Task-Oriented Contact Optimization for Pushing Manipulation with Mobile Robots

Filippo Bertoncelli[1], Mario Selvaggio[2], Fabio Ruggiero[2], Lorenzo Sabattini[1]

***Abstract*— This work addresses the problem of transporting an object along a desired planar trajectory by pushing with mobile robots. More specifically, we concentrate on establishing optimal contacts between the object and the robots to execute the given task with minimum effort. We present a task-oriented contact placement optimization strategy for object pushing that allows calculating optimal contact points minimizing the amplitude of forces required to execute the task. Exploiting the optimized contact configuration, a motion controller uses the computed contact forces in feed-forward and position error feedback terms to realize the desired trajectory tracking task. Simulations and real experiments results confirm the validity of our approach.**

I. Introduction

Manipulating objects in the real world is one of the primal tasks of a robotic system, and how the robot interacts with the environment has been a subject of several research studies [1]. Frequently, especially in industrial scenarios, the robot is equipped with an ad-hoc end-effector that has been purposefully designed to fulfil a specific task. More generic end-effectors, like robotic hands, are employed in scenarios where this is not possible due to a lack of knowledge about the shape of the object to manipulate or when restricting the operational capabilities of the manipulator to specific classes of objects is not convenient. Not having a pre-designed way of manipulating an object opened a different class of engineering problems: what is the best way to grasp a given object? For many years, the solutions have been based on the concept of *force closure* or *form closure* [2]. These grasp configuration properties ensure that the object is rigidly attached to the manipulator and that the mechanical connection can sustain, to some extent, the forces required by the manipulation task. However, the inertial and gravitational effects during manipulation can severely affect the stability [3], dexterity [4] and disturbance resistance capability [2] of the grasp, especially with heavy objects. In these scenarios, inadequate grasp configurations could mean failing the manipulation or even rupture of the end-effector mechanism. Grasp synthesis algorithms can be employed to consider these factors [5] and obtain a resilient grasp configuration. When the task is known, it is intuitively convenient to include its information in the grasp generation procedure. The authors of [6] propose a task-oriented quality measure to assess how suitable a grasp is with respect to a given task. In [7], the authors propose a grasp synthesis algorithm that computes the optimal grasp for a task whose force/torque requirements are extracted from a human demonstration.

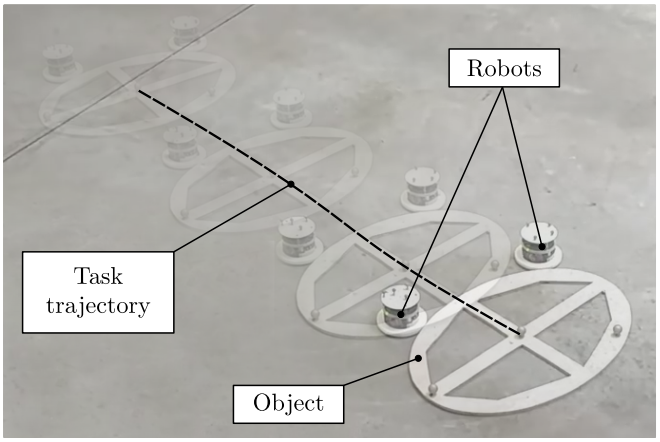

Fig. 1. Picture of the problem addressed in this work: a group of mobile robots has to manipulate an object by pushing it along a trajectory. The aim is to establish optimal contact points to simultaneously minimize the trajectory tracking error and the contact forces along the task execution.

However, these strategies cannot be applied if the target object is too large, too heavy to be transported, or too delicate to handle by a firm grasp. In these scenarios, non-prehensile manipulation approaches [8] can be exploited to achieve the desired task by removing some of the intrinsic limitations of grasping. In detail, these are strategies where one or multiple robots impose the motion to the object through unilateral constraints only: the pushing manipulation primitive considered in this work is a typical example of non-prehensile manipulation. However, the application of non-prehensile manipulation techniques is non-trivial since they require taking into account several factors, such as the dynamics and the kinematics of both the object and the robots, as well as the environment since each exchanged force can significantly impact the outcome of the task. One of the most common approaches to non-prehensile manipulation involves the computation of a motion model given a contact configuration [9]–[11] to design a planning or control architecture to fulfil the manipulation task. In this work, we tackle the non-prehensile manipulation problem from a different point of view. Starting from a manipulation

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony) and the COWBOT project, in the frame of the PRIN 2020 research program, grant number 2020NH7EAZ_002.

[1]Filippo Bertoncelli and Lorenzo Sabattini are with the Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy. {filippo.bertoncelli, lorenzo.sabattini}@unimore.it

[2]Mario Selvaggio and Fabio Ruggiero are with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Napoli, Italy. {mario.selvaggio, fabio.ruggiero}@unina.it

task encoded by a planar trajectory, we identify a non-prehensile contact configuration established by a group of robots that is best suited to track the task trajectory (see Fig. 1). We then employ a control strategy to command a team of mobile robots to execute the desired pushing action onto the object.

II. Related works and Contribution

Pushing manipulation belongs to the broader class of non-prehensile manipulation actions [8]. These include throwing [12], catching [13], batting [14], rolling [15], and so on. The main advantage of non-prehensile manipulation lies in using simple robot end-effector designs to perform various manipulation tasks robustly [16]. However, this comes at the expense of endowing robots with good physics models. The mechanics of pushing manipulation have been explored in the literature to identify a usable motion model for practical applications. The works presented in [17] and [9] have enabled the design of planning algorithms for pushing, such as the one presented in [18] that exploits a two-layer random sampling strategy to generate a feasible manipulation sequence. The field of control algorithms has also benefited from the ability to predict the motion of a pushed object, designing controllers specific for pushing applications. One example is presented in [10], where the authors integrated the pushing stability constraints into a model predictive control scheme to control a mobile robot. With a similar approach, the authors of [11] apply a predictive control strategy to the manipulation of a planar object by pushing with a robotic arm, considering the possibility for the contact point to slide onto the surface of the object.

Non-prehensile manipulation has been recently realized by exploiting the properties of flexible elements. For example, the robots are equipped with a flexible cable in [19], in which the authors propose a planning method to define the motion of the robot exploiting the cable for pushing an object. In a more generic scenario, the authors of [20] proposed a strategy for instructing a pair of mobile robots connected through a cable to pull a heavy object cooperatively. However, the interconnection between robots can significantly reduce their freedom of motion.

By having the robots push directly onto the object, it is possible to avoid these restraints [21], given that the robots can freely navigate the environment without colliding with possible obstacles and therefore change their relative position with respect to the object. The authors of [22] achieve this behaviour by dealing with uncertainties in both control and motion execution with an appropriate planning strategy that considers an increased size of the pushed object to accommodate re-positioning manoeuvres of the pushing robot. An artificial potential field is designed in [23] to let a robot, or a team of robots, push an object by only measuring its instantaneous direction of motion.

Most of the works mentioned above, especially those related to planning, have in common the presence of a task generated to be feasible for a specific pushing configuration. In the literature, the problem of how to grasp a given object to perform a specific task, which is usually encoded through its desired trajectory while satisfying some optimal criteria, is known as *task-oriented grasping*. This concept has been successfully applied in numerous application scenarios [24]–[28]. However, all the previous works considered either force- or form- closure grasps, which allow the robot to treat the object as entirely restrained. Indeed, none of the above works has addressed the task-oriented optimal placement of multiple robots around an object for non-prehensile pushing.

In this paper, we extend the use of such a concept to non-prehensile pushing manipulation performed by a fleet of autonomous mobile robots. Our main contributions can be listed as follows:

- we define a task-oriented optimization procedure to optimally position the robots around the object to perform a given task (tracking of the desired trajectory);
- we design a control algorithm that instructs the robots on how to push the object to track the assigned task trajectory properly;
- we thoroughly validate the presented procedures using physical simulation and real-world experiments using a non-optimized case as a baseline.

III. Problem statement and assumptions

The first challenge addressed here is the optimal positioning of the robots along the object perimeter to enhance the object's trajectory tracking performances and the pushing robustness. More specifically, given a manipulation task, we aim at identifying the optimal deployment of n robots in contact with the object, also denoted as the *grasp configuration*, that can fulfil the task in the most robust way possible. The second challenge is designing a control algorithm for the motion of the robots performing the manipulation of the object along the desired trajectory. In detail, the algorithm aims at reducing the tracking error of the object's configuration with respect to the desired trajectory, having each robot push on the object from the predetermined optimized contact position. To fulfil the presented challenges, we will consider a system described as follows.

Let $T_o \in SE(2)$ be the planar configuration of the *manipulandum*'s (the object to be manipulated) centre of mass (CoM). Let m_o, $J_o \in \mathbb{R} > 0$ be the mass and inertia moment of the object, respectively. The position of the i-th robot on the plane is identified by $p_i \in \mathbb{R}^2$, expressed in $\{\mathcal{W}\}$, the global fixed reference frame. The robots are modelled as circular objects of radius $r_i > 0$ and can establish point contacts with friction with the object's perimeter [29]. We assume the robots move on the plane following the motion law of a single integrator

$$\dot{p}_i = u_i. \tag{1}$$

Figure 2 visualizes the considered system with an elliptical manipulandum (grey) laying on the horizontal plane with two robots (yellow) represented in the contact position. The considered manipulation task is described as a trajectory Γ composed of a timed sequence of object configurations and the associated velocity at each generic time instant $t \in$

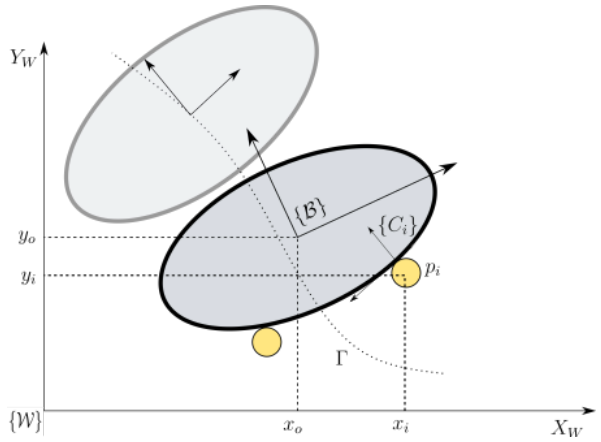

Fig. 2. Visual representation of the considered system. Symbols are explained in Sec. III.

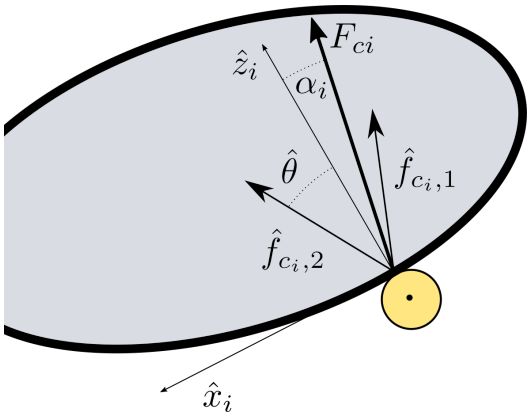

Fig. 3. Close up visualization of the force applied on the object by robot i as well as the borders of the associated friction cone. Symbols are explained in Sec. IV-A.

$[0,\, T]$, with T being the trajectory duration. The following additional assumptions are considered:

- all the contacts are modelled as *point contacts with friction* [29], and interactions obey Coulomb's friction model with a known friction coefficient between the parts in contact;
- the manipulated object and the robots lay in a planar environment, with all the frictional forces acting on the plane and gravity acting downward on the vertical axis;
- the frictional properties of the ground are uniform across the environment;
- the motion is slow enough, such that inertial forces are negligible or instantly absorbed by the frictional effects (*quasi-static* assumption);
- the object's perimeter is represented as a closed smooth curve enclosing a convex region of the space.

IV. Task-oriented optimization and control architecture

A. Task-oriented optimal robots positioning

This section describes the calculation procedure used to optimize the robots' contact positions around the object. The closed curve representing the object's perimeter at a given configuration $T_o \in SE(2)$ is parametrized by θ, i.e., the angle formed by the segment connecting the object CoM and a point on the object's perimeter. Given the object convexity assumption, the value of θ corresponds to a given position belonging to the object's perimeter. Thus, a contact point between the i-th robot and the object is identified by a particular value of θ_i. We aim to calculate optimized values of θ_i for each robot to perform the pushing task optimally.

At each contact point, we define a reference frame $\{\mathcal{C}_i\}$, whose origin lies on the object's perimeter, the $\hat{x}$-axis is directed along the perimeter tangent, and the $\hat{z}$-axis is directed along the inward normal. The number of contact points is n_c, equal to the number of robots in contact with the object. Denoting with $\Theta = [\theta_1, \ldots, \theta_{n_c}]^{\mathrm{T}} \in \mathbb{R}^{n_c}$, the grasp matrix $G(\Theta) \in \mathbb{R}^{3\times 2n_c}$ maps the contact forces and the velocities between the object's CoM, identified by the frame $\{\mathcal{B}\}$, and the contact frames $\{\mathcal{C}_1\}, \ldots, \{\mathcal{C}_{n_c}\}$. Generally, G can be constructed given the generic $i-$th contact point pose (p_{c_i}, R_{c_i}), with $R_{c_i} \in \mathrm{SO}(2)$ the rotation matrix of the frame $\{\mathcal{C}_i\}$ and $p_{c_i} = [p_{c_i,x},\, p_{c_i,y}]^{\mathrm{T}} \in R^2$ the position of the origin of $\{\mathcal{C}_i\}$ expressed in $\{\mathcal{B}\}$, as follows (see [29])

$$G = \begin{bmatrix} \ldots & R_{c_i} & 0 & \ldots \\ & [-p_{c_i,y}\ p_{c_i,x}]R_{c_1} & 1 & \end{bmatrix}. \tag{2}$$

Under the stated quasi-static assumption, the object dynamic model can be formulated as follows

$$\dot{x}_o = R_o H G(\Theta) F_c, \tag{3}$$

where $x_o = [p_{o,x},\, p_{o,y},\, \theta_o]^{\mathrm{T}} \in \mathbb{R}^3$ denotes the minimal representation of the object pose and $\dot{x}_o$ its time derivative, while $F_c = [f_{c_1}^{\mathrm{T}}, \ldots, f_{c_{n_c}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2n_c}$ is the vector stacking the contact forces, $f_{c_i} \in \mathbb{R}^2$ with $i = 1, \ldots, n_c$, representing the overall system's input.

To be realizable, the contact forces F_c must belong to the friction cone space. To facilitate the integration of this constraint in the optimization problem introduced later, a contact force parametrization is introduced. The $i-$th contact force, f_{c_i}, can be expressed as a non-negative linear combination of unit vectors $\hat{f}_{c_i,j} \in \mathbb{R}^2$, with $j = 1, 2$, denoting the $i-$th friction cone boundaries. These vectors can be calculated, using geometrical considerations (see Fig. 3), starting from the friction coefficient $\mu \geq 0$ as follows

$$\hat{f}_{c_i,j} = R_y(\pm\hat{\theta})\hat{z}_i, \quad \hat{\theta} = \arctan\mu, \tag{4}$$

where $R_y(\cdot) \in SO(3)$ is the rotation matrix around the $\hat{y}-$axis, $\hat{z}_i = [0,\, 0,\, 1]^{\mathrm{T}} \in \mathbb{R}^3$ is the contact normal expressed in $\{\mathcal{C}_i\}$, $\hat{\theta} \in \mathbb{R}$ is the semi-aperture angle of the friction cone.

By denoting with $\hat{F}_c = \mathrm{blockdiag}\left(\hat{F}_{c,1}, \ldots, \hat{F}_{c,n_c}\right) \in \mathbb{R}^{2n_c\times 2n_c}$ the matrix encoding all the unit vectors representing the boundary of the friction cones, i.e., $\hat{F}_{c,i} = \left[\hat{f}_{c_i,1}, \hat{f}_{c_i,2}\right] \in \mathbb{R}^{2\times 2}$, and with $\Lambda = \left[\lambda_{c_1,1}, \lambda_{c_1,2}, \ldots, \lambda_{c_{n_c},1}, \lambda_{c_{n_c},2}\right]^{\mathrm{T}} \in \mathbb{R}^{2n_c}$ the associated coefficients used to decompose the contact forces along the friction cone boundaries, we can compactly express the vector of stacked contact forces as follows

$$F_c = \hat{F}_c \Lambda. \tag{5}$$

At this point, the contact forces belong to the composite friction cone space (Cartesian product of all the friction cone spaces) if $\Lambda \succeq 0$, i.e., all the $\lambda_{i,j}$, with $i = 1, \ldots, n_c$ and $j = 1, 2$, must be non-negative [29].

A task-oriented optimal criterion is established to optimize the positions of the robots around the object (and thus the contact points). Given a task trajectory Γ to be realized (tracked) expressed as a timed sequence of desired poses for the object $x_{o,d}(t)$, $0 \leq t \leq T$, we choose to optimize the position of the robots (through the optimization of Θ) to simultaneously minimize the task tracking error and the associated contact forces (coefficients) to realize it. Mathematically, the discretized problem is formulated as follows

$$\min_{\Theta, [\Lambda_1, \ldots, \Lambda_N]} \quad \sum_{k=1}^{N} \frac{1}{2} ||x_{o,d}(k) - x_o(k)||_Q^2 + \frac{1}{2} ||\Lambda(k)||_R^2 \quad (6)$$

$$s.t. \quad x_o(k+1) = x_o(k) + R_o(k) G(\Theta) \hat{F}_c \Lambda(k), \quad (7)$$

$$D\Theta + \bar{\Theta} \succeq \epsilon, \quad (8)$$

$$\Lambda(k) \succeq 0, \quad \forall k = 1, \ldots, N \quad (9)$$

where k represents the generic time instant, N is the total number of steps taken to accomplish the task (and it is related to the trajectory duration T and the adopted discretization step), $x_{o,d}(k)$ and $x_o(k)$ are the desired and the current object positions at a given time step $1 \leq k \leq N$, $Q \in \mathbb{R}^{3\times3}$ and $R \in \mathbb{R}^{2n_c \times 2n_c}$ are diagonal and positive-definite matrices. The constraint in (7) is obtained by combining the object dynamic model (3) and the contact force parametrization (5), while the constraint (9) denotes the feasibility of the contact forces discussed above. The constraint (8) is used to avoid collisions between the robots and is described hereafter. The matrix $D \in \mathbb{R}^{n_c \times n_c}$ is the *collision matrix* of the multi-robot system deployed around the object (encoding potential pairwise collisions among the robots), $\bar{\Theta} \in \mathbb{R}^{n_c}$ is the vector taking into account the modulo 2π property of the parametrized object perimeter, and are specified as follows

$$D = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ldots & 0 \\ \vdots & & & \ddots & \vdots \\ 1 & 0 & 0 & \ldots & -1 \end{bmatrix}, \qquad \bar{\Theta} = \begin{bmatrix} 0 \\ \vdots \\ 2\pi \\ \vdots \\ 0 \end{bmatrix},$$

where $\epsilon \geq 0$ is a non-negative constant vector denoting the collision bounds to be opportunely chosen. The matrix has a pair of 1 and -1 for every pair of adjacent robots around the object (the first and last robots are considered adjacents).

The value 2π occupies the $k-$th component of the vector $\bar{\Theta}$, and it is used to take the shorter distance between the $k-$th and $k+1-$th contact points when this last crosses the value $\theta = 2\pi$. This is used to calculate the right minimum (angular) distance between the robots. The value of ϵ_i (i.e., the i-th component of vector ϵ) represents the lower bound on the angle between the computed configurations of the two successive contact points/robots i and $i+1$. This parametrization can only be used in the case of convex objects to improve performance. For concave objects, adding self-collision avoidance will require more precise geometry information.

B. Motion control for trajectory tracking

This section reports the proposed control scheme that commands the robots to execute the considered pushing manipulation task. More specifically, the objective of the manipulation is to move the object on the plane following a predefined trajectory Γ. To achieve this objective, we propose to define the desired body force according to a proportional-derivative control scheme. In particular, let $x_{o,d} \in \mathbb{R}^3$ be the current desired pose for the object along Γ, and $v_{o,d} = [v_{o,d,x}, v_{o,d,y}, \omega_{o,d}]^{\mathrm{T}} \in \mathbb{R}^3$ be the associated desired velocity. The desired body force $\mathcal{F}_o^* \in \mathbb{R}^3$ for the problem at hand is computed as

$$\mathcal{F}_{o,d} = K_{P_O}(x_{o,d} - x_o) + K_{D_O}(v_{o,d} - v_o) + f_\mu(v_{o,d}), \quad (10)$$

where $K_P, K_D \in \mathbb{R}^{3\times3}$ are positive-definite diagonal matrices. The last term, $f_\mu(v_d) \in \mathbb{R}^3$, is a feed-forward component for sliding friction compensation. This component is computed as the wrench caused by the friction effect on the object sliding with velocity $v_{o,d}$. More specifically, $f_\mu(v_d)$ is computed using the following ellipsoidal approximation for the limit surface [30]

$$f_\mu(v_{o,d}) = \frac{\mu_f m_o g}{\sqrt{v_{o,d,x}^2 + v_{o,d,y}^2 + (\gamma\omega_{o,d})^2}} \begin{bmatrix} v_{o,x} \\ v_{o,y} \\ \gamma^2\omega \end{bmatrix}, \quad (11)$$

where $\gamma = \sqrt{J_o/m_o}$ and $\mu_f > 0$ is the friction coefficient between the object and the floor. The body force $\mathcal{F}_{o,d}$ computed in (10) is then used to online extrapolate feasible contact forces to be applied to the object using the optimization problem described below

$$\min_{F_c} \quad \frac{1}{2} ||\mathcal{F}_{o,d} - G(\Theta^*) F_c||^2 \quad (12)$$

$$s.t. \quad F_c = \hat{F}_c \Lambda, \quad (13)$$

$$\Lambda \succeq 0, \quad (14)$$

where $\Theta^* \in \mathbb{R}^{n_c}$ is the optimal solution of (6). The input velocity for the i-th robot is thus computed using the following control law

$$u_i = K_{P_R}(p_{c_i}^* - p_i) + K_{V_R} R_o G(\Theta) F_{ci}, \quad (15)$$

where the first term provides proportional position feedback for the robot since $p_{c_i}^*$ is the desired optimal position for the i-th robot, calculated using θ_i^*, and p_i is its current position. The second addendum is a feed-forward component instructing the robots to move in the direction of the contact force needed to accomplish the task. The coefficients $K_{P_R}, K_V > 0$ are gains of the control law to be opportunely selected. The proposed online procedure uses the optimal robots' positioning Θ^*, which guarantees minimal contact forces and tracking error in the ideal case while providing robustness through position feedback terms.

V. Simulation and Experimental Results

In this section, we present the evaluation method used to assess the validity of the proposed strategy. In order to properly evaluate the presented optimization and control architecture, a series of simulations have been performed. We propose three different manipulation tasks carried out by some robots pushing an elliptical prism. For each trajectory, the optimal positioning for each robot is computed using the procedure presented in Section IV-A. The control law proposed in Section IV-B is then used to track the task trajectory by having the robots push onto the object after they have reached the desired contact position. The validation results will be detailed in Section V-C. Performed trajectories are also shown in the accompanying video.

A. Simulation setup

The simulations have been performed using the CoppeliaSim physical simulation software [31]. Two or three robots are placed in the simulated environment with an elliptical object, as visualized in Fig. 4. The robots are a modelled version of the e-puck [32] differential drive mobile robot and are equipped with bumper rings with radius $r_i = 5$ cm, to provide a consistent contact surface. In order to provide the single integrator behaviour described in (1), the velocities for the wheels of each robot are obtained through input-output feedback linearization [33]. The object's mass is set to $m_o = 0.3$ kg while the inertia moment is set to $J_o = 0.0072$ kg m^2. The principal axes of the ellipse are 0.5 m and 0.2 m, respectively. The friction coefficient between the object and the floor is set to $\mu_f = 0.6$ while the robot-object coefficient is set to $\mu = 0.2$. All the used parameters are set according to the best available estimate from the available real-world equipment described in the next section. The purpose of testing the proposed controller in a simulated environment is to verify its behaviour within ideal conditions, while the real-world tests will assess its robustness. In order to fully test our algorithm in simulation, we would have to take into account uncertainties of the parameters, e.g., the non-uniformity of the friction coefficient across the environment or the floor not being perfectly flat. The data required to compute the control laws is transmitted through ROS communication channels to a MATLAB script implementing the proposed methodology. The simulations are performed on a laptop equipped with an Intel Core I7-9750H and 16GB of RAM.

B. Experimental setup

The real-world experiment is performed on a honed concrete floor using the same robot employed in the simulation environment, the e-puck differential drive robots, equipped with a 3D printed bumper ring. The elliptical manipulandum is obtained from a 5 mm plywood sheet. The principal axes of the ellipse are 0.255 m and 0.175 m, respectively. The global pose of the robots and the object is obtained using an Optitrack™ motion capture system [34] composed of seven Prime13 cameras. The information about the velocity of the object, required to compute (10), is obtained by feeding the pose information into an extended Kalman filter generating the 3D linear and angular velocities of the object. The system inside the experiment area is visualized in Fig. 5. All the required data is sent to the PC, computing the control law through ROS channels. The input for each robot is sent through a Bluetooth connection to the robot's control board.

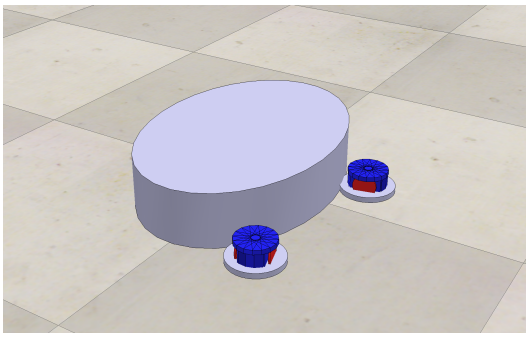

Fig. 4. The robots and object in the CoppeliaSim simulation environment.

Fig. 5. The robots and object in the real-world environment.

C. Results

This section shows the results of the performed tests. First, the results of two different simulated tasks are presented, followed by a manipulation task performed in the real environment. For each instant of each assigned trajectory, the reference orientation for the object is kept equal to the orientation of the global reference frame. Figure 6 top graphs show the execution of a desired b-spline trajectory (black dashed line), in the optimal robots' configuration case (left) and in the non-optimal case(right). The bottom left graph shows the norm of the tracking error $e_p = p_{o,d} - p_o$ while the bottom graph shows the contact force $(f_{c_1}, f_{c_2}, f_{c_3})$ norms along the performed trajectory. The continuous lines are associated with results with the robots in the optimal configuration $\Theta^* = [5.112, 4.623, 3.569]^{\mathrm{T}}$ found solving the problem in (6), while dashed lines are associated with non-optimal robots' configuration $\Theta = [7/4\pi, 3/2\pi, 5/4\pi]^{\mathrm{T}}$, which is also used as a starting point of the optimization. As it is possible to note from the upper graph, the optimal solution increases the performance of the tracking task showing a lower and less varying error norm (continuous line) compared to the non-optimal configuration (dashed line). This holds especially in the second half of the trajectory, where the non-optimized trajectory oscillates significantly

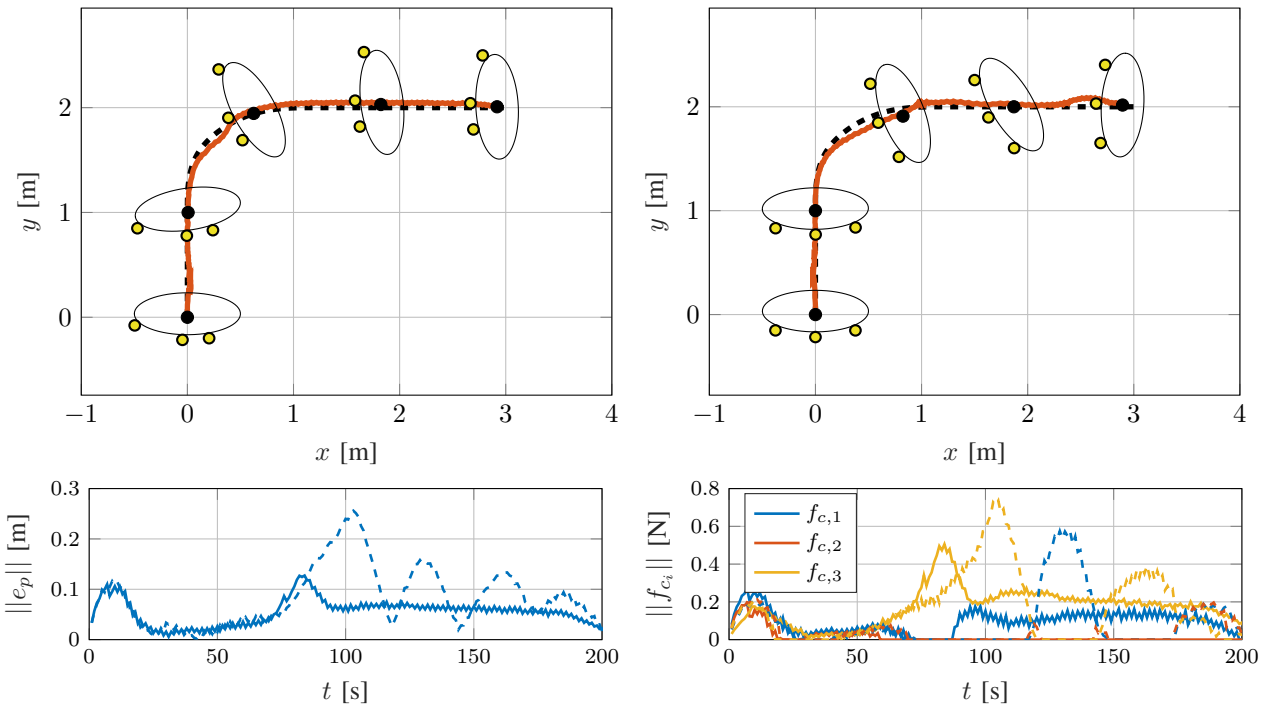

Fig. 6. Upper graphs: desired (black dashed line) and executed trajectory (red) in the optimal robots' configuration (left) and non-optimal robots' configuration (right). Bottom left graph: norm of the tracking error $e_p = p_{o,d} - p_o$. Bottom right graph: norm the contact forces $f_{c_1}, f_{c_2}, f_{c_3}$ along the performed trajectory. Results of the optimal robots' configuration are shown with continuous lines, non optimal robots' configuration results are shown with dashed lines.

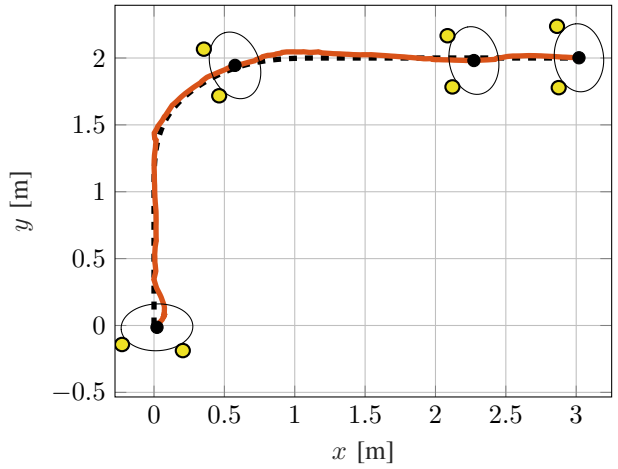

Fig. 7. Tracking performance along the first proposed test trajectory, depicted as a dashed blue line. The orange line shows the motion of the object during the simulated manipulation.

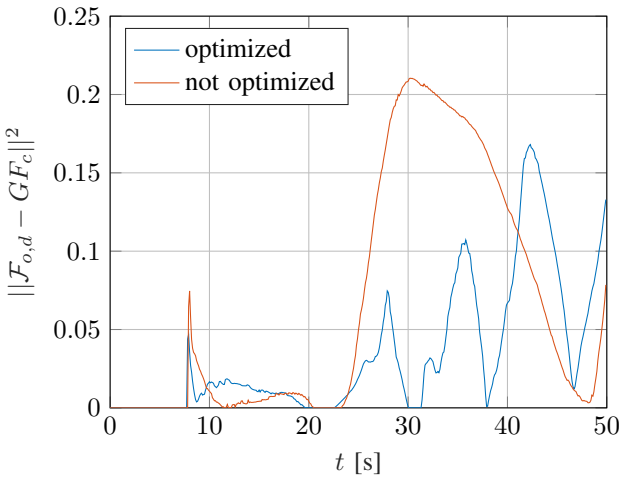

Fig. 8. Comparison of the error between the requested body force and the force that can be applied by the contacts.

more than the optimized one. Analogous considerations can be drawn by looking at the bottom graphs that show the norm of the three contact forces $f_{c_1}, f_{c_2}, f_{c_3}$ for the optimal configuration (continuous lines) and the non-optimal configuration (dashed lines). In particular, it can be noted as the contact forces are more uniformly distributed among the robots in the optimal case, especially in the second half of the trajectory. It is worth remarking that, although successful in the considered case, non-optimal configurations can lead to even larger tracking errors and contact forces norms, which may lead to task failures.

Figure 7 show the tracking performance of the same trajectory using two robots. The orange line shows that the robots can closely track the assigned trajectory. However, this is achieved by sacrificing the orientation tracking performance as the robots align the object to mostly push along the motion direction. The optimal configuration of the robots is $\Theta = [5.497, 3.734]^{\mathrm{T}}$, and corresponds to both robots actively pushing along the entire trajectory.

Figure 8 shows the performance difference between an

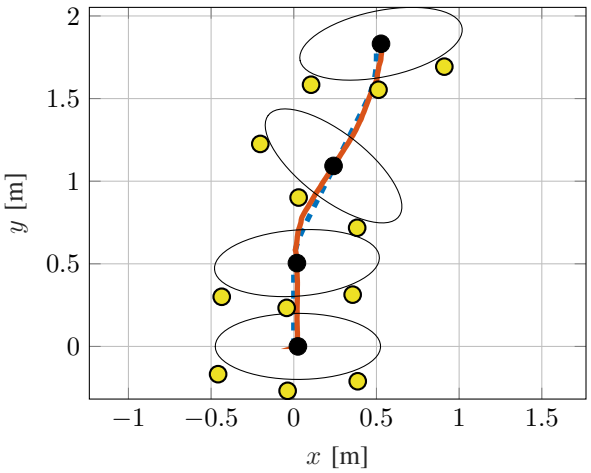

Fig. 9. Tracking performance along the second proposed test trajectory (dashed blue) using 3 robots in a simulated environment. The orange line shows the motion of the object during the simulated manipulation.

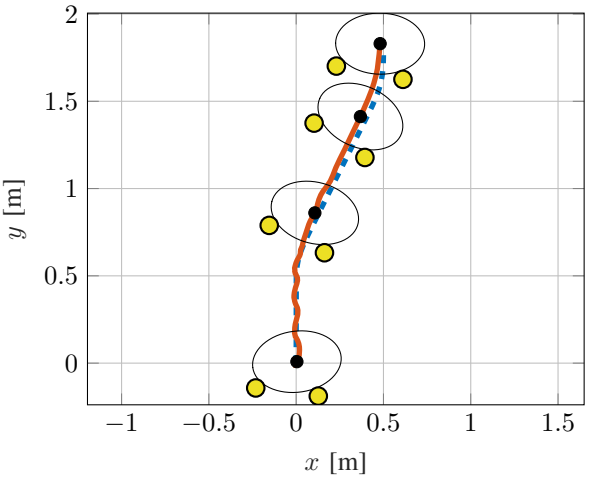

Fig. 10. Tracking performance along the third proposed test trajectory, depicted as a dashed blue line. The orange line shows the motion of the object during the simulated manipulation.

optimized configuration and a non-optimized one in terms of the norm of the difference between the desired body force $\mathcal{F}_{o,d}$ and the cumulative force applied at the contacts GF_c, obtained tracking the same b-spline trajectory. Even if the magnitude of the error is relatively small in both cases, the optimized configuration is more capable of quickly reducing the error, therefore obtaining higher tracking performance. It is crucial to notice how similar results are at the beginning of the tracking. Both configurations can apply a cumulative force directed in the object's local y direction. The difference in performance arises when the trajectory starts to turn, indicating that the optimized configuration can apply a force closer to the force requested by the task.

The second manipulation task is defined by the polynomial trajectory visualized in Figure 9. In this three-robot scenario, the optimal configuration is $\Theta = [5.1121, 4.6237, 3.5693]^{\mathrm{T}}$. The obtained performance matches the results of the previous scenarios in which the orientation tracking is sacrificed.

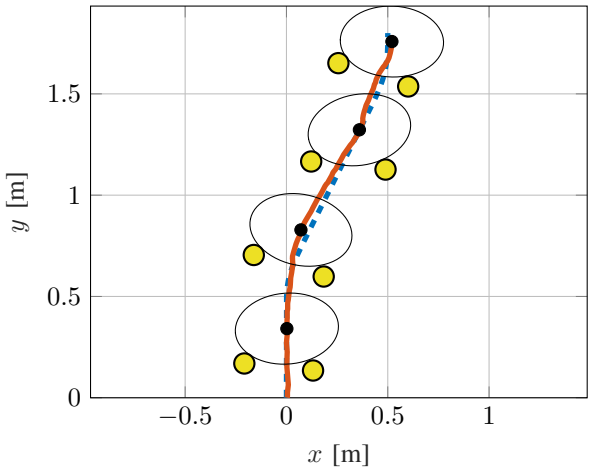

Fig. 11. Tracking performance along the third proposed test trajectory, depicted as a dashed blue line. The orange line shows the motion of the object during the real world manipulation.

The tracking performances depicted in Figure 10 and Figure 11 show the manipulation of the same elliptical object in simulation and the real environment, respectively. The optimal contact configuration is defined as $\Theta = [5.1868, 3.7398]^{\mathrm{T}}$. In this two-robot scenario, it is possible to note how in the optimal robot configuration, one robot mostly pushes from below while the other aligns itself with the diagonal portion of the trajectory. The great similarity between the simulation and the real environment results further validates the proposed approach.

VI. Conclusions and Future Works

In this work, we presented a task-oriented optimization and control strategy for object pushing with mobile robots. In particular, we proposed a method for calculating optimal non-colliding contact points for a group of mobile robots pushing an object that minimizes the trajectory tracking error and the norm of contact forces to perform the task. Exploiting the optimized contact configuration, a motion controller was developed that exploits online computed contact forces in feed-forward and position error feedback terms to realize the desired trajectory tracking task. Results were validated through simulations and experiments on a real robotic system composed of multiple robots.

Although effective, our method still has several limitations that must be addressed in future works. For instance, using a local optimization method to calculate the contact points might lead to sub-optimal results. A global optimization method might produce better results at the expense of a more significant computing time. A trade-off can be found depending on the application. Besides, we aim to reformulate the problem in (6) to find possibly time-varying contact configurations and the corresponding pushing forces for a given trajectory using the task-oriented approach presented here. Finally, further experimental validations and tuning will be carried out in the future, testing against other methods from recent literature and possibly regulating pushing forces at their optimal values by feedback.

REFERENCES

[1] I. Kao, K. M. Lynch, and J. W. Burdick, *Contact Modeling and Manipulation*. Cham: Springer International Publishing, 2016, pp. 931–954.

[2] A. Bicchi, "On the closure properties of robotic grasping," *Int. J. Robot. Res.*, vol. 14, no. 4, pp. 319–334, 1995.

[3] W. Howard and V. Kumar, "On the stability of grasped objects," *IEEE Trans. Rob. Autom.*, vol. 12, no. 6, pp. 904–917, 1996.

[4] Y. Park and G. Starr, "Optimal grasping using a multifingered robot hand," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 689–694 vol.1.

[5] V. Lippiello, B. Siciliano, and L. Villani, "Fast multi-fingered grasp synthesis based on object dynamic properties," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2010, pp. 1134–1139.

[6] R. Haschke, J. J. Steil, I. Steuwer, and H. Ritter, "Task-oriented quality measures for dextrous grasping," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation,*, 2005.

[7] S. El-Khoury, R. D. Souza, and A. Billard, "On computing task-oriented grasps," *Rob. Auton. Syst.*, vol. 66, pp. 145–158, 2015.

[8] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1711–1718, 2018.

[9] K. M. Lynch, "The mechanics of fine manipulation by pushing," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 1992, pp. 2269–2276.

[10] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying mpc for nonprehensile object manipulation with a nonholonomic mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 11 032–11 038.

[11] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 755–773, 2020.

[12] A. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, "Coordinate-free framework for robotic pizza tossing and catching," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3932–3939.

[13] M. M. Schill, F. Gruber, and M. Buss, "Quasi-direct nonprehensile catching with uncertain object states," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2468–2474.

[14] C. Liu, Y. Hayakawa, and A. Nakashima, "Racket control and its experiments for robot playing table tennis," in *Proc. IEEE Int. Conf. Robot. Biomim.*, 2012, pp. 241–246.

[15] D. Serra, F. Ruggiero, A. Donaire, L. Buonocore, V. Lippiello, and B. Siciliano, "Control of nonprehensile planar rolling manipulation: A passivity-based approach," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 317–329, 2019.

[16] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P. Robuffo Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Trans. Rob.*, vol. 38, no. 1, pp. 569–583, 2022.

[17] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *Int. J. Rob. Res.*, pp. 53–71, 1986.

[18] C. Zito, R. Stolkin, M. Kopicki, and J. L. Wyatt, "Two-level rrt planning for robotic push manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 2012, pp. 678–685.

[19] Y.-H. Kim and D. A. Shell, "Using a compliant, unactuated tail to manipulate objects," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 223–230, 2017.

[20] T. Maneewarn and P. Detudom, "Mechanics of cooperative nonprehensile pulling by multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 2004–2009.

[21] P. Kolhe, N. Dantam, and M. Stilman, "Dynamic pushing strategies for dynamically stable mobile manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3745–3750.

[22] S. Krivic, E. Ugur, and J. Piater, "A robust pushing skill for object delivery between obstacles," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, 2016, pp. 1184–1189.

[23] T. Igarashi, Y. Kamiyama, and M. Inami, "A dipole field for object delivery by pushing on a flat surface," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 5114–5119.

[24] M. Prats, P. J. Sanz, and A. P. del Pobil, "Task-oriented grasping using hand preshapes and task frames," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1794–1799.

[25] E. A. M. Ghalamzan, F. Abi-Farraj, P. R. Giordano, and R. Stolkin, "Human-in-the-loop optimisation: Mixed initiative grasping for optimally facilitating post-grasp manipulative actions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3386–3393.

[26] M. Selvaggio, E. A. M. Ghalamzan, R. Moccia, F. Ficuciello, and B. Siciliano, "Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3617–3623.

[27] R. Detry, J. Papon, and L. Matthies, "Task-oriented grasping with semantic and geometric scene understanding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3266–3273.

[28] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *Int. J. Robot. Res.*, vol. 39, no. 2-3, pp. 202–216, 2020.

[29] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[30] A. Fakhari, M. Keshmiri, and M. Keshmiri, "Dynamic modeling and slippage analysis in object manipulation by soft fingers," in *Proc. ASME International Mechanical Engineering Congress and Exposition*, vol. 4, Nov. 2014.

[31] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013.

[32] P. Gonçalves, P. Torres, C. Alves, F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," *Proc. 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, 2009.

[33] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[34] "Motion capture systems." [Online]. Available: https://optitrack.com/