, \breve{\mathbf{u}}_{1:m}) \\ s.t. \quad & ||\mathbf{u}_i|| \leq u_{max}, \, ||\breve{\mathbf{u}}_i|| \leq u_{max} \quad \forall i \in \mathcal{M}, \\ & \text{linear constraints 1, 3–7,} \\ & \text{convex quadratic constraints 2 and 8.} \end{aligned} \tag{11}$$

Here, all constraints are written in their centralized form, for all neighbors' edges and all manipulators, resulting in a $4m$-dimensional convex optimization.

***Algorithm** 2 (Distributed optimization):* Each manipulator $i \in \mathcal{M}$ independently solves an optimization, Fig. 4(b), where its new velocity commands $\mathbf{u}_i$ and $\breve{\mathbf{u}}_i$ are computed. The variable ξ_i is added to the optimization and represents the amount the manipulation constraints are relaxed to render the optimization feasible.

$$\begin{aligned} \underset{\mathbf{u}_i, \breve{\mathbf{u}}_i, \xi_i}{\arg\min} \quad & \mathcal{C}(\mathbf{u}_i, \breve{\mathbf{u}}_i) + \xi_i \\ s.t. \quad & ||\mathbf{u}_i|| \leq u_{max}, ||\breve{\mathbf{u}}_i|| \leq u_{max}, \\ & 0 \leq \xi_i \leq u_{max}, \\ & \text{soft linear constraint 1 and 3,} \\ & \text{soft convex quadratic constraint 2,} \\ & \text{hard linear constraints 4–7,} \\ & \text{hard convex quadratic constraint 8,} \end{aligned} \tag{12}$$

Here, all constraints are written in their distributed form for manipulator i. For fixed ξ_i, a convex optimization is obtained. The variable ξ_i takes the minimum value that renders the optimization feasible and is obtained through an iterative search on the parameter ξ_i, by sequentially solving a 4-dimensional quadratic optimization with linear and quadratic constraints.

F. Theoretical guarantees

Assumptions. The following assumptions are made: errors in sensing of positions, velocities, and forces are negligible; agents in $\mathcal{R}$ maintain a constant velocity during the local planning interval; and the object maintains a constant angular velocity and shape for a short time horizon $t \leq \tau_c$. These assumptions are reasonable thanks to the fast replanning cycle and the short time horizon employed.

Centralized algorithm guarantees. *By construction, if Algorithm 1 is feasible and the assumptions hold, the commands $\boldsymbol{u}_{1:m}$ guarantee that the shape of the object is maintained within the limits up to time τ_s and that the motions are collision-free up to time τ_c.*

Distributed algorithm guarantees. *Collision-free motion and object shape maintenance can be guaranteed with additional assumptions.*

If Algorithm 2 is feasible for manipulator i, its motion (given by $\mathbf{u}_i$) is collision-free up to τ_c. Collision-free motion

for the object is only guaranteed if the assumptions hold and if all manipulators select the same side for avoidance (see Remark 1) with respect to static and dynamic obstacles.

Recall that a conservative hypothesis is made regarding the velocity change of the other agents' velocities ($\Delta\mathbf{v}_i = \Delta\mathbf{v}_{ij}/2$ in Sec. IV-B). If $\xi_i = 0$, then it is guaranteed that the object shape remains within its limits, otherwise forces are transmitted (Sec. III-E.3) and the other manipulators would react accordingly in the subsequent planning steps to compensate. Nonetheless, if reaction time is too slow, then collisions may still arise.

***Remark** 2 (Infeasibility):* If the optimization is infeasible, not all the constraints can be satisfied and a collision might be imminent. The robots decrease their speed to guarantee passive safety.

***Remark** 3 (Global planning):* In both cases, the receding-horizon nature of the controller results in a lack of long term guarantees, thus the need for a global algorithm for guidance and coordination, which replans regularly with a simplified model of the object.

V. Discussion and Experimental Results

Data was collected over a total of more than three hours in a series of experiments involving a variety of deformable objects: a rope (two robots), a foam mat (two robots), a queen-size bed sheet (three robots), and a bath towel (three robots). We kept the manipulators (KUKA youBots) moving by assigning random goals (position/orientation), at about one per minute, in a room of dimensions 5.5 m on each side. For each goal, an approximate trajectory for the object was obtained via random sampling. In some experiments, an iRobot Create followed a predictable, pre-programmed path through the room that was unknown to the robots. We also introduced a human as a less predictable obstacle. Since our robots do not currently support force sensing, our implementation relies on a exchange of virtual forces.

Localization was provided by an external tracking system and the local motion planner of Section IV computed new velocity commands for the robots at 10Hz frequency. For most of the experiments the distributed version was employed. We collected data on the position and velocity of the mobile manipulators, dynamic and static obstacles. Examples are shown in the attached video and in Figs. 1 and 7.

The experimenter manually supplied minimum and maximum object bounds for each object, from which the robots automatically computed object limits (Definition 1). The performance of the robots in maintaining the object within its limits is given in Figure 8 and Table I. The robots spend a significant amount of time at the upper bound of their distance ranges because the transmission of forces is effective only when the robots are spaced at the bounds. The addition of virtual spring forces, which can be easily added, would induce the robots to maintain a preferred distance within bounds. Distributed local planners violate the bounds less often – likely because the implementation is more conservative than for the centralized case. Also of note are the experiments carrying the foam mat, which show significantly worse performance than the other objects - likely due to the strong constraints. Since our robots do not currently support force sensing, our implementation relies on velocity control. The velocity controller, implemented with PID control law, can introduce errors with the stiffer foam material due to a buildup in the integral term.

For the 117 random goals assigned to the robots; there were zero failures to converge to the goal. In 35% of trials, the robots reached the goal and came to a halt. For the remaining 65% of cases, a new goal was assigned before the old one could be achieved. However, the system never became stuck, and it continued to make progress in all cases, due to the guidance from the global planner.

Collisions during the experiments were minimal. Zero collisions between the manipulator robots and static obstacles were observed during all experiments. In two cases, a dynamic obstacle collided transiently with a stationary manipulator robot. We believe that such collisions could occur due to unmodeled experimental parameters such as lag in communicating state information to the robots and to the system not being able to react fast enough.

We observed that in some situations (for example when two manipulators held a rope and a dynamic obstacle came perpendicularly towards its middle point), and especially if no consistent global plan for the object was available, the manipulators may disagree on the avoidance side. Stronger coordination is thus required in those situations.

Fig. 7. Collaborative manipulation of deformable objects. Mobile manipulators carry the object while avoiding static and dynamic obstacles.

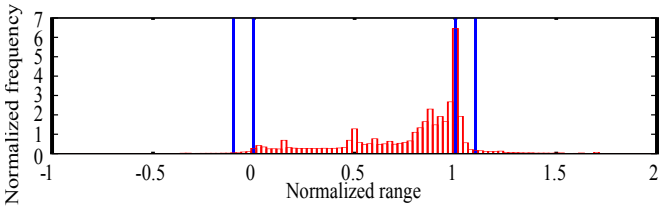


Fig. 8. Histogram of inter-robot distances aggregated and normalized across all experiments. Each object configuration dictates upper and lower bounds between pairs of manipulator robots. Each VICON reading at 120 Hz is reflected in this plot, in which the inter-robot distances are normalized onto a common range. Here, zero represents the lower bound and one represents the upper bound. Vertical lines partition the data as indicated in Table I.

TABLE I

STATISTICS ON EXPERIMENTAL DATA. COLUMNS INDICATE PERCENTAGE OF TOTAL DISTANCE READINGS WITHIN SPECIFIED SHAPE BOUNDS; LESS THAN 10% ABOVE UPPER OR BELOW LOWER BOUNDS (+10% UB AND -10% LB, RESPECTIVELY); AND MORE THAN 10% ABOVE UPPER OR BELOW LOWER BOUNDS ($\gg$UB AND $\ll$LB).

Experiments	Total readings	$\ll$ LB	-10% LB	Within Bounds	+10% UB	$\gg$ UB
Rope	63,808	0.0	3.7	78.5	6.1	11.7
Foam mat	114,371	14.2	2.5	56.7	5.5	21.1
Sheet	243,798	0.9	1.2	92.2	2.0	3.6
Towel	1,039,491	0.5	0.4	73.6	23.3	2.1
Centralized	284,649	5.7	1.8	66.5	14.2	11.8
Distributed	1,176,819	0.7	0.6	77.8	18.4	2.5
All exp.	1,461,468	1.7	0.9	75.6	17.6	4.3

Overall, we observed no deadlocks, low collision rates, and proper shape maintenance, as shown in Table I.

VI. CONCLUSION

In this paper, the problem of multi-robot coordination for transportation of deformable objects among static and dynamic obstacles is addressed. A local planner is presented and successfully tested with a team of mobile manipulators. Constraints for both collision avoidance and manipulation are seamlessly integrated in velocity space and a convex optimization is solved, either centralized or distributed. For the latter, transmission of forces to the object is integrated to show motion intent and for coordination.

Regarding future work, more must be done to help ensure consensus among the manipulators of the global trajectory for the object. The associated shape bounds could be automatically determined from an object and the dynamics of the robots and object could be modeled.

The method may apply to manipulators carrying an object together with a human operator. It would apply without modifications if the trajectory of the human is known by the manipulators. If the human operator acts as a leader and has freedom of movement, the manipulators can move the object accordingly by minimizing the change in velocity subject to the shape maintenance and collision avoidance constraints. This extension has been successfully tested in simulation, but experimental evaluation is left as future works.

APPENDIX

In practice, $\mathbf{u}_i$ and $\breve{\mathbf{u}}_i$ would be commanded to robot i and $\underline{\mathbf{f}}_i$ measured via force sensors or motor current. In simulation, the executed velocities are computed by simulating the deformation of the object subject to the the inputs from all manipulators, imposing that it stays within its limits and that the executed velocity of each robot is at most equal to its commanded one (manipulators can not be pushed directly).

***Algorithm** 3 (Simulation of object constraints):* Given the input velocities $\mathbf{u}_{1:m}$ and $\breve{\mathbf{u}}_{1:m}$ for all manipulators, the executed velocities $\mathbf{v}_{1:m}$ and $\breve{\mathbf{v}}_{1:m}$ are given by the following optimization and the transmitted forces by Eq. (2).

$$
\begin{aligned}
\underset{\mathbf{v}_{1:m},\breve{\mathbf{v}}_{1:m}}{\text{argmin}} \quad & ||\breve{\mathbf{v}}_{1:m} - \breve{\mathbf{u}}_{1:m}||^2 + ||\mathbf{v}_{1:m} - \mathbf{u}_{1:m}||^2 \\
s.t. \quad & \min(0, u^s) \leq v^s \leq \max(0, u^s) \\
& \forall u^s, v^s \text{ components of } \mathbf{u}_{1:m}, \mathbf{v}_{1:m}, \breve{\mathbf{u}}_{1:m}, \breve{\mathbf{v}}_{1:m} \\
& \mathbf{v}_{1:m}, \breve{\mathbf{v}}_{1:m} \text{ satisfy constraints of Section IV-B} \\
& \mathbf{v}_{1:m}, \breve{\mathbf{v}}_{1:m} \text{ satisfy constraints of Section IV-D.}
\end{aligned}
$$

In our setting with virtual forces, at the gripping position $\breve{\mathbf{p}}_i$, the resultant force $\underline{\mathbf{f}}_i$ and moment $\underline{\mathbf{m}}_i$ are given as a function of the manipulator exerted forces $\bar{\mathbf{f}}_i$ and their position $\breve{\mathbf{p}}_j$ relative to the object's center $\mathbf{p}_B$,

$$
\underline{\mathbf{f}}_i = \sum_{j \in \mathcal{M}-\{i\}} \bar{\mathbf{f}}_i \,, \quad \underline{\mathbf{m}}_i = \sum_{j \in \mathcal{M}-\{i\}} \bar{\mathbf{f}}_j \cdot (\breve{\mathbf{p}}_j - \mathbf{p}_B)/||\breve{\mathbf{p}}_j - \mathbf{p}_B||.
$$

REFERENCES

[1] R. Bischoff, U. Huggenberger, and E. Prassler, "KUKA youBot-a mobile manipulator for research and education," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[2] F. I. Magdeburg, "Factories of the future: Mobile manipulators for aerospace production," *Press release*, pp. 1–2, Apr. 2013.

[3] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," *Intelligent Robots and Systems, Proc. 1995 IEEE/RSJ Int. Conf. on*, 1995.

[4] Z. Li, S. S. Ge, and Z. Wang, "Robust adaptive control of coordinated multiple mobile manipulators," *Mechatronics*, vol. 18, June 2008.

[5] A. Yamashita, T. Arai, J. O., and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *Robotics and Automation, IEEE Trans. on*, vol. 19, Apr. 2003.

[6] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-robot manipulation via caging in environments with obstacles," in *Robotics and Automation, ICRA IEEE International Conference on*, 2008.

[7] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators," *Journal of Robotic Systems*, vol. 13, no. 11, pp. 755–764, 1996.

[8] T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *Robotics and Automation, IEEE Trans.*, vol. 18, no. 1, 2002.

[9] H. G. Tanner, K. J. Kyriakopoulos, and N. J. Krikelis, "Modeling of multiple mobile manipulators handling a common deformable object," *Journal of Robotic Systems*, vol. 15, no. 11, pp. 599–623, 1998.

[10] J. P. Desai and V. Kumar, "Motion planning for cooperating mobile manipulators," *Journal of Robotic Systems*, vol. 16, no. 10, 1999.

[11] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 19, pp. 53–64, Feb. 2003.

[12] K. Sreenath and V. Kumar, "Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots," *Robotics: Science and Systems*, 2013.

[13] R. Ritz, M. W. Müller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *Intelligent Robots and Systems (IROS), IEEE/RSJ Int. Conf. on*, 2012.

[14] F. F. Khalil and P. Payeur, "Dexterous robotic manipulation of deformable objects with multi-sensory feedback-a review," in *Robot Manipulators Trends and Development*, INTECH Open Access, 2010.

[15] R. Gayle, P. Segars, M. C. Lin, and D. Manocha, "Path Planning for Deformable Robots in Complex Environments.," 2005.

[16] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, vol. 25, no. 4, pp. 809–836, 2005.

[17] P. Fiorini and Z. Shillert, "Motion Planning in Dynamic Environments using Velocity Obstacles," *Int. Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[18] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body Collision Avoidance," in *Int. Symp. on Robotics Research*, 2009.

[19] J. Alonso-Mora, M. Rufli, R. Siegwart, and P. Beardsley, "Collision Avoidance for Multiple Agents with Joint Utility Maximization," *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2013.

[20] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Rob. & Autom. Mag.*, vol. 19, no. 4, pp. 72–82, 2012.