# Communication-based Decentralized Cooperative Object Transportation Using Nonlinear Model Predictive Control

Christos K. Verginis, Alexandros Nikou and Dimos V. Dimarogonas

*Abstract*— **This paper addresses the problem of cooperative transportation of an object rigidly grasped by $N$ robotic agents. We propose a decentralized Nonlinear Model Predictive Control (NMPC) scheme that guarantees the navigation of the object to a desired pose in a bounded workspace with obstacles, while complying with certain input saturations of the agents. The control scheme is based on inter-agent communication and is decentralized in the sense that each agent calculates its own control signal. Moreover, the proposed methodology ensures that the agents do not collide with each other or with the workspace obstacles as well as that they do not pass through singular configurations. Finally, simulation results illustrate the validity and efficiency of the proposed method.**

## I. Introduction

Over the last years, multi-agent systems have gained a significant amount of attention, due to the advantages they offer with respect to single-agent setups. Robotic manipulation is a field where the multi-agent formulation can play a critical role, since a single robot might not be able to perform manipulation tasks that involve heavy payloads and challenging maneuvers.

Regarding cooperative manipulation, the literature is rich with works that employ control architectures where the robotic agents communicate and share information with each other as well as completely decentralized schemes, where each agent uses only local information or observers [1]–[6]. The most common methodology used in the related literature constitutes of impedance and force/motion control [1], [7]–[13]. Most of the aforementioned works employ force/torque sensors to acquire knowledge of the manipulator-object contact forces/torques, which, however, may result to performance decline due to sensor noise or mounting difficulties.

Moreover, in manipulation tasks, such as pose/force or trajectory tracking, collision with obstacles in the environment has been dealt with only by exploiting the potential extra degrees of freedom of over-actuated agents, or by using potential field-based algorithms. These methodologies, however, may suffer from local minima, even in single-agent cases, and in many cases they yield high control inputs that do not comply with the saturation of actual motor inputs, especially close to collision configurations. In our previous works, [14], [15], we considered the problem of trajectory tracking for decentralized robust cooperative manipulation, without taking into account singularity- or collision avoidance.

Another important property that concerns robotic manipulators is the singularities of the Jacobian matrix, which maps the joint velocities of the agent to a 6D vector of generalized velocities. Such *singular kinematic* configurations, that indicate directions towards which the agent cannot move, must be always avoided, especially when dealing with task-space control in the end-effector [16]. In the same vein, *representation* singularities can also occur in the mapping from coordinate rates to angular velocities of a rigid body.

The main contribution of this work is to provide decentralized feedback control laws that guarantee the cooperative manipulation of an object in a bounded workspace with obstacles. In particular, given $N$ agents that rigidly grasp an object, we design decentralized control inputs for the navigation of the object to a final pose, while avoiding inter-agent collisions as well as collisions with obstacles. Moreover, we take into account constraints that emanate from control input saturation as well kinematic and representation singularities. The proposed approach to address this problem is the repeated solution of a Finite-Horizon Open-loop Optimal Control Problem (FHOCP) of each agent, by assigning a set of priorities. Control approaches using this strategy are referred to as Nonlinear Model Predictive Control (NMPC) (see e.g. [17]–[23]). A decentralized NMPC scheme has been considered in our submitted work [24], which concerns multi-agent navigation with inter-agent connectivity maintenance and collision avoidance.

In our previous work [25], a similar problem was considered in a centralized way. However, the computation burden is high, due to the fact that the number of states in the centralized case increases proportionally with the number of agents. In this work, we decouple the dynamic model among the object and the agents by using certain load-sharing coefficients and consider a communication-based leader agent formulation, where a leader agent determines the followed trajectory for the object and the follower agents comply with it through appropriate constraints. To the best of the authors' knowledge, this is the first time that the problem of decentralized object transportation with singularity, obstacle and collision avoidance is addressed.

The remainder of the paper is structured as follows. Section II provides the preliminary background. The system dynamics and the formal problem statement are given in Section III. Section IV discusses the technical details of the solution and Section V is devoted to a simulation example. Conclusions and future work are discussed in Section VI.

Christos K. Verginis, Alexandros Nikou and Dimos V. Dimarogonas are with the ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden and with the KTH Center for Autonomous Systems. Email: {cverginis, anikou, dimos}@kth.se. This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the EU H2020 Co4Robots Project, the EU H2020 AEROWORKS project, the swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

## II. Notation and Preliminaries

The set of positive integers is denoted as $\mathbb{N}$ and the real $n$-coordinate space, with $n \in \mathbb{N}$, as $\mathbb{R}^n$; $\mathbb{R}^n_{\geq 0}$ and $\mathbb{R}^n_{>0}$ are the sets of real $n$-vectors with all elements nonnegative and positive, respectively; $I_n \in \mathbb{R}^{n\times n}$ and $0_{m\times n} \in \mathbb{R}^{m\times n}$ are the identity matrix and the $m \times n$ matrix with all entries zeros, respectively. Given a set $S$, we denote by $S^N := S \times \cdots \times S$ its $N$-fold Cartesian product. The vector connecting the origins of coordinate frames $\{A\}$ and $\{B\}$ expressed in frame $\{C\}$ coordinates in 3-D space is denoted as $p^C_{B/A} = [x_{B/A}, y_{B/A}, z_{B/A}]^\top \in \mathbb{R}^3$. Given a vector $a \in \mathbb{R}^3, S(a)$ is the skew-symmetric matrix defined according to $S(a)b = a \times b$. We further denote by $\eta_{A/B} = [\phi_{A/B}, \theta_{A/B}, \psi_{A/B}]^\top \in \mathbb{T} \subseteq \mathbb{R}^3$ the $x$-$y$-$z$ Euler angles representing the orientation of frame $\{A\}$ with respect to frame $\{B\}$, where $\mathbb{T} := (-\pi, \pi) \times (-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\pi, \pi)$; Moreover, $R^B_A \in SO(3)$ is the rotation matrix associated with the same orientation and $SO(3)$ is the 3-D rotation group. The angular velocity of frame $\{B\}$ with respect to $\{A\}$, expressed in frame $\{C\}$ coordinates, is denoted as $\omega^C_{B/A} \in \mathbb{R}^3$ and it holds that $\dot{R}^B_A = S(\omega^A_{B/A})R^B_A$ [16]. Define also the sets $\mathbb{M} := \mathbb{R}^3 \times \mathbb{T}$, $\mathcal{N} := \{1, \ldots, N\}$. We define also the set $\mathcal{O}_z := \mathcal{O}(c_z, \beta_{1,z}, \beta_{2,z}, \beta_{3,z}) := \{p \in \mathbb{R}^3 : (p - c_z)^\top P (p - c_z) \leq 1\}$, as the set of an *ellipsoid* in 3D, where $c_z \in \mathbb{R}^3$ is the center of the ellipsoid, $\beta_{1,z}, \beta_{2,z}, \beta_{3,z} \in \mathbb{R}_{>0}$ the lengths of its three semi-axes and $z \geq 1$ is an index term. The eigenvectors of matrix $P \in \mathbb{R}^{3\times 3}$ define the principal axes of the ellipsoid, and the eigenvalues of $P$ are: $\beta^{-2}_{1,z}, \beta^{-2}_{2,z}$ and $\beta^{-2}_{3,z}$. For notational brevity, when a coordinate frame corresponds to an inertial frame of reference $\{I\}$, we will omit its explicit notation (e.g., $p_B = p^I_{B/I}, \omega_B = \omega^I_{B/I}, R_A = R^I_A$). Finally, all vector and matrix differentiations will be with respect to an inertial frame $\{I\}$, unless otherwise stated.

**Definition 1.** [26] A continuous function $\alpha : [0, a) \to \mathbb{R}_{\geq 0}$ belongs to *class* $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. If $a = \infty$ and $\lim\limits_{r\to\infty} \alpha(r) = \infty$, then function $\alpha$ belongs to class $\mathcal{K}_\infty$.

## III. Problem Formulation

The formulation we adopt in this paper follows from the one from our previous work [25]. Consider a bounded and convex workspace $\mathcal{W} := \mathcal{B}(0_{3\times 1}, r_w) \subseteq \mathbb{R}^3$, where $r_w > 0$ is the workspace radius, consisting of $N$ robotic agents rigidly grasping an object, as shown in Fig. 1, and $Z$ static obstacles described by the ellipsoids $\mathcal{O}_z, z \in \mathcal{Z} := \{1, \ldots, Z\}$. The free space is denoted as $\mathcal{W}_{\text{free}} := \mathcal{W} \backslash \bigcup_{z\in\mathcal{Z}} \mathcal{O}_z$. The agents are considered to be fully actuated and they consist of a base that is able to move around the workspace (e.g., mobile or aerial vehicle) and a robotic arm. The reference frames corresponding to the $i$-th end-effector and the object's center of mass are denoted with $\{E_i\}$ and $\{O\}$, respectively, whereas $\{I\}$ corresponds to an inertial reference frame. The rigidity of the grasps implies that the agents can exert any forces/torques along every direction to the object. We consider that each agent $i$ knows the position and velocity only of its own state as well as its own and the object's geometric parameters. Moreover, no interaction force/torque measurements or on-line communication is required.

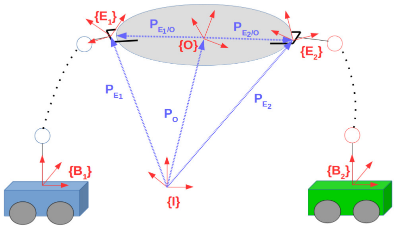

Fig. 1: Two robotic arms rigidly grasping an object with the corresponding frames.

### A. System model

*1) Robotic Agents:* We denote by $q_i \in \mathbb{R}^{n_i}$ the joint space variables of agent $i \in \mathcal{N}$, with $n_i = n_{\alpha_i} + 6$, $q_i = [p^\top_{B_i}, \eta^\top_{B_i}, \alpha^\top_i]^\top$, where $p_{B_i} = [x_{B_i}, y_{B_i}, z_{B_i}]^\top \in \mathbb{R}^3, \eta_{B_i} = [\phi_{B_i}, \theta_{B_i}, \psi_{B_i}]^\top \in \mathbb{T}$ is the position and Euler-angle orientation of the agent's base, and $\alpha_i \in \mathbb{R}^{n_{\alpha_i}}, n_{\alpha_i} > 0$, are the degrees of freedom of the robotic arm. The overall joint space configuration vector is denoted as $q := [q^\top_1, \ldots, q^\top_N]^\top \in \mathbb{R}^n$, with $n := \sum_{i\in\mathcal{N}} n_i$. The linear and angular velocities of the agents' base are described by the functions $v_{L,B_i} : \mathbb{R}^{n_i} \to \mathbb{R}^3$, with $v_{L,B_i}(\dot{q}_i) := \dot{p}_{B_i}$ and $\omega_{B_i} : \mathbb{R}^{2n_i} \to \mathbb{R}^3$, with $\omega_{B_i}(q_i, \dot{q}_i) := J_{B_i}(\eta_{B_i})\dot{\eta}_{B_i}$, where $J_{B_i} : \mathbb{T} \to \mathbb{R}^{3\times 3}$ is the representation Jacobian matrix, with $J_{B_i}(\eta_{B_i}) = \begin{bmatrix} 1 & 0 & \sin(\theta_{B_i}) \\ 0 & \cos(\phi_{B_i}) & -\cos(\theta_{B_i})\sin(\phi_{B_i}) \\ 0 & \sin(\phi_{B_i}) & \cos(\theta_{B_i})\cos(\phi_{B_i}) \end{bmatrix}$. We consider that each agent $i \in \mathcal{N}$ has access to its own state $q_i, \dot{q}_i$, and can compute, therefore, the terms $v_{L,B_i}(\dot{q}_i), \omega_{B_i}(q_i, \dot{q}_i)$.

In addition, we denote as $p_{E_i} : \mathbb{R}^{n_i} \to \mathbb{R}^3, \eta_{E_i} : \mathbb{R}^{n_i} \to \mathbb{T}$ the position and Euler-angle orientation of agent $i$'s end-effector. More specifically, it holds that: $p_{E_i}(q_i) = p_{B_i} + R_{B_i}(\eta_{B_i})k_{p_i}(\alpha_i)$, $\eta_{E_i}(q_i) = \eta_{B_i} + k_{\eta_i}(\alpha_i)$, where $k_{p_i} : \mathbb{R}^{n_{\alpha_i}} \to \mathbb{R}^3, k_{\eta_i} : \mathbb{R}^{n_{\alpha_i}} \to \mathbb{T}$ are the forward kinematics of the robotic arm [16], and $R_{B_i} : \mathbb{T} \to SO(3)$ is the rotation matrix of the agent $i$'s base.

Let also $v_i = [\dot{p}^\top_{E_i}, \omega^\top_i]^\top : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^6$ denote a function that represents the generalized velocity of agent $i$'s end-effector, with $\omega_i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^3$ being the angular velocity. Then, $v_i$ can be computed as:

$$v_i(q_i, \dot{q}_i) = \begin{bmatrix} \dot{p}_{E_i}(q_i) \\ \omega_i(q_i, \dot{q}_i) \end{bmatrix} =$$
$$\begin{bmatrix} \dot{p}_{B_i} - S(R_{B_i}(\eta_{B_i})k_{p_i}(\alpha_i))\omega_{B_i}(q_i, \dot{q}_i) + R_{B_i}(\eta_{B_i})\dfrac{\partial k_{p_i}(\alpha_i)}{\partial \alpha_i} \\ \omega_{B_i}(q_i, \dot{q}_i) + R_{B_i}(\eta_{B_i})J_{A_i}(q_i)\dot{\alpha}_i \end{bmatrix}, \tag{1}$$

where $J_{A_i} : \mathbb{R}^{n_{\alpha_i}} \to \mathbb{R}^{3\times n_{\alpha_i}}$ is the angular Jacobian of the robotic arm with respect to the agent's base [16]. The differential kinematics (1) can be written as:

$$v_i(q_i, \dot{q}_i) = J_i(q_i)\dot{q}_i, \tag{2}$$

where the Jacobian matrix $J_i : \mathbb{R}^{n_i} \to \mathbb{R}^{6\times n_i}$ is given by: $J_i(q_i) := \begin{bmatrix} I_3 & -S(R_{B_i}(\eta_{B_i})k_{p_i}(\alpha_i))J_{B_i}(\eta_{B_i}) & R_{B_i}(\eta_{B_i})\frac{\partial k_{p_i}(\alpha_i)}{\partial \alpha_i} \\ 0_{3\times 3} & J_{B_i}(\eta_{B_i}) & R_{B_i}(\eta_{B_i})J_{A_i}(q_i) \end{bmatrix}$.

**Remark 1.** Note that $J_{B_i}(\phi_{B_i}, \theta_{B_i}, \psi_{B_i})$ becomes singular at representation singularities, when $\theta_{B_i} = \pm\frac{\pi}{2}$ and $J_i(q_i)$ becomes singular at kinematic singularities defined by the set $\mathcal{Q}_i = \{q_i \in \mathbb{R}^{n_i} : \det(J_i(q_i)[J_i(q_i)]^\top) = 0\}$, $i \in \mathcal{N}$. In the following, we will aim at guaranteeing that $q_i$ will always be in the closed set: $\widetilde{\mathcal{Q}}_i = \{q_i \in \mathbb{R}^{n_i} : |\det(J_i(q_i)[J_i(q_i)]^\top)| \geq \varepsilon > 0\}$, $i \in \mathcal{N}$, for a small positive constant $\varepsilon$.

The joint-space dynamics for agent $i \in \mathcal{N}$ can be computed using the Lagrangian formulation:

$$B_i(q_i)\ddot{q}_i + N_i(q_i, \dot{q}_i)\dot{q}_i + g_{q_i}(q_i) = \tau_i - [J_i(q_i)]^\top \lambda_i, \quad (3)$$

where $B_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i \times n_i}$ is the joint-space positive definite inertia matrix, $N_i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^{n_i \times n_i}$ represents the joint-space Coriolis matrix, $g_{q_i} : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ is the joint-space gravity vector, $\lambda_i \in \mathbb{R}^6$ is the generalized force vector that agent $i$ exerts on the object and $\tau_i \in \mathbb{R}^{n_i}$ is the vector of generalized joint-space inputs, with $\tau_i = [\lambda_{B_i}^\top, \tau_{\alpha_i}^\top]^\top$, where $\lambda_{B_i} = [f_{B_i}^\top, \mu_{B_i}^\top]^\top \in \mathbb{R}^6$ is the generalized force vector on the center of mass of the agent's base and $\tau_{\alpha_i} \in \mathbb{R}^{n_{\alpha_i}}$ is the torque inputs of the robotic arms' joints. By differentiating (2) we obtain: $\dot{v}_i(q_i, \dot{q}_i) = J_i(q_i)\ddot{q}_i + \dot{J}_i(q_i)\dot{q}_i$. By inverting (3) and using (2) and $\dot{v}_i$, we can obtain the task-space agent dynamics [16]:

$$\begin{aligned} M_i(q_i)\dot{v}_i(q_i, \dot{q}_i) + C_i(q_i, \dot{q}_i)v_i(q_i, \dot{q}_i) + g_i(q_i) \\ = u_i - \lambda_i, \quad (4) \end{aligned}$$

with the corresponding task-space terms $M_i : \mathbb{R}^{n_i} \backslash \mathcal{Q}_i \to \mathbb{R}^{6\times 6}$, $C_i : \mathbb{R}^{n_i} \backslash \mathcal{Q}_i \times \mathbb{R}^{n_i} \to \mathbb{R}^{6\times 6}$, $g_i : \mathbb{R}^{n_i} \backslash \mathcal{Q}_i \to \mathbb{R}^6$: $M_i(q_i) = \left[J_i(q_i)B_i^{-1}(q_i)J_i^\top(q_i)\right]^{-1}$, $C_i(q_i, \dot{q}_i)J_i(q_i)\dot{q}_i = M_i(q_i)\left[J_i(q_i)B_i^{-1}(q_i)N_i - \dot{J}_i(q_i)\right]\dot{q}_i$, $g_i(q_i) = M_i(q_i)J_i(q_i)B_i^{-1}(q_i)g_{q_i}(q_i)$. The task-space input wrench $u_i$ can be translated to the joint space inputs $\tau_i \in \mathbb{R}^{n_i}$ via $\tau_i = [J_i(q_i)]^\top u_i + \bar{\tau}_i(q_i)$, where $\bar{\tau}_i$ belongs to the nullspace of $[J_i(q_i)]^\top$ and concerns over-actuated agents [16]. We define by $\mathcal{A}_i : \mathbb{R}^{n_i} \rightrightarrows \mathbb{R}^3, i \in \mathcal{N}$, the union of the ellipsoids that bound the $i$-th agent's volume, i.e., which is essentially the union of the ellipsoids that bound the volume of the agents' links.

*2) Object Dynamics:* Regarding the object, we denote its state as $x_O \in \mathbb{M}$, $v_O = [v_{L,O}^\top, \omega_O^\top]^\top \in \mathbb{R}^6$, representing the pose and velocity of the object's center of mass, with $x_O = [p_O^\top, \eta_O^\top]^\top$, $p_O \in \mathbb{R}^3$, $\eta_O = [\phi_O, \theta_O, \psi_O]^\top \in \mathbb{T}$. The second order Newton-Euler dynamics of the object are given by:

$$\dot{x}_O = [J_{O_r}(\eta_O)]^{-1}v_O, \quad (5a)$$

$$\lambda_O = M_O(x_O)\dot{v}_O + C_O(x_O, v_O)v_O + g_O(x_O), \quad (5b)$$

where $M_O : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is the positive definite inertia matrix, $C_O : \mathbb{M} \times \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ is the Coriolis matrix, and $g_O : \mathbb{M} \to \mathbb{R}^6$ is the gravity vector. In addition, $J_{O_r} : \mathbb{T} \to \mathbb{R}^{6\times 6}$ is the object representation Jacobian $J_{O_r}(\eta_O) := \text{diag}\{I_3, J_{O_{r,\theta}}(\eta_O)\}$, with $J_{O_{r,\theta}}(\eta_O) = \begin{bmatrix} 1 & 0 & \sin(\theta_O) \\ 0 & \cos(\phi_O) & -\cos(\theta_O)\sin(\phi_O) \\ 0 & \sin(\phi_O) & \cos(\theta_O)\cos(\phi_O) \end{bmatrix}$, which is singular when $\theta_O = \pm\frac{\pi}{2}$. Finally, $\lambda_O \in \mathbb{R}^6$ is the force vector acting on the object's center of mass. Also, similarly to the robotic agents, we define by $\mathcal{C}_O : \mathbb{M} \rightrightarrows \mathbb{R}^3$ the bounding ellipsoid of the object.

*3) Coupled Dynamics:* Consider $N$ robotic agents rigidly grasping an object. Then, the coupled system object-agents behaves like a closed-chain robot and we can express the object's pose and velocity as a function of $q_i$ and $\dot{q}_i$, $\forall i \in \mathcal{N}$. Hence, In view of Fig. 1, we conclude that:

$$\begin{aligned} p_O = p_{O_i}(q_i) &:= p_{E_i}(q_i) + p_{O/E_i}(q_i) \\ &:= p_{E_i}(q_i) + R_{E_i}(q_i)p_{O/E_i}^{E_i}, \quad (6a) \end{aligned}$$

$$\eta_O = \eta_{O_i}(q_i) = \eta_{E_i}(q_i) + \eta_{O/E_i}, \quad (6b)$$

for every $i \in \mathcal{N}$, where $p_{O_i} : \mathbb{R}^{n_i} \to \mathbb{R}^3, \eta_{O_i} : \mathbb{R}^{n_i} \to$ are local functions of the agents that provide the object's pose, $p_{O/E_i}^{E_i}$ represents the constant distance and $\eta_{O/E_i}$ the relative orientation offset between the $i$th agent's end-effector and the object's center of mass, which are considered known. The grasp rigidity implies that $\omega_{E_i}(q_i) = \omega_O$, $\forall i \in \mathcal{N}$. Therefore, by differentiating (6a), we can also express $v_O$ as a function of $q_i, \dot{q}_i$ as

$$v_O = v_{O_i}(q_i, \dot{q}_i) := J_{i_O}(q_i)v_i(q_i, \dot{q}_i), \quad (7)$$

from which, we obtain: $\dot{v}_{O_i}(q_i, \dot{q}_i) = J_{i_O}(q_i)\dot{v}_i(q_i, \dot{q}_i) + \dot{J}_{i_O}(q_i)v_i(q_i, \dot{q}_i)$, where $J_{i_O} : \mathbb{R}^{n_i} \to \mathbb{R}^{6\times 6}$ is a smooth mapping representing the Jacobian from the object to the $i$-th agent: $J_{i_O}(q_i) = \begin{bmatrix} I_3 & S(p_{E_i/O}(q_i)) \\ 0_{3\times 3} & I_3 \end{bmatrix}$, and is always full rank due to the grasp rigidity.

**Remark 2.** Since the geometric object parameters $p_{O/E_i}^{E_i}$ and $\eta_{O/E_i}$ are known, each agent can compute $p_O(q_i), \eta_O(q_i)$ and $v_O(q_i, \dot{q}_i)$ by (6) and (7), respectively, without employing any sensory data. In the same vein, all agents can also compute the object's bounding ellipsoid $\mathcal{C}_O(x_O(q_i))$.

The Kineto-statics duality [16] along with the grasp rigidity suggest that the force $\lambda_O$ acting on the object center of mass and the generalized forces $\lambda_i, i \in \mathcal{N}$, exerted by the agents at the contact points are related through:

$$\lambda_O = [G(q)]^\top \lambda, \quad (8)$$

where $\lambda = [\lambda_1^\top, \cdots, \lambda_N^\top]^\top \in \mathbb{R}^{6N}$ and $G : \mathbb{R}^n \to \mathbb{R}^{6N\times 6}$ is the grasp matrix, with $G(q) = [[J_{O_1}(q_1)]^\top, \cdots, [J_{O_N}(q_N)]^\top]^\top$ and $J_{O_i}(q_i) := [J_{i_O}(q_i)]^{-1}$ the matrix inverse of $J_{i_O}(q_i)$, $i \in \mathcal{N}$.

Consider now the constants $c_i$, with $0 < c_i < 1$ and $\sum\limits_{i\in\mathcal{N}} c_i = 1$, that play the role of load sharing coefficients for the agents. Then (5b) can be written as: $\sum\limits_{i\in\mathcal{N}} c_i \Big\{ M_O(x_{O_i}(q_i))v_{O_i}^d(q_i, \dot{q}_i, \ddot{q}_i) + g_O(x_{O_i}(q_i))C_O(x_{O_i}(q_i), v_{O_i}(q_i, \dot{q}_i))v_{O_i}(q_i, \dot{q}_i) \Big\} = \sum\limits_{i\in\mathcal{N}} [J_{O_i}(q_i)]^\top \lambda_i$, from which, by employing (8), (2), (7), and after straightforward algebraic manipulations, we obtain the coupled dynamics

$$\sum_{i\in\mathcal{N}} \Big\{ \widetilde{M}_i(q_i)\ddot{q}_i + \widetilde{C}_i(q_i, \dot{q}_i)\dot{q}_i + \widetilde{g}_i(q_i) \Big\} = \sum_{i\in\mathcal{N}} [J_{O_i}(q_i)]^\top u_i, \quad (9)$$

where:

$$\widetilde{M}_i(q_i) \coloneqq c_i M_O(x_{O_i}(q_i)) J_{i_O}(q_i) J_i(q_i) + [J_{O_i}(q_i)]^\top M_i(q_i) J_i(q_i),$$
$$\widetilde{C}_i(q_i, \dot{q}_i) \coloneqq [J_{O_i}(q_i)]^\top \Big( M_i(q_i) \dot{J}_i(q_i) + C_i(q_i, \dot{q}_i) J_i(q_i) \Big) +$$
$$c_i M_O(x_{O_i}(q_i)) J_{i_O}(q_i) \dot{J}_i(q_i) + c_i M_O(x_{O_i}(q_i)) \dot{J}_{i_O}(q_i) J_i(q_i),$$
$$+ c_i C_O(x_{O_i}(q_i), v_{O_i}(q_i, \dot{q}_i)),$$
$$\widetilde{g}_i(q_i) \coloneqq c_i g_O(x_{O_i}(q_i)) + [J_{O_i}(q_i)]^\top g_i(q_i), i \in \mathcal{N},$$

and $x_{O_i} \coloneqq [\dot{p}_{O_i}^\top, \eta_{O_i}^\top]^\top \in \mathbb{M}$, $\forall i \in \mathcal{N}$.

**Problem 1.** Consider $N$ robotic agents, rigidly grasping an object, governed by the coupled dynamics (9). Given a desired pose $x_{\text{des}}$ for the object, design the control inputs $u_i \in \mathbb{R}^{6N}$ such that $\lim\limits_{t \to \infty} \|x_O(t) - x_{\text{des}}\| \to 0$, while ensuring the satisfaction of the following collision avoidance and singularity properties:

1) $\mathcal{A}_i(q_i(t)) \cap \mathcal{O}_z = \emptyset, \forall i \in \mathcal{N}, z \in \mathcal{Z}$,
2) $\mathcal{C}_O(x_O(t)) \cap \mathcal{O}_z = \emptyset, \forall z \in \mathcal{Z}$,
3) $\mathcal{A}_i(q_i(t)) \cap \mathcal{A}_j(q_j(t)) = \emptyset, \forall i, j \in \mathcal{N}, i \neq j$,
4) $-\frac{\pi}{2} < -\bar{\theta} \le \theta_O(t) \le \bar{\theta} < \frac{\pi}{2}$, $\forall i \in \mathcal{N}$,
5) $-\frac{\pi}{2} < -\bar{\theta} \le \theta_{B_i}(t) \le \bar{\theta} < \frac{\pi}{2}$, $\forall i \in \mathcal{N}$,
6) $q_i \in \widetilde{\mathcal{Q}}_i$.

$\forall t \in \mathbb{R}_{\ge 0}$, for $0 < \bar{\theta} < \frac{\pi}{2}$, as well as the velocity and input constraints: $|\tau_{i_k}(t)| \le \bar{\tau}_i, |\dot{\tau}_{i_k}(t)| \le \bar{\dot{\tau}}_i, |\dot{q}_{i_k}(t)| \le \bar{\dot{q}}_i, \forall k \in \{1, \ldots, n_i\}, i \in \mathcal{N}$, for some positive constants $\bar{\tau}_i, \bar{\dot{q}}_i, i \in \mathcal{N}$.

Regarding the joint velocity constraints, we impose for the arm joint velocities the constraint $\|\dot{\alpha}_i(t)\| \le 1, \forall i \in \mathcal{N}$, which maximizes the manipulability ellipsoid of the arms [16], and hence increases robotic manipulability. Specifications $1) - 3)$ in the aforementioned problem stand for collision avoidance between the agents, the objects, and the workspace obstacles and specifications $4) - 6)$ stand for representation and kinematic singularities.

In order to solve the aforementioned problem, we need the following assumption regarding the workspace and the agent communication:

**Assumption 1.** (Problem feasibility) The set $\{q \in \mathbb{R}^n : \mathcal{A}_i(q_i) \cap \mathcal{O}_z = \emptyset, \mathcal{A}_i(q_i) \cap \mathcal{A}_\ell(q_\ell) = \emptyset, \mathcal{C}_i(x_{O_i}(q_i)) \cap \mathcal{O}_z = \emptyset, \forall i, \ell \in \mathcal{N}, i \neq \ell, z \in \mathcal{Z}\}$, is connected.

**Assumption 2.** (Sensing and communication capabilities) Each agent $i \in \mathcal{N}$ is able to continuously measure the other agents' state $q_j, \dot{q}_j, j \in \mathcal{N} \backslash \{i\}$. Moreover, each agent $i \in \mathcal{N}$ is able to communicate with the other agents $j \in \mathcal{N} \backslash \{i\}$ without any delays.

Note that the aforementioned sensing assumption is reasonable, since in cooperative manipulation tasks, the agents are sufficiently close to each other, and therefore potential sensing radii formed by realistic sensors are large enough to cover them. Moreover, each agent $i \in \mathcal{N}$ can construct at every time instant the set-valued functions $\mathcal{A}_j(q_j)$, $\forall j \in \mathcal{N} \backslash \{i\}$, whose structure can be transmitted off-line to all agents.

Define also the sets: $\mathcal{S}_{i,O} \coloneqq \{q_i \in \mathbb{R}^{n_i} : \mathcal{A}_i(q_i) \cap \mathcal{O}_z \neq \emptyset, \forall z \in \mathcal{Z}\}$, $\mathcal{S}_{i,A} \coloneqq \{q \in \mathbb{R}^n : \mathcal{A}_i(q_i) \cap \mathcal{A}_j(q_\ell) \neq \emptyset, \forall \ell \in \mathcal{N} \backslash \{i\}\}$, as well as $\mathcal{S}_{O_i} \coloneqq \{q_i \in \mathbb{R}^{n_i} : \mathcal{C}_O(x_{O_i}(q_i)) \cap \mathcal{O}_z \neq \emptyset\}$, $\forall i \in \mathcal{N}$, associated with the desired collision-avoidance properties. Moreover, define the projection sets for agent $i$ as the set-valued functions $\widetilde{\mathcal{S}}_{i,A}([q_\ell]_{\ell \in \mathcal{N} \backslash \{i\}}) \coloneqq \{q_i \in \mathbb{R}^{n_i} : q \in \mathcal{S}_{i,A}\}$, $\forall i \in \mathcal{N}$, where the notation $[q_\ell]_{\ell \in \mathcal{N} \backslash \{i\}}$ stands for the stack vector of all $q_\ell, \ell \in \mathcal{N} \backslash \{i\}$.

## IV. Main Results

In this section, a systematic solution to Problem 1 is introduced. Our overall approach builds on designing a NMPC scheme for the system of the manipulators and the object. The proposed methodology is decentralized, since we do not consider a centralized system that calculates all the control signals and transmits them to the agents, like in our previous work [25]. To achieve that, we employ a leader-follower perspective. More specifically, as will be explained in the sequel, at each sampling time, a leader agent solves part of the coupled dynamics (9) via an NMPC scheme, and transmits its predicted variables to the rest of the agents. Assume, without loss of generality, that the leader corresponds to agent $i = 1$. Loosely speaking, the proposed solution proceeds as follows: agent 1 solves, at each sampling time step, the receding horizon model predictive control subject to the forward dynamics:

$$\widetilde{M}_1(q_1)\ddot{q}_1 + \widetilde{C}_1(q_1, \dot{q}_1)\dot{q}_1 + \widetilde{g}(q_1) = [J_{O_1}(q_1)]^\top u_1, \quad (10)$$

and a number of inequality constraints, as will be clarified later. After obtaining a control input sequence and a set of predicted variables for $q_1, \dot{q}_1$, denoted as $\hat{q}_1, \hat{\dot{q}}_1$, it transmits the corresponding predicted state for the object $x_{O_1}(\hat{q}_1), v_{O_1}(\hat{q}_1, \hat{\dot{q}}_1)$ for the control horizon to the other agents $\{2, \ldots, N\}$. Then, the followers solve the receding horizon NMPC subject to the forward dynamics:

$$\widetilde{M}_i(q_i)\ddot{q}_i + \widetilde{C}_i(q_i, \dot{q}_i)\dot{q}_i + \widetilde{g}(q_i) = [J_{O_i}(q_i)]^\top u_i, \quad (11)$$

the state equality constraints:

$$x_{O_i}(q_i) = x_{O_1}(\hat{q}_1), v_{O_i}(q_i, \dot{q}_i) = v_{O_1}(\hat{q}_1, \hat{\dot{q}}_1), \quad (12)$$

$i \in \{2, \ldots, N\}$ as well as a number of inequality constraints that incorporate obstacle and inter-agent collision avoidance. More specifically, we consider that there is a priority sequence among the agents, which we assume, without loss of generality, that is defined by $\{1, \ldots, N\}$. Each agent, after solving its optimization problem, transmits its calculated predicted variables to the agents of lower priority, which take them into account for collision avoidance. Note that the coupled object-agent dynamics are implicitly taken into account in equations (10), (11) in the following sense. Although the coupled model (9) does not imply that each one of these equations is satisfied, by forcing each agent to comply with the specific dynamics through the optimization procedure, we guarantee that (9) is satisfied, since it's the result of the addition of (10) and (11), for every $i = 1$ and $i \in \{2, \ldots, N\}$, respectively. Intuitively, the leader agent is the one that determines the path that the object will navigate through, and the rest of the agents are the followers that contribute to the transportation. Moreover, the equality constraints (12) guarantee that the predicted variables of the agents $\{2, \ldots, N\}$ will comply with the rigidity at the grasping points through the equality constraints (12).

By using the notation $x_i := [x_{i_1}^\top, x_{i_2}^\top]^\top := [q_i^\top, \dot{q}_i^\top]^\top \in \mathbb{R}^{2n_i}$, $i \in \mathcal{N}$, the nonlinear dynamics of each agent can be written as:

$$\dot{x}_i = f_i(x_i, u_i) := \begin{bmatrix} f_{i_1}(x_i) \\ f_{i_2}(x_i, u_i) \end{bmatrix}, \tag{13}$$

where $f_i : E_i \times \mathbb{R}^6 \to \mathbb{R}^{2n_i}$ is the locally Lipschitz function: $f_{i_1}(x_i, u_i) = x_{i_2}$, $f_{i_2}(x_i, u_i) = \widehat{M}_i(q_i)\Big([J_{O_i}(q_i)]^\top u_i - \widetilde{C}_i(q_i, \dot{q}_i)\dot{q} - \widetilde{g}_i(q_i)\Big)$, $i \in \mathcal{N}$, where $\widehat{M}_i : \mathbb{R}_{n_i} \backslash \mathcal{Q}_i \to \mathbb{R}^{n_i \times 6}$, is the pseudo-inverse $\widehat{M}_i(q_i) := \widetilde{M}_i(q_i)\Big(\widetilde{M}_i(q_i)[\widetilde{M}_i(q_i)]^\top\Big)^{-1}$, and $E_i := \mathbb{R}^{n_i} \backslash \mathcal{Q}_i \times \mathbb{R}^{n_i}$, $\forall i \in \mathcal{N}$. It can be proved that in the set $\mathbb{R}^{n_i} \backslash \mathcal{Q}_i$ the matrix $\widetilde{M}_i(q_i)[\widetilde{M}_i(q_i)]^\top$ has full rank and hence, $\widehat{M}_i(q_i)$ is well defined for all $q \in \mathbb{R}^{n_i} \backslash \mathcal{Q}_i$. We define then the error vector $e_1 : E_1 \to \mathbb{M} \times \mathbb{R}^6$, as: $e_1(x_1) := \begin{bmatrix} x_{O_1}(q_1) - x_{\text{des}} \\ v_{O_1}(q_1, \dot{q}_1), \end{bmatrix}$ which gives us the *error dynamics*:

$$\dot{e}_1 = g_1(x_1, u_1), \tag{14}$$

with $g_1 : E_1 \times \mathbb{R}^6 \to \mathbb{R}^{2n_i}$: $g_1(x_1, u_1) := \begin{bmatrix} [J_{O_r}(\eta_{O_1}(q_1))]^{-1} J_{1_O}(q_1) J_1(q_1)\dot{q}_1 \\ J_{1_O}(q_1)J_1(q_1)f_{1_2}(x_1, u_1) + \Big(J_{1_O}\dot{J}_1(q_1) + \dot{J}_{1_O}(q_1)J_1(q_1)\Big)\dot{q}_1. \end{bmatrix}$, where we employed (14) and (5a).

**Remark 3.** It can be concluded that $g_1(\cdot, u_1)$ is *Lipschitz continuous* in $E_1$ since it is continuously differentiable in its domain. Thus, for every $x_1, x_1' \in E_1$, with $x_1 \neq x_1'$, there exists a Lipschitz constant $L_g$ such that: $|g(x_1, u) - g(x_1', u)| \leq L_g \|x_1 - x_1'\|$.

The time derivative of joint space inputs is given by: $\dot{\tau}_i = [\dot{J}_i(q_i)]^\top u_i + [J_i(q_i)]^\top \dot{u}_i$. Hence, the constraints for $\tau_{i_k}$ and $\dot{\tau}_{i_k}$, $k \in \mathbb{R}^{n_i}$,$i \in \mathcal{N}$, can be written as coupled state-input constraints: $\|\tau_i\| \leq \bar{\tau}_i \Leftrightarrow \|[J(q_i)]^\top u_i\| \leq \bar{\tau}_i$, $\|\dot{\tau}_i\| \leq \bar{\dot{\tau}}_i \Leftrightarrow \|[\dot{J}_i(q_i)]^\top u_i + [J_i(q_i)]^\top \dot{u}_i\| \leq \bar{\dot{\tau}}_i$. Let us now define the following sets $U_i \subseteq \mathbb{R}^{6 \times 6 \times (2n_i)}$:

$$U_i := \Big\{(u_i, \dot{u}_i, x_i) \in \mathbb{R}^{6 \times 6 \times (2n_i)} : \|[J(q_i)]^\top u_i\| \leq \bar{\tau}_i, \\ \|[\dot{J}_i(q_i)]^\top u_i + [J_i(q_i)]^\top \dot{u}_i\| \leq \bar{\dot{\tau}}_i\Big\}, i \in \mathcal{N}, \tag{15}$$

as the sets that capture the control input constraints of (13), as well as their projections

$$U_{i,u} := \left\{u_i \in \mathbb{R}^6 : (u_i, \dot{u}_i, x_i) \in U_i\right\}, i \in \mathcal{N}. \tag{16}$$

Define also the set-valued functions $X_i : \mathbb{R}^{n-n_i} \rightrightarrows \mathbb{R}^{2n_i}$, $i \in \mathcal{N}$, by: $X_1([q_\ell]_{\ell \in \{2,\ldots,N\}}) := \Big\{x_1 \in \mathbb{R}^{2n_1} : \theta_{O_1}(q_1) \in [-\bar{\theta}, \bar{\theta}]$, $\theta_{B_1} \in [-\bar{\theta}, \bar{\theta}]$, $|\dot{q}_{k_1}| \leq \bar{\dot{q}}_1$, $q_1 \in \widetilde{\mathcal{Q}}_1 \backslash \Big(\mathcal{S}_{1,O} \cup \widetilde{\mathcal{S}}_{1,A}([q_\ell]_{\ell \in \{2,\ldots,N\}})\Big)$, $x_{O_1}(q_1) \in \mathbb{R}^3 \backslash S_{O_1}\Big\}$, $X_i([q_\ell]_{\ell \in \mathcal{N} \backslash \{i\}}) := \Big\{x_i \in \mathbb{R}^{2n_i} : \theta_{B_i} \in [-\bar{\theta}, \bar{\theta}], |\dot{q}_{k_i}| \leq \bar{\dot{q}}_i, q_i \in \widetilde{\mathcal{Q}}_i \backslash (\mathcal{S}_{i,O} \cup \mathcal{S}_{i,A}([q_\ell]_{\ell \in \mathcal{N} \backslash \{i\}}))\Big\}$, $i \in \{2, \ldots, N\}$. Note that $q_i \in X_i([q_\ell]_{\ell \in \mathcal{N} \backslash \{i\}}) \implies q_i \notin \mathcal{Q}_i$, $\forall i \in \mathcal{N}$.

The sets $X_i$ capture all the state constraints of the system dynamics (13), i.e., representation- and singularity-avoidance, collision avoidance among the agents and the obstacles, as well as collision avoidance of the object with the obstacles, which is assigned to the leader agent only. We further define the set-valued functions $\mathcal{E}_1 : \mathbb{R}^{n-n_1} \rightrightarrows \mathbb{M} \times \mathbb{R}^6$ as $\mathcal{E}_1([q_\ell]_{\ell \in \{2,\ldots,N\}}) := \{e_1(x_1) \in \mathbb{M} \times \mathbb{R}^6 : x_1 \in X_1([q_\ell]_{\ell \in \{2,\ldots,N\}})\}$.

The main problem at hand is the design of a *feedback control law* $u_1 \in U_1$ for agent 1 which guarantees that the error signal $e_1$ with dynamics given in (14), satisfies $\lim_{t \to \infty} \|e_1(x_1(t))\| \to 0$, while ensuring singularity avoidance, collision avoidance between the agents, between the agents and the obstacles as well as the object and the obstacles. The role of the followers $\{2, \ldots, N\}$ is, through the load-sharing coefficients $c_2, \ldots, c_N$ in (9), to contribute to the object trajectory execution, as derived by the leader agent 1. In order to solve the aforementioned problem, we propose a NMPC scheme, that is presented hereafter.

Consider a sequence of sampling times $\{t_j\}$, $j \in \mathbb{N}$ with a constant sampling period $h$, $0 < h < T_p$, where $T_p$ is the prediction horizon, such that: $t_{j+1} = t_j + h$, $j \in \mathbb{N}$. Hereafter we will denote by $j$ the sampling instant. In sampled-data NMPC, a FHOCP is solved at the discrete sampling time instants $t_j$ based on the current state error information $e_1(x_1(t_j))$. The solution is an optimal control signal $\hat{u}_1^\star(s)$, computed over $s \in [t_j, t_j + T_p]$. For agent 1, the open-loop input signal applied in between the sampling instants is given by the solution of the following FHOCP:

$$\min_{\hat{u}_1(\cdot)} J_1(e_1(x_1(t_j)), \hat{u}_1(\cdot)) = \min_{\hat{u}_1(\cdot)} \Bigg\{ V_1(e_1(\hat{x}_1(t_j + T_p))) \\ + \int_{t_j}^{t_j + T_p} \Big[F_1(e_1(\hat{x}_1(s)), \hat{u}_1(s))\Big] ds \Bigg\} \tag{17a}$$

subject to:

$$\dot{e}(\hat{x}_1(s)) = g_1(\hat{x}_1(s), \hat{u}_1(s)), \; e_1(\hat{x}_1(t_j)) = e_1(x_1(t_j)), \tag{17b}$$

$$e_1(\hat{x}_1(s)) \in \mathcal{E}_1([q_\ell(t_j)]_{\ell \in \{2,\ldots,N\}}), s \in [t_j, t_j + T_p], \tag{17c}$$

$$(\hat{u}_1(s), \dot{\hat{u}}_1(s), \hat{x}_1(s)) \in U_1, s \in [t_j, t_j + T_p], \tag{17d}$$

$$e_1(\hat{x}_1(t_j + T_p)) \in \mathcal{F}_1([q_\ell]_{\ell \in \{2,\ldots,N\}}). \tag{17e}$$

At a generic time $t_j$ then, agent 1 solves the aforementioned FHOCP. The notation $\hat{(\cdot)}$ is used to distinguish the predicted variables which are internal to the controller, corresponding to the system (17b). This means that $e_1(\hat{x}_1(\cdot))$ is the solution of (17b) driven by the control input $\hat{u}_1(\cdot) : [t_j, t_j + T_p] \to U_1$ with initial condition $e_1(x_1(t_j))$. Note that, since the prediction horizon is finite, the predicted values are not the same with the actual closed-loop values (see [18]). In the following, we use the notation $\mathcal{E}_1(\cdot)$ instead of $\mathcal{E}_1([q_\ell]_{\ell \in \{2,\ldots,N\}})$ for brevity. The functions $F_1 : \mathcal{E}_1(\cdot) \times U_{1,u} \to \mathbb{R}_{\geq 0}$, $V_1 : \mathcal{E}_1(\cdot) \to \mathbb{R}_{\geq 0}$ stand for the *running cost* and the *terminal penalty cost*, respectively, and they are defined as: $F_1\big(e_1, u_1\big) = e_1^\top Q_1 e_1 + u_1^\top R_1 u_1$, $V_1\big(e_1\big) = e_1^\top P_1 e_1$; $R_1 \in \mathbb{R}^{6 \times 6}$ and $P_1 \in \mathbb{R}^{(2n_1) \times (2n_1)}$ are symmetric and positive definite controller gain matrices to be appropriately tuned; $Q_1 \in \mathbb{R}^{(2n_1) \times (2n_1)}$ is a symmetric and positive semi-definite controller gain matrix to be appropriately tuned. The *terminal* set $\mathcal{F}_1$ will be defined later. For the running and terminal penalty costs $F_1$ and $V_1$, respectively, the following hold:

**Lemma 1.** *There exist functions* $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ *such that:* $\alpha_1\big(\|z\|\big) \leq F_1\big(e_1, u_1\big) \leq \alpha_2\big(\|z\|\big)$, *for every* $z := \left[e_1^\top, u_1^\top\right]^\top \in \mathcal{E}_1(\cdot) \times U_{1,u}$.

*Proof.* The proof can be found in [24, App. C, p. 22]. □

**Lemma 2.** *The running cost* $F_1$ *is Lipschitz continuous in* $\mathcal{E}_1(\cdot) \times U_{1,u}$. *Thus, it holds that:* $\big|F_1(e_1, u_1) - F_1(e_1', u_1)\big| \leq L_{F_1}\|e_1 - e_1'\|, \forall e_1, e_1' \in \mathcal{E}_1(\cdot), u_1 \in U_{1,u}$, *where* $L_{F_1} := 2\sigma_{\max}(Q_1) \sup\limits_{e_1 \in \mathcal{E}_1(\cdot)} \|e_1\|$.

*Proof.* The proof can be found in [24, App. D, p. 22]. □

The *terminal set* $\mathcal{F}_1(\cdot) \subseteq \mathcal{E}_1(\cdot)$ is chosen as: $\mathcal{F}_1([q_\ell]_{\ell \in \{2,\dots,N\}}) = \{e_1 \in \mathcal{E}_1([q_\ell]_{\ell \in \{2,\dots,N\}}) : V_1(e_1) \leq \epsilon_1\}$, where $\epsilon_1 \in \mathbb{R}_{>0}$ is an arbitrarily small constant to be appropriately tuned. The terminal set is used to enforce the stability of the closed-loop system. For the terminal penalty costs $V_1$, the following holds:

**Lemma 3.** *The terminal penalty function* $V_1$, *is Lipschitz continuous in* $\mathcal{F}_1(\cdot)$, *thus it holds that:* $\big|V_1(e_1) - V_1(e_1')\big| \leq L_{V_1}\|e_1 - e_1'\|, \forall e_1, e_1' \in \mathcal{F}_1(\cdot)$, *where* $L_{V_1} := 2\sigma_{\max}(P_1) \sup\limits_{e_1 \in \mathcal{F}_1} (\cdot)\|e_1\|$.

*Proof.* The proof is similar to the proof of Lemma 2 and it is omitted. □

The solution to FHOCP (17a) - (17e) at time $t_j$ provides an optimal control input, denoted by $\hat{u}_1^\star(s;\ e_1(x_1(t_j)), x_1(t_j))$, $s \in [t_j, t_j + T_p]$. This control input is then applied to the system until the next sampling instant $t_{j+1}$:

$$u_1\left(s;\ x_1(t_j), e_1(x_1(t_j))\right) = \hat{u}_1^\star\left(s;\ x_1(t_j), e_1(x_1(t_j))\right), \tag{18}$$

for every $s \in [t_j, t_j + h)$. At time $t_{j+1} = t_j + h$ a new FHOCP is solved in the same manner, leading to a receding horizon approach. The control input $u_1(\cdot)$ is of feedback form, since it is recalculated at each sampling instant based on the then-current state. The solution of (14) at time $s$, $s \in [t_j, t_j + T_p]$, starting at time $t_j$, from an initial condition $x_1(t_j), e_1(x_1(t_j))$, by application of the control input $u_1 : [t_j, s] \to U_{1,u}$ is denoted by $e_1\big(x_1(s);\ u_1(\cdot);\ x_1(t_j), e_1(x_1(t_j))\big)$, $s \in [t_j, t_j + T_p]$. The *predicted* state of the system (17b) at time $s$, $s \in [t_j, t_j + T_p]$ based on the measurement of the state at time $t_j$, $x_1(t_j)$, by application of the control input $u_1\big(t;\ x_1(t_j), e_1(x_1(t_j))\big)$ as in (18), is denoted by $\hat{x}_1\big(s;\ u_1(\cdot);\ x_1(t_j), e_1(x_1(t_j))\big)$, and the corresponding predicted error by $e_1\big(\hat{x}_1(\cdot);\ u_1(\cdot);\ x_1(t_j), e_1(x_1(t_j))\big)$, $s \in [t_j, t_j + T_p]$.

After the solution of the FHOCP and the calculation of the predicted states $\hat{x}_1\big(s;\ u_1(\cdot), e_1(x_1(t_j)), x_1(t_j)\big)$, $s \in [t_j, t_j + T_p]$ at each time instant $t_j$, agent 1 transmits the values $\hat{q}_1(s, \cdot)$, $\hat{\dot{q}}_1(s, \cdot)$ as well as $x_{O_1}(\hat{q}_1(s, \cdot))$ and $v_{O_1}(\hat{q}_1(s, \cdot), \hat{\dot{q}}_1(s, \cdot))$, as computed by (6), (7), $\forall s \in [t_j, t_j + T_p]$ to the rest of the agents $\{2, \dots, N\}$. The rest of the agents then proceed as follows. Each agent $i \in \{2, \dots, N\}$, solves the following FHOCP:

$$\min_{\hat{u}_i(\cdot)} J_i(x_i(t_j)), \hat{u}_i(\cdot)) \tag{19a}$$

subject to:

$$\dot{x}_i = f_i(x_i(s), u_i(s)), \tag{19b}$$

$$x_i(s) \in X_i\Big([q_\ell(t_j)]_{\ell \in \{i+1,\dots,N\}}\Big), \tag{19c}$$

$$x_i(s) \in X_i\Big([\hat{q}_\ell(s, \cdot)]_{j \in \{1,\dots,i-1\}}\Big), \tag{19d}$$

$$x_{O_i}(q_i(s)) = x_{O_1}(\hat{q}_1(s; \cdot)), \tag{19e}$$

$$v_{O_i}(q_i(s), \dot{q}_i(s)) = v_{O_1}(\hat{q}_1(s; \cdot), \hat{\dot{q}}_1(s; \cdot)), \tag{19f}$$

$$(u_i(s), \dot{u}_i(s), x_i(s)) \in U_i, s \in [t_j, t_j + T_p], \tag{19g}$$

at every sampling time $t_j$. Note that, through the equality constraints (19e), (19f), the follower agents must comply with the trajectory computed by the leader $\hat{q}_1(s, \cdot), \hat{\dot{q}}_1(s, \cdot)$. This can be problematic in the sense that this trajectory might drive the followers to collide with an obstacle or among each other. Resolution of such cases is not in the scope of this paper and constitutes part of future research. Here, we assume here that there are no such cases:

**Assumption 3.** The sets $\{(q, s) \in \mathbb{R}^n \times [t_j, t_j + T_p] : x_{O_i}(q_i(s)) = x_{O_1}(\hat{q}_1(s; \cdot)), v_{O_i}(q_i(s), \dot{q}_i(s)) = v_{O_1}(\hat{q}_1(s; \cdot), \hat{\dot{q}}_1(s; \cdot)) \cap \mathcal{S}_{i,O} \cap \widetilde{\mathcal{S}}_{i,A}([q_\ell(t_j)]_{\ell \in \{i+1,\dots,N\}} \cap \widetilde{\mathcal{S}}_{i,A}([q_\ell(s)]_{\ell \in \{1,\dots,i-1\}})\}$ are nonempty, $\forall i \in \{2, \dots, N\}$.

Next, similarly to the leader agent $i = 1$, it calculates the predicted states $\hat{q}_i(s, \cdot), \hat{\dot{q}}_i(s, \cdot), s \in [t_j, t_j + T_p]$, which then transmits to the agents $\{i + 1, \dots, N\}$. In that way, at each time instant $t_j$, each agent $i \in \{2, \dots, N\}$ measures the other agents' states (as stated in Assumption 2), incorporates the constraint (19c) for the agents $\{i + 1, \dots, N\}$, receives the predicted states $\hat{q}_\ell(s, \cdot), \hat{\dot{q}}_\ell(s, \cdot)$ from the agents $\ell \in \{2, \dots, i - 1\}$ and incorporates the collision avoidance constraint (19d) for the entire horizon. Loosely speaking, we consider that each agent $i \in \mathcal{N}$ takes into account the first state of the next agents in priority $(q_\ell(t_j), \ell \in \{i+1, \dots, N\})$, as well as the transmitted predicted variables $\hat{q}_\ell(s, \cdot), \ell \in \{1, \dots, i-1\}$ of the previous agents in priority, for collision avoidance. Intuitively, the leader agent executes the planning for the followed trajectory of the object's center of mass (through the solution of the FHOCP (17a)-(17e)), the follower agents contribute in executing this trajectory through the load sharing coefficients $c_i$ (as indicated in the coupled model (9)), and the agents low in priority are responsible for collision avoidance with the agents of higher priority. Moreover, the aforementioned equality constraints (19e), (19f) as well as the forward dynamics (19a) guarantee the compliance of all the followers with the model (9). For the followers, the cost $J_i(x_i(t_j), \hat{u}_i(\cdot))$ can be selected as any function of $x_i, u_i$, $\forall i \in \{2, \dots, N\}$.

Therefore, given the constrained FHOCP (19a)-(19g), the solution of problem lies in the capability of the leader agent to produce a state trajectory that guarantees $x_{O_1}(q_1(t)) \to x_{\text{des}}$, by solving the FHOCP (17a)-(17e), which is discussed in Theorem 1.

**Remark 4.** Note that, if the satisfaction of the equality constraints (19e), (19f) guarantees that there is no collision among the agents (e.g., in the case that two agents grasp a large object from two symmetrical - with respect to the object's center of mass - grasping points), then the transmission of the predicted variables among the follower agents $\{2, \dots, N\}$ is not needed. In that case, the followers can solve the problem (19a) - (19g) simultaneously, reducing thus the overall computation time.

**Definition 2.** A control input $u_1 : [t_j, t_j + T_p] \to \mathbb{R}^m$ for a state $e_1(x_1(t_j))$ is called *admissible* for the FHOCP (17a)-(17e) if the following hold: 1) $u_1(\cdot)$ is piecewise continuous; 2) $u_1(s) \in U_{1,u}, \forall s \in [t_j, t_j + T_p]$; 3) $e_1\big(x_1(s);\ u_1(\cdot);\ x_1(t_j), e_1(x_1(t_j))\big) \in \mathcal{E}_1(\cdot), \forall\ s \in [t_j, t_j + T_p]$, and 4) $e_1\big(x_1(t_j + T_p);\ u_1(\cdot);\ x_1(t_j), e_1(x_1(t_j))\big) \in \mathcal{F}_1(\cdot)$.

**Theorem 1.** *Suppose that:* 1) *Assumption 1 - 3 hold;* 2) *The FHOCP* (17a)-(17e) *is feasible for the initial time $t = 0$;* 3) *There exists an admissible control input $\kappa_1 : [t_j + T_p, t_{j+1} + T_p] \to U_1$ such that for all $e_1 \in \mathcal{F}_1(\cdot)$ and for every $s \in [t_j + T_p, t_{j+1} + T_p]$ it holds that: $e_1(x_1(s)) \in \mathcal{F}_1(\cdot)$ and $\dfrac{\partial V_1}{\partial e_1} g_1(e_1(x_1(s)), \kappa_1(s)) + F_1(e_1(x_1(s)), h_1(s)) \leq 0$. Then, the system* (14)*, under the control input* (18)*, converges to the set $\mathcal{F}_1(\cdot)$ when $t \to \infty$.*

*Proof.* The proof of the theorem consists of two parts: firstly, recursive feasibility is established, that is, initial feasibility is shown to imply subsequent feasibility; secondly, and based on the first part, it is shown that the error state $e_1(t)$ reaches the terminal set $\mathcal{F}_1(\cdot)$. The feasibility and the convergence analysis is similar with the proof of Theorem 1 in [25, Section IV, p. 6]. □

## V. Simulation Results

To demonstrate the efficiency of the proposed control protocol, we consider a simulation example with $N = 3$ ground vehicles equipped with 2 DOF manipulators, rigidly grasping an object with $n_1 = n_2 = n_3 = 4$, $n = n_1 + n_2 + n_3 = 12$. The states of the agents are given as: $q_i = [p_{B_i}^\top, \alpha_i^\top]^\top \in \mathbb{R}^4$, $p_{B_i} = [x_{B_i}, y_{B_i}]^\top \in \mathbb{R}^2$, $\alpha_i = [\alpha_{i_1},\ \alpha_{i_2}]^\top \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$. The state of the object is $x_O = [p_O^\top, \phi_O]^\top \in \mathbb{R}^4$ and it is calculated though the states of the agents. The manipulators become singular when $\sin(\alpha_{i_1}) = 0\}, i \in \{1, 2\}$, thus the state constraints for the manipulators are set to: $\varepsilon < \alpha_{1_1} < \frac{\pi}{2} - \varepsilon$, $-\frac{\pi}{2} + \varepsilon < \alpha_{1_2} < \frac{\pi}{2} - \varepsilon$, $-\frac{\pi}{2} + \varepsilon < \alpha_{2_1} < -\varepsilon$, $-\frac{\pi}{2} + \varepsilon < \alpha_{2_2} < \frac{\pi}{2} - \varepsilon$. We also consider the input constraints: $-8.5 \leq u_{i,j}(t) \leq 8.5$, $i \in \{1, 2\}$, $j \in \{1, \dots, 4\}$. The initial conditions of agents and the object are set to: $q_1(0) = [0.5, 0, \frac{\pi}{4}, \frac{\pi}{4}]^\top$, $q_2(0) = [0, -4.4142, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $q_3(0) = [-0.50, -4.4142, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $\dot{q}_1(0) = \dot{q}_2(0) = \dot{q}_3(0) = [0, 0, 0, 0]^\top$ and $x_O(0) = [0, -2.2071, 0.9071, \frac{\pi}{2}]^\top$. The desired goal state the object is set to $x_{O,\text{des}} = [5, -2.2071, 0.9071, \frac{\pi}{2}]^\top$, which, due to the structure of the considered robots, corresponding uniquely to $q_{1,\text{des}} = [5.5, 0, \frac{\pi}{4}, \frac{\pi}{4}]^\top$, $q_{2,\text{des}} = [5, -4.4142, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $q_{3,\text{des}} = [4.5, 0, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $\dot{q}_{3,\text{des}} = [0, 0, 0, 0]^\top$ and $\dot{q}_{1,\text{des}} = \dot{q}_{2,\text{des}} = \dot{q}_{3,\text{des}} = [0, 0, 0, 0]^\top$. We set an obstacle between the initial and the desired pose of the object. The obstacle is spherical with center $[2.5, -2.2071, 1]$ and radius $\sqrt{0.2}$. The sampling time is $h = 0.1\,\text{sec}$, the horizon is $T_p = 0.5\,\text{sec}$, and the total simulation time is $60\,\text{sec}$; The matrices $P$, $Q$, $R$ are set to: $P = Q = 0.5 I_{8\times 8}$, $R = 0.5 I_{4\times 4}$ and the load sharing coefficients as $c_1 = 0.3$, $c_2 = 0.5$, and $c_3 = 0.2$. The simulation results are depicted in Fig. 2- Fig. 9; Fig. 2, Fig. 3 and Fig. 4 show the error states of agent 1, 2 and 3, respectively, which converge to 0; Fig. 5 depicts the states of the objects; Fig. 9 shows the collision-avoidance constraint with the obstacle; Fig. 6 - Fig. 8 depict the control inputs of the three agents. Note that the different load-sharing coefficients produce slightly different inputs. The simulation was carried out by using the NMPC toolbox given in [21] and it took $13450\,\text{sec}$ in MATLAB $R2015\text{a}$ Environment on a desktop with 8 cores, 3.60 GHz CPU and 16GB of RAM. In our previous work [25], the same simulation was *centralized* and it took $45547\,\text{sec}$ on the same computer.

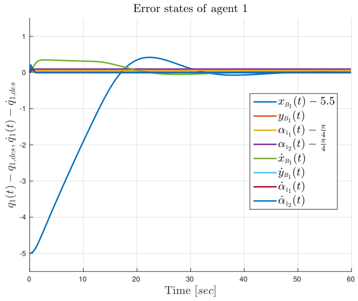

Fig. 2: The error states of agent 1.

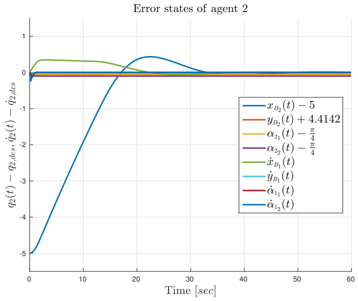

Fig. 3: The error states of agent 2.

## VI. Conclusions and Future Work

In this work we proposed a NMPC scheme for decentralized cooperative transportation of an object rigidly grasped by $N$ robotic agents. The proposed control scheme deals with singularities of the agents, inter-agent collision avoidance as well as collision avoidance between the agents and the object with the workspace obstacles. We proved the feasibility and convergence analysis of the proposed methodology and simulation results verified the efficiency of the approach. Future efforts will be devoted towards reconfiguration in case of task infeasibility for the followers, event-triggered communication between the agents so as to reduce the communication burden that is required for solving the FHOCP at every sampling time, and real-time experiments.

## References

[1] S. Schneider and R. H. R. Cannon, "Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 383–394, 1992.

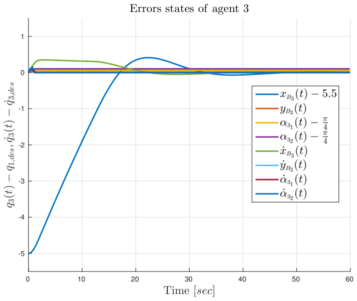

Fig. 4: The error states of agent 3.

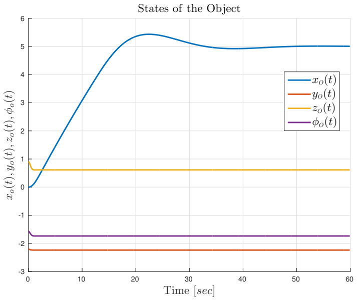

Fig. 5: The states of object converging to the desired ones.

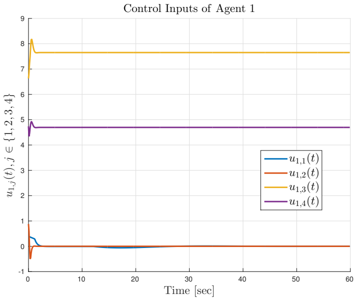

Fig. 6: The control inputs of agent 1 with $-8.5 \leq u_{1,j}(t) \leq 8.5$.

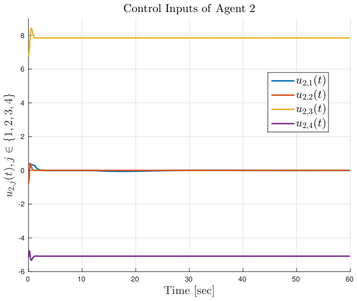

Fig. 7: The control inputs of agent 2 with $-8.5 \leq u_{2,j}(t) \leq 8.5$.

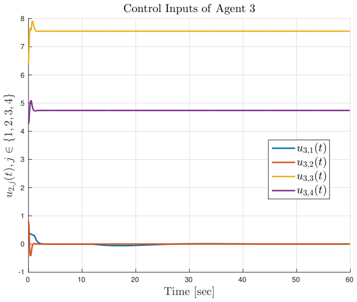

Fig. 8: The control inputs of agent 3 with $-8.5 \leq u_{3,j}(t) \leq 8.5$.

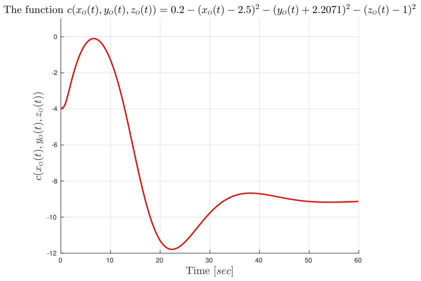

Fig. 9: The function $c(x_O(t), y_O(t), z_O(t)) = 0.2 - (x_O - 2.5)^2 - (y_O + 2.2071)^2 - (z_O - 1)^2$ is always negative, indicating collision avoidance.

[2] Y.-H. Liu and S. Arimoto, “Decentralized Adaptive and Nonadaptive Position/Force Controllers for Redundant Manipulators in Cooperations,” *The International Journal of Robotics Research*, 1998.

[3] M. Zribi and S. Ahmad, “Adaptive Control for Multiple Cooperative Robot Arms,” *IEEE Conference of Decision and Control (CDC)*, 1992.

[4] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, “Decentralized Cooperation Between Multiple Manipulators,” *IEEE International Workshop on Robot and Human Communication*, pp. 183–188, 1996.

[5] F. Caccavale, P. Chiacchio, and S. Chiaverini, “Task-Space Regulation of Cooperative Manipulators,” *Automatica*, 2000.

[6] L. Gudiño-Lau, M. Arteaga, L. Munoz, and V. Parra-Vega, “On the Control of Cooperative Robots Without Velocity Measurements,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 600–608, 2004.

[7] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, “Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators,” *IEEE/ASME Transactions On Mechatronics*, vol. 13, no. 5, pp. 576–586, 2008.

[8] D. Heck, D. Kostić, A. Denasi, and H. Nijmeijer, “Internal and External Force-Based Impedance Control for Cooperative Manipulation,” *IEEE European Control Conference (ECC)*, pp. 2299–2304, 2013.

[9] S. Erhart and S. Hirche, “Adaptive force/velocity control for multi-robot cooperative manipulation under uncertain kinematic parameters,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 307–314, 2013.

[10] J. Szewczyk, F. Plumet, and P. Bidaud, “Planning and controlling cooperating robots through distributed impedance,” *Journal of Robotic Systems*, vol. 19, no. 6, pp. 283–297, 2002.

[11] F. Ficuciello, A. Romano, L. Villani, and B. B. Siciliano, “Cartesian Impedance Control of Redundant Manipulators for Human-Robot co-manipulation,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2120–2125, 2014.

[12] A. Ponce-Hinestroza, J. Castro-Castro, H. Guerrero-Reyes, V. Parra-Vega, and E. Olguŷn-Dỳaz, “Cooperative Redundant Omnidirectional Mobile Manipulators: Model-Fee Decentralized Integral Sliding Modes and Passive Velocity Fields,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2375–2380, 2016.

[13] W. Gueaieb, F. Karray, and S. Al-Sharhan, “A Robust Hybrid Intelligent Position/Force Control Scheme for Cooperative Manipulators,” *IEEE/ASME Transactions on Mechatronics*, pp. 109–125, 2007.

[14] C. K. Verginis and D. V. Dimarogonas, “Distributed Cooperative Manipulation under Timed Temporal Specifications,” *American Control Conference (ACC)*, 2017.

[15] C. K. Verginis, M. Mastellaro, and D. V. Dimarogonas, “Robust Quaternion-based Cooperative Manipulation without Force/Torque Information,” *IFAC World Congress, Toulouse, France*, 2017.

[16] B.Siciliano, L. Sciavicco, and L. Villani, *“Robotics: Modelling, Planning and Control”*. Advanced Textbooks in Control and Signal Processing, Springer, 2009.

[17] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained Model Predictive Control: Stability and Optimality,” *Automatica*, 2000.

[18] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, “State and Output Feedback Nonlinear Model Predictive Control: An Overview,” *European Journal of Control*, vol. 9, no. 2-3, pp. 190–206, 2003.

[19] H. Chen and F. Allgöwer, “A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability,” *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[20] R. Findeisen, L. Imsland, F. Allgöwer, and B. Foss, “Towards a Sampled-Data Theory for Nonlinear Model Predictive Control,” *New Trends in Nonlinear Dynamics and Control and their Applications*, pp. 295–311, 2003.

[21] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer London, 2011.

[22] E. Camacho and C. Bordons, “Nonlinear Model Predictive Control: An Introductory Review,” pp. 1–16, 2007.

[23] F. Fontes, “A General Framework to Design Stabilizing Nonlinear Model Predictive Controllers,” *Systems and Control Letters*, vol. 42, no. 2, pp. 127–143, 2001.

[24] A. Filotheou, A. Nikou, and D. V. Dimarogonas, “Decentralized Control of Uncertain Multi-Agent Systems with Connectivity Maintenance and Collision Avoidance,” *IEEE European Control Conference (ECC), Under Review, Arxiv Link: https://arxiv.org/abs/1710.09204,*, 2017.

[25] A. Nikou, C. K. Verginis, S. Heshmati-alamdari, and D. V. Dimarogonas, “A Nonlinear Model Predictive Control Scheme for Cooperative Manipulation with Singularity and Collision Avoidance,” *25th IEEE*

*Mediterranean Conference on Control and Automation (MED), Valletta, Malta, Arxiv Link: https://arxiv.org/abs/1705.01426*, 2017.

[26] H. Khalil, *Noninear Systems*. Prentice-Hall, New Jersey, 1996.