

Control of Cooperative Manipulator-Endowed Systems under High-Level Tasks and Uncertain Dynamics*

Christos K. Verginis^{a,*}, Dimos V. Dimarogonas^b

^a*Division of Signals and Systems, Department of Electrical Engineering, Uppsala University, Uppsala, Sweden*

^b*Division of Decision and Control, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden*

Abstract

This paper considers the problem of distributed motion- and task-planning of multi-agent and multi-agent-object systems under temporal-logic-based tasks and uncertain dynamics. We focus on manipulator-endowed robotic agents that can interact with their surroundings. We present first continuous control algorithms for multi-agent navigation and cooperative object manipulation that exhibit the following properties. First, they are distributed in the sense that each agent calculates its own control signal from local interaction with the other agents and the environment. Second, they guarantee safety properties in terms of inter-agent collision avoidance and obstacle avoidance. Third, they adapt on-the-fly to dynamic uncertainties and are robust to exogenous disturbances. The aforementioned algorithms allow the abstraction of the underlying system to a finite-state representation. Inspired by formal-verification techniques, we use such a representation to derive plans for the agents that satisfy the given temporal-logic tasks. Various simulation results and hardware experiments verify the efficiency of the proposed algorithms.

*This document is the results of the research project supported by the H2020 ERC Starting Grant BUCOPHSYS, the European Union's Horizon 2020 Research and Innovation Programmes under the GA No. 731869 (Co4Robots) and 644128 (AEROWORKS), the Swedish Research Council (VR), the Knut och Alice Wallenberg Foundation (KAW) and the Swedish Foundation for Strategic Research (SSF).

*Corresponding author

Email addresses: `christos.verginis@angstrom.uu.se` (Christos K. Verginis), `dimos@kth.se` (Dimos V. Dimarogonas)

Keywords: Multi-agent systems, multi-agent navigation, cooperative manipulation, uncertain dynamics, temporal-logics

1. Introduction

The technological developments have been increasing exponentially during the last century, with an evident peak in the last few decades. The recent need for development of smart cities (including autonomy in industrial buildings, houses, highways, as well as automated rescue missions) calls for wider deployment of autonomous agents that must coordinate with each other to achieve a specific task. Additionally, noteworthy is the increasing evolution of wireless communication technology that results in the low-cost massive development of (internal and external) sensor devices. Along with the incapability of the corresponding computing units to process very large amounts of data in small amounts of time, this has given rise to a special case of systems that consist of multiple agents, namely multi-agent systems. Multi-agent systems consist of autonomous agents that rely solely on local sensor information with respect to the rest of the agents to determine their actions, which is often called *distributed* or *decentralized control*.

The main focus of multi-agent systems is the design of distributed control protocols in order to achieve global tasks, such as *consensus* [1, 2, 3, 4, 5], in which all the agents are required to converge to a specific point, and *formation* [6, 7], in which all the agents aim to form a predefined geometric shape. At the same time, the agents might need to fulfill certain transient properties, such as *network connectivity* [8, 9, 10] and/or *collision avoidance* [11]. Additionally, multi-agent systems entail several application in the fields of robotics, which is the main focus of this paper. More specifically, we consider robotic agents that consist of mobile platforms and robotic manipulators, having hence access to the entire workspace and being able to interact with the environment.

Another topic that has troubled researchers the last decades is the control of multi-agent systems such that each agent fulfills desired tasks given by high-level

specifications expressed as temporal logic formulas. Temporal-logic-based motion planning has gained a significant amount of attention over the last decade, since it provides a fully automated correct-by-design controller synthesis approach for autonomous agents. Temporal logics, such as linear temporal logic (LTL), provide formal high-level languages that can describe planning objectives more complex than the well-studied navigation algorithms, and have been used extensively both in single- as well as in multi-agent setups. The objectives are given as a temporal logic formula with respect to a discretized abstraction of the system (usually a finite transition system), and then, a high-level discrete path is found by off-the-shelf model-checking algorithms, given the abstracted system and the task specification. Consider, for instance, a robot operating in a workspace which is partitioned into 6 rooms and a corridor consisting of three regions. A high-level task for the robot might have the following form: “Periodically visit rooms R_1, R_4, R_6 , in this order, while avoiding rooms R_2, R_3 and R_5 ”, or “Grab the ball that lies in room R_6 and deliver it in room R_3 between 10 and 20 seconds”. The aforementioned specifications include complex tasks where *time* might play an important role.

One of the main problems that arise when dealing with high-level tasks based on temporal-logic formulas is the construction of a discrete abstracted representation of the continuous system. More specifically, given a temporal-logic formula over a continuous workspace/state space, how does one partition this space into discrete states? Moreover, given a predefined partition, what are the control inputs of the agents that guarantee well-defined transitions among the discrete states? When multiple robotic agents are concerned, the aforementioned specifications must also incorporate collision-avoidance as well as connectivity-maintenance properties among the agents, which brings the problem of abstraction to a new level of complexity. Furthermore, consider a case where some unactuated objects must undergo a series of processes in a workspace with autonomous agents (e.g., car factories), expressed as temporal-logic high-level specifications. In such cases, the agents, except for satisfying their own motion specifications, are also responsible for coordinating with each other in

order to transport the objects around the workspace.

The aforementioned problems become even more challenging when one takes into account system uncertainty. The dynamic model of real robotic systems cannot be accurately known by the user/designer, since it includes terms that might not be easy to identify, e.g., dynamic parameters (mass, inertia), friction, and other external disturbances. This becomes more apparent as the complexity of the considered systems increases (consider, e.g., a mobile robot vs a 6-DoF robotic manipulator). Such uncertainties are expected to affect the performance of the system, and since they cannot be accurately canceled by the control design, the latter must render the closed-loop system adaptable and robust to them [12].

This paper summarizes the main results of the PhD Thesis [13]¹. We consider the problem of distributed motion- and task-planning of uncertain multi-agent and multi-agent-object systems under complex task specifications expressed via temporal logic constraints. We present robust abstractions of the continuous system dynamics into a discrete representation (e.g., transition systems) and the application of formal verification methodologies towards the satisfaction of temporal-logic-based tasks. More specifically, we break down the problem into three main subproblems.

First, motivated by the need of transition design for multi-agent systems, we develop distributed control protocols for the safe multi-agent navigation subject to collision and connectivity constraints as well as model uncertainties. Second, in view of the incorporation of unactuated objects in the task specifications, we consider the problem of cooperative object manipulation. We design control protocols for distributed cooperative manipulation of an object grasped by multiple robotic agents subject to dynamic uncertainties. The third part draws from the previous ones to design well-defined discrete abstractions for multi-agent and multi-agent-objects systems. In that way, we allow the expression

¹The first author was the awardee of the 2020 EECI European PhD Award on Control for Complex and Heterogeneous Systems.

of complex desired tasks as temporal logic specifications, for which we provide suitable controller-synthesis algorithms.

2. Related Work

2.1. Multi-Agent Navigation

Multi-agent navigation with collision avoidance, possibly also with workspace obstacles, is a special instance of the motion planning problem [14, 15]. Several techniques have been developed in the related literature for robot motion planning with obstacle avoidance, such as discretization of the continuous space and employment of discrete algorithms (e.g., Dijkstra, A^*), probabilistic roadmaps, sampling-based motion planning, and feedback-based motion planning [16].

Feedback-based motion planning has been receiving attention for more than two decades. Early works [17, 18, 19], relied on the Koditschek-Rimon navigation function (KRNF) [20, 21], where the robots successfully converge to their goals while avoiding collisions in obstacle-free workspaces from almost all initial conditions (in the sense of a measure-zero set), if a control gain is chosen large enough. The idea of gain tuning has been also employed to an alternative KRNF in [22]. Tuning-free controllers are presented in [23, 24]. Other works employ control barrier functions for multi-agent collision avoidance [25] and optimization-based techniques via model predictive control (MPC) [26]; [27] proposes reciprocal collision obstacle by local decision making for the desired velocity of the agents and sensing uncertainties are taken into account in [28]. The work [29] develops a workspace decomposition methodologies with a hybrid controller.

A common assumption that most of the aforementioned works consider is the simplified robot dynamics, i.e., single integrators/unicycle kinematics, without taking into account any robot dynamic parameters and where the control input is the robot velocity. Hence, indirectly, the schemes depend on an embedded internal system that converts the desired signal to the actual robot actuation

command. The above imply that the actual agent trajectories might deviate from the desired ones, jeopardizing safety and possibly resulting in collisions.

Multi-agent navigation can be also achieved via means of sequential leader-follower coordination, where an assigned leader agent has priority over the other agents for goal convergence. A large variety of works, however, do not consider collision-avoidance specifications among the agents [30, 31, 32, 33], which is crucial when considering real physical systems. Moreover, as discussed before, many of the works in the related literature consider simplified/known dynamics ([34, 6, 22, 35, 31, 36, 33, 37, 38]), which can have crucial effects on the actual behaviour of real robotic systems.

2.2. Cooperative Manipulation

Cooperative manipulation is a well-studied topic, with numerous works in the last three decades [39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55]). Impedance and hybrid force/position control is the most common methodology used in the related literature, where a desired impedance behavior is imposed potentially with force regulation. Most of the aforementioned works employ force/torque sensors to acquire feedback of the object-robots contact forces/torques, which however may result in a performance decline due to sensor noise or mounting difficulties. Recent technological advances allow manipulator grippers to grasp rigidly certain objects (see e.g., [56]), which can render the use of force/torque sensors unnecessary. Force/Torque sensor-free methodologies can be found in [42, 57, 49]. Moreover, [51] uses an external force estimator, without employing force sensors, [40] presents a force sensor-free control protocol with gain tuning, and [44] considers the object regulation problem without force/torque feedback. Finally, force/torque sensor-free methodologies are developed in [58], where the robot dynamics are not taken into account.

Most works in the related literature consider known dynamic parameters regarding the object and the robotic agents. However, the accurate knowledge of such parameters, such as masses or moments of inertia, can be a challenging issue, especially for complex robotic manipulators; adaptive control protocols are

proposed in [41] with a gain tuning scheme, in [44], where the object regulation problem is considered, and in [42], [52]. An estimation of parameters is included in [58, 59], whereas [53] and [54] employ fuzzy mechanisms to compensate for model uncertainties. In [55, 60], the authors develop a task-oriented adaptive control protocol using observers. Kinematic uncertainties and joint limits are handled in [61], [47], and [62], respectively.

Other works consider internal force and load distribution analysis [63], leader-follower schemes [50], or intermittent contact [64]; [65] proposes a kinematic-based multi-robot manipulation scheme, and [66, 67] address the problem from a formation-control point of view. Finally, an important property not addressed in the cooperative-manipulation related works is the establishment of transient constraints for the object and the robotic agents. Such constraints might entail obstacle avoidance, input saturation constraints, or singularities of the agents' Jacobian matrix, which maps the joint velocities of each agent to a 6D vector of generalized velocities.

2.3. Abstractions of Multi-Agent Systems

Temporal-logic-based planning has gained significant attention in recent years, as it provides a fully automated correct-by-design controller synthesis approach for autonomous agents. Temporal logics such as linear temporal logic (LTL) and metric interval temporal logic (MITL) provide formal high-level languages that can describe complex planning objectives. There exists a wide variety of works that employ temporal logic languages for multi-agent systems, e.g., [68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78]. The discretization of a multi-agent system to an abstracted finite transition system necessitates the design of appropriate continuous-time controllers for the transition of the agents among the states of the transition system [79]. Most works in the related literature, however, adopt point-mass agents with perfectly known dynamics, without taking into account inter-agent collision specifications and facilitating, thus, the control design. Furthermore, the vast majority of temporal-logic-based works consider LTL-based objectives; LTL languages, in contrast, e.g., to MITL, lack the abil-

ity to encode *time constraints*, which provide a larger and more expressive variety of tasks. Time constraints are efficiently encoded in temporal-logic-based tasks through signal temporal logic (STL) languages [76, 77, 78]. However, STL tasks are tackled purely with feedback-control algorithms, inheriting thus their drawbacks, such as local minima. Finally, most works in the related literature consider temporal logic-based motion planning for fully actuated, autonomous agents. In many applications, however, *unactuated* objects must undergo a series of processes in a workspace with autonomous agents (e.g., car factories). When the unactuated objects’ specifications are expressed using temporal logics, then the abstraction of the agents’ behavior becomes much more complex, since it has to take into account the objects’ goals.

3. Preliminaries

3.1. Notation

The sets of real and natural numbers are denoted by \mathbb{R} and \mathbb{N} , respectively; $\|x\|$ and $\|x\|_1$ denote the 2-norm and 1-norm, respectively, of a vector $x \in \mathbb{R}^n$. The open and closed balls with respect to the 2-norm and with radius $\delta > 0$, centered at $x \in \mathbb{R}^n$, are denoted by $\mathcal{B}(x, \delta)$ and $\bar{\mathcal{B}}(x, \delta)$, respectively; $S_s(\cdot)$ is the skew-symmetric operator defined according to the cross product $S_s(a)b = a \times b$ for vectors $a, b \in \mathbb{R}^3$. Given the sets $S_1, S_2 \subset \mathbb{R}^n$ and the matrix $A \in \mathbb{R}^{n \times m}$, we define the Minkowski addition as $S_1 \oplus S_2 := \{s_1 + s_2 \in \mathbb{R}^n : s_1 \in S_1, s_2 \in S_2\}$, the Pontryagin difference as $S_1 \ominus S_2 := \{s_1 \in \mathbb{R}^n : s_1 + s_2 \in S_1, \forall s_2 \in S_2\}$, and the matrix-set multiplication as $A \circ S := \{a : \exists s \in S : a = As\}$. Finally, the 3D rotation group is denoted by $\text{SO}(3)$.

3.2. Task Specification in LTL

We focus on the task specification Φ given as a Linear Temporal Logic (LTL) formula. The basic ingredients of an LTL formula are a set of atomic propositions Ψ and several boolean and temporal operators. LTL formulas are formed according to the following grammar [79]: $\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \bigcirc \Phi \mid \Phi_1 \mathcal{U} \Phi_2$, where $a \in \Psi$, Φ_1 and Φ_2 are LTL formulas and \bigcirc, \mathcal{U} are the *next* and *until*

operators, respectively. Definitions of other useful operators like \square (*always*), \diamond (*eventually*) and \Rightarrow (*implication*) are omitted and can be found at [79]. The semantics of LTL are defined over infinite words over 2^Ψ . Intuitively, an atomic proposition $\psi \in \Psi$ is satisfied on a word $w = w_1w_2w_3\dots$ if it holds at its first position w_1 , i.e. $\psi \in w_1$, denoted as $w \models \Phi$. Formula $\bigcirc\Phi$ holds true if Φ is satisfied on the word suffix that begins in the next position w_2 , whereas $\Phi_1 \cup \Phi_2$ states that Φ_1 has to be true until Φ_2 becomes true. Finally, $\diamond\Phi$ and $\square\Phi$ holds on w eventually and always, respectively. For a full definition of the LTL semantics, the reader is referred to [79].

3.3. Task Specification in MITL

Similar to LTL, the basic ingredients of MITL are a set of atomic propositions Ψ , boolean and temporal operators, and intervals of time. MITL formulas are formed according to the following grammar [79]: $\Phi := \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \bigcirc_I \Phi \mid \diamond_I \Phi \mid \square_I \Phi \mid \Phi_1 \mathcal{U}_I \Phi_2$, where $a \in \Psi$, and $\bigcirc, \diamond, \square$ and \mathcal{U} are the next, future, always and until operators, respectively, as in LTL case; I is a nonempty time interval in one of the following forms: $[i_1, i_2], [i_1, i_2), (i_1, i_2], (i_1, i_2), [i_1, \infty), (i_1, \infty)$ with $i_2 > i_1 \geq 0$. MITL can be interpreted either in continuous or point-wise semantics. We utilize the latter and interpret MITL formulas over timed runs. The semantics of MITL are defined over infinite timed words over 2^Ψ . More specifically, the satisfaction of an atomic proposition $\psi \in \Psi$ by a timed word $w = (w_1, t_1)(w_2, t_2)\dots$ at index j is as follows:

$$\begin{aligned}
(w, j) \models p &\Leftrightarrow p \in \mathcal{L}(r(j)), \\
(w, j) \models \neg\Phi &\Leftrightarrow (w, j) \not\models \Phi \\
(w, j) \models \Phi_1 \wedge \Phi_2 &\Leftrightarrow (w, j) \models \Phi_1 \text{ and } (w, j) \models \Phi_2 \\
(w, j) \models \bigcirc_I \Phi &\Leftrightarrow (w, j+1) \models \Phi \text{ and } t_{j+1} - t_j \in I \\
(w, j) \models \Phi_1 \mathcal{U}_I \Phi_2 &\Leftrightarrow \exists k, j, \text{ with } j \leq k, \text{ s.t. } (w, k) \models \Phi_2, t_k - t_j \in I \\
&\text{and } (w, m) \models \Phi_1, \forall m \in \{j, \dots, k\}
\end{aligned}$$

For more details, the reader is referred to [80, 81].

4. Multi-Agent Navigation

Safe, distributed navigation of multiple agents is of paramount importance and entails numerous applications. In the special case of temporal-logic specifications, each agent’s task requires it to navigate around a workspace while avoiding collisions with other agents and potential obstacles. In this section, we address the problem of distributed multi-agent navigation subject to collision constraints and uncertain dynamics. First, we consider a setup where the agents operate in an obstacle-free environment, but have to comply with connectivity constraints among them. Second, we relax the connectivity constraints and generalize our results to obstacle-cluttered environments. Both setups consider a leader-follower setting, where, at every time instant, a leader agent is prioritized over the rest of the agents to navigate to its goal. Such a prioritization can be given a priori to the agents and is used online to guarantee their sequential navigation to the respective goals.

4.1. Leader-follower navigation with collision avoidance and connectivity maintenance

Consider $N > 1$ autonomous agents, with $\mathcal{N} := \{1, \dots, N\}$, operating in \mathbb{R}^n and characterized by the closed spheres $\bar{\mathcal{B}}_i(p_i, r_i) = \{y \in \mathbb{R}^n : \|p_i - y\| \leq r_i\}^2$, with $p_i \in \mathbb{R}^n$ being agent i ’s center, and $r_i > 0$ its bounding radius. We consider the following Lagrangian dynamics for the agents:

$$M_i(p_i)\ddot{p}_i + C_i(p_i, \dot{p}_i)\dot{p}_i + g_i(p_i) + f_i(p_i, \dot{p}_i) + d_i(t) = u_i, \quad (1a)$$

where $M_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ are positive definite (and bounded) inertia matrices, $C_i : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{n \times n}$ are the Coriolis terms, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are the gravity vectors, $f_i : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$ are unknown vector fields that represent friction-like terms, $d_i : [0, \infty) \rightarrow \mathbb{R}^n$ are unknown external disturbances and modeling uncertainties, and $u_i \in \mathbb{R}^n$ are the agents’ control inputs, $\forall i \in \mathcal{N}$. All the aforementioned dynamic terms are continuous in their arguments, and d_i are uniformly bounded.

²Ellipsoid-shaped agents are considered in [82]

Moreover, we consider that the dynamic terms M_i , C_i , and g_i include unknown constant dynamic parameters of the agents (e.g., masses, moments of inertia), denoted by the vectors $\theta_i \in \mathbb{R}^\ell$, $\ell \in \mathbb{N}$, for all $i \in \mathcal{N}$. The Lagrangian system (1) can be linearly parameterized with respect to these parameters, i.e., for any vectors $x, y, z, w \in \mathbb{R}^n$, it holds that $M_i(x)y + C_i(x, z)w + g_i(x) = Y_i(x, z, w, y)\theta_i$, $\forall x, y, z, w \in \mathbb{R}^n$, where $Y_i : \mathbb{R}^{4n} \rightarrow \mathbb{R}^{n \times \ell}$ are known regressor matrices, $i \in \mathcal{N}$. We further assume that the terms f_i satisfy the growth condition: $\|f_i(p_i, \dot{p})\|_1 \leq \alpha_i \|\dot{p}_i\|_1$, $\|d_i(t)\|_1 \leq d_{b_i}$, for all $x_i, v_i \in \mathbb{R}^{2n}$, $t \geq 0$, where α_i, d_{b_i} are *unknown* positive constants, $i \in \mathcal{N}$.

Each agent has a limited sensing radius $\varsigma_i > 0$, with $\varsigma_i > \max_{j \in \mathcal{N}} \{r_i + r_j\}$, which implies that the agents can sense each other without colliding. Based on this, we model the topology of the multi-agent network through the undirected graph $\mathcal{G}(p) := (\mathcal{N}, \mathcal{E}(p))$, with $\mathcal{E}(p) := \{(i, j) \in \mathcal{N}^2 : \|p_i - p_j\| \leq \min\{\varsigma_i, \varsigma_j\}\}$, where $p := [p_1^\top, \dots, p_N^\top]^\top \in \mathbb{R}^{nN}$. We further denote $K(p) := |\mathcal{E}(p)|$. Given the k -th edge in the edge set $\mathcal{E}(p)$, we use the notation $(k_1, k_2) \in \mathcal{N}^2$ that gives the agent indices that form edge $k \in \mathcal{K}(p)$, where k_1 is the tail and k_2 is the head of edge k , and $\mathcal{K}(p) := \{1, \dots, K(p)\}$ is an arbitrary numbering of the edges $\mathcal{E}(p)$.

We consider that the agents aim at safely navigating to their goals, denoted by x_{d_i} , $i \in \mathcal{N}$. As mentioned before, we assume that the agents have been assigned with a certain prioritization. The first agent in priority is a leader agent, while the rest are the follower agents. The follower agents give priority to the navigation of the leader agent, while avoiding collisions with each other and maintaining connectivity, as defined by the initial graph $\mathcal{G}(p(0))$. Once the leader navigates to its goal, the next agent in priority assumes the role of the leader and so forth. Without loss of generality, we assume that agent $i = 1$ corresponds to the leader agent, whereas $i > 1$ are the followers, which belong to the set $\mathcal{N}_{\mathcal{F}} := \{2, \dots, N\}$. Hence, the problem at hand is the design of a distributed control protocol such that leader navigates to a desired pose $p_d := p_{d_1}$, i.e., $\lim_{t \rightarrow \infty} (p_1(t) - p_d) = 0$, while guaranteeing collision avoidance and connectivity maintenance, i.e., $\bar{\mathcal{B}}_i(p_i(t), r_i) \cap \bar{\mathcal{B}}_j(p_j(t), r_j) = \emptyset$, for all $t \geq 0$, $i, j \in \mathcal{N}$, $i \neq j$, and $\|p_{k_1}(t) - p_{k_2}(t)\| \leq \min\{\varsigma_{m_1}, \varsigma_{m_2}\}$, for all $t \geq 0$, $k \in \mathcal{K}_0 \subset \mathcal{K}(p(0))$,

where $\mathcal{K}_0 := \{1, \dots, K_0\}$ is an edge numbering for the edge set $\mathcal{E}_0 := \mathcal{E}(p(0))$, with $K_0 := |\mathcal{E}_0|$.

Besides the edge set \mathcal{E}_0 , with edge numbering \mathcal{K}_0 and K_0 edges, which needs to remain connected, we consider also the complete graph $\bar{\mathcal{G}} := (\mathcal{N}, \bar{\mathcal{E}})$, with $\bar{\mathcal{E}} := \{(i, j), \forall i, j \in \mathcal{N}, i < j\}$, $\bar{K} := |\bar{\mathcal{E}}| = \frac{N(N-1)}{2}$, and the edge numbering $\bar{\mathcal{K}} := \{1, \dots, K_0, K_0 + 1, \dots, \bar{K}\}$, where $\{K_0 + 1, \dots, \bar{K}\}$ corresponds to the edges in $\bar{\mathcal{E}} \setminus \mathcal{E}_0$. We construct now the local collision and connectivity functions for all edges $\bar{\mathcal{K}}$ and \mathcal{K}_0 , respectively. Given positive constants $\bar{\beta}_c$ and $\bar{\beta}_n$, define $\beta_{c,k} : \mathbb{R}_{\geq 0} \rightarrow [0, \bar{\beta}_c]$ and $\beta_{n,l} : \mathbb{R}_{\geq 0} \rightarrow [0, \bar{\beta}_n]$, with

$$\beta_{c,k}(x) := \begin{cases} \vartheta_{c,k}(x) & 0 \leq x < \bar{d}_{c,k}, \\ \bar{\beta}_c & \bar{d}_{c,k} \leq x \end{cases}, \quad \beta_{n,l}(x) := \begin{cases} \vartheta_{n,l}(x) & 0 \leq x < \underline{d}_{n,l}^2, \\ \bar{\beta}_n & \underline{d}_{n,l}^2 \leq x \end{cases},$$

for all $k \in \bar{\mathcal{K}}$, $l \in \mathcal{K}_0$, where $\vartheta_{c,k} : \mathbb{R}_{\geq 0} \rightarrow [0, \bar{\beta}_c]$, $\vartheta_{n,l} : \mathbb{R}_{\geq 0} \rightarrow [0, \bar{\beta}_n]$ are polynomials that guarantee that $\beta_{c,k}$ and $\beta_{n,l}$, respectively, are twice continuously differentiable, for all $k \in \bar{\mathcal{K}}$, $l \in \mathcal{K}_0$. The aforementioned functions are smooth switches. Then, we choose

$$\begin{aligned} \beta_{c,k} &:= \beta_{c,k}(\iota_k), & \iota_k &:= \iota_k(p_{k_1}, p_{k_2}) := \|p_{k_1} - p_{k_2}\|^2 - (r_{k_1} + r_{k_2})^2 \\ \beta_{n,l} &:= \beta_{n,l}(\nu_l), & \nu_l &:= \nu_l(p_{l_1}, p_{l_2}) := \underline{d}_{n,l}^2 - \|p_{l_1} - p_{l_2}\|^2 \end{aligned}$$

with $\underline{d}_{n,k} := \min\{\varsigma_{k_1}, \varsigma_{k_2}\}$ and we also set $\bar{d}_{c,k} := \underline{d}_{n,k}^2 - (r_{k_1} + r_{k_2})^2$, $k \in \bar{\mathcal{K}}$, $l \in \mathcal{K}_0$. The terms $\bar{\beta}_c$, $\bar{\beta}_n$ can be any positive constants. Note that $\beta_{c,k}$ and $\beta_{n,l}$ take into account the limited sensing capabilities of the agents and their derivatives vanish at collisions and connectivity breaks, respectively, of the respective edges. All the parameters for the construction of $\beta_{c,k}$, $\beta_{n,l}$ can be transmitted off-line to the agents.

Regarding the uncertain terms of (1), we define estimates $\hat{\theta}_i \in \mathbb{R}^\ell$, $\hat{\alpha}_i \in \mathbb{R}$, $\hat{d}_{b_i} \in \mathbb{R}$, $\forall i \in \mathcal{N}$, with the respective errors $\tilde{\theta}_i := \hat{\theta}_i - \theta_i$, $\tilde{\alpha}_i := \hat{\alpha}_i - \alpha_i$, $\tilde{d}_{b_i} := \hat{d}_{b_i} - d_{b_i}$, $\forall i \in \mathcal{N}$. By using adaptive control techniques, these estimations compensate appropriately for the unknown terms, without necessarily converging to them. In addition, we define the leader error signal $s_e := x_1 - x_d$ and the constants

$\alpha_{i,k}^c$ and $\alpha_{i,l}^n$ as:

$$\alpha_{i,k}^c := \begin{cases} -1, & i = k_1 \\ 1, & i = k_2 \\ 0, & \text{otherw.} \end{cases} \quad \alpha_{i,l}^n := \begin{cases} -1, & i = l_1 \\ 1, & i = l_2 \\ 0, & \text{otherw.} \end{cases}$$

for all $k \in \bar{\mathcal{K}}, l \in \mathcal{K}_0, i \in \mathcal{N}$, which provide boolean values depending on whether agent i is part (head or tail) of edge m and l . Finally, we define, for all $k \in \bar{\mathcal{K}}, l \in \mathcal{K}_0$, the terms

$$\beta'_{c,k} := \frac{\partial}{\partial \nu_k} \left(\frac{1}{\beta_{c,k}(\nu_k)} \right), \quad \beta'_{n,l} := \frac{\partial}{\partial \nu_l} \left(\frac{1}{\beta_{n,l}(\nu_l)} \right), \quad (2)$$

which diverge to infinity in a collision and a connectivity break of the agents k_1, k_2 and l_1, l_2 , respectively. We propose now the following distributed adaptive control protocol. Choose the agents' desired velocity as

$$v_{d_1} = -\gamma_e s_e + \sum_{k \in \bar{\mathcal{K}}} \alpha_{1,k}^c \beta'_{c,k} \frac{\partial \nu_k}{\partial x_{k_1}} + \sum_{l \in \mathcal{K}_0} \alpha_{1,l}^n \beta'_{n,l} \frac{\partial \nu_l}{\partial x_{l_1}} \quad (3a)$$

$$v_{d_i} = k_i \left(\sum_{k \in \bar{\mathcal{K}}} \alpha_{i,k}^c \beta'_{c,k} \frac{\partial \nu_k}{\partial x_{k_1}} + \sum_{l \in \mathcal{K}_0} \alpha_{i,l}^n \beta'_{n,l} \frac{\partial \nu_l}{\partial x_{l_1}} \right), \forall i \in \mathcal{N}_{\mathcal{F}} \quad (3b)$$

that concerns the collision avoidance and connectivity maintenance properties, with the extra term $\gamma_e s_e$ for the leader to guarantee the navigation to x_d . The terms γ_e, k_i are positive constants, for $i \in \mathcal{N}_{\mathcal{F}}$. Since v_{d_i} is not the actual velocity of the agents, we define the errors $e_{v_i} := \dot{p}_i - v_{d_i}$, for all $i \in \mathcal{N}$, and design the distributed control laws

$$\begin{aligned} u_i := & \sum_{k \in \bar{\mathcal{M}}} \alpha_{i,k}^c \beta'_{c,k} \frac{\partial \nu_k}{\partial p_{k_1}} + \sum_{l \in \mathcal{K}_0} \alpha_{i,l}^n \beta'_{n,l} \frac{\partial \nu_l}{\partial p_{l_1}} - k_{v_i} e_{v_i} - \tilde{s}_{e_i} + Y_{r_i} \hat{\theta}_i \\ & - \text{sgn}(e_{v_i}) \|\dot{p}_i\|_1 \hat{\alpha}_i - \text{sgn}(e_{v_i}) \hat{d}_{b_i}, \end{aligned} \quad (4)$$

for all $i \in \mathcal{N}$, where $\tilde{s}_{e_1} = \gamma_e s_e, \tilde{s}_{e_i} = 0, i \in \mathcal{N}_{\mathcal{F}}, Y_{r_i} := Y_i(p_i, \dot{p}_i, v_{d_i}, \dot{v}_{d_i})$, and k_{v_i} are positive gains. Moreover, we design the adaptation signals

$$\dot{\hat{d}}_{b_i} = \gamma_{i,d} \|e_{v_i}\|_1, \quad \dot{\hat{\alpha}}_i = \gamma_{i,f} \|e_{v_i}\|_1 \|\dot{p}_i\|_1, \quad \dot{\hat{\theta}}_i = -\gamma_{i,\theta} Y_{r_i}^\top e_{v_i}, \quad (5)$$

for all $i \in \mathcal{N}$, with arbitrary bounded initial conditions, and positive constants $\gamma_{i,d}, \gamma_{i,f}, \gamma_{i,\theta}, i \in \mathcal{N}$. The following theorem establishes the leader navigation while ensuing collision avoidance and connectivity maintenance.

Theorem 1 ([83]). *Consider a multi-agent system described by the dynamics (1). Then, application of the control and adaptation laws (4), (5) guarantees: 1) navigation of the leader agent to x_d , 2) connectivity maintenance of the subset \mathcal{E}_0 of the initial edges, 3) inter-agent collision avoidance, and 4) boundedness of all closed loop signals, from all collision-free initial configurations. Moreover, it holds that $\lim_{t \rightarrow \infty} v_i(t) = 0, i \in \mathcal{N}$.*

Once the leader agent converges to its goal, e.g., when $\|p_1 - p_d\|$ becomes smaller than a small positive constant, it broadcasts it and the next agent in priority becomes the leader. Therefore, repetitive application of the aforementioned theorem for all agents guarantees their sequential navigation to their respective goals.

4.2. Leader-follower navigation in obstacle-cluttered environments

In this section, we consider the more challenging problem of distributed and safe multi-agent navigation in a bounded workspace with obstacles. More specifically, the workspace is now assumed to be the open ball $\mathcal{W} := \mathcal{B}(0, r_{\mathcal{W}}) = \{z \in \mathbb{R}^n : \|z\| < r_{\mathcal{W}}\}$, with radius $r_{\mathcal{W}} > 0$. The workspace contains $M > 0$ closed sets $\mathcal{O}_j, j \in \mathcal{J} := \{1, \dots, J\}$, corresponding to obstacles. Each obstacle is a closed ball centered at $c_j \in \mathbb{R}^n$, with radius $r_{c_j} > 0$, i.e., $\mathcal{O}_j := \bar{\mathcal{B}}(c_j, r_{c_j}) = \{z \in \mathcal{W} : \|z - c_j\| \leq r_{c_j}\}$, for $j \in \mathcal{J}$. The agents are now assumed to obey the simplified - with respect to (1) - dynamics:

$$m_i \ddot{p}_i + f_i(p_i, \dot{p}_i) + m_i g = u_i, \quad (6)$$

for all $i \in \mathcal{N}$, where m_i is the i th agent's mass, which is considered unknown, g is a gravity constant, and $f_i(\cdot)$ are unknown functions satisfying $\|f_i(p_i, \dot{p}_i)\| \leq \alpha_i \|\dot{p}_i\|$, for unknown constants $\alpha_i, i \in \mathcal{N}$, similarly to the previous section. Without loss of generality, we assume that $i = 1$ is the leader agent, aiming to safely converge to its goal $p_d = p_{d_1}$. The large complexity of the considered problem, imposed by the addition of static obstacles, prevents us from considering connectivity properties while keeping the convergence guarantees.

Moreover, in contrast to the previous section, we consider that when the leader agent arrives at its goal, it stays there idle, representing an additional obstacle to the rest of the agents. We further make the following assumption regarding the workspace, the initial states, and goals:

Assumption 1. The workspace \mathcal{W} , obstacles $\mathcal{O}_j, j \in \mathcal{J}$, and destinations $p_{d_i}, i \in \mathcal{N}$, satisfy:

$$\begin{aligned} \|c_j - c_k\| &> r_{c_j} + r_{c_k} + 2r_M, & r_{\mathcal{W}} - \|c_j\| &> r_{c_j} + 2r_M \\ \|c_j - p_{d_i}\| &> r_{c_j} + r_i + 2r_M + \varepsilon_d, & \|p_{d_i} - p_{d_\ell}\| &> r_i + r_\ell + 2r_M + 2\varepsilon_d, \\ r_{\mathcal{W}} - \|p_{d_i}\| &> r_i + 2r_M + \varepsilon_d, \end{aligned}$$

for all $i, \ell \in \mathcal{N}, i \neq \ell, j, k \in \mathcal{J}, j \neq k$, for an arbitrarily small positive constant ε_d , where $r_M := \max_{i \in \mathcal{N}}\{r_i\}$.

Loosely speaking, the aforementioned assumption states that the pairwise distances among obstacles and workspace boundary are large enough so that one agent can always navigate between them. Since the convergence of the agents to their destinations is asymptotic, we incorporate the threshold ε_d , which is the desired proximity we want the agents to achieve to the destination.

In view of Assumption 1, we can find a positive constant \bar{r} such that

$$\|c_j - c_k\| > r_{c_j} + r_{c_k} + 2r_M + 2\bar{r}, \quad (7a)$$

$$r_{\mathcal{W}} - \|c_j\| > r_{c_j} + 2r_M + 2\bar{r}, \quad (7b)$$

$$\|c_j - p_{d_i}\| > r_{c_j} + r_i + 2r_M + \varepsilon_d + 2\bar{r}, \quad (7c)$$

$$\|p_{d_i} - p_{d_\ell}\| > r_i + r_\ell + 2r_M + 2\varepsilon_d + 2\bar{r}, \quad (7d)$$

$$r_{\mathcal{W}} - \|p_{d_i}\| > r_i + 2r_M + \varepsilon_d + 2\bar{r}, \quad (7e)$$

for all $i, \ell \in \mathcal{N}, i \neq \ell, j, k \in \mathcal{J}, j \neq k$.

The proposed control algorithm relies on the fact that the agents actively avoid collisions with each other and the obstacles only when they are sufficiently close to them. Such a property is encoded via a designed switching function. Before providing it, we define the associated distance metrics for the leader

agent

$$d_{1,c_j} := d_{1,c_j}(p_1) := \|p_1 - c_j\|^2 - (r_1 + r_{c_j})^2, \quad (8a)$$

$$d_{1,i} := d_{1,i}(p_1, p_i) := \|p_1 - p_i\|^2 - (r_1 + r_i)^2, \quad (8b)$$

$$d_{1,c_0} := d_{1,c_0}(p_1) := (r_{\mathcal{W}} + r_1)^2 - \|p_1\|^2 \quad (8c)$$

for all $j \in \mathcal{J}$, $i \in \mathcal{N}_{\mathcal{F}}$, and the follower distance metrics

$$d_{i,c_j} := d_{i,c_j}(p_i) := \|p_i - c_j\|^2 - (r_i + r_{c_j} + 2r_M + 2\bar{r})^2, \quad (8d)$$

$$d_{i,1} := d_{i,1}(p_i, p_1) := \|p_i - p_1\|^2 - (r_i + r_1)^2 = d_{1,i}(p_1, p_i) \quad (8e)$$

$$d_{i,\ell} := d_{i,\ell}(p_i, p_\ell) := \|p_i - p_\ell\|^2 - (r_i + r_\ell + 2r_M + 2\bar{r})^2, \quad (8f)$$

$$d_{i,d_\ell} := d_{i,d_\ell}(p_i) := \|p_i - p_{d_\ell}\|^2 - (r_i + r_\ell + 2r_M + 2\bar{r} + \varepsilon_d)^2, \quad (8g)$$

$$d_{i,c_0} := d_{i,c_0}(p_i) := (r_{\mathcal{W}} - r_i - 2r_M - 2\bar{r})^2 - \|p_i\|^2, \quad (8h)$$

for all $i, \ell \in \mathcal{N}_{\mathcal{F}}$, $i \neq \ell$, $j, k \in \mathcal{J}$, $j \neq k$. The aforementioned distances encode the collisions among the agents, with the obstacles and the workspace boundary. Note that the follower agents need to keep a larger distance among each other, with the obstacles, and the workspace boundary. This is needed so that the leader agent is able to navigate among them to reach its goal. Intuitively, the follower agents will reach a local minimum from the conflicting objectives of driving to their goal and avoiding the other agents and obstacles; the leader then will be able to navigate among them and the obstacles to its goal. Moreover, as mentioned before and unlike the previous section, where the agents were required to stay connected to one another, here we consider that the agents aim to navigate to their goals and remain there. Therefore, the control algorithm uses the distances d_{i,d_ℓ} , $i, \ell \in \mathcal{N}$, $i \neq \ell$ so that a follower agent i does not reach a local minimum too close to a goal p_{d_ℓ} for some agent $\ell \neq i$ higher in priority (which would need to navigate to its goal before agent i). Finally, we assume that all aforementioned distances in (8) are positive initially at $t = 0$ and we define the constant $\bar{r}_d := \min \left\{ \bar{r}_{\mathcal{W}}^2 - \|p_{d_1}\|^2, \min_{j \in \mathcal{J}} \{ \|p_{d_1} - c_j\|^2 - \bar{r}_{c_j}^2 \} \right\}$, where $\bar{r}_{\mathcal{W}} := r_{\mathcal{W}} - r_1$, $\bar{r}_{c_j} := r_{c_j} + r_1$, $j \in \mathcal{J}$.

As in the previous section, we consider that each agent $i \in \mathcal{N}$ has a limited

sensing radius ς_i and has access to (p_j, \dot{p}_j) , $\forall j \in \{j \in \mathcal{N} : \|p_i - p_j\| \leq \varsigma_i\}$. We now assume that ς_i satisfies $\varsigma_i > \sqrt{\min(\bar{r}^2, \bar{r}_d)} + r_i + 3r_M + 2\bar{r}$, with \bar{r}_d, \bar{r} . The aforementioned specification on ς_i will be needed later for the feasibility of the control protocol.

A critical property for the correctness of the proposed algorithm is the fact that the leader agent actively avoid collisions with the followers and the obstacles only when it is sufficiently close to them. This is encoded through a twice contin. differentiable function $\beta : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$, with the properties

1. $\beta((0, \tau])$ is strictly decreasing, $\lim_{x \rightarrow 0} \beta(x) = \infty$, and $\beta(x) = \beta(\tau)$, $\forall x \geq \tau$, $j \in \bar{\mathcal{J}}$, for some $\tau > 0$.
2. The function $\tilde{\beta} : (0, \tau) \rightarrow \mathbb{R}_{\geq 0}$, with $\tilde{\beta}(x) := \beta''(x)x\sqrt{x}$ is decreasing.

An example for a function β is $\beta(x) = \frac{1}{6x^5 - 15x^4 + 10x^3}$ for $x \leq \bar{\beta}$ and $\beta(x) = 1$, for $x \geq \bar{\beta}$, for any positive $\bar{\beta}$. Note that $\beta'(x) = \beta''(x) = 0$, for $x \geq \tau$.

Let now a function β that satisfies the aforementioned properties. We define the 2nd-order navigation functions for the agents as

$$\begin{aligned} \phi_i(p) &:= k_{1_i} \|p_i - p_{d_i}\|^2 + k_{2_i} \left(b_{1_i}(p_i) + b_{2_i}(p) + k_{f_i} b_{3_i}(p_i) \right) \\ b_{1_i} &:= b_{1_i}(p_i) := \sum_{j \in \bar{\mathcal{J}}} \beta_i(d_{i,c_j}(p_i)) \\ b_{2_i} &:= b_{2_i}(p) := \sum_{\ell \in \mathcal{N} \setminus \{i\}} \beta(d_{i,\ell}(p_i, p_\ell)) \\ b_{3_i} &:= b_{3_i}(p_i) := \sum_{\ell \in \mathcal{N} \setminus \{i\}} \beta_i(d_{i,d_\ell}(p_i)), \end{aligned}$$

for all $i \in \mathcal{N}$.

The convergence of the leader to its goal is based on the fact that it actively avoids collision with other agents and obstacles only when it is sufficiently close to them. In that way, *at most* one of the terms $\beta_1(d_{1,c_j})$, $\beta(d_{1,\ell})$, $\beta_1(d_{1,d_\ell})$, appearing in ϕ_1 , *has non-zero gradient* at each $x \in \mathbb{R}^{Nn}$. We achieve that by setting τ in the functions β as $\tau \in (0, \min\{\bar{r}^2, \bar{r}_d\})$, as proven in [84]. Regarding the ability of the agents to sense each other when $d_{i,\ell}(p_i, p_\ell) < \tau$, which is when

they avoid collisions with other, it holds that

$$\begin{aligned} d_{i,\ell}(p_i, p_\ell) < \tau &\Leftrightarrow \|p_i - p_\ell\|^2 \leq \tau + (r_i + r_\ell + 2r_M + 2\bar{r})^2 \Rightarrow \\ \|p_i - p_\ell\| &\leq \sqrt{\tau} + r_i + r_\ell + 2r_M + 2\bar{r} \Rightarrow \\ \|p_i - p_\ell\| &\leq \sqrt{\min\{\bar{r}^2, \bar{r}_d\}} + r_i + r_\ell + 2r_M + 2\bar{r} < \varsigma_i, \end{aligned}$$

for all $i, \ell \in \mathcal{N}$, $i \neq \ell$, since $\varsigma_i > \sqrt{\min\{\bar{r}^2, \bar{r}_d\}} + r_i + 3r_M + 2\bar{r}$ for all $i \in \mathcal{N}$.

We next define the distributed control law as

$$u_i := u_i(p, \dot{p}, \hat{m}_i, \hat{\alpha}_i) := -k_{\phi_i} \nabla_{p_i} \phi_i(p) + \hat{m}_i (\dot{v}_{d_i} + g) - \left(k_{v_i} + \frac{3}{2} \hat{\alpha}_i \right) e_{v_i}, \quad (9)$$

for all $i \in \mathcal{N}$; k_{ϕ_i} , k_{v_i} are positive constants, v_{d_i} is the i th agent's reference velocity, $v_{d_i} := v_{d_i}(x) := -\nabla_{p_i} \phi_i(p)$, e_{v_i} are the velocity errors $e_{v_i} := v_i - v_{d_i}$, and \hat{m}_i , $\hat{\alpha}_i$ denote the estimates of m_i and α_i , respectively, by agent i , evolving according to

$$\dot{\hat{m}}_i := -k_{m_i} e_{v_i}^\top (\dot{v}_{d_i} + g), \quad \dot{\hat{\alpha}}_i := k_{\alpha_i} \|e_{v_i}\|^2, \quad (10)$$

with k_{m_i} , k_{α_i} positive gain constants, $\hat{\alpha}_i(t_0) \geq 0$, and arbitrary initial conditions $\hat{m}_i(t_0)$, for $i \in \mathcal{N}$.

The correctness of the proposed control algorithm is given in the subsequent theorem.

Theorem 2 ([84]). *Consider N robots operating in \mathcal{W} , subject to the uncertain 2nd-order dynamics (6), and a leader agent $i = 1$. Then, the control protocol (9), (10) guarantees collision avoidance between the agents and the agents and obstacles/workspace boundary as well as convergence of p_1 to $p_d = p_{d_1}$ from almost all initial conditions that satisfy positivity of the distance metrics (8) at $t = 0$, given sufficiently small τ and that $k_{\phi_i} > \frac{\alpha_i}{2}$, $i \in \mathcal{N}$. Moreover, all closed loop signals remain bounded, $\forall t \geq 0$.*

Similarly to the previous section, once the leader agent converges close to its goal, i.e., when $\|p_1 - p_d\| \leq \varepsilon_d$, it broadcasts it and the next agent in priority becomes the leader. This occurs iteratively until all the robotic agents reach their destinations. Finally, the proposed algorithm can be extended to environments with obstacles that are more complicated (e.g., star-shaper obstacles [85]) in the single-agent case, i.e., when the follower robots are completely immobilized

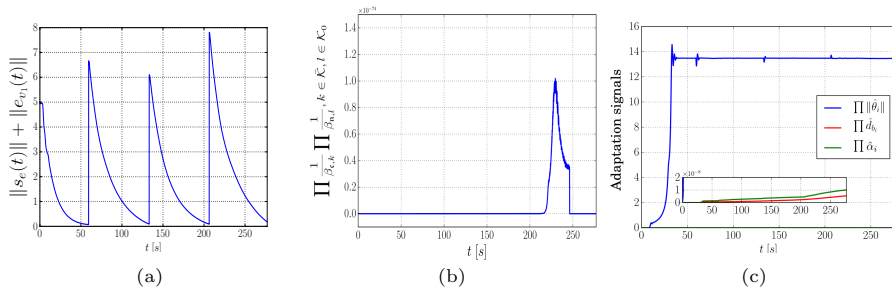


Figure 1: (a): The leader signal $\|s_e(t)\| + \|e_{v_1}(t)\|$, which converges to zero for every navigation objective; (b) the product $\prod_{k \in \mathcal{K}} \frac{1}{\beta_{c,k}(t_k(t))} \prod_{l \in \mathcal{K}_0} \frac{1}{\beta_{n,l}(v_l(t))}$, which remains bounded, proving thus the collision and connectivity properties (the zero values stem from the computer's lower numerical limits); (c) the adaptation signals $\prod_{i \in \{1, \dots, 6\}} \|\hat{\theta}_i(t)\|$, $\prod_{i \in \{1, \dots, 6\}} \hat{d}_{b_i}(t)$, $\prod_{i \in \{1, \dots, 6\}} \hat{\alpha}_i(t)$, which remain bounded, $\forall t \in [0, 277]$ s.

[84]. For more complex workspaces and simultaneous multi-agent navigation, an algorithm combining potential fields and sampling-based motion planning is developed in [86], requiring however a central computer unit to monitor and plan the motion for the agents.

4.3. Simulation Results

We first demonstrate the control algorithm of Section 4.1. We conduct simulations with $N = 6$ UAVs in \mathbb{R}^3 using the realistic robotic simulator Gazebo [87]. We consider bounding radii $r_i = 0.35$ m, sensing ranges $\varsigma_i = 3$ m, $\forall i \in \mathcal{N}$, and initial positions $x_1(0) = [0, 0, 0.1]^\top$, $x_2(0) = [2, -0.5, 0.1]^\top$, $x_3(0) = [-1.5, 1.5, 0.1]^\top$, $x_4(0) = [1, 2, 0.1]^\top$, $x_5(0) = [-1.5, -1, 0.1]^\top$, and $x_6(0) = [0.5, -1.5, 0.1]^\top$ m. We also consider that the leader has 4 navigation objectives, that is, to sequentially navigate to the points $x_{d,1} = [0, 0, 5]^\top$, $x_{d,2} = [4, 5, 3]^\top$, $x_{d,3} = [-2, 4, 2]^\top$, $x_{d,4} = [3, -2, 3]^\top$ m. Since this work provides asymptotic results with respect to the error s_e , the leader switches navigation goal each time it gets closer than 0.075m to the current goal, i.e., $\|s_e\| \leq 0.075$ m. We also consider $\mathcal{E}_0 = \{(1, 2), (1, 3), (1, 4), (3, 4), (3, 5), (5, 6), (2, 6)\}$. The unknown parameters θ_i concern the UAVs' mass and the gravity constant. The control gains and parameters are set as $\gamma_e = 0.7$, $k_i = 5$, $\forall i \in \{2, \dots, 6\}$, and $\gamma_{i,\theta} = 0.1$, $\gamma_{i,d} = 0.01$, $\gamma_{i,f} = 0.1$, $k_{v_i} = 2$, $\forall i \in \{1, \dots, 6\}$. The simulation results are shown in Fig. 1 for $t \in [0, 277]$ s. More specifically, Fig. 1 shows (a) the

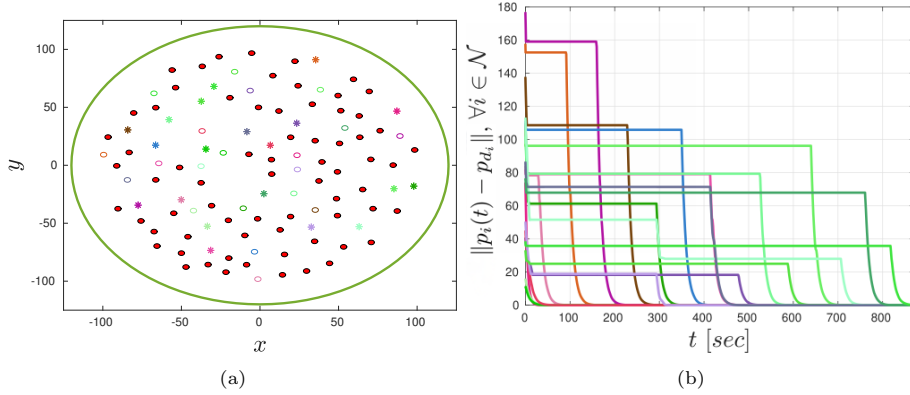


Figure 2: (a): The initial configurations of the multi-agent scenario. The obstacles are depicted as filled red disks whereas the agents as circles. The destinations are shown with asterisk; (b) The resulting signals $\|p_i(t) - p_{d_i}\|, \forall i \in \mathcal{N}$, shown to converge to zero.

evolution of the signal $\|s_e(t)\| + \|e_{v_1}(t)\|$, which converges to zero for each navigation objective, (b) the evolution of the product $\prod_{k \in \bar{\mathcal{K}}} \frac{1}{\beta_{c,k}(\nu_k(t))} \prod_{l \in \mathcal{K}_0} \frac{1}{\beta_{n,l}(\nu_l(t))}$, which remains bounded, verifying thus the collision avoidance and connectivity maintenance properties, and (c) the evolution of the products of the adaptation signals $\prod_{i \in \{1, \dots, 6\}} \|\hat{\theta}_i(t)\|, \prod_{i \in \{1, \dots, 6\}} \hat{d}_{b_i}(t), \prod_{i \in \{1, \dots, 6\}} \hat{\alpha}_i(t)$, which remain bounded, verifying thus the boundedness of the individual signals. An illustrating video can be found in <https://youtu.be/bzzXC-v2hEM>.

Next, we demonstrate the control algorithm of Section 4.2. We consider 20 agents in a 2D workspace of $r_{\mathcal{W}} = 120$, populated with 70 obstacles, as depicted in Fig. 2a. The radius of the agents and the obstacles is chosen as $r_i = r_{c_j} = 2, \forall i \in \mathcal{N}, j \in \mathcal{J}$, and the sensing radius of the agents is taken as $\varsigma_i = 20, \forall i \in \mathcal{N}$. The masses, and functions $f_i(p, \dot{p})$, both *unknown* to the agent, are taken as $m_i = 1$, and $f_i(p, \dot{p}) = \frac{\alpha}{16} \sin(0.5(p_{i_x} + p_{i_y})) F(\dot{p}_i) v$, with $F(\dot{p}_i) = \text{diag}\{\exp(-\text{sgn}(\dot{p}_{i_z}) \dot{p}_{i_z}) + 1\}_{z \in \{x, y\}}$, and $\alpha = 10$, where we denote $(p_{i_x}, p_{i_y}) = p_i, (\dot{p}_{i_x}, \dot{p}_{i_y}) = \dot{p}_i$. We choose β as $\beta(x) = \frac{1}{6x^5 - 15x^4 + 10x^3}$ for $x \leq 100$ and $\beta(x) = 1$, for $x \geq 100$, with $\tau = 0.25$. The control gains are chosen as $k_{1_i} = 0.04, k_{2_i} = 5, k_{v_i} = 20, k_{\phi_i} = 1$, and $k_{m_i} = k_{\alpha_i} = 0.01$, for all $i \in \mathcal{N}$. The results are depicted in Fig. 2b for 870 seconds, which shows the convergence of the distance errors $\|p_i(t) - p_{d_i}\|$ to zero, $\forall i \in \mathcal{N}, t \in [0, 870]$. A video illustrating

the simulation can be found on <https://vimeo.com/393443782>.

5. Cooperative Manipulation

As discussed in Section 1, we aim to incorporate high-level temporal tasks for unactuated objects. Therefore, apart from multi-agent navigation, it is equally important to establish control algorithms for cooperative object manipulation. In this section, we address the problem of cooperative manipulation of a single object by multiple robotic agents. We consider that the agents grasp the object via means of rigid contacts. Such an assumption is relaxed in our work [88, 89], where rolling contacts are employed, but is omitted from the current paper for ease of exposition.

We provide first the model of the cooperative manipulation system. We denote by $q_i, \dot{q}_i \in \mathbb{R}^{n_i}$, with $n_i \in \mathbb{N}, \forall i \in \mathcal{N} := \{1, \dots, N\}$, the generalized joint-space variables and their time derivatives of agent i , with $q_i := [q_{i_1}, \dots, q_{i_{n_i}}]$. The overall joint configuration is then $q := [q_1^\top, \dots, q_N^\top]^\top, \dot{q} := [\dot{q}_1^\top, \dots, \dot{q}_N^\top]^\top \in \mathbb{R}^n$, with $n := \sum_{i \in \mathcal{N}} n_i$. In addition, the inertial position and orientation of the i th end-effector, denoted by p_{E_i} and η_{E_i} , respectively, can be derived by the forward kinematics and are smooth functions of q_i , i.e. $p_{E_i} := p_{E_i}(q_i) : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^3, \eta_{E_i} := \eta_{E_i}(q_i) : \mathbb{R}^{n_i} \rightarrow \mathbb{T}$, where \mathbb{T} is an appropriate orientation space. The generalized velocity of each agent's end-effector $v_i := [\dot{p}_{E_i}^\top, \omega_{E_i}^\top]^\top \in \mathbb{R}^6$, where $\omega_{E_i} \in \mathbb{R}^3$ is the respective angular velocity, can be considered as a transformed state through the differential kinematics $v_i = J_i \dot{q}_i$ [90], where $J_i := J_i(q_i) : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{6 \times n_i}$ is a smooth function representing the geometric Jacobian matrix, $i \in \mathcal{N}$ [90]. We define also the sets $\mathcal{S}_i := \{q_i \in \mathbb{R}^{n_i} : \det(J_i(q_i)J_i(q_i)^\top) > 0\}$, which contains all the singularity-free configurations. The differential equation describing the task-space dynamics of each agent is [90]:

$$M_i(q_i)\dot{v}_i + C_i(q_i, \dot{q}_i)v_i + g_i(q_i) + d_i(q_i, \dot{q}_i, t) = u_i - h_i, \quad (11)$$

where $M_i : \mathcal{S}_i \rightarrow \mathbb{R}^{6 \times 6}$ is the positive definite inertia matrix, $C_i : \mathcal{S}_i \times \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{6 \times 6}$ is the Coriolis matrix, $g_i : \mathcal{S}_i \rightarrow \mathbb{R}^6$ is the gravity vector, $d_i : \mathcal{S}_i \times \mathbb{R}^{n_i} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^6$ is a disturbance term, and $u_i = [u_{i,1}, \dots, u_{i,6}]^\top \in \mathbb{R}^6$ is the task

space wrench, representing the control input. All the aforementioned dynamic terms are continuous, and $d_i(q_i, \dot{q}_i, t)$ are bounded in t , for all $\forall i \in \mathcal{N}$.

Regarding the object, we denote by $x_o := [p_o^\top, \eta_o^\top]^\top \in \mathbb{M} := \mathbb{R}^3 \times \mathbb{T}$, $v_o := [\dot{p}_o^\top, \omega_o^\top]^\top \in \mathbb{R}^6$ the pose and generalized velocity of its center of mass; η_o here denotes explicitly Euler angles $\eta_o := [\varphi_o, \theta_o, \psi_o]^\top \in \mathbb{T} = (-\pi, \pi) \times (-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\pi, \pi)$. We consider the following second-order dynamics, which can be derived based on the Newton-Euler formulation:

$$\dot{x}_o = J_o(\eta_o)v_o, \quad (12a)$$

$$M_o(\eta_o)\dot{v}_o + C_o(\eta_o, \omega_o)v_o + g_o + d_o(x_o, \dot{x}_o, t) = h_o, \quad (12b)$$

where $M_o : \mathbb{T} \rightarrow \mathbb{R}^{6 \times 6}$ is the positive definite inertia matrix, $C_o : \mathbb{T} \times \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ is the Coriolis matrix, $g_o \in \mathbb{R}^6$ is the gravity vector, $d_o : \mathbb{M} \times \mathbb{R}^6 \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^6$ a bounded vector representing modeling uncertainties and external disturbances, and $h_o \in \mathbb{R}^6$ is the vector of generalized forces acting on the object's center of mass. Moreover, $J_o := J_o(\eta_o) : \mathbb{T} \rightarrow \mathbb{R}^{6 \times 6}$ is the object representation Jacobian [90], which is not well-defined when $\theta_o = \pm \frac{\pi}{2}$, which is referred to as *representation singularity*. Similarly to the robotic agents, the aforementioned dynamic terms are continuous and $d_o(x_o, \dot{x}_o, t)$ is bounded in t . We also denote by $R_o : \mathbb{T} \rightarrow \mathbb{SO}(3)$ the object's rotation matrix.

The pose of the agents and the object's center of mass are related as

$$p_{E_i}(q_i) = p_o + R_i(q_i)p_{E_i/O}^{E_i}, \quad (13a)$$

$$\eta_{E_i}(q_i) = \eta_o + \eta_{E_i/O}, \quad (13b)$$

for all $i \in \mathcal{N}$, where $R_i : \mathbb{R}^{n_i} \rightarrow \mathbb{SO}(3)$ is the i 's end-effector rotation matrix, and $p_{E_i/O}^{E_i}, \eta_{E_i/O} \in \mathbb{R}^3$ are the *constant* distance and orientation offset vectors between $\{O\}$ and $\{E_i\}$. Following (13), along with the fact that, due to the grasping rigidity, it holds that $\omega_{E_i} = \omega_o, i \in \mathcal{N}$, one obtains

$$v_i = J_{O_i}v_o, \quad (14)$$

where $J_{O_i} : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{6 \times 6}$ is the object-to-agent Jacobian matrix, with

$$J_{O_i}(q_i) := \begin{bmatrix} I_3 & -S_s(R_i(q_i)p_{E_i/O}^{E_i/O}) \\ 0 & I_3 \end{bmatrix}, \forall q_i \in \mathbb{R}^{n_i}, \quad (15)$$

which is always full-rank.

The kineto-statics duality along with the grasp rigidity suggest that the force h_o acting on the object's center of mass and the generalized forces $h_i, i \in \mathcal{N}$, exerted by the agents at the grasping points, are related through:

$$h_o = G(q)h, \quad (16)$$

where $G : \mathbb{R}^n \rightarrow \mathbb{R}^{6 \times 6N}$, with $G(q) := [J_{O_1}^\top, \dots, J_{O_N}^\top]$, is the full row-rank grasp matrix. By combining (16), (14), (11), and (12), we obtain the coupled dynamics

$$\widetilde{M}(x)\dot{v}_o + \widetilde{C}(x)v_o + \widetilde{g}(x) + \widetilde{d}(x, t) = Gu, \quad (17)$$

where $\widetilde{M} := M_o + GMG^\top$, $\widetilde{C} := C_o + GCG^\top + GM\dot{G}^\top$, $\widetilde{g} := g_o + Gg$, $\widetilde{d} := d_o + Gd$, x is the overall state $x := [q^\top, \dot{q}^\top, \eta_o^\top, \omega_o^\top]^\top \in \mathbb{X} := \mathbf{S} \times \mathbb{R}^{n+3} \times \mathbb{T}$, $\mathbf{S} := \mathbf{S}_1 \times \dots \times \mathbf{S}_N$, and we further use $v := [v_1^\top, \dots, v_N^\top]^\top \in \mathbb{R}^{6N}$, $M := \text{diag}\{[M_i]_{i \in \mathcal{N}}\} \in \mathbb{R}^{6N \times 6N}$, $C := \text{diag}\{[C_i]_{i \in \mathcal{N}}\} \in \mathbb{R}^{6N \times 6N}$, $h := [h_1^\top, \dots, h_N^\top]^\top$, $u := [u_1^\top, \dots, u_N^\top]^\top$, $g := [g_1^\top, \dots, g_N^\top]^\top$, $d := [d_1^\top, \dots, d_N^\top]^\top \in \mathbb{R}^{6N}$.

The problem we consider in this section is the tracking of a trajectory by the object. That is, we aim to design distributed, feedback control laws u such that $\lim_{t \rightarrow \infty} (p_o(t) - p_d(t)) = \lim_{t \rightarrow \infty} (\eta_o(t) - \eta_d(t)) = 0$, for reference trajectories $p_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^3$, $\eta_d := [\varphi_d, \theta_d, \psi_d] := [\varphi_d(t), \theta_d(t), \psi_d(t)] : \mathbb{R}_{\geq 0} \rightarrow \mathbb{T}$, with bounded first and second derivatives, satisfying $\theta_d(t) \in [-\bar{\theta}, \bar{\theta}] \subset (-\frac{\pi}{2}, \frac{\pi}{2})$, for all $t \in \mathbb{R}_{\geq 0}$; η_d is associated with a desired angular velocity ω_d , as in (12a).

We assume that each robotic agent has continuous feedback of its own state q_i, \dot{q}_i and knows the constant offsets $p_{E_i/O}^{E_i/O}$ and $\eta_{E_i/O}$, implying it can compute the pose and velocity of the object's center of mass via (13) and (14), for $i \in \mathcal{N}$.

As previously mentioned, the dynamics of real robotic systems cannot be accurately known; therefore, we consider that the object and robot dynamic terms in (12), (11) are uncertain, i.e., they are not fully available for feedback

in the control design. Furthermore, as mentioned in Section 1, cooperative object manipulation is part of a larger framework that consists of the *safe* and possibly *timed* execution of high-level tasks. Therefore, we present in the following two *constrained*-control methodologies based on Prescribed Performance Control (PPC) and nonlinear Model Predictive Control (MPC).

5.1. Prescribed Performance Control

We first adopt the concepts and techniques of PPC, proposed in [91], in order to achieve predefined transient and steady-state response for the derived error, as well as ensure that $\theta_o(t)$ evolves in a subset of $(-\frac{\pi}{2}, \frac{\pi}{2})$, for all $t \geq 0$. Prescribed performance characterizes the behavior where a signal evolves strictly within a predefined region that is bounded by absolutely decaying functions of time, called performance functions. This signal is represented by the object's pose error

$$e_s := \left[e_{s_x}, e_{s_y}, e_{s_z}, e_{s_\varphi}, e_{s_\theta}, e_{s_\psi} \right]^\top := x_o - x_d \quad (18)$$

where $x_d := x_d(t) := [(p_d)^\top, (\eta_d)^\top]^\top := [(p_d(t))^\top, (\eta_d(t))^\top]^\top : \mathbb{R}_{\geq 0} \rightarrow \mathbb{M}$. We further assume here that the robotic agents operate away from their kinematic singularities, i.e., each $q_i(t)$ evolves in a closed subset of \mathbf{S}_i , for $i \in \mathcal{N}$. Finally, we consider that the dynamic terms and disturbances appearing in the robot and object models (11) and (12), respectively, are completely unknown and cannot be used in the control design.

The mathematical expressions of prescribed performance are given by the following inequalities:

$$-\rho_{s_k}(t) < e_{s_k}(t) < \rho_{s_k}(t), \forall k \in \mathcal{K}, \quad (19)$$

where $\mathcal{K} := \{x, y, z, \varphi, \theta, \psi\}$ and $\rho_k : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}$, with

$$\rho_{s_k}(t) := (\rho_{s_k,0} - \rho_{s_k,\infty}) \exp(-l_{s_k} t) + \rho_{s_k,\infty}, \quad \forall k \in \mathcal{K}, \quad (20)$$

are designer-specified, smooth, bounded and decreasing positive functions of time with $l_{s_k}, \rho_{s_k,\infty}, k \in \mathcal{K}$, positive parameters incorporating the desired transient and steady-state performance respectively. The terms $\rho_{s_k,\infty}$ can be set

arbitrarily small, achieving thus practical convergence of the errors to zero. Next, we propose a state feedback control protocol that does not incorporate any information on the agents' or the object's dynamics or the external disturbances and guarantees (19) for all $t \geq 0$. More specifically, given the errors (18):

Step I-a. Select the functions ρ_{s_k} as in (20) with

- (i) $\rho_{s_{\theta},0} = \rho_{s_{\theta}}(0) = \theta^*$, $\rho_{s_k,0} = \rho_{s_k}(0) > |e_{s_k}(0)|, \forall k \in \mathcal{K} \setminus \{\theta\}$,
- (ii) $l_{s_k} > 0, \forall k \in \mathcal{K}$,
- (iii) $\rho_{s_k,\infty} \in (0, \rho_{s_k,0}), \forall k \in \mathcal{K}$,

where θ^* is a positive constant satisfying $\theta^* + \bar{\theta} < \frac{\pi}{2}$.

Step I-b. Introduce the normalized errors

$$\xi_s := \left[\xi_{s_x}, \dots, \xi_{s_\psi} \right]^\top := \rho_s^{-1} e_s, \quad (21)$$

where $\rho_s := \text{diag}\{[\rho_{s_k}]_{k \in \mathcal{K}}\} \in \mathbb{R}^{6 \times 6}$, as well as the transformed state functions $\varepsilon_s : (-1, 1)^6 \rightarrow \mathbb{R}^6$, and signals $r_s : (-1, 1)^6 \rightarrow \mathbb{R}^{6 \times 6}$, with

$$\varepsilon_s(\xi_s) := \left[\varepsilon_{s_x}, \dots, \varepsilon_{s_\psi} \right]^\top := \left[\ln \left(\frac{1+\xi_{s_x}}{1-\xi_{s_x}} \right), \dots, \ln \left(\frac{1+\xi_{s_\psi}}{1-\xi_{s_\psi}} \right) \right]^\top \quad (22)$$

$$r_s(\xi_s) := \text{diag}\{[r_{s_k}(\xi_{s_k})]_{k \in \mathcal{K}}\} := \text{diag} \left\{ \left[\frac{\partial \varepsilon_{s_k}}{\partial \xi_{s_k}} \right]_{k \in \mathcal{K}} \right\}, \quad (23)$$

and design the reference velocity vector

$$v_r := -g_s J_O(\eta_O)^{-1} \rho_s(t)^{-1} r_s(\xi_s) \varepsilon_s(\xi_s), \quad (24)$$

where g_s is a positive gain constant.

Step II-a. Define the velocity error vector

$$e_v := \left[e_{v_x}, \dots, e_{v_\psi} \right]^\top := v_O - v_r, \quad (25)$$

and select the corresponding positive performance functions $\rho_{v_k}(t) := (\rho_{v_k,0} - \rho_{v_k,\infty}) \exp(-l_{v_k} t) + \rho_{v_k,\infty}$, such that $\rho_{v_k,0} > \|e_v(0)\|, l_{v_k} > 0$ and $\rho_{v_k,\infty} \in (0, \rho_{v_k,0}), k \in \mathcal{K}$.

Step II-b. Define the normalized velocity error

$$\xi_v := \left[\xi_{v_x}, \dots, \xi_{v_\psi} \right]^\top := \rho_v^{-1} e_v, \quad (26)$$

where $\rho_v := \rho_v(t) := \text{diag}\{[\rho_{v_k}]_{k \in \mathcal{K}}\}$, as well as the transformed states $\varepsilon_v : (-1, 1)^6 \rightarrow \mathbb{R}^6$ and signals $r_v : (-1, 1)^6 \rightarrow \mathbb{R}^{6 \times 6}$, with

$$\varepsilon_v := \varepsilon_v(\xi_v) := [\varepsilon_{v_x}, \dots, \varepsilon_{v_\psi}]^\top := \left[\ln \left(\frac{1+\xi_{v_x}}{1-\xi_{v_x}} \right), \dots, \ln \left(\frac{1+\xi_{v_\psi}}{1-\xi_{v_\psi}} \right) \right]^\top \quad (27)$$

$$r_v(\xi_v) := \text{diag}\{[r_{v_k}(\xi_{v_k})]_{k \in \mathcal{K}}\} := \text{diag} \left\{ \left[\begin{array}{c} \frac{\partial \varepsilon_{v_k}}{\partial \xi_{v_k}} \end{array} \right]_{k \in \mathcal{K}} \right\}, \quad (28)$$

and design the distributed feedback control protocol for each agent $i \in \mathcal{N}$ as

$$u_i := -g_v J_{\mathcal{O}_i}(q_i)^{-1} \rho_v(t)^{-1} r_v(\xi_v) \varepsilon_v(\xi_v), \quad (29)$$

where g_v is a positive constant gain.

Remark 1. The PPC technique guarantees predefined transient and steady-state performance specifications by enforcing the normalized errors ξ_{s_k} , and ξ_{v_k} , $k \in \mathcal{K}$, to remain strictly in $(-1, 1)$ for all $t \geq 0$. Owing to (22) and (27), the proposed control algorithm achieves such a containment simply by maintaining the boundedness of the modulated errors ε_{s_k} , ε_{v_k} , $k \in \mathcal{K}$; a careful inspection of (24) and (29) reveals that the PPC algorithm operates similarly to reciprocal barrier functions in constrained optimization, admitting high negative or positive values depending on whether $e_{s_k}(t) \rightarrow \pm \rho_{s_k}(t)$ and $e_{v_k}(t) \rightarrow \pm \rho_{v_k}(t)$, $k \in \mathcal{K}$, eventually preventing $e_{s_k}(t)$ and $e_{v_k}(t)$ from reaching the respective boundaries.

Remark 2. Notice from (29) that each agent $i \in \mathcal{N}$ calculates its own control signal, rendering thus the overall control scheme distributed. The terms l_k , $\rho_{k,0}$, $\rho_{k,\infty}$, l_{v_k} , and $\rho_{v_k,\infty}$, $k \in \mathcal{K}$ needed for the calculation of the performance functions can be transmitted off-line to the agents. Moreover, the PPC protocol is robust to model uncertainties and external disturbances. In particular, note that the control laws do not even require the structure of the terms \tilde{M} , \tilde{C} , \tilde{g} , \tilde{d} .

The correctness of the PPC algorithm is established in the following theorem.

Theorem 3 ([92]). *Consider N agents rigidly grasping an object with unknown coupled dynamics (17). Then, the distributed control protocol (21)-(29) guarantees that $-\rho_{s_k}(t) < e_{s_k}(t) < \rho_{s_k}(t)$, for all $k \in \mathcal{K}$, $t \geq 0$, from all initial conditions satisfying $|\theta_o(0) - \theta_d(0)| < \theta^*$ (from **Step I-a** (i)), with all closed loop signals being bounded.*

The incorporation of θ^* in the aforementioned theorem stems from the need to comply with the singularity constraints $-\frac{\pi}{2} < \theta_o(t) < \frac{\pi}{2}$. One can avoid such singularities by using different representation for the object orientation, such as unit quaternions or rotation matrices. Algorithms for multi-agent control using

such representations have been developed in [93, 94]. Finally, the developed PPC algorithm can be extended to achieve simultaneous *asymptotic* stability, except for compliance with the performance bounds [95, 96].

5.2. Model Predictive Control

In this section, we consider a more general problem, where we take into account explicit constraints regarding the object and the agents [97, 98, 99]. Such constraints consist of inter-robot collision avoidance, collision avoidance with obstacles, singularity avoidance, and control-input saturation constraints. In particular, consider $J \in \mathbb{N}$ obstacles $\mathcal{O}_j \subset \mathbb{R}^3$, $j \in \mathcal{J} = \{1, \dots, J\}$, as in Section 4, and denote by $\mathcal{A}_i(q_i) \subset \mathbb{R}^3$, $i \in \mathcal{N}$, $\mathcal{A}_o(x_o) \subset \mathbb{R}^3$ the physical volumes occupied by agent i , at state q_i , $i \in \mathcal{N}$, and the object, at state x_o , respectively. The problem here consists of designing the control input u in a distributed way, such that $\lim_{t \rightarrow \infty} x_o(t) = x_d$, while ensuring the satisfaction of the following collision avoidance and singularity properties:

1. $\mathcal{A}_i(q_i(t)) \cap \mathcal{O}_j = \emptyset$, $\mathcal{A}_o(x_o(t)) \cap \mathcal{O}_j = \emptyset, \forall i \in \mathcal{N}, j \in \mathcal{J}$,
2. $\mathcal{A}_i(q_i(t)) \cap \mathcal{A}_\ell(q_\ell(t)) = \emptyset, \forall i, \ell \in \mathcal{N}, i \neq \ell$,
3. $-\frac{\pi}{2} < -\bar{\theta} \leq \theta_o(t) \leq -\bar{\theta} < \frac{\pi}{2}$, $q_i(t) \in \mathcal{S}_i, \forall i \in \mathcal{N}$

for all $t \geq 0$, as well as the input and velocity magnitude constraints: $|\tau_{i,k}| \leq \bar{\tau}_i, |\dot{q}_{i_k}| \leq \bar{q}_i, \forall k \in \{1, \dots, n_i\}, i \in \mathcal{N}$, for some positive constants $\bar{\tau}_i, \bar{q}_i, i \in \mathcal{N}$.

In the following, we assume that the free space is connected, i.e., the set $\{(q, x_o) \in \mathbb{R}^n \times \mathbb{M} : \mathcal{A}_i(q_i) \cap \mathcal{O}_j = \emptyset, \mathcal{A}_i(q_i) \cap \mathcal{A}_\ell(q_\ell) = \emptyset, \mathcal{A}_o(x_o) \cap \mathcal{O}_j = \emptyset, \forall i, \ell \in \mathcal{N}, i \neq \ell, j \in \mathcal{J}\}$ is nonempty and connected and that the each agent can continuously communicate with the other agents and transmit appropriate information.

We further define the sets

$$S_{i,o} := \{q_i \in \mathbb{R}^{n_i} : \mathcal{A}_i(q_i) \cap \mathcal{O}_j = \emptyset, \forall j \in \mathcal{J}\}, \quad \forall i \in \mathcal{N}$$

$$S_A := \{q \in \mathbb{R}^n : \mathcal{A}_i(q_i) \cap \mathcal{A}_\ell(q_\ell) = \emptyset, \forall i, \ell \in \mathcal{N}, i \neq \ell\}$$

$$S_o := \{x_o \in \mathbb{M} : \mathcal{A}_o(x_o) \cap \mathcal{O}_j = \emptyset\}$$

$$S_{o_i} := \{q_i \in \mathbb{R}^{n_i} : \mathcal{A}_o(x_{o_i}(q_i)) \cap \mathcal{O}_j = \emptyset, \forall j \in \mathcal{J}\}, \quad \forall i \in \mathcal{N}$$

$$S_{i,A}(q_{-i}) := \{q_i \in \mathbb{R}^{n_i} : \mathcal{A}_i(q_i) \cap \mathcal{A}_\ell(q_\ell) = \emptyset, \forall \ell \in \mathcal{N} \setminus \{i\}\}, \quad \forall i \in \mathcal{N}$$

associated with the desired collision-avoidance properties, where $q_{-i} := [q_1^\top, \dots, q_{i-1}^\top, q_{i+1}^\top, \dots, q_N^\top]^\top$, for $i \in \mathcal{N}$. The presented results in this section concern a constant reference pose x_d , but can be extended to account for time-varying reference trajectories. Further, we consider that the dynamic uncertainties are represented by the state- and time-dependent functions $d_i(\cdot)$ and $d_o(\cdot)$ of (11) and (12), respectively, while the rest of the dynamics terms are assumed known. Finally, for ease of exposition, we adopt agents with 6 degrees of freedom, i.e., $n_i = 6$, $i \in \mathcal{N}$; the analysis can be extended, however, to redundant agents ($n_i > 6$).

In order to develop a distributed MPC algorithm, we modify the representation of the coupled object-agents dynamics presented so far. We first decouple the dynamics (17) for each agent's MPC. We define $x_{o_i} : \mathbb{R}^{n_i} \rightarrow \mathbb{M}$, $v_{o_i} : \mathbb{R}^{2n_i} \rightarrow \mathbb{R}^6$ with $x_{o_i}(q_i) := [p_{o_i}(q_i)^\top, \eta_{o_i}(q_i)^\top]^\top \in \mathbb{M}$,

$$p_{o_i}(q_i) := p_{E_i}(q_i) + R_i(q_i) p_{O/E_i}^{E_i} \quad (30a)$$

$$\eta_{o_i}(q_i) := \eta_{E_i}(q_i) + \eta_{O/E_i} \quad (30b)$$

for all $i \in \mathcal{N}$, as well as

$$v_{o_i}(q_i, \dot{q}_i) := [\dot{p}_{o_i}(q_i)^\top, \omega_{o_i}(q_i, \dot{q}_i)^\top]^\top := J_{i_o}(q_i) v_i(q_i, \dot{q}_i), \quad \forall i \in \mathcal{N}, \quad (31)$$

where $J_{i_o}(q_i) := J_{o_i}(q_i)^{-1}$, $\forall i \in \mathcal{N}$, which are derived from (13) and (14), respectively; x_{o_i} and v_{o_i} are the pose and velocity of the object as computed by agent $i \in \mathcal{N}$.

Consider now the constants c_i , with $0 < c_i < 1$ and $\sum_{i \in \mathcal{N}} c_i = 1$ that play the role of load sharing coefficients for the agents. Then, the object dynamics (12)

can be written as:

$$\sum_{i \in \mathcal{N}} c_i \left\{ M_O(\eta_{O_i}(q_i)) \dot{v}_{O_i}(q_i, \dot{q}_i) + C_O(\eta_{O_i}(q_i), \omega_{O_i}(q_i, \dot{q}_i)) v_{O_i}(q_i, \dot{q}_i) + g_O + d_O(\cdot) \right\} = \sum_{i \in \mathcal{N}} J_{O_i}(q_i)^\top h_i,$$

from which, by employing the grasp coupling (see (16)), the differential kinematics of the agents, (31), and after straightforward algebraic manipulations, we obtain the coupled dynamics

$$\sum_{i \in \mathcal{N}} \left\{ M_{D_i}(q_i) \ddot{q}_i + C_{D_i}(q_i, \dot{q}_i) \dot{q}_i + g_{D_i}(q_i) + d_{D_i}(q_i, t) \right\} = \sum_{i \in \mathcal{N}} J_{O_i}(q_i)^\top u_i, \quad (32)$$

where

$$\begin{aligned} M_{D_i} &:= c_i M_O J_{i_O} J_i + J_{O_i}^\top M_i J_i, \\ C_{D_i} &:= J_{O_i}^\top \left(M_i \dot{J}_i + C_i J_i \right) + c_i M_O J_{i_O} \dot{J}_i + c_i M_O \dot{J}_{i_O} J_i + c_i C_O, \\ g_{D_i} &:= c_i g_O + J_{O_i}^\top g_i, \\ d_{D_i} &:= c_i d_O + J_{O_i}^\top d_i \end{aligned}$$

for all $i \in \mathcal{N}$.

To design a distributed NMPC control scheme, 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 (32) 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 *nominal* dynamics:

$$M_{D_1} \ddot{q}_1 + C_{D_1} \dot{q}_1 + g_{D_1} = J_{O_1}^\top u_1, \quad (33)$$

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 , it transmits the corresponding predicted state for the object $x_{O_1}(q_1), v_{O_1}(q_1, \dot{q}_1)$

for the control horizon to the other agents $\{2, \dots, N\}$. Then, the followers solve the receding horizon NMPC subject to the forward nominal dynamics:

$$M_{D_i} \ddot{q}_i + C_{D_i} \dot{q}_i + g_{D_i} = J_{O_i}^\top u_i, \quad (34)$$

the state equality constraints:

$$x_{O_i}(q_i) = x_{O_1}(q_1), v_{O_i}(q_i, \dot{q}_i) = v_{O_1}(q_1, \dot{q}_1), \quad (35)$$

for $i \in \{2, \dots, N\}$, 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 $\{1, \dots, N\}$. 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 (33), (34) in the following sense. Although the coupled model (32) 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 (32) is satisfied, since it's the result of the addition of (33) and (34), for $i = 1$ and every $i \in \{2, \dots, N\}$, 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 (35) guarantee that the predicted variables of the agents $\{2, \dots, N\}$ will comply with the rigidity at the grasping points.

By using the notation $x_{q_i} := [x_{q_i,1}^\top, x_{q_i,2}^\top]^\top := [q_i^\top, \dot{q}_i^\top]^\top \in \mathbb{R}^{2n_i}$, $i \in \mathcal{N}$, the nonlinear dynamics of each agent can be written as:

$$\dot{x}_{q_i} = f_{q_i}(x_{q_i}, u_i) + d_{q_i}(x_{q_i}, t) := \begin{bmatrix} f_{q_i,1}(x_{q_i}) \\ f_{q_i,2}(x_{q_i}, u_i) \end{bmatrix} + \begin{bmatrix} 0 \\ -\widehat{M}_{D_i}(q_i) d_{D_i}(q_i, t) \end{bmatrix}, \quad (36)$$

where

$$\begin{aligned} f_{q_i,1}(x_{q_i}) &:= x_{q_i,2}, \\ f_{q_i,2}(x_{q_i}, u_i) &:= \widehat{M}_{D_i}(q_i) \left(J_{O_i}(q_i)^\top u_i - C_{D_i}(q_i, \dot{q}_i) \dot{q} - g_{D_i}(q_i) \right), \end{aligned}$$

and $\widehat{M}_{D_i}(q_i) := M_{D_i}(q_i)^\top (M_{D_i}(q_i)M_{D_i}(q_i)^\top)^{-1}$. for all $i \in \mathcal{N}$. The *nominal* dynamics of each agent are given by omitting the disturbance $d_{q_i}(\cdot)$:

$$\dot{\bar{x}}_{q_i} = f_{q_i}(\bar{x}_{q_i}, \bar{u}_i) := \begin{bmatrix} f_{q_i,1}(\bar{x}_{q_i}) \\ f_{q_i,2}(\bar{x}_{q_i}, \bar{u}_i) \end{bmatrix}, \quad (37)$$

and $\bar{\cdot}$ is used to denote the nominal signals. It can be concluded that the matrices $\widehat{M}_{D_i}(q_i) + \widehat{M}_{D_i}(q_i)^\top$ are positive definite when $q_i \in \mathcal{S}_i$, $i \in \mathcal{N}$. Therefore, one also concludes that the linearized nominal systems are stabilizable around $\bar{x}_{q_i} = 0$.

We define now the error and nominal error as:

$$e_{D_1}(x_{q_1}) := \begin{bmatrix} x_{o_1}(q_1) - x_d \\ v_{o_1}(q_1, \dot{q}_1) \end{bmatrix}, \quad \bar{e}_{D_1}(\bar{x}_{q_1}) := \begin{bmatrix} x_{o_1}(\bar{q}_1) - x_d \\ v_{o_1}(\bar{q}_1, \dot{\bar{q}}_1) \end{bmatrix} \quad (38)$$

which gives us the dynamics:

$$\dot{e}_{D_1} = f_{D_1}(x_{q_1}, t, u_1), \quad (39a)$$

$$\dot{\bar{e}}_{D_1} = \bar{f}_{D_1}(\bar{x}_{q_1}, \bar{u}_1), \quad (39b)$$

with

$$\begin{aligned} f_{D_1}(x_{q_1}, u_1) &:= \\ &\begin{bmatrix} J_{o_1}(\eta_{o_1}(q_1))J_{1o}(q_1)J_1(q_1)\dot{q}_1 \\ J_{1o}(q_1)J_1(q_1)(f_{q_1,2}(x_{q_1}, u_1) - \widehat{M}_{D_1}(q_1)d_{D_1}(q_1, t)) + (J_{1o}(q_1)\dot{J}_1(q_1) + \dot{J}_{1o}(q_1)J_1(q_1))\dot{q}_1 \end{bmatrix} \\ \bar{f}_{D_1}(\bar{x}_{q_1}, \bar{u}_1) &:= \\ &\begin{bmatrix} J_{o_1}(\eta_{o_1}(\bar{q}_1))J_{1o}(\bar{q}_1)J_1(\bar{q}_1)\dot{\bar{q}}_1 \\ J_{1o}(\bar{q}_1)J_1(\bar{q}_1)f_{q_1,2}(\bar{x}_{q_1}, \bar{u}_1) + (J_{1o}(\bar{q}_1)\dot{J}_1(\bar{q}_1) + \dot{J}_{1o}(\bar{q}_1)J_1(\bar{q}_1))\dot{\bar{q}}_1 \end{bmatrix} \end{aligned}$$

where we employed (39) and the object dynamics.

The input-constraint sets are now defined as

$$U_{D_i} := \left\{ u_i \in \mathbb{R}^6 : \|u_i\| \leq \frac{\bar{\tau}_i}{\sigma_{\min}(J_i^\top)} \right\},$$

where $\sigma_{\min}(\cdot)$ denotes the minimum singular value. Define also the sets

$$X_{D_1}(q_{-1}) := \left\{ x_{q_1} \in \mathbb{R}^{2n_1} : \theta_{o_1}(q_1) \in [-\bar{\theta}, \bar{\theta}], |\dot{q}_{1k}| \leq \bar{q}_1, \forall k \in \{1, \dots, n_1\}, \right. \\ \left. q_1 \in \mathcal{S}_1 \cap \mathcal{S}_{1,A}(q_{-1}) \cap \mathcal{S}_{1,o} \cap \mathcal{S}_{o_1} \right\}$$

$$X_{D_i}(q_{-i}) := \left\{ x_{q_i} \in \mathbb{R}^{2n_i} : |\dot{q}_{ik}| \leq \bar{q}_i, \forall k \in \{1, \dots, n_i\}, q_i \in \mathcal{S}_i \cap \mathcal{S}_{i,A} \cap \mathcal{S}_{i,o} \right\},$$

for all $i \in \{2, \dots, N\}$. The sets X_{D_i} capture all the state constraints of the system dynamics (36), i.e., representation- and singularity-avoidance, collision avoidance among the agents and the obstacles, and collision avoidance of the object with the obstacles. Such constraints are assigned to the leader agent only. We further define the set

$$E_{D_1}(q_{-1}) := \{e_{D_1}(x_{q_1}) \in \mathbb{R}^{12} : x_{q_1} \in X_{D_1}(q_{-1})\},$$

which now represents the constraints set for the NMPC scheme of the leader.

The main problem at hand is the design of a *feedback control law* $u_1 \in U_{D_1}$ for agent 1 that guarantees that the error signal e_{D_1} with dynamics given in (39), satisfies $\lim_{t \rightarrow \infty} \|e_{D_1}(x_{q_1}(t))\| \rightarrow 0$, while ensuring the aforementioned constraints. The role of the followers $\{2, \dots, N\}$ is, through the load-sharing coefficients c_2, \dots, c_N in (32), to contribute to the object trajectory execution, as derived by the leader agent 1, while also avoiding collisions. In order to solve the aforementioned problem, we propose a NMPC scheme, which is presented hereafter.

The control law for the leader agent consists of a nominal control action \bar{u}_1 and a state feedback law $\kappa_1(e_{D_1}, \bar{e}_{D_1})$. As it will be presented hereafter, \bar{u}_1 will be the outcome of a nominal finite-horizon optimal control problem (FHOC), solved at each sampling time instant; and the feedback law $\kappa_1(e_{D_1}, \bar{e}_{D_1})$ is used to guarantee that the real trajectory $e_{D_1}(x_{q_1}(t))$ remains in bounded hyper-tubes centered in the nominal trajectories $\bar{e}_{D_1}(\bar{x}_{q_1}(t))$, for all times. The volume of the hyper-tubes depends on the upper bound of the disturbances d_{D_1} as well as the agent dynamics. The state feedback law is designed as

$$\kappa_1(e_{D_1}, \bar{e}_{D_1}) = -k_{D_1}(e_{D_1} - \bar{e}_{D_1}),$$

where k_{D_1} is a positive control gain. By appropriately setting k_{D_1} , we guarantee that $e_{D_1} - \bar{e}_{D_1} \in \mathcal{Z}_1$, where \mathcal{Z}_1 is a closed ball in \mathbb{R}^{12} . More details are provided in [100].

Next, we describe the design of the component \bar{u}_1 . Consider a sequence of sampling times $\{t_j\}$, $j \in \mathbb{N}$ with $t_{j+1} = t_j + h_s$, $h_s \in (0, T_p)$, and T_p the respective horizon. For agent 1, the open-loop input signal applied in between the sampling instants is given by the solution of the following FHOC:

$$\begin{aligned} \min_{\bar{u}_1(\cdot)} J_{D_1}(\bar{e}_{D_1}(\bar{x}_{q_1}(t_j)), \bar{u}_1(\cdot)) &:= \min_{\bar{u}_1(\cdot)} \left\{ V_{D_1}(\bar{e}_{D_1}(\bar{x}_{q_1}(t_j + T_p))) \right. \\ &\quad \left. + \int_{t_j}^{t_j + T_p} [F_{D_1}(\bar{e}_{D_1}(\bar{x}_{q_1}(s)), \bar{u}_1(s))] ds \right\} \end{aligned} \quad (40a)$$

subject to:

$$\dot{\bar{e}}_{D_1}(\bar{x}_{q_1}(s)) = \bar{f}_{D_1}(\bar{x}_{q_1}(s), \bar{u}_1(s)), \quad \bar{e}_{D_1}(\bar{x}_{q_1}(t_j)) = \bar{f}_{D_1}(\bar{x}_{q_1}(t_j)), \quad (40b)$$

$$\bar{e}_{D_1}(\bar{x}_{q_1}(s)) \in \bar{E}_{D_1}(q_{-1}(t_j)), \quad s \in [t_j, t_j + T_p], \quad (40c)$$

$$\bar{u}_1(s) \in \bar{U}_{D_1}, \quad s \in [t_j, t_j + T_p], \quad (40d)$$

$$\bar{e}_{D_1}(\bar{x}_{q_1}(t_j + T_p)) \in \mathcal{E}_{D_1}. \quad (40e)$$

At a generic time t_j then, agent 1 solves the aforementioned FHOCP. The functions $F_{D_1} : \mathbb{R}^{12} \times \mathbb{R}^6 \rightarrow \mathbb{R}_{\geq 0}$, $V_{D_1} : \mathbb{R}^{12} \rightarrow \mathbb{R}_{\geq 0}$ stand for the *running cost* and the *terminal penalty cost*, respectively, and they are defined as: $F_{D_1}(e_{D_1}, u_1) := e_{D_1}^\top Q_{D_1} e_{D_1} + u_1^\top R_{D_1} u_1$, $V_{D_1}(e_{D_1}) := e_{D_1}^\top P_{D_1} e_{D_1}$, where $R_{D_1} \in \mathbb{R}^{6 \times 6}$ and $P_{D_1} \in \mathbb{R}^{12 \times 12}$ are symmetric and positive definite gain matrices; $Q_{D_1} \in \mathbb{R}^{12 \times 12}$ is a symmetric and positive semi-definite controller gain matrix and \mathcal{E}_{D_1} is the terminal set used to force stability of the closed-loop system.

We will explain hereafter the form of the sets $\bar{E}_{D_1}(q_{-1}(t_j))$ and \bar{U}_{D_1} . In order to guarantee that, while agent 1 is solving the FHOCP (40) for its nominal system, the real trajectory $e_{D_1}(x_{q_1}(t))$ and the control inputs $u_1(t)$ satisfy the state and input constraints $E_{D_1}(q_{-1}(t_j))$ and U_{D_1} , respectively, the sets $E_{D_1}(q_{-1}(t_j))$ and U_{D_1} need to be properly modified. In view of the fact that $e_{D_1} - \bar{e}_{D_1} \in \mathcal{Z}_1$ under the feedback control law $\kappa(e_{D_1}, \bar{e}_{D_1})$, it holds that $e_{D_1}(x_{q_1}(s)) \in \mathcal{Z}_1 \oplus \{\bar{e}_{D_1}(\bar{x}_{q_1}(s))\}$, for $s \in [t_j, t_j + T_p]$. Therefore, $\bar{E}_{D_1}(q_{-1}(t_j))$ is defined as $\bar{E}_{D_1}(q_{-1}(t_j)) := E_{D_1}(q_{-1}(t_j)) \ominus \mathcal{Z}_1$. By following a similar reasoning, \bar{U}_{D_1} is defined as $\bar{U}_{D_1} := U_{D_1} \ominus [(-k_{D_1}) \circ \mathcal{Z}_1]$. Intuitively, the sets $E_{D_1}(q_{-1}(t_j))$ and U_{D_1} are tightened due to the difference between $e_{D_1}(x_{q_1}(t))$ and $\bar{e}_{D_1}(\bar{x}_{q_1}(t))$.

The solution to FHOCP (40a) - (40e) starting at time t_j provides an optimal control input, denoted by $\bar{u}_1^*(s; e_{D_1}(x_{q_1}(t_j)), x_q(t_j))$, $s \in [t_j, t_j + T_p]$, $x_q := [x_{q_1}^\top, \dots, x_{q_N}^\top]^\top$. This control input is then applied to the system until the next sampling instant t_{j+1} :

$$\bar{u}_1(s; \bar{e}_{D_1}(\bar{x}_{q_1}(t_j)), \bar{x}_q(t_j)) = \bar{u}_1^*(s; e_{D_1}(x_{D_1}(t_j)), x_q(t_j)), \quad \forall s \in [t_j, t_{j+1}]. \quad (41)$$

At time $t_{j+1} = t_j + h_s$, a new FHOCP is solved in the same manner, leading to a receding horizon approach. The control input $\bar{u}_1(\cdot)$ is of feedback form, since it is recalculated at each sampling instant based on the then-current state.

After the solution of the FHOCP and the calculation of the predicted states $\bar{x}_{q_1}(s)$, $s \in [t_j, t_{j+1})$, at each time instant t_j , agent 1 transmits the values $\bar{q}_1(s)$, $\dot{\bar{q}}_1(s)$ as well as $x_{o_1}(\bar{q}_1(s))$ and $v_{o_1}(\bar{q}_1(s), \dot{\bar{q}}_1(s))$, computed by (30), (31), $s \in [t_j, t_j + T_p]$, to the rest of the agents $\{2, \dots, N\}$. The rest of the agents then proceed as follows. Each agent $i \in \{2, \dots, N\}$ applies the control input $u_i = \bar{u}_i + \kappa_i(\cdot)$, similarly to the leader; $\kappa_i(\cdot)$ is a closed-form state feedback law associated with some error function, such as e_{D_i} defined as in (38), and \bar{u}_i is the solution of the FHOCP:

$$\min_{\bar{u}_i(\cdot)} J_{D_i}(\bar{x}_{q_i}(t_j), \bar{u}_i(\cdot)) \quad (42a)$$

subject to:

$$\dot{\bar{x}}_{q_i} = f_{q_i}(\bar{x}_{q_i}(s), \bar{u}_i(s)), \quad (42b)$$

$$\bar{x}_{q_i}(s) \in X_i(\bar{q}_1(s), \dots, \bar{q}_{i-1}(s), q_{i+1}(t_j), \dots, q_N(t_j)), \quad (42c)$$

$$x_{o_i}(\bar{q}_i(s)) = x_{o_1}(\bar{q}_1(s)), \quad (42d)$$

$$v_{o_i}(\bar{q}_i(s), \dot{\bar{q}}_i(s)) = v_{o_1}(\bar{q}_1(s), \dot{\bar{q}}_1(s)), \quad (42e)$$

$$\bar{u}_i(s) \in U_{D_i}, \quad (42f)$$

for $s \in [t_j, t_j + T_p]$ and at every sampling time t_j , where J_{D_i} is an associated cost function. The constraint (42c) guarantees that agent i will obtain a trajectory that does not collide with the predicted trajectories of the agents higher in priority, or the agents lower in priority at t_j . Note that, through the equality constraints (42d), (42e), the follower agents must comply with the trajectory computed by the leader $\bar{q}_1(s), \dot{\bar{q}}_1(s)$. We further assume that the aforementioned constraints are feasible, i.e., the sets $\{q \in \mathbb{R}^n : x_{o_i}(\bar{q}_i(s)) = x_{o_1}(\bar{q}_1(s)), v_{o_i}(\bar{q}_i(s), \dot{\bar{q}}_i(s)) = v_{o_1}(\bar{q}_1(s), \dot{\bar{q}}_1(s)), q_i \in X_i(\bar{q}_1(s), \dots, \bar{q}_{i-1}(s), q_{i+1}(t_j), \dots, q_N(t_j))\}$ are nonempty, for all $i \in \{2, \dots, N\}$, $s \in [t_j, t_j + T_p]$, $j \in \mathbb{N}$.

Next, similarly to the leader agent, agent $i > 1$ calculates the predicted states $\bar{q}_i(s), \dot{\bar{q}}_i(s)$, $s \in [t_j, t_j + T_p]$, which then transmits to the agents $\{i + 1, \dots, N\}$. In that way, at each time instant t_j , each agent $i \in \{2, \dots, N\}$ receives the other agents' states, incorporates the constraint (42c) for the agents $\{i + 1, \dots, N\}$, receives the predicted states $\bar{q}_\ell(s), \dot{\bar{q}}_\ell(s)$ from the agents $\ell \in \{2, \dots, i - 1\}$ and incorporates the collision avoidance constraint (42c) 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 $\bar{q}_\ell(s), \ell \in \{1, \dots, i - 1\}$ 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 (40a)-(40e)), the follower agents contribute in executing this trajectory through the load sharing coefficients c_i (as indicated in the coupled model (32)), and the agents low in priority are responsible for collision avoidance with the agents of higher priority. Moreover, the aforementioned equality constraints (42d), (42e) as well as the forward dynamics (42a) guarantee the compliance of all the followers with the model (32).

Given the constrained FHOCP (42a)-(42f), the solution of the problem lies in the capability of the leader agent to produce a state trajectory that guarantees $x_{o_1}(q_1(t)) \rightarrow x_{\text{des}}$, by solving the FHOCP (40a)-(40e) [98]. A formal proof is similar to the ones presented in [101, 100] and is therefore omitted.

5.3. Energy-Optimal Cooperative Manipulation

An important aspect in cooperative manipulation systems is the regulation of internal forces. Internal forces are forces exerted by the agents at the grasping points that do not contribute to the motion of the object. While a certain amount of such forces is required in many cases (e.g., to avoid contact loss in multi-fingered manipulation), they need to be minimized in order to prevent object damage and unnecessary effort of the agents. In particular, the forces $h = [h_1^\top, \dots, h_N^\top]^\top$ of (11) between the object and the agents can be decoupled into motion-induced and internal forces

$$h = h_{\text{in}} + h_{\text{int}}.$$

The internal forces h_{int} are squeezing forces that the agents exert to the object and belong to the nullspace of $G(x)$ (i.e., $G(x)h_{\text{int}} = 0$). When $h = h_{\text{int}}$, it holds that $G(x)(u - M\dot{v} - Cv - g - d) = 0$ and the object moves according to

$h_o = M_o \dot{v}_o + C_o v_o + g_o = 0$. Hence, h_{int} does not contribute to the object's motion and results in internal stresses that might damage it.

In cooperative manipulation schemes, the most energy-efficient way of transporting an object is to exploit the full potential of the cooperating robotic agents, i.e., each agent does not exert less effort at the expense of other agents, which might then potentially exert more effort than necessary. For instance, consider a rigid cooperative manipulation scheme, with only one agent (a leader) working towards bringing the object to a desired location, whereas the other agents have zero inputs. Since the grasps are rigid, if the leader is equipped with sufficiently powerful actuators, it will achieve the task by “dragging” the rest of the agents, compensating for their dynamics, and creating non-negligible internal forces. In such cases, when the cooperative manipulation system is rigid (i.e., the grasps are considered to be rigid), the optimal strategy of transporting an object is achieved by regulating the internal forces to zero. Therefore, from a control perspective, the goal of a rigid cooperative manipulation system is to design a control protocol that achieves a desired cooperative manipulation task, while guaranteeing that the internal forces remain zero.

By employing Gauss' principle for constrained motion and exploiting the grasp rigidity, we obtain a closed-form expression for the internal forces as $h_{\text{int}} = -M^{\frac{1}{2}}(A_R M^{-\frac{1}{2}})^{\dagger}(A_R v + A_R \alpha_R)$, where $\alpha_R := M^{-1}(u - C v - g - d)$ and A_R is the constraint matrix

$$A_R = \begin{bmatrix} I_3 & S_s(p_{12}) & -I_3 & 0_{3 \times 3} & \dots & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 3} & -I_3 & \dots & 0_{3 \times 3} & 0_{3 \times 3} \\ \vdots & \vdots & \dots & \ddots & \ddots & \vdots & \vdots \\ I_3 & S_s(p_{1N}) & 0_{3 \times 3} & 0_{3 \times 3} & \dots & -I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 3} & \dots & 0_{3 \times 3} & -I_3 \end{bmatrix}$$

with $p_{ij} := p_{E_i} - p_{E_j}$, $i, j \in \mathcal{N}^3$. Based on the aforementioned expression, we derive new results on the expression of h_{int} as well as the internal-force-free distribution of an object force to the agents.

Theorem 4 ([102, 103]). *Consider N robotic agents rigidly grasping an object. Then it holds that*

$$h_{\text{int}} = (I_{6N} - M G^{\top} (G M G^{\top})^{-1} G) h.$$

Further, let a desired force $h_{o,d} \in \mathbb{R}^6$ to be applied to the object, which is distributed to the agents' desired forces as $h_d = G^* h_{o,d}$, and where G^* is a right-inverse of G , i.e., $G G^* = I_6$. Then it holds that

$$h_{\text{int}} = 0 \Leftrightarrow G^* = M G^{\top} (G M G^{\top})^{-1}.$$

³It can be proven that the matrix A_R is equivalent to the *rigidity* matrix in rigid formations when the manipulator end-effectors are viewed as the nodes of a graph [102]

Based on the aforementioned result, we provide next a control protocol that guarantees convergence to a reference x_d while guaranteeing regulation of internal forces to zero. In order to avoid the representation singularity of $\theta_o = \frac{\pi}{2}$ (see (12a)), we use the orientation error metric $e_o = \frac{1}{2}\text{tr}(I_3 - R_d^\top R_o)$, where $R_d : \mathbb{R}_{\geq 0} \rightarrow \text{SO}(3)$ is a desired rotation matrix, corresponding to η_d and evolving according to $\dot{R}_d = S_s(\omega_d)R_d$. which we desire to drive to zero [102]. The aforementioned singularity is translated now to an undesired equilibrium located at the configuration that satisfies $e_o = 2$. The proposed control protocol guarantees $e_o(t) < 2$ for all $t \geq 0$.

Corollary 1 ([102, 94]). *Consider N robotic agents rigidly grasping an object, with coupled dynamics (17). Further assume that $d_o(\cdot) = d_i(\cdot) = 0$, for all $i \in \mathcal{N}$, and $e_o(0) < 2$. Consider the control law*

$$u = g + (CG^\top + MG^\top)v_o + G^*(g_o + C_o v_o) + (MG^\top + G^*M_o)(\dot{v}_d - K_d e_v - K_p e_x) \quad (43)$$

where $e_v := v_o - v_d$, $v_d := [\dot{p}_d^\top, \omega_d^\top]^\top \in \mathbb{R}^6$, $e_x := [e_p^\top, \frac{1}{2(2-e_o)^2} e_R^\top R_o^\top]^\top$, $K_p := \text{diag}\{K_{p_1}, k_{p_2} I_3\}$, where $K_{p_1} \in \mathbb{R}^{3 \times 3}$, $K_d \in \mathbb{R}^{6 \times 6}$ are positive definite matrices, and $k_{p_2} \in \mathbb{R}_{>0}$ is a positive constant. Then the solution of the closed-loop coupled system satisfies: 1) $e_o(t) < 2$, for all $t \in \mathbb{R}_{\geq 0}$, 2) $\lim_{t \rightarrow \infty} (p_o(t) - p_d(t)) = 0$, $\lim_{t \rightarrow \infty} R_d(t)^\top R_o(t) = I_3$, and 3) there are no internal forces, i.e., $h_{\text{int}}(t) = 0$, for all $t \in \mathbb{R}_{\geq 0}$, if and only if $G^* = MG^\top(GMG^\top)^{-1}$.

Note that the employed inverse dynamics controller requires knowledge of the agent and object dynamics. In case of dynamic parameter uncertainty, standard adaptive and/or robust control schemes (like the ones of the previous sections) that attempt to compensate for potential uncertainties in the model would intrinsically create internal forces, since the dynamics of the system would not be explicitly accounted for. Further, note that $G^* = MG^\top(GMG^\top)^{-1}$ induces an implicit and natural load-sharing scheme via the incorporation of M . More specifically, note that the force distribution to the robotic agents via $Gh_{o,d}$ yields for each agent $M_i J_{o_i} (\sum_{i \in \mathcal{N}} J_{o_i}^\top M_i J_{o_i})^{-1}$, for all $i \in \mathcal{N}$. Hence, larger values of M_i will produce larger inputs for agent i , implying that agents with larger inertia characteristics will take on a larger share of the object load. Note that this is also a desired load-sharing scheme, since larger dynamic values usually imply more powerful robotic agents.

5.4. Simulation and Experimental Results

We first present experimental results for the PPC scheme of Section 5.1. The tested scenario for the experimental setup consists of two WidowX Robot Arms rigidly grasping a wooden cuboid object of initial pose $x_o(0) = [0.3, 0, 0.15, 0, 0, 0]^\top$ ([m], [rad]), which has to track a planar time trajectory $p_d(t) = [0.3 + 0.05 \sin(\frac{2\pi t}{35}), 0.15 - 0.05 \cos(\frac{2\pi t}{35})]^\top$, $\eta_d(t) = \frac{\pi}{20} \sin(\frac{5\pi t}{35})$. We set the performance functions as $\rho_{s_x}(t) = \rho_{s_z}(t) = 0.03 \exp(-0.2t) + 0.02$ [m], $\rho_{s_\theta}(t) = 0.2 \exp(-0.2t) + 0.2$ [rad], $\rho_{v_x}(t) = 5 \exp(-0.2t) + 5$ [m/s], $\rho_{v_z}(t) = 5 \exp(-0.2t) + 10$ [m/s], and $\rho_{v_\theta}(t) = 4 \exp(-0.2t) + 3$ [m/s], and the control gains of (24) and (29) as

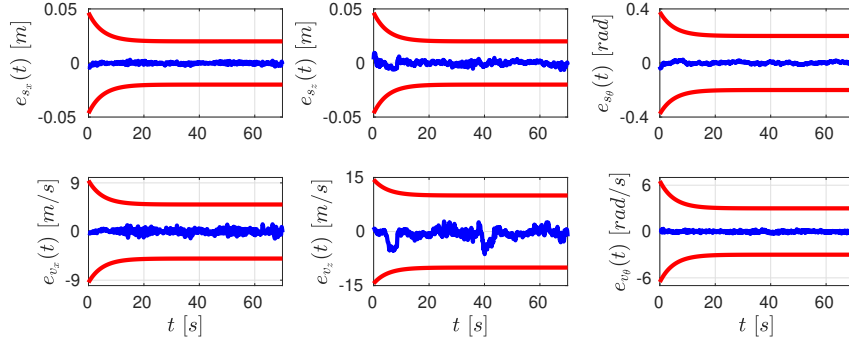


Figure 3: Experimental results for the PPC of Section 5.1. Top: the pose errors $e_{s_x}(t)$, $e_{s_z}(t)$, $e_{s_\theta}(t)$ (with blue) along with the respective performance functions (with red); Bottom: The velocity errors $e_{v_x}(t)$, $e_{v_z}(t)$, $e_{v_\theta}(t)$ (with blue) along with the respective performance functions (with red), $\forall t \in [0, 70]$.

$g_s = 0.05$ and $g_v = 10$, respectively. The experimental results are depicted in Fig. 3, which shows the pose and velocity errors $e_s(t)$, $e_v(t)$ along with the respective performance functions, for $t \in [0, 70]$ seconds. A video illustrating the experimental results can be found on <https://youtu.be/jJWeI5ZvQPY>.

Next, we demonstrate the MPC scheme of Section 5.2. We consider a simulation example with $N = 3$ ground vehicles equipped with 2-DOF manipulators, rigidly grasping an object with $n_1 = n_2 = n_3 = 4$. The states of the agents are given as: $q_i = [p_{B_i}^\top, \alpha_i^\top]^\top \in \mathbb{R}^4$, where $p_{B_i} = [x_{B_i}, y_{B_i}]^\top \in \mathbb{R}^2$ are the vehicles' position and $\alpha_i = [\alpha_{i_1}, \alpha_{i_2}]^\top \in \mathbb{R}^2$ are the manipulators' joint angles, $i \in \mathcal{N} = \{1, 2, 3\}$. The manipulators become singular when $\sin(\alpha_{i_1}) = 0$, $i \in \mathcal{N}$, thus the state constraints for the manipulators are set to $\epsilon \leq \alpha_{i_1} \leq \frac{\pi}{2} - \epsilon$, $-\frac{\pi}{2} + \epsilon \leq \alpha_{i_2} \leq \frac{\pi}{2} - \epsilon$, $i \in \mathcal{N}$. We also consider the input constraints: $-10 \leq u_{i,j}(t) \leq 10$, $i \in \mathcal{N}$, $j \in \{1, \dots, 4\}$. The initial conditions of agents and the object are set to: $q_1(0) = [0.5, 0, \frac{\pi}{4}, \frac{\pi}{4}]^\top$, $q_2(0) = [0, -4.4142, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $q_3(0) = [-0.50, -4.4142, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $\dot{q}_1(0) = \dot{q}_2(0) = \dot{q}_3(0) = [0, 0, 0, 0]^\top$, $x_o(0) = [0, -2.2071, 0.9071, \frac{\pi}{2}]^\top$, and $\dot{x}_o(0) = [0, 0, 0, 0]^\top$ (in [rad], [rad/s] ([m], [rad]), ([m/s], [rad/s]), respectively). The desired goal state the object is set to $x_d = [5, -2.2071, 0.9071, \frac{\pi}{2}]^\top$ ([m], [rad]), which, due to the structure of the considered robots, corresponds uniquely to $q_{1,d} = [5.5, 0, \frac{\pi}{4}, \frac{\pi}{4}]^\top$, $q_{2,d} = [5, -4.4142, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $q_{3,d} = [4.5, 0, -\frac{\pi}{4}, -\frac{\pi}{4}]^\top$, $\dot{q}_{1,d} = \dot{q}_{2,d} = \dot{q}_{3,d} = [0, 0, 0, 0]^\top$. ([rad] and [rad/s], respectively). We set an obstacle between the initial and the desired pose of the object. The obstacle is spherical with center $(2.5, -2.2071, 1)$ m and radius $\sqrt{0.2}$ m. The sampling time is $h_s = 0.1$ seconds, the horizon is $T_p = 0.5$ seconds, and the total simulation time is 60 seconds; The matrices P_{D_i} , Q_{D_i} , R_{D_i} are set to: $P_{D_i} = Q_{D_i} = 0.5I_8$, $R_{D_i} = 0.5I_4$, $\forall i \in \mathcal{N}$, 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. 4, which shows the error states of the agents, successfully converging to zero. The simulation was carried out by using the NMPC toolbox given in [104] and it took 13450 sec in

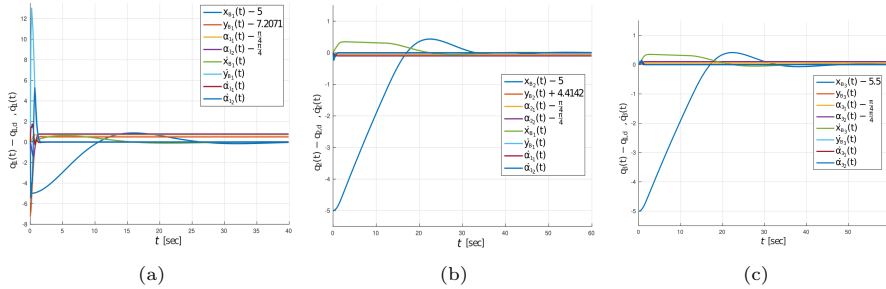


Figure 4: The error states of the three agent for the MPC scheme.

MATLAB Environment on a desktop computer with 8 cores, 3.60 GHz CPU and 16GB of RAM.

Finally, we demonstrate the internal-force-free control algorithm of Section 5.3. We consider 4 6-DOF UR5 robotic manipulators in the realistic dynamic environment CoppeliaSim [105]. The 4 agents are rigidly grasping an object of 40 kg at an initial configuration $p_o(0) = [-0.225, -0.612, 0.161]^T$, $\eta_o(0) = [0, 0, 0]^T$. In order to verify the theoretical findings of the previous sections, we apply the control law (43) to achieve tracking of a desired trajectory by the object's center of mass. We simulate the closed-loop system for two cases of G^* , namely the proposed one $G_1^* = MG^T(GMG^T)^{-1}$ as well as the more standard choice $G_2^* = G^T(GG^T)^{-1}$. We set the desired trajectory as $p_d(t) = p_o(0) + [0.2 \sin(w_p t + \varphi_d), 0.2 \cos(w_p t + \varphi_d), 0.09 + 0.1 \sin(w_p t + \varphi_d)]^T$, $\eta_d(t) = [0.15 \sin(w_\phi t + \varphi_d), 0.15 \sin(w_\theta t + \varphi_d), 0.15 \sin(w_\psi t + \varphi_d)]^T$ (in [m] and [rad], respectively), where $\varphi_d = \frac{\pi}{6}$, $w_p = w_\phi = w_\psi = 1$, $w_\theta = 0.5$, and $\eta_d(t)$ is transformed to the respective $R_d(t)$. The control gains are set as $K_{p_1} = 15$, $k_{p_2} = 75$, and $K_d = 40I_6$. The results are given in Figs. 5 and 6 for 15 seconds. Fig. 5 depicts the pose and velocity errors $e_p(t)$, $e_o(t)$, $e_v(t)$, which are shown to converge to zero for both choices of G^* , as expected. Fig. 6 depicts the norm of the internal forces, $\|h_{\text{int}}(t)\|$. It is clear that G_2^* yields significantly large internal forces, whereas G_1^* keeps them very close to zero, as proven in the theoretical analysis. A video illustrating the aforementioned simulation can be found on <https://youtu.be/a31LTBBkE-Q>.

6. Abstractions of Multi-Agent and Multi-Agent-Object Systems

The control algorithms of the previous sections offer a way to abstract the underlying continuous system to a discrete representation. Such a discrete representation can be used to synthesize a higher-level controller such that the system satisfies a temporal-logic task. In the following, we present such an abstraction and controller synthesis for both multi-agent and multi-agent-object systems.

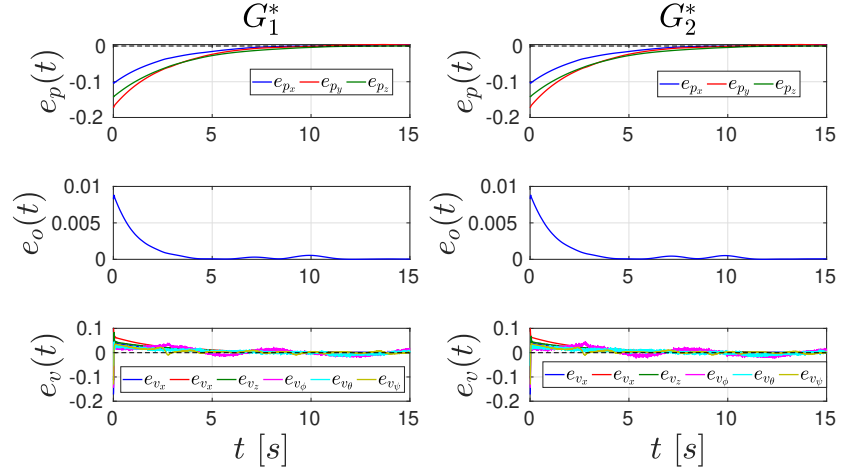


Figure 5: The error metrics $e_p(t)$, $e_o(t)$, $e_v(t)$, respectively, top to bottom, for the two choices G_1^* and G_2^* and $t \in [0, 15]$ seconds.

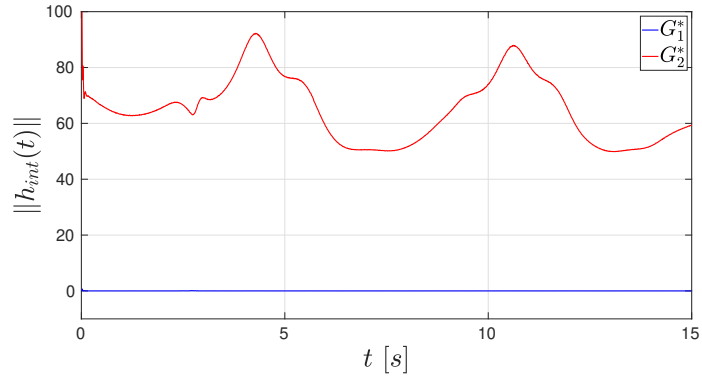


Figure 6: The norm of the internal forces $\|h_{int}(t)\|$ for the two cases of G^* and $t \in [0, 15]$ seconds.

6.1. Multi-Agent Systems

We first consider the distributed control-synthesis problem for a multi-agent system subject to linear temporal logic (LTL) tasks [106, 107]. Formally, we consider N agents, characterized by $p_i \in \mathbb{R}^n$, $i \in \mathcal{N} = \{1, \dots, N\}$ and operating in a static workspace \mathcal{W} . Similarly to Section 4, we consider that the agents evolve subject to 2nd-order uncertain dynamics and their physical volume is bounded by the closed spheres $\bar{\mathcal{B}}(p_i, r_i)$, where r_i is the radius of agent $i \in \mathcal{N}$. The workspace is populated with $J \geq 0$ obstacles at $c_j \in \mathbb{R}^n$ represented by the closed balls $\bar{\mathcal{B}}(c_j, r_{c_j})$, $j \in \mathcal{J} = \{1, \dots, J\}$. We further assume that there exist $K_R > 1$ points in \mathcal{W} , denoted by $p_{\pi_k} \in \mathbb{R}^n$, corresponding to certain properties of interest (e.g., gas station, repairing area, etc.), with $\mathcal{K}_R := \{1, \dots, K_R\}$. Since, in practise, these properties are naturally inherited to some neighbourhood of the respective point of interest, we define for each $k \in \mathcal{K}_R$ the *region of interest* π_k , corresponding to p_{π_k} , as the the closed ball $\pi_k := \bar{\mathcal{B}}(p_{\pi_k}, r_{\pi_k})$ with radius $r_{\pi_k} > 0$, for $k \in \mathcal{K}_R$. These properties of interest are expressed as boolean variables via finite, disjoint sets of atomic propositions Ψ_i , $i \in \mathcal{N}$. The properties satisfied at each region π_k are provided by the labelling functions $\mathcal{L}_i : \Pi \rightarrow 2^{\Psi_i}$, which assigns to each region π_k the subset of the atomic propositions Ψ_i that are true in that region. We provide now some tools that relate the continuous multi-agent system with the navigation of the agents to the regions of interest.

Definition 1. Let $\bar{\mathcal{B}}(p_i(t_0), r_i) \subset \pi_{k_0}$, i.e., agent i is in region π_{k_0} , $k_0 \in \mathcal{K}_R$, for some $t_0 \geq 0$. Then, agent executes a transition from π_{k_0} to π_{k_1} , $k_1 \in \mathcal{K}_R \setminus \{k_0\}$, denoted by $\pi_{k_0} \rightarrow_i \pi_{k_1}$, if there exists a finite time instant $t_1 \geq t_0$ such that $\bar{\mathcal{B}}(p_i(t_1), r_i) \subset \pi_{k_1}$ and $\bar{\mathcal{B}}(p_i(t), r_i) \cap (\bigcup_{\ell \in \mathcal{N} \setminus \{i\}} \bar{\mathcal{B}}(p_\ell(t), r_\ell)) = \bar{\mathcal{B}}(p_i(t), r_i) \cap (\bigcup_{j \in \mathcal{J}} \bar{\mathcal{B}}(c_j, r_{c_j})) = \emptyset$, $r_{\mathcal{W}} > \|p_i(t)\| + r_i$, for all $t \in [t_0, t_1]$.

Loosely speaking, agent i can transit between two regions in Π if it can navigate among them while avoiding collisions with other agents and obstacles. We note that, if the agent's task requires it to avoid a certain region of Π , then the latter can be considered as an obstacle.

Our goal here is to control the multi-agent system so that each agent's behaviour obeys a given temporal-logic specification over its atomic propositions Ψ_i . Formally, the behaviour of agent i over its trajectory $p_i(t)$ is the infinite sequence $\mathbf{b}_i(\psi_i) := (p_i(t_{i_1}), \psi_{i_1})(p_i(t_{i_2}), \psi_{i_2}) \dots$, with $\psi_{i_m} \in 2^{\Psi_i}$ and $\bar{\mathcal{B}}(p_i(t_{i_m}), r_i) \subset \pi_{k_m}$, $\psi_{i_m} \in \mathcal{L}_i(\pi_{k_m})$, with $k_m \in \mathcal{K}_R$ and an increasing sequence of positive time instants t_{i_m} , $m \in \mathbb{N}$. The control objectives are given for each agent separately as LTL formulas Φ_i over Ψ_i , $i \in \mathcal{N}$ (see Section 3). Agent i satisfies the LTL formula Φ_i if it produces a behaviour $\mathbf{b}_i(\psi_i)$ that satisfies Φ_i , i.e., if $\psi_i = \psi_{i_1} \psi_{i_2} \dots \models \Phi_i$.

By using the control algorithm of Section 4, one guarantees the transitions $\pi_k \rightarrow_i \pi_{k'}$, for all $k, k' \in \mathcal{K}_R$ and agents $i \in \mathcal{N}$, which gives rise to the following finite transition system that encodes the motion of the agents in the workspace:

$$\mathcal{T}_i := \{\Pi, \pi_i^{\text{init}}, \rightarrow_i, \Psi_i, \mathcal{L}_i\}$$

where we use $\pi_i^{\text{init}} \in \Pi$ to denote the i th agent’s initial region.

After defining the transition systems \mathcal{T}_i , the problem at hand reduces to deriving high-level paths for the agents, as sequences of regions of \mathcal{T}_i to be visited, that satisfy their individual specifications encoded by Φ_i . In order to derive such paths, we follow an algorithm inspired by formal-verification methodologies [79]. The algorithm translates each formula Φ_i to a Büchi Automaton \mathcal{C}_i and computes the product $\tilde{\mathcal{T}}_i := \mathcal{T}_i \times \mathcal{C}_i$. One can compute the accepting runs of $\tilde{\mathcal{T}}_i$ by viewing it as a weighted digraph and using standard graph-search methods. Such accepting runs satisfy Φ_i and are directly projected to a sequence of waypoints to be visited, providing therefore a desired path for agent i . Although the semantics of LTL is defined over infinite sequences of atomic propositions, it can be proven that there always exists a high-level path that takes a form of a finite state sequence followed by an infinite repetition of another finite state sequence. For more details on the followed technique, we refer the reader to the related literature, e.g. [79].

6.2. Multi-Agent-Object Systems

As mentioned in Section 1, numerous applications require that, except for the agents, unactuated objects undergo certain processes. Therefore, this section considers the control-synthesis problem under tasks expressed as temporal-logic specifications for a team of robotic agents *and a set of unactuated objects* [108, 109].

The problem setup follows Section 6.1, i.e. we consider a set of N agents operating in a workspace $\mathcal{W} \subset \mathbb{R}^n$ with a set Π of K_R of spherical regions around points of interest p_{π_k} , $k \in \mathcal{K}_R$ and the respective atomic proposition set Ψ_i and labelling functions \mathcal{L}_i , $i \in \mathcal{N}$. We further consider M_R unactuated objects, characterized by the variables p_{o_j} (e.g., center of mass), for $j \in \mathcal{M}_R := \{1, \dots, M_R\}$. The objects have their own set of properties of interest over Π , as expressed by the disjoint sets Ψ_{o_j} of atomic propositions and the respective labelling functions $\mathcal{L}_{o_j} : \Pi \rightarrow 2^{\Psi_{o_j}}$, $j \in \mathcal{M}_R$. Similarly to the agents, the objects’ volume is modelled by the spheres $\bar{\mathcal{B}}(p_{o_j}, r_{o_j})$, $j \in \mathcal{M}_R$. We note that the agents are equipped with suitable equipment, e.g., robotic manipulators, in order to be able to interact with the objects. For ease of exposition, however, we consider that each agent’s volume is represented by the sphere $\bar{\mathcal{B}}(p_i, r_i)$, where p_i denotes a point on agent i (e.g., its center of mass).

We further introduce some additional tools for the interaction among the agents and the objects. First, we consider that the agents have specific power capabilities, which for simplicity, we associate to positive integers $\zeta_i > 0$, $i \in \mathcal{N}$, via an analogous relation. Hence, we define a function $\Lambda \in \{\text{True}, \text{False}\}$ that outputs whether the agents that potentially grasp an object are able to transport it, based on their power capabilities. For instance, $\Lambda(j, \mathcal{A}) = \text{True}$, implies that the agents with indices in $\mathcal{A} \subseteq \mathcal{N}$ have sufficient power capabilities to (cooperatively) transport object j . Second, we define the boolean functions $\mathcal{AG}_j : \mathbb{R}_{\geq 0} \rightarrow 2^{\mathcal{N}_0}$, with $\mathcal{N}_0 := \mathcal{N} \cup \{0\}$, to denote which agents grasp an object $j \in \mathcal{M}_R$ at a specific time instant; $\mathcal{AG}_j(t_s) = \{0\}$ means that no agents grasp

object j at time t_s . Note also that $i \in \mathcal{AG}_j(t_s) \Leftrightarrow i \notin \mathcal{AG}_{j'}(t_s) = \text{False}$, for all $j' \in \mathcal{M}_R \setminus \{j\}$, i.e., agent i can grasp at most one object at a time. We further denote $\mathcal{AG} := [\mathcal{AG}_1, \dots, \mathcal{AG}_{M_R}]^\top \in (2^{\mathcal{N}_0})^{M_R}$. Finally, we represent the volume of a set of agents $\mathcal{A} \subseteq \mathcal{N}$ grasping an object $j \in \mathcal{M}_R$ by the sphere $\tilde{\mathcal{B}}(p_{o_j}, r_{\mathcal{A},j})$, with $r_{\mathcal{A},j}$ being the bounding radius, which is large enough to cover the volume of the coupled system.

We now define action primitives for the multi-agent-object systems, similarly to Definition 1. In these definitions, we use the term ‘‘entity’’ to refer to single agent, object as well as systems comprised by agents that grasp an object (agents-object systems). The number of these systems depends on the variables \mathcal{AG} . Given a grasping configuration $\mathcal{AG} \in (2^{\mathcal{N}_0})^{M_R}$, consider $\bar{T}_R(\mathcal{AG})$ number of entities, indexed by the set $\mathcal{T}_R(\mathcal{AG}) := \{1, \dots, \bar{T}_R(\mathcal{AG})\}$. Each entity (agent, object, or coupled agents-object system) is characterized by the respective configuration (e.g., p_i, p_o_j , or $p_{o_{j'}}$) and radius (e.g., r_i, r_o_j , or $r_{\mathcal{A},j'}$), respectively, which we denote for simplicity by the generic variables p_i^e, r_i^e , for all $i \in \mathcal{T}_R(\mathcal{AG})$.

Definition 2. (Navigation) Let $\mathcal{AG}(t_0) \in (2^{\mathcal{N}_0})^M$ and $i \in \mathcal{T}(\mathcal{AG}(t_0))$ such that $\tilde{\mathcal{B}}(p_i^e(t_0), r_i^e) \subset \pi_{k_0}$, for some $k_0 \in \mathcal{K}_R$ and $t_0 \geq 0$. Then, entity i executes a transition from π_{k_0} to π_{k_1} , with $k_1 \in \mathcal{K}_R$, if there exists a finite time instant $t_1 \geq t_0$ such that $\tilde{\mathcal{B}}(x_i^e(t_1), r_i^e) \subset \pi_{k_1}$ and $\tilde{\mathcal{B}}(x_i^e(t_1), r_i^e) \cap (\bigcup_{\ell \in \mathcal{N} \setminus \{i\}} \tilde{\mathcal{B}}(p_\ell^e(t), r_\ell^e)) = \bigcap (\bigcup_{j \in \mathcal{J}} \tilde{\mathcal{B}}(c_j, r_{c_j})) = \emptyset$, $r_{\mathcal{W}} > \|p_i^e(t)\| + r_i^e$, for all $t \in [t_0, t_1]$.

Definition 2 corresponds to the action primitives of agent navigation and (cooperative) object transportation, depending on what entity i is, from π_{k_0} to π_{k_1} . When the transition corresponds to a single-agent navigation for agent $i \in \mathcal{N}$, we denote it by $\pi_{k_0} \rightarrow_i \pi_{k_1}$; when it corresponds to a transportation of object $j \in \mathcal{M}_R$ by a subset of agents $\mathcal{A} \subset \mathcal{N}$, we denote it by $\pi_{k_0} \xrightarrow{\mathcal{A},j} \pi_{k_1}$. We further need the definition of *grasping* an object.

Definition 3. (Grasping) Let $\mathcal{AG}(t_0) \in (2^{\mathcal{N}_0})^M$ and $i \in \mathcal{N}$, $j \in \mathcal{M}_R$ such that $i \notin \mathcal{AG}_j(t_0)$. Then, agent i grasps object j , denoted by $i \xrightarrow{g} j$, if there exists a finite time instant $t_1 \geq t_0$ such that $i \in \mathcal{AG}_j(t_1)$ and $r_{\mathcal{W}} > \|p_i^e(t)\| + r_i^e$, $\tilde{\mathcal{B}}(p_i^e(t_1), r_i^e) \cap (\bigcup_{\ell \in \mathcal{N} \setminus \{i\}} \tilde{\mathcal{B}}(p_\ell^e(t), r_\ell^e)) = \bigcap (\bigcup_{j \in \mathcal{J}} \tilde{\mathcal{B}}(c_j, r_{c_j})) = \emptyset$, for all $t \in [t_0, t_1]$.

Similarly, we can define the *releasing* action $i \xrightarrow{r} j$ for an agent $i \in \mathcal{N}$ and object $j \in \mathcal{M}_R$.

Next, we assume the existence of a procedure \mathcal{P}_s that outputs whether or not a set of non-intersecting spheres fits in a larger sphere as well as possible positions of the spheres in the case they fit. More specifically, given a region of interest π_k and a number $\tilde{N} \in \mathbb{N}$ of sphere radii (of robots, objects, or object-robots systems) the procedure can be seen as a function $\mathcal{P}_s := [\mathcal{P}_{s,0}, \mathcal{P}_{s,1}^\top]^\top$, where $\mathcal{P}_{s,0} : \mathbb{R}_{\geq 0}^{\tilde{N}+1} \rightarrow \{\text{True}, \text{False}\}$ outputs whether the spheres fit in the region π_k whereas $\mathcal{P}_{s,1}$ provides possible configurations of the robots and the objects or 0 in case the spheres do not fit. For instance, $\mathcal{P}_{s,0}(r_{\pi_2}, r_1, r_3, r_{o_1}, r_{o_5})$ determines whether the robots 1, 3 and the objects 1, 5 fit in region π_2 , without colliding

with each other; $(p_1^e, p_2^e, p_3^e, p_4^e) = (p_1, p_3, p_{o_1}, p_{o_5}) = P_{s,1}(r_{\pi_2}, r_1^e, r_2^e, r_3^e, r_4^e) = P_{s,1}(r_{\pi_2}, r_1, r_3, r_{o_1}, r_{o_5})$ provides a set of configurations such that the pairwise intersections among $\mathcal{B}_i(p_i^e, r_i^e)$, $i \in \{1, \dots, 4\}$, are empty. The problem of finding an algorithm \mathcal{P}_s is a special case of the sphere packing problem [110]. Note, however, that we are not interested in finding the maximum number of spheres that can be packed in a larger sphere but, rather, in the simpler problem of determining whether a set of spheres can be packed in a larger sphere.

Our goal is to control the multi-agent-object system defined above such that the agent and the objects obey a given specification over their atomic propositions $\Psi_i, \Psi_{o_j}, \forall i \in \mathcal{N}, j \in \mathcal{M}$. Similar to the agents' behaviour, defined in Section 6.1, we define the behaviour of object j , over its trajectory p_{o_j} , as the infinite sequence $\mathbf{b}_{o_j}(\psi_{o_j}) := (p_{o_j}(t_{o_j,1}), \psi_{o_j,1})(p_{o_j}(t_{o_j,2}), \psi_{o_j,2}) \dots$, with $\psi_{o_j,m} \in 2^{\Psi_{o_j}}$ and $\mathcal{B}(p_{o_j}(t_{o_j,m})) \subset \pi_{k_m}$, $\psi_{o_j,m} \in \mathcal{L}_{o_j}(\pi_{k_m})$, with $k_m \in \mathcal{K}_R$, and an increasing sequence of time instants $t_{o_j,m}$, $m \in \mathbb{N}$.

As in Section 6.1, the control objectives are given as LTL formulas Φ_i, Φ_{o_j} over Ψ_i, Ψ_{o_j} , respectively, for all $i \in \mathcal{N}, j \in \mathcal{M}_R$. The control-synthesis problem is the derivation of actions for the agents such that each agent $i \in \mathcal{N}$ and each object $j \in \mathcal{M}_R$ satisfy their LTL formulas Φ_i and Φ_{o_j} , respectively, i.e., actions that produce behaviours $\mathbf{b}_i(\psi_i)$ and $\mathbf{b}_{o_j}(\psi_{o_j})$ such that $\psi_i = \psi_{i_1} \psi_{i_2} \dots \models \Phi_i$ and $\psi_{o_j} = \psi_{o_j,1} \psi_{o_j,2} \dots \models \Phi_{o_j}$, for all $i \in \mathcal{N}, j \in \mathcal{M}_R$.

As in Section 6.1, we abstract the continuous multi-agent-object system into a discrete transition system. For the execution of the navigation primitives (Def. 2), one can use the multi-agent navigation and cooperative-manipulation control algorithms of Sections 4 and 5, respectively. For the execution of the grasping/releasing primitives (Def. 3), one can employ existing methodologies that can derive the respective control actions (e.g., [111]). Therefore, we can abstract the behaviour of the multi-agent-object system using the finite transition system

$$\mathcal{TO} = (\Pi_s, \Pi_s^{\text{init}}, \rightarrow_s, \mathcal{AG}, \Psi, \mathcal{L}, \Lambda, P_s, \chi)$$

where

1. $\Pi_s \subset \bar{\Pi} \times \bar{\Pi}_o \times (2^{\mathcal{N}_0})^M$ is the set of states; $\bar{\Pi} := \Pi_1 \times \dots \times \Pi_N$ and $\bar{\Pi}_o := \Pi_{o_1} \times \dots \times \Pi_{o_{M_R}}$ are the set of states-regions that the agents and the objects can be at, with $\Pi_i = \Pi_{o_j} = \Pi$, for all $i \in \mathcal{N}, j \in \mathcal{M}_R$;

By defining $\bar{\pi} := (\pi_{k_1}, \dots, \pi_{k_N})$, $\bar{\pi}_o := (\pi_{k_{o,1}}, \dots, \pi_{k_{o,M_R}})$, then the coupled state $\pi_s := (\bar{\pi}, \bar{\pi}_o, \mathcal{AG})$ belongs to Π_s , i.e., $(\bar{\pi}, \bar{\pi}_o, \mathcal{AG}) \in \Pi_s$ if

- (a) $\mathcal{P}_{s,0}(r_{\pi_k}, [r_i]_{i \in \{i \in \mathcal{N}: k_i=k\}}, [r_{o_j}]_{j \in \{j \in \mathcal{M}_R: k_{o,j}=k\}}) = \top$, i.e., the respective agents and objects fit in the region, for all $k \in \mathcal{K}_R$,
- (b) $k_i = k_{o_j}$ for all $i \in \mathcal{N}, j \in \mathcal{M}_R$ such that $i \in \mathcal{AG}_j$, i.e., an agent must be in the same region with the object it grasps,

2. $\Pi_s^{\text{init}} \in \Pi_s$ is the initial state at $t = 0$,

3. $\rightarrow_s \subset \Pi_s \times \Pi_s$ is a transition relation defined as follows: given the states $\pi_s, \tilde{\pi}_s \in \Pi$, with

$$\begin{aligned}\pi_s &:= (\bar{\pi}, \bar{\pi}_o, \mathcal{AG}) \\ &:= (\pi_{k_1}, \dots, \pi_{k_N}, \pi_{k_{o,1}}, \dots, \pi_{k_{o,M_R}}, \mathcal{AG}_1, \dots, \mathcal{AG}_M), \\ \tilde{\pi}_s &:= (\tilde{\pi}, \tilde{\pi}_o, \widetilde{\mathcal{AG}}) \\ &:= (\pi_{\tilde{k}_1}, \dots, \pi_{\tilde{k}_N}, \pi_{\tilde{k}_{o,1}}, \dots, \pi_{\tilde{k}_{o,1}}, \widetilde{\mathcal{AG}}_1, \dots, \widetilde{\mathcal{AG}}_M),\end{aligned}\quad (44)$$

a transition $\pi_s \rightarrow_s \pi'_s$ occurs if all the following hold:

- (a) $\nexists i \in \mathcal{N}, j \in \mathcal{M}_R$ such that $i \in \mathcal{AG}_j, i \notin \widetilde{\mathcal{AG}}_j$ (or $i \notin \mathcal{AG}_j, i \in \widetilde{\mathcal{AG}}_j$), and $k_i \neq \tilde{k}_i$, i.e., there are no simultaneous grasp/release and navigation actions,
 - (b) $\nexists i \in \mathcal{N}, j \in \mathcal{M}_R$ such that $i \in \mathcal{AG}_j, i \notin \widetilde{\mathcal{AG}}_j$ (or $i \notin \mathcal{AG}_j, i \in \widetilde{\mathcal{AG}}_j$), and $k_i = k_{o,j} \neq \tilde{k}_i = \tilde{k}_{o,j}$, i.e., there are no simultaneous grasp/release and transportation actions,
 - (c) $\nexists i \in \mathcal{N}, j, j' \in \mathcal{M}_R$, with $j \neq j'$, such that $i \in \mathcal{AG}_j$ and $i \in \widetilde{\mathcal{AG}}_{j'}$, i.e., there are no simultaneous grasp and release actions,
 - (d) $\nexists j \in \mathcal{M}_R$ such that $k_{o,j} \neq \tilde{k}_{o,j}$ and $i \notin \mathcal{AG}_j, \forall i \in \mathcal{N}$ (or $i \notin \widetilde{\mathcal{AG}}_j, \forall i \in \mathcal{N}$), i.e., there is no transportation of a non-grasped object,
 - (e) $\nexists j \in \mathcal{M}_R, \mathcal{T} \subseteq \mathcal{N}$ such that $k_{o,j} \neq \tilde{k}_{o,j}$ and $\Lambda(j, \zeta_{\mathcal{T}}) = \perp$, with $(i \in \mathcal{AG}_j, i \in \widetilde{\mathcal{AG}}_j) \Leftrightarrow i \in \mathcal{T}$, i.e., the agents grasping an object are powerful enough to transfer it,
4. $\Psi := \bar{\Psi} \cup \bar{\Psi}^o$ with $\bar{\Psi} = \bigcup_{i \in \mathcal{N}} \Psi_i$ and $\bar{\Psi}^o = \bigcup_{j \in \mathcal{M}_R} \Psi_{o_j}$, are the atomic propositions of the agents and objects, respectively.
 5. $\mathcal{L} : \Pi_s \rightarrow 2^\Psi$ is a labeling function defined as follows: Given a state π_s as in (44) and $\psi_s := \left(\bigcup_{i \in \mathcal{N}} \psi_i \right) \cup \left(\bigcup_{j \in \mathcal{M}_R} \psi_{o_j} \right)$ with $\psi_i \in 2^{\Psi_i}, \psi_{o_j} \in 2^{\Psi_{o_j}}$, then $\psi_s \in \mathcal{L}(\pi_s)$ if $\psi_i \in \mathcal{L}_i(\pi_{k_i})$ and $\psi_{o_j} \in \mathcal{L}_{o_j}(\pi_{k_{o,j}})$, for all $i \in \mathcal{N}, j \in \mathcal{M}_R$.
 6. Λ and P_s as defined previously.
 7. $\chi : (\rightarrow_s) \rightarrow \mathbb{R}_{\geq 0}$ is a function that assigns a cost to each transition $\pi_s \rightarrow_s \tilde{\pi}_s$. This cost might be related to the distance of the robots' regions in π_s to the ones in $\tilde{\pi}_s$, combined with the cost efficiency of the robots involved in transport tasks (according to $\zeta_i, i \in \mathcal{N}$).

Next, we proceed as in Section 6.1; we form the global LTL formula $\Phi := (\wedge_{i \in \mathcal{N}} \Phi_i) \wedge (\wedge_{j \in \mathcal{M}_R} \Phi_{o_j})$ over the set Ψ , and we translate it to a Büchi Automaton \mathcal{CO} . Next, we compute the product $\widetilde{\mathcal{TO}} := \mathcal{TO} \times \mathcal{CO}$ and use graph-search-based methodologies to find an accepting path that satisfies Φ . This path is projected to the original transition system \mathcal{TO} and provides a sequence of

action primitives for the agents to execute (navigation, grasping, releasing, or cooperative transportation). Note that, unlike Section 6.1, where the tasks were specified independently for each agent, the object-related LTL tasks require the coupling and collaboration of more than one agents. Such a coupling results in a centralized path-planning procedure, i.e., a central computer unit computes $\widehat{\mathcal{TO}}$ and the accepting path for all the agents. We note that such a centralized procedure might yield significant computational complexity, in terms of computer memory and running time. One can, however, employ procedures more sophisticated than simple graph search to compute an accepting path of $\widehat{\mathcal{TO}}$ in order to reduce the computational complexity (such as STyLuS* [109]).

6.3. Timed Abstractions for Cooperative Manipulation

In the previous sections, we considered the motion planning and control of multi-agent and multi-agent-object systems subject to tasks expressed as LTL constraints. In this section, we consider the problem of controlling a cooperatively manipulated object such that it satisfies a task expressed as metric interval temporal logic (MITL) constraints [112, 113]; such constraints allow the incorporation of *time* constraints (deadlines), enriching thus the range of available tasks (see Section 3).

We consider here N robotic agents rigidly grasping an object in a bounded workspace $\mathcal{W} \subset \mathbb{R}^3$, as in Section 5. The agent, object, and coupled dynamics are given by (11), (12), and (17), respectively. In contrast to section 6.1 and 6.2, where the LTL tasks were specified over specific “regions of interest” π_k , we consider here a different discretization of the workspace, described next.

We denote by \mathcal{S}_q the set that consists of all points $p_s \in \mathcal{W}$ that physically belong to the coupled system, i.e., they consist part of either the volume of the agents or the volume of the object. Note that these points depend on the agents’ configuration variables q . We further define the constant $\hat{L} \geq \sup_{\substack{q \in \mathbb{R}^n \\ p_s \in \mathcal{S}_q}} \|p_s - p_o(q)\|$, where we express p_o as a function of q . Note that, although the explicit computation of \mathcal{S}_q may not be possible, \hat{L} is an upper bound of the maximum distance between the object center of mass and a point in the coupled system’s volume over all possible configurations q , and thus, it can be measured. For instance, Fig. 7 shows \hat{L} for a system comprised by two robotic agents and an object. It is straightforward to conclude that $\mathcal{S}_q \subset \bar{\mathcal{B}}(p_o(q), \hat{L}), \forall q \in \mathbb{R}^n$. Next, we partition the workspace \mathcal{W} into $R_{\mathcal{W}}$ equally sized rectangular regions $\Pi = \{\pi_1, \dots, \pi_{R_{\mathcal{W}}}\}$, whose geometric centers are denoted by $p_{\pi_j}^c \in \mathcal{W}, j \in \{1, \dots, R_{\mathcal{W}}\}$. The length of the region sides is set to $D = 2\hat{L} + 2l_0$, where l_0 is an arbitrary positive constant. Hence, each region π_j can be formally defined as follows:

$$\pi_j := \{p \in \mathcal{W} \text{ s.t. } (p)_k \in [(p_{\pi_j}^c)_k - \hat{L} - l_0, (p_{\pi_j}^c)_k + \hat{L} + l_0], \forall k \in \{x, y, z\}\},$$

with $\|p_{\pi_{j+1}}^c - p_{\pi_j}^c\| = (2\hat{L} + 2l_0), \forall j \in \{1, \dots, R - 1\}$, and $(p_{\pi_j}^c)_z := \hat{L} + l_0, \forall j \in \{1, \dots, R_{\mathcal{W}}\}$; $(\cdot)_k, k \in \{x, y, z\}$, denotes the k -th coordinate. An illustration of

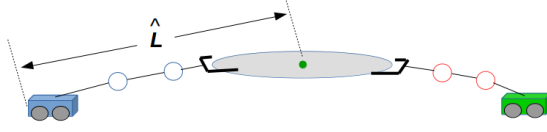


Figure 7: An example of a cooperative manipulation system in the configuration that produces \hat{L} .

the aforementioned partition is depicted in Fig. 8. We also define the neighborhood \mathcal{D} of region π_j as the set of its adjacent regions, i.e., $\mathcal{D}(\pi_j) := \{\pi_{j'} \in \Pi \text{ s.t. } \|p_{\pi_j}^c - p_{\pi_{j'}}^c\| = (2\hat{L} + 2l_0)\}$, which is symmetric, i.e., $\pi_{j'} \in \mathcal{D}(\pi_j) \Leftrightarrow \pi_j \in \mathcal{D}(\pi_{j'})$.

We write $\mathcal{A}(q) \in \pi_j$ to denote that the coupled object-agents system is in region π_j , equivalent to $\mathcal{S}_q \subset \pi_j$ and $\|p_o(q) - p_{\pi_j}^c\| < l_0$. We now define the timed transition of the coupled system between two regions $\pi_j, \pi_{j'}$:

Definition 4. Assume that $\mathcal{A}(q(t_0)) \in \pi_j$ for some $j \in \{1, \dots, R_{\mathcal{W}}\}$, $t_0 \geq 0$. Then, there exists a transition for the coupled object-agents system from π_j to $\pi_{j'}, j' \in \{1, \dots, R_{\mathcal{W}}\}$, with time duration $\delta t_{j,j'} \geq 0$, denoted as $\pi_j \xrightarrow{\mathcal{T}} \pi_{j'}$, if

1. $\mathcal{A}(q(t_0 + \delta t_{j,j'})) \in \pi_{j'}$,
2. $\mathcal{S}_{q(t)} \subset \pi_i \cup \pi_j, \forall t \in [t_0, t_0 + \delta t_{j,j'}]$.

Note that the entire system object-agents must remain in $\pi_j, \pi_{j'}$ during the transition and therefore the requirement $\pi_{j'} \in \mathcal{D}(\pi_j)$ is implicit in Definition 4.

Given the aforementioned workspace partition, we introduce a set of atomic propositions Ψ_o for the object, such as ‘‘Obstacle region’’, ‘‘Goal region’’, with the associated labeling function $\mathcal{L} : \Pi \rightarrow 2^{\Psi_o}$.

Similarly to Sections 6.1, 6.2, we define the *timed behaviour* of the coupled object-agents system over a trajectory $q(t)$ as the infinite *timed* sequence $\mathbf{b}_t := (q(t_1), \psi_1, t_1)(q(t_2), \psi_2, t_2) \dots$, where t_m is an increasing sequence of positive time instants, $m \in \mathbb{N}$; $q(t_m)$ and ψ_m satisfy $\mathcal{A}(q(t_m)) \in \pi_{j_m}$, $\psi_m \in 2^{\Psi_o}$, and $\psi_m \in \mathcal{L}(\pi_{j_m})$, with $j_m \in \{1, \dots, R_{\mathcal{W}}\}$, for all $m \in \mathbb{N}$. The problem consists of computing the agents’ control actions such that the object satisfies a given MITL formula Φ , i.e., control actions that produce a timed behaviour \mathbf{b}_t such that $(\psi_1, t_1)(\psi_2, t_2) \dots \models \Phi$.

The first ingredient for the solution of the aforementioned problem is the design of a control protocol such that a transition relation between two adjacent regions according to Definition 4 is established. We first associate to the transition a smooth and bounded trajectory with bounded time derivatives, defined by the line segment that connects $p_{\pi_j}^c$ and $p_{\pi_{j'}}^c$, i.e. define $p_{j,j'} : [t_0, \infty) \rightarrow \mathbb{R}^3$, such that $p_{j,j'}(t_0) = p_{\pi_j}^c, p_{j,j'}(t) = p_{\pi_{j'}}^c, \forall t \geq t_0 + \delta t_{j,j'}$ and $\mathcal{B}(p_{j,j'}(t), \hat{L} + l_0) \subset \pi_j \cup \pi_{j'}, \forall t \geq t_0$. An example of $p_{j,j'}$ is

$$p_{j,j'}(t) = \begin{cases} \frac{p_{\pi_{j'}}^c - p_{\pi_j}^c}{\delta t_{j,j'}} t + \frac{p_{\pi_j}^c (\delta t_{j,j'} - 1) - p_{\pi_{j'}}^c}{\delta t_{j,j'}} t_0, & t \in [t_0, t_0 + \delta t_{j,j'}) \\ p_{\pi_{j'}}^c, & t \in [t_0 + \delta t_{j,j'}, \infty) \end{cases} \quad (45)$$

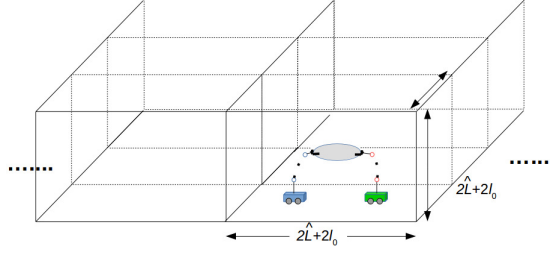


Figure 8: The workspace partition according to the bounding box of the coupled system.

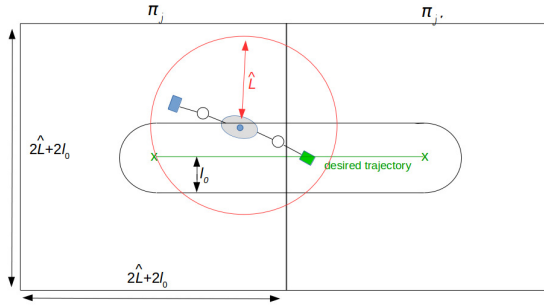


Figure 9: Top view of a transition between two adjacent regions π_j and $\pi_{j'}$. Since $p_O \in \bar{\mathcal{B}}(p_{j,j'}(t), l_0)$, we conclude that $\mathcal{S}_q \subset \bar{\mathcal{B}}(p_O, \hat{L}) \subset \bar{\mathcal{B}}(p_{j,j'}(t), l_0 + \hat{L}) \subset \pi_j \cup \pi_{j'}$.

Intuitively, if we guarantee that the object's center of mass stays l_0 -close to $p_{j,j'}$, i.e., $\|p_O(t) - p_{j,j'}(t)\| < l_0$, for all $t \geq t_0$, then $\|p_O(t_0 + \delta t_{j,j'}) - p_{\pi_{j'}}^c\| < l_0$, we obtain $\mathcal{S}_{q(t)} \subset \bar{\mathcal{B}}(p_O(t), \hat{L}) \subset \bar{\mathcal{B}}(p_{j,j'}(t), \hat{L} + l_0) \subset \pi_j \cup \pi_{j'}$, for all $t \geq t_0$ (and therefore $t \in [t_0, t_0 + \delta t_{j,j'}]$), and thus the requirements of Definition 4 for the transition relation are met. Fig. 9 illustrates the aforementioned reasoning.

Along with $p_{j,j'}$, we consider that the object has to comply with certain specifications associated with its orientation. Therefore, we also define a smooth and bounded orientation trajectory $\eta_{j,j'} := [\varphi_{j,j'}, \theta_{j,j'}, \psi_{j,j'}]^\top : [t_0, \infty) \rightarrow \mathbb{T}$ with bounded time derivative, that has to be tracked by the object's center of mass. We form, therefore, the desired pose trajectory $x_{j,j'} : [t_0, \infty) \rightarrow \mathbb{M}$, with $x_{j,j'}(t) := [p_{j,j'}(t)^\top, \eta_{j,j'}(t)^\top]^\top$. In case of multiple consecutive transitions $\dots \pi_h \xrightarrow{T} \pi_j \xrightarrow{T} \pi_{j'} \xrightarrow{T} \pi_{h'} \dots$ over the intervals $\dots, \delta t_{h,j}, \delta t_{j,j'}, \delta t_{j',h'}, \dots$, the desired orientation trajectories $\dots, \eta_{h,j}(t), \eta_{j,j'}(t), \eta_{j',h'}(t), \dots$ must be continuous at the transition points, i.e., $\eta_{h,j}(t_0) = \eta_{j,j'}(t_0)$ and $\eta_{j,j'}(t_0 + \delta t_{j,j'}) = \eta_{j',h'}(t_0 + \delta t_{j,j'})$.

The considered problem is equivalent to a problem of trajectory tracking within certain bounds. A suitable methodology for the control design in hand is that of prescribed performance control (PPC), which was used for the cooperative manipulation problem in Section 5.1. More specifically, the distributed control law (29) guarantees the tracking of the desired trajectory $p_{j,j'}(t)$ by the object with prescribed performance, i.e., $-\rho_{s_k}(t) < p_O(t) - p_{j,j'}(t) < \rho_{s_k}(t)$,

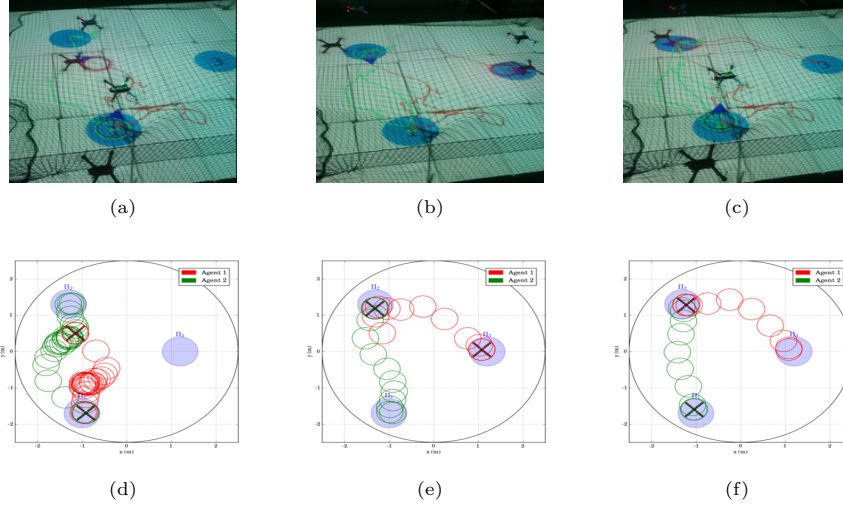


Figure 10: Execution of the paths $(\pi_1\pi_2\pi_3\pi_2)^1$ and $(\pi_2\pi_1)^2$ by agents 1 and 2, respectively for the second experimental scenario. (a), (d): $\pi_1 \rightarrow_1 \pi_2, \pi_2 \rightarrow_2 \pi_1$, (b), (e): $\pi_2 \rightarrow_1 \pi_3, \pi_3 \rightarrow_2 \pi_1$, (c), (f): $\pi_3 \rightarrow_1 \pi_2, \pi_2 \rightarrow_2 \pi_1$.

for user-defined performance functions $\rho_{s,k}(t)$ and all $k \in \{x, y, z, \varphi, \theta, \psi\}$. Now, by selecting $\rho_{s,k}(t_0) = l_0 \geq \rho_{s,k}(t)$, for $t \geq t_0$ and $k \in \{x, y, z\}$, the PPC algorithm guarantees that $p_o(q(t)) \in \mathcal{B}(p_{j,j'}(t), l_0), \forall t \geq t_0$ and, consequently, $p_o(q(t_0 + \delta t_{j,j'})) \in \mathcal{B}(p_{\pi_{j'}}^c, l_0)$, since $p_{j,j'}(t_0 + \delta t_{j,j'}) = p_{\pi_{j'}}^c$. Moreover, since $p_o(q(t)) \in \mathcal{B}(p_{j,j'}(t), l_0)$, we deduce that $\mathcal{B}(p_o(q(t)), \hat{L}) \subset \mathcal{B}(p_{j,j'}(t), l_0 + \hat{L})$ and $\mathcal{S}_{q(t)} \subset \pi_j \cup \pi_{j'}, \forall t \in [t_0, t_0 + \delta t_{j,j'}] \subset [t_0, \infty)$, and therefore a transition relation with time duration $\delta t_{j,j'}$ is successfully established. More details can be found in [112, 113]. Therefore, we can abstract the motion of the coupled agents-object system as a *timed* transition system

$$\mathcal{T} = \{\Pi, \Pi_0, \xrightarrow{\mathcal{T}}, \Psi, \mathcal{L}, \gamma_{\mathcal{T}}\},$$

where we further introduce the map $\gamma_{\mathcal{T}} : (\xrightarrow{\mathcal{T}}) \rightarrow \mathbb{R}_{\geq 0}$ that assigns to each transition its time duration, i.e., $\gamma_{\mathcal{T}}(\pi_j \xrightarrow{\mathcal{T}} \pi_{j'}) = \delta t_{j,j'}$.

The next step of the algorithm follows a similar procedure as that of Sections 6.1 and 6.2. Namely, we translate the MITL formula Φ into a *Timed Büchi Automaton* \mathcal{C}_{Φ}^t [114], we compute the product $\tilde{\mathcal{T}} = \mathcal{T} \otimes \mathcal{C}_{\Phi}^t$ [79], and find its accepting runs. The projection of these accepting runs onto \mathcal{T} provides a infinite sequence of regions in Π to be visited at specific time instants; such a sequence satisfies Φ . More details on the technique can be found in [79, 115, 114].

6.4. Simulation and Experimental Results

We first demonstrate the algorithm of Section 6.1 through experiments with *IRIS+* quadrotors from 3D Robotics. We consider that the quadrotors have

a sensing range of $\varsigma_i = 0.65\text{m}$, and bounding spheres with radius $r_i = 0.3\text{m}$, $i \in \{1, 2\}$. We consider a 2-dimensional scenario in a workspace \mathcal{W} with radius of 2.5m and 3 regions of interest $\Pi = \{\pi_1, \dots, \pi_3\}$, with $r_{\pi_k} = 0.4$, $k \in \{1, \dots, 3\}$ and $p_{\pi_1} = [-1, -1.7]^\top\text{m}$, $p_{\pi_2} = [-1.3, 1.3]^\top\text{m}$ and $p_{\pi_3} = [1.2, 0]^\top\text{m}$. The agents are initially in π_1 and π_2 , respectively. We define the atomic propositions $\Psi_1 = \Psi_2 = \{\text{“res}_a\text{”}, \text{“res}_b\text{”}, \text{“base”}\}$, corresponding to a base and several resources in the workspace, with $\mathcal{L}_1(\pi_1) = \mathcal{L}_2(\pi_1) = \{\text{“res}_a\text{”}\}$, $\mathcal{L}_1(\pi_2) = \mathcal{L}_2(\pi_2) = \{\text{“base”}\}$, $\mathcal{L}_1(\pi_3) = \mathcal{L}_2(\pi_3) = \{\text{“res}_b\text{”}\}$. The agents have to transfer the resources to the “base” in π_2 ; both agents are responsible for “res_a” but only agent 1 should access “res_b”. The specifications are translated to the formulas $\Phi_1 = \square(\diamond(\text{“res}_a\text{”} \circ \text{“base”}) \wedge \diamond(\text{“res}_b\text{”} \circ \text{“base”}))$, $\Phi_2 = \square\neg\text{“res}_b\text{”} \wedge \square\diamond(\text{“res}_a\text{”} \circ \text{“base”})$ and the derived paths are $(\pi_1\pi_2\pi_3\pi_2)^\omega$ for agent 1, and $(\pi_1\pi_2)^\omega$ for agent 2. For the continuous control, we use the algorithm of Section 4.2. The execution of the paths $(\pi_1\pi_2\pi_3\pi_2)^1$ and $(\pi_2\pi_1)^2$ by agents 1 and 2, respectively, are depicted in Fig. 10, showing the successful execution and satisfaction of the paths and formulas. A video demonstrating the experiment can be found on <https://youtu.be/dO77ZYEFHIE>.

Next, we demonstrate the multi-agent-object algorithm of Section 6.2 in an obstacle-cluttered office environment. We choose the atomic propositions for the agents and objects as $\Psi_i = \{\text{“}i\text{-}\pi_1\text{”}, \dots, \text{“}i\text{-}\pi_K\text{”}\}$ and $\Psi_{o_j} = \{\text{“}O_j\text{-}\pi_1\text{”}, \dots, \text{“}O_j\text{-}\pi_K\text{”}\}$, respectively, for $i \in \mathcal{N}$, $j \in \mathcal{M}_R$, indicating their presence in the regions of interest. For the constructed transition systems, we set the cost χ as the Euclidean distance among the RoI contained in the nodes of the transitions. We consider $N = 3$ robots, $K = 4$ regions and $M_R = 2$ objects. The regions of interest are circular centered at $(88, -280)\text{m}$, $(100, -160)\text{m}$, $(200, -130)\text{m}$, and $(250, -285)\text{m}$, with radius equal to 4m. The agents’ and object’ mass is taken as 1kg and 0.25kg, respectively, while their spherical volume’s radii as 0.75m and 0.2m, respectively. The power capabilities of the agents are 2, 3, 4, respectively, and the power required for each object is 5, 6, respectively. Initially, the agents are located in regions $\pi_{init(1)} = \pi_1$, $\pi_{init(2)} = \pi_3$, and $\pi_{init(3)} = \pi_4$, respectively, whereas the objects are located in regions $\pi_{init_o(1)} = \pi_2$, $\pi_{init_o(2)} = \pi_1$, respectively. For the continuous control design, we choose the algorithm of Section 4.2, both for multi-agent navigation and object transportation; for the latter, we treat the coupled agents-object system as a sphere and we distribute the control law (9) equally to the agents grasping the object. To deal with the non-spherical obstacles, we use the workspace transformation from [116] that converts the obstacles to points.

The resulting transition system consists of 3, 112 reachable states and 154, 960 transitions and it was created within 1.71 minutes. We set the following formula for the multi-robot-object system:

$$\phi = \square\diamond\bar{\psi}_1 \wedge \square\diamond\bar{\psi}_2 \wedge \square\diamond\bar{\psi}_2 \wedge \diamond\bar{\psi}_4 \wedge (\neg\bar{\psi}_4\mathcal{U}\bar{\psi}_5) \quad (46)$$

with

- $\bar{\psi}_1 = \{\text{“}1\text{-}\pi_1\text{”}, \text{“}2\text{-}\pi_1\text{”}, \text{“}3\text{-}\pi_1\text{”}, \text{“}O_1\text{-}\pi_1\text{”}, \text{“}O_2\text{-}\pi_4\text{”}\}$
- $\bar{\psi}_2 = \{\text{“}1\text{-}\pi_1\text{”}, \text{“}2\text{-}\pi_3\text{”}, \text{“}3\text{-}\pi_1\text{”}, \text{“}O_1\text{-}\pi_1\text{”}, \text{“}O_2\text{-}\pi_3\text{”}\}$

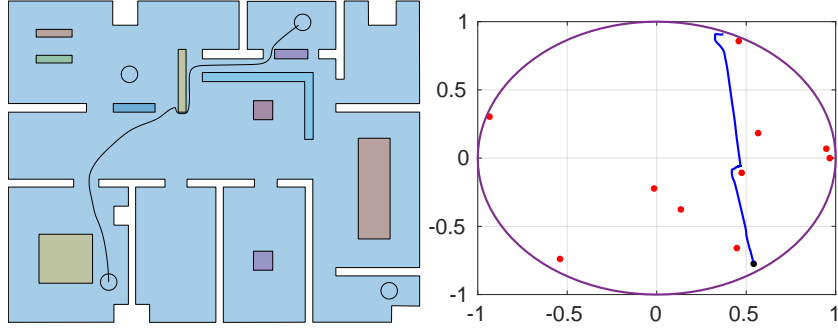


Figure 11: Navigation of agent 1 from π_1 to π_3 in the original (left) and the transformed point-world (right) environment. In the original environment, the agent’s circular volume has been transferred to the obstacles and workspace boundary.

- $\bar{\psi}_3 = \{“1-\pi_3”, “2-\pi_3”, “3-\pi_3”, “O_1-\pi_4”, “O_2-\pi_3”\}$
- $\bar{\psi}_4 = \{“1-\pi_2”, “2-\pi_4”, “3-\pi_1”, “O_1-\pi_4”, “O_2-\pi_2”\}$
- $\bar{\psi}_5 = \{“1-\pi_4”, “2-\pi_4”, “3-\pi_2”, “O_1-\pi_3”, “O_2-\pi_4”\}$

In words, the mission specification in (46) requires the robots and objects to satisfy $\bar{\psi}_1$, $\bar{\psi}_2$, and $\bar{\psi}_3$ infinitely often, satisfy $\bar{\psi}_4$ eventually, and satisfy $\bar{\psi}_5$ before satisfying $\bar{\psi}_4$. The LTL formula in (46) corresponds to a non-deterministic Büchi automaton with 6 states - among which one is a final state - and 18 transitions. For the multi-agent-object planning, we use the STyLuS* algorithm [117], which found the first feasible prefix and suffix path within 1.23 minutes and 0.64 minutes, respectively. The action path of the agents is depicted in Tables 1-2, starting from $\pi_{s,1}$ and satisfying $\{“1-\pi_1”, “2-\pi_2”, “3-\pi_4”, “O_1-\pi_2”, “O_2-\pi_1”\}$. We further illustrate the continuous control design. In particular, we consider the navigation of agent 1 from π_1 to π_3 . The results are depicted in Figs. 11. The left part of Fig. 11 shows the trajectory of agent 1 in the environment, where the obstacles and boundary have absorbed the the spherical volume of the agent; the right part of Fig. 11 shows the trajectory of agent 1 in the transformed point world, where the obstacles are represented by points.

Finally, we demonstrate the algorithm of Section 6.3 in the CoppeliaSim simulator [105]. We consider a rectangular rigid body of dimensions $0.025 \times 0.2 \times 0.025$ m³ representing the object that is rigidly grasped by two agents. Each agent consists of a quadrotor base $\{B_i\}$ and a 2-DOF robotic arm. We consider that the quadrotor is fully actuated and that there exists an embedded algorithm that translates the generalized force of the quadrotor base to the actual motor inputs. The initial conditions of the system are taken as $p_o(0) = [0, 0, 1.5]^\top$ m, $\eta_o(0) = [0, 0, 0]^\top$ rad. The workspace is partitioned into $R_{\mathcal{W}} = 16$ regions, with $\bar{L} = 0.75$ m and $l_0 = 0.5$ m. Initially, the coupled system is located in π_1 . We further define the atomic propositions $\Psi = \{“green_1”, “green_2”, “red”, “obs”\}$, representing goal (“green₁”, “green₂”) and obstacle (“obs”) regions with $\mathcal{L}(\pi_5) = \{“green_1”\}$, $\mathcal{L}(\pi_{14}) = \{“green_2”\}$, $\mathcal{L}(\pi_6) = \mathcal{L}(\pi_{10}) = \{“obs”\}$ and $\mathcal{L}(\pi_j) = \emptyset$, for

Table 1: The agent actions for the discrete path of the first case study (Part 1)

$\pi_{s,\ell}$	Actions	$\pi_{s,\ell}$	Actions
$\pi_{s,1}$	—	$\pi_{s,17}$	$2 \xrightarrow{g} 1, 3 \xrightarrow{g} 1$
$\pi_{s,2}$	$1 \xrightarrow{g} 2, \pi_4 \rightarrow_3 \pi_1$	$\pi_{s,18}$	$\pi_3 \xrightarrow{T}_{\{2,3\},1} \pi_4$
$\pi_{s,3}$	$\pi_3 \rightarrow_2 \pi_2, 3 \xrightarrow{g} 2$	$\pi_{s,19}$	$2 \xrightarrow{r} 1, 3 \xrightarrow{r} 1$
$\pi_{s,4}$	$\pi_1 \xrightarrow{T}_{\{1,3\},2} \pi_4, 2 \xrightarrow{g} 1$	$\pi_{s,20}$	$1 \xrightarrow{g} 2, 2 \xrightarrow{g} 1, \pi_4 \rightarrow_3 \pi_1$
$\pi_{s,5}$	$1 \xrightarrow{r} 2, 2 \xrightarrow{r} 1$	$\pi_{s,21}$	$\pi_1 \rightarrow_3 \pi_4$
$\pi_{s,6}$	$\pi_4 \rightarrow_1 \pi_2, 2 \xrightarrow{g} 1, 3 \xrightarrow{r} 2$	$\pi_{s,22}$	$3 \xrightarrow{g} 1$
$\pi_{s,7}$	$1 \xrightarrow{g} 1$	$\pi_{s,23}$	$1 \xrightarrow{r} 2, \pi_4 \xrightarrow{T}_{\{2,3\},1} \pi_1$
$\pi_{s,8}$	$\pi_2 \xrightarrow{T}_{\{1,2\},1} \pi_3$	$\pi_{s,24}$	$2 \xrightarrow{r} 1, 3 \xrightarrow{r} 1$
$\pi_{s,9}$	$1 \xrightarrow{r} 1, \pi_4 \rightarrow_3 \pi_2$	$\pi_{s,25}$	$1 \xrightarrow{g} 2, \pi_1 \rightarrow_3 \pi_2$
$\pi_{s,10}$	$\pi_3 \rightarrow_1 \pi_4, 2 \xrightarrow{r} 1$	$\pi_{s,26}$	$1 \xrightarrow{g} 2, \pi_1 \rightarrow_2 \pi_3, 3 \xrightarrow{g} 2$
$\pi_{s,11}$	$\pi_3 \rightarrow_2 \pi_4$	$\pi_{s,27}$	$\pi_2 \xrightarrow{T}_{\{1,3\},2} \pi_4$
$\pi_{s,12}$	$1 \xrightarrow{g} 2, \pi_2 \rightarrow_3 \pi_4$	$\pi_{s,28}$	$1 \xrightarrow{r} 2, \pi_3 \rightarrow_2 \pi_1, 3 \xrightarrow{r} 2$
$\pi_{s,13}$	$\pi_4 \rightarrow_2 \pi_3, 3 \xrightarrow{g} 2$	$\pi_{s,29}$	$\pi_4 \rightarrow_3 \pi_1$
$\pi_{s,14}$	$\pi_4 \xrightarrow{T}_{\{1,3\},2} \pi_2$	$\pi_{s,30}$	$\pi_4 \rightarrow_1 \pi_1, 3 \xrightarrow{g} 1$
$\pi_{s,15}$	$1 \xrightarrow{r} 2, 2 \xrightarrow{g} 1, 3 \xrightarrow{r} 2$	$\pi_{s,31}$	$\pi_1 \rightarrow_2 \pi_4, 3 \xrightarrow{r} 1$
$\pi_{s,16}$	$2 \xrightarrow{r} 1, \pi_2 \rightarrow_3 \pi_3$	$\pi_{s,32}$	$\pi_1 \rightarrow_3 \pi_4$

the remaining regions. We consider the MITL formula

$$\Phi = (\Box_{[0,\infty)} \neg \text{“obs”}) \wedge \Diamond_{[0,60]} (\text{“green}_1\text{”} \wedge \text{“green}_2\text{”}),$$

which describes the following behavior: the coupled system must always avoid the obstacle regions and it must visit the greens region in the first 60 seconds. By following the presented algorithm, we obtain the accepting timed run $(\pi_1, 0)(\pi_2, 6)(\pi_3, 12)(\pi_4, 18)(\pi_5, 24)(\pi_{12}, 30)(\pi_{13}, 36)(\pi_{14}, 42)(\pi_{11}, 48)(\pi_{12}, 54)(\pi_5, 60)$.

Regarding each transition $\pi_{r_j} \xrightarrow{T} \pi_{r_{j+1}}, j \in \{1, \dots, 10\}$, we choose $\delta t_{r_j, r_{j'}} = 6$ s, $p_{r_j, r_{j'}}(t)$ as in (45) and $\eta_{r_j, r_{j'}}(t) = [0, 0, \frac{\pi}{4} \sin(\frac{\pi}{3}(t - t_{r_j}))]^T$, where $t_{r_j} = j \delta t_{r_j, r_{j'}} = 6j$ plays the role of t_0 for each transition. Regarding the parameters of the performance functions, we choose $\rho_{s_k, 0} = \rho_{s_k}(t_{r_j}) = l_0 = 0.5$ [m], $l_{s_k} = 0.5$, $\rho_{s_k, \infty} = \lim_{t \rightarrow \infty} \rho_{s_k}(t) = 0.1$ [m], $\forall k \in \{x, y, z\}$, $\rho_{s_k, 0} = \rho_{s_k}(t_{r_j}) = \frac{\pi}{2}$ [rad], $l_{s_k} = 0.5$, $\rho_{s_k, \infty} = \lim_{t \rightarrow \infty} \rho_{s_k}(t) = \frac{\pi}{12}$ r, $\forall k \in \{\phi, \theta, \psi\}$, $\rho_{v_k, 0} = \rho_{v_k}(t_{r_j}) = 2|e_{v_k}(t_{r_j})| + 0.5$, $l_{v_k} = 0.5$ and $\rho_{v_k, \infty} = \lim_{t \rightarrow \infty} \rho_{v_k}(t) = 0.1$, $k \in \mathcal{K}, j \in \{1, \dots, 10\}$. The control gains are chosen as $g_s = 1$, $g_v = 10$, and the agents are set to contribute equally to the object motion. The simulation results are depicted in Figs. 12-13. More specifically, Fig. 12 depicts the timed transi-

Table 2: The agent actions for the discrete path of the first case study (Part 2)

$\pi_{s,\ell}$	Actions	$\pi_{s,\ell}$	Actions
$\pi_{s,33}$	$\pi_1 \rightarrow_1 \pi_2, 2 \xrightarrow{g} 2, 3 \xrightarrow{g} 2$	$\pi_{s,49}^*$	$\pi_3 \xrightarrow{T} \{1,3\}, 2 \pi_4$
$\pi_{s,34}$	$\pi_4 \xrightarrow{T} \{2,3\}, 2 \pi_3$	$\pi_{s,50}^*$	$3 \xrightarrow{r} 2$
$\pi_{s,35}$	$\pi_2 \rightarrow_1 \pi_1, 2 \xrightarrow{r} 2, 3 \xrightarrow{r} 2$	$\pi_{s,51}^*$	$1 \xrightarrow{r} 2, 2 \xrightarrow{r} 1, \pi_4 \rightarrow_3 \pi_1$
$\pi_{s,36}$	$1 \xrightarrow{g} 1, \pi_3 \rightarrow_3 1$	$\pi_{s,52}^*$	$\pi_4 \rightarrow_1 \pi_1, 3 \xrightarrow{g} 1$
$\pi_{s,37}$	$\pi_3 \rightarrow_2 \pi_2, 3 \xrightarrow{g} 1$	$\pi_{s,53}^*$	$\pi_1 \rightarrow_2 \pi_4, 3 \xrightarrow{r} 1$
$\pi_{s,38}$	$\pi_1 \xrightarrow{T} \{1,3\}, 1 \pi_4$	$\pi_{s,54}^*$	$\pi_1 \rightarrow_3 \pi_4$
$\pi_{s,39}$	$\pi_2 \rightarrow_2 \pi_3, 1 \xrightarrow{r} 1, 3 \xrightarrow{r} 1$	$\pi_{s,55}^*$	$1 \xrightarrow{g} 1, 2 \xrightarrow{g} 2, 3 \xrightarrow{g} 2$
$\pi_{s,40}$	$\pi_4 \rightarrow_3 \pi_3$	$\pi_{s,56}^*$	$1 \xrightarrow{r} 1, \pi_4 \xrightarrow{T} \{2,3\}, 2 \pi_3$
$\pi_{s,41}$	$\pi_4 \rightarrow_1 \pi_3, 3 \xrightarrow{g} 2$	$\pi_{s,57}^*$	$2 \xrightarrow{r} 2, 3 \xrightarrow{r} 2$
$\pi_{s,42}^*$	$\pi_3 \rightarrow_2 \pi_4, 3 \xrightarrow{r} 2$	$\pi_{s,58}^*$	$1 \xrightarrow{g} 1, \pi_3 \rightarrow_3 \pi_1$
$\pi_{s,43}^*$	$\pi_3 \rightarrow_3 \pi_4$	$\pi_{s,59}^*$	$\pi_3 \rightarrow_2 \pi_2, 3 \xrightarrow{g} 1$
$\pi_{s,44}^*$	$2 \xrightarrow{g} 1, 3 \xrightarrow{g} 1$	$\pi_{s,60}^*$	$\pi_1 \xrightarrow{T} \{1,3\}, 1 \pi_4$
$\pi_{s,45}^*$	$\pi_4 \xrightarrow{T} \{2,3\}, 1 \pi_1$	$\pi_{s,61}^*$	$1 \xrightarrow{r} 1, \pi_2 \rightarrow_2 \pi_3, 3 \xrightarrow{r} 1$
$\pi_{s,46}^*$	$2 \xrightarrow{r} 1, 2 \xrightarrow{r} 3$	$\pi_{s,62}^*$	$\pi_4 \rightarrow_3 \pi_3$
$\pi_{s,47}^*$	$\pi_1 \rightarrow_3 \pi_3$	$\pi_{s,63}^*$	$\pi_4 \rightarrow_1 \pi_3, 3 \xrightarrow{g} 2$
$\pi_{s,48}^*$	$1 \xrightarrow{g} 2, 2 \xrightarrow{g} 1, 3 \xrightarrow{g} 2$		

tions of the coupled object-agents system, from which it can be deduced that $p_o(t) \in \mathcal{B}(p_{r_j, r_{j'}}, l_0)$ and therefore $\mathcal{S}_q(t) \subset \pi_{r_j} \cup \pi_{r_{j'}}, \forall j \in \{1, \dots, 10\}$. Moreover, Fig. 13 illustrate the errors $e_s(t)$ and $e_v(t)$ along with the performance functions $\rho_s(t)$, respectively, for all the transitions $\pi_{r_j} \rightarrow \pi_{r_{j'}}, j \in \{1, \dots, 10\}$. A video showing the aforementioned simulation paradigm can be found on <https://youtu.be/AiAt9NqL1jo>.

7. Conclusion and Discussion

This paper presented control and planning algorithms for multi-agent and multi-agent-object systems subject to temporal-logic-based tasks. Temporal logic tasks offer a wide range of complex objectives and, except for the robotic applications presented in this paper, they entail numerous applications, such as biological systems or epidemic strategies [118, 119]. By using formal-verification techniques, the problem often boils down to creating a finite-state abstraction of the continuous system, which, in turn, requires appropriate control algorithms for the transition of the system among the abstraction's states (see Sections 4, 5). Regarding *timed* transitions, required by timed temporal tasks, we adopt a

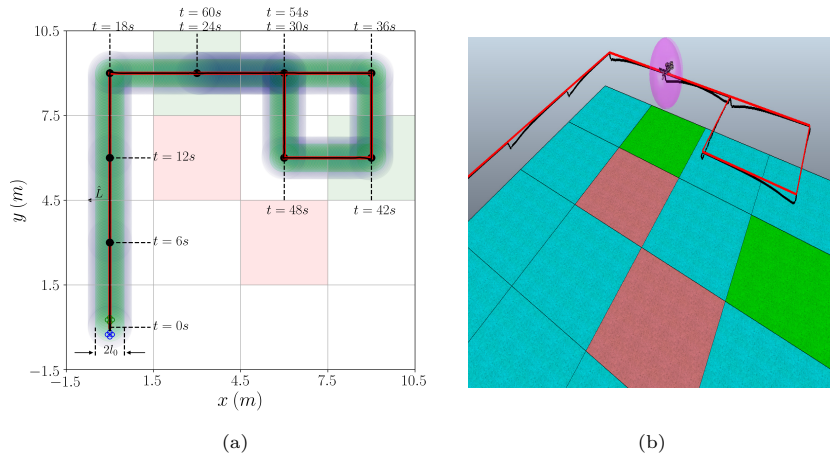


Figure 12: (a): The overall desired object trajectory (with red), the actual object trajectory (with black), the domain specified by $\mathcal{B}(p_{r_j, r_{j'}}(t), l_0), \forall j \in \{1, \dots, 10\}$ (with green), and the domain specified by $\mathcal{B}(p_O(t), \hat{L})$ (with blue), for $t \in [0, 60]$ s. (b): Illustration of the system at the final region at $t = 60$ s in the CoppeliaSim environment along with the ball $\mathcal{B}(p_O(60), \hat{L})$. Since $p_O \in \mathcal{B}(p_{r_j, r_{j'}}(t), l_0)$, the desired timed run is successfully executed.

complete partition of the state space, ensuring timed navigation among neighbouring cells. More general workspaces with arbitrarily-shaped obstacles can be found in [120, 121, 122] for single-agent systems. The extension to multiple agents, however, is significantly challenging since the inter-agent interactions and potential actuation constraints may prevent them from achieving navigation in pre-defined time intervals. Finally, the presented algorithms focus on the safety of the underlying system and on the adaptability and robustness to dynamic uncertainties and exogenous disturbances. Safety is a crucial property when it comes to physical dynamical systems that interact with each other, such as robots. Being an invariance-like property, however, safety concerns a much wider range of systems, including for instance sensor networks, biological systems, or power systems, where several state variables must operate within certain regions of operation. Robustness and adaptability to model uncertainties is an equally important property; the vast majority of dynamical systems cannot be modelled accurately and suffer, in many cases, from unknown disturbances or unexpected faults.

One can identify numerous future directions of the presented results. Regarding the control algorithms of Sections 4, 5, an important direction consists of the incorporation of explicit control-input constraints. For the multi-agent navigation (Section 4), an open problem is the simultaneous and *timed* navigation of the agents in complex environments. Such a problem is also interesting for the cooperative manipulation problem (Section 5), since the complex structure of robotic manipulators brings significant challenges when it comes to collision avoidance. Regarding the planning algorithms of Section 6, an important direc-

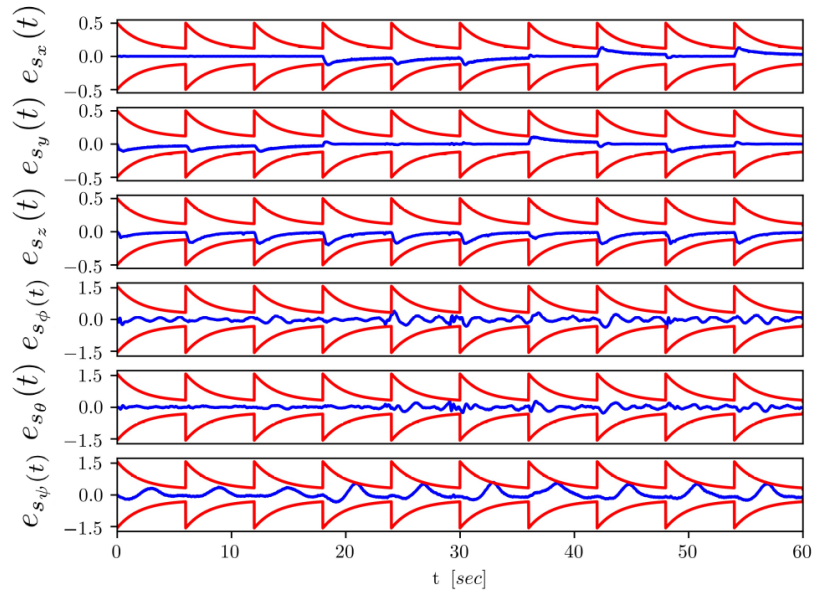


Figure 13: The pose errors $e_s(t)$ (with blue) along with the performance functions $\rho_s(t)$ (with red).

tion is the development of reconfigurable and more decentralized procedures for the derivation of high-level paths in multi-agent-object systems.

References

- [1] W. Ren, R. Beard, Consensus Seeking in Multi-agent Systems under Dynamically Changing Interaction Topologies, *IEEE Transactions on Automatic Control (TAC)* 50 (5) (2005) 655–661.
- [2] R. Olfati-Saber, R. Murray, Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control (TAC)* 49 (9) (2004) 1520–1533.
- [3] A. Jadbabaie, J. Lin, S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Transactions on Automatic Control (TAC)* 48 (6) (2003) 988–1001.
- [4] H. Tanner, A. Jadbabaie, G. J. Pappas, Flocking in fixed and switching networks, *IEEE Transactions on Automatic Control (TAC)* 52 (5) (2007) 863–868.
- [5] D. V. Dimarogonas, K. Kyriakopoulos, On the rendezvous problem for multiple nonholonomic agents, *IEEE Transactions on Automatic Control (TAC)* 52 (5) (2007) 916–922.

- [6] M. Egerstedt, X. Hu, Formation constrained multi-agent control, *IEEE Transactions on Robotics and Automation* 17 (6) (2001) 947–951.
- [7] K. Oh, M. Park, H. Ahn, A survey of multi-agent formation control, *Automatica* 53 (2015) 424–440.
- [8] M. Ji, M. Egerstedt, Distributed Coordination Control of Multi-Agent Systems While Preserving Connectedness, *IEEE Transactions on Robotics (TRO)* 23 (4) (2007) 693–703.
- [9] M. Zavlanos, G. J. Pappas, Potential Fields for Maintaining Connectivity of Mobile Networks, *IEEE Transactions on Robotics (TRO)* 23 (4) (2007) 812–816.
- [10] M. Zavlanos, G. J. Pappas, Distributed connectivity control of mobile networks, *IEEE Transactions on Robotics (TRO)* 24 (6) (2008) 1416–1428.
- [11] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, M. Zavlanos, A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Non-Point Agents, *Automatica* 42 (2) (2006) 229–243.
- [12] M. Krstic, I. Kanellakopoulos, P. Kokotovic, *Nonlinear and Adaptive Control Design*, Publisher: Wiley New York (1995).
- [13] C. Verginis, Planning and control of uncertain cooperative mobile manipulator-endowed systems under temporal logic tasks, Ph.D. thesis, KTH Royal Institute of Technology (2020).
- [14] J.-C. Latombe, *Robot motion planning*, Vol. 124, Springer Science & Business Media, 2012.
- [15] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*, MIT press, 2005.
- [16] S. M. LaValle, *Planning algorithms*, Cambridge university press, 2006.
- [17] S. G. Loizou, K. J. Kyriakopoulos, Closed loop navigation for multiple holonomic vehicles, *IEEE/RSJ International Conference on Intelligent Robots and Systems* 3 (2002) 2861–2866.
- [18] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, M. M. Zavlanos, A feedback stabilization and collision avoidance scheme for multiple independent non-point agents, *Automatica* 42 (2) (2006) 229–243.
- [19] G. Roussos, K. J. Kyriakopoulos, Decentralized and prioritized navigation and collision avoidance for multiple mobile robots, *Distributed Autonomous Robotic Systems* (2013) 189–202.

- [20] D. Koditschek, E. Rimon, Robot navigation functions on manifolds with boundary, *Advances in Applied Mathematics* 11 (4) (1990) 412–442.
- [21] E. Rimon, D. E. Koditschek, Exact robot navigation using artificial potential functions, *IEEE Transactions on Robotics and Automation* 8 (5) (1992) 501–518.
- [22] H. G. Tanner, A. Kumar, Towards decentralization of multi-robot navigation functions., *IEEE International Conference on Robotics and Automation (ICRA)* 4 (2005) 4132.
- [23] S. G. Loizou, The multi-agent navigation transformation: Tuning-free multi-robot navigation., *Robotics: Science and Systems* 6 (2014) 1516–1523.
- [24] D. Panagou, A distributed feedback motion planning protocol for multiple unicycle agents of different classes, *IEEE Transactions on Automatic Control* 62 (3) (2017) 1178–1193.
- [25] L. Wang, A. D. Ames, M. Egerstedt, Safety barrier certificates for collisions-free multirobot systems, *IEEE Transactions on Robotics* 33 (3) (2017) 661–674.
- [26] A. Filotheou, A. Nikou, D. V. Dimarogonas, Decentralized control of uncertain multi-agent systems with connectivity maintenance and collision avoidance, *European Control Conference* (2018).
- [27] J. Van Den Berg, J. Snape, S. J. Guy, D. Manocha, Reciprocal collision avoidance with acceleration-velocity obstacles, *IEEE International Conference on Robotics and Automation (ICRA)* (2011).
- [28] E. J. Rodríguez-Seda, D. M. Stipanović, M. W. Spong, Collision avoidance control with sensing uncertainties, *Proceedings of the 2011 American Control Conference* (2011) 3363–3368.
- [29] O. Arslan, D. P. Guralnik, D. E. Koditschek, Coordinated robot navigation via hierarchical clustering, *IEEE Transactions on Robotics* 32 (2) (2016) 352–371.
- [30] X. Zhang, L. Liu, G. Feng, Leader–follower consensus of time-varying nonlinear multi-agent systems, *Automatica* 52 (2015) 8–14.
- [31] Z. Li, X. Liu, W. Ren, L. Xie, Distributed tracking control for linear multiagent systems with a leader of bounded unknown input, *IEEE Transactions on Automatic Control* 58 (2) (2013).
- [32] W. Liu, J. Huang, Adaptive leader-following consensus for a class of higher-order nonlinear multi-agent systems with directed switching networks, *Automatica* 79 (2017) 84–92.

- [33] J. Hu, G. Feng, Distributed tracking control of leader–follower multi-agent systems under noisy measurement, *Automatica* 46 (8) (2010) 1382–1387.
- [34] M. Turpin, N. Michael, V. Kumar, Capt: Concurrent assignment and planning of trajectories for multiple robots, *The International Journal of Robotics Research* 33 (1) (2014) 98–112.
- [35] M. M. Zavlanos, G. J. Pappas, Distributed connectivity control of mobile networks, *IEEE Transactions on Robotics* 24 (6) (2008) 1416–1428.
- [36] M. Ji, M. Egerstedt, Distributed coordination control of multiagent systems while preserving connectedness, *IEEE Transactions on Robotics* 23 (4) (2007) 693–703.
- [37] L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, Distributed control of multirobot systems with global connectivity maintenance, *IEEE Transactions on Robotics* 29 (5) (2013) 1326–1332.
- [38] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, M. W. Spong, Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments, *The International Journal of Robotics Research* 27 (1) (2008) 107–126.
- [39] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, Decentralized cooperation between multiple manipulators, *IEEE International Workshop on Robot and Human Communication* (1996) 183–188.
- [40] Y.-H. Liu, S. Arimoto, T. Ogasawara, Decentralized cooperation control: non-communication object handling, *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)* 3 (1996) 2414–2419.
- [41] Y.-H. Liu, S. Arimoto, Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations, *The International Journal of Robotics Research* 17 (3) (1998) 232–247.
- [42] M. Zribi, S. Ahmad, Adaptive control for multiple cooperative robot arms, *Proceedings of the IEEE Conference on Decision and Control (CDC)* (1992) 1392–1398.
- [43] L. Gudiño-Lau, M. Arteaga, L. Munoz, V. Parra-Vega, On the control of cooperative robots without velocity measurements, *IEEE Transactions on Control Systems Technology* 12 (4) (2004) 600–608.
- [44] F. Caccavale, P. Chiacchio, S. Chiaverini, Task-space regulation of cooperative manipulators, *Automatica* 36 (6) (2000) 879–887.
- [45] F. Caccavale, P. Chiacchio, A. Marino, L. Villani, Six-dof impedance control of dual-arm cooperative manipulators, *IEEE/ASME Transactions On Mechatronics* 13 (5) (2008) 576–586.

- [46] D. Heck, D. Kostic, A. Denasi, H. Nijmeijer, Internal and external force-based impedance control for cooperative manipulation, Proceedings of the IEEE European Control Conference (ECC) (2013) 2299–2304.
- [47] S. Erhart, S. Hirche, Adaptive force/velocity control for multi-robot cooperative manipulation under uncertain kinematic parameters, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2013) 307–314.
- [48] S. Erhart, D. Sieber, S. Hirche, An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2013) 315–322.
- [49] Y. Kume, Y. Hirata, K. Kosuge, Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2007) 4077–4082.
- [50] A. Tsiamis, C. K. Verginis, C. P. Bechlioulis, K. J. Kyriakopoulos, Cooperative manipulation exploiting only implicit communication, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015) 864–869.
- [51] F. Ficuciello, A. Romano, L. Villani, B. Siciliano, Cartesian impedance control of redundant manipulators for human-robot co-manipulation, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2014) 2120–2125.
- [52] A.-N. Ponce-Hinestroza, J.-A. Castro-Castro, H.-I. Guerrero-Reyes, V. Parra-Vega, E. Olguyn-Dyaz, Cooperative redundant omnidirectional mobile manipulators: Model-free decentralized integral sliding modes and passive velocity fields, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2016) 2375–2380.
- [53] W. Gueaieb, F. Karray, S. Al-Sharhan, A robust hybrid intelligent position/force control scheme for cooperative manipulators, IEEE/ASME Transactions on Mechatronics 12 (2) (2007) 109–125.
- [54] Z. Li, C. Yang, C. Y. Su, S. Deng, F. Sun, W. Zhang, Decentralized fuzzy control of multiple cooperating robotic manipulators with impedance interaction, IEEE Transactions on Fuzzy Systems 23 (4) (2015) 1044–1056.
- [55] A. Marino, Distributed adaptive control of networked cooperative mobile manipulators, IEEE Transactions on Control Systems Technology 26 (5) (2017) 1646–1660.
- [56] M. Ciocarlie, F. Hicks, R. Holmberg, J. Hawke, M. Schlicht, J. Gee, S. Stanford, R. Bahadur, The velo gripper: A versatile single-actuator design for enveloping, parallel and fingertip grasps, The International Journal of Robotics Research (2014).

- [57] J. T. Wen, K. Kreutz-Delgado, Motion and force control of multiple robotic manipulators, *Automatica* 28 (4) (1992) 729–743.
- [58] A. Petitti, A. Franchi, D. D. Paola, A. Rizzo, Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2016) 441–446.
- [59] A. Marino, F. Pierri, A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots, *Robotics and Autonomous Systems* 103 (2018) 122–133.
- [60] A. Marino, A decentralized adaptive control for tightly connected networked lagrangian systems, *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (2017) 4656–4661.
- [61] F. Aghili, Self-tuning cooperative control of manipulators with position/orientation uncertainties in the closed-kinematic loop, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2011) 4187–4193.
- [62] D. Ortenzi, R. Muthusamy, A. Freddi, A. Monteriù, V. Kyrki, Dual-arm cooperative manipulation under joint limit constraints, *Robotics and Autonomous Systems* 99 (2018) 110–120.
- [63] S. Erhart, S. Hirche, Internal force analysis and load distribution for cooperative multi-robot manipulation, *IEEE Transactions on Robotics* 31 (5) (2015) 1238–1243.
- [64] T. D. Murphey, M. Horowitz, Adaptive cooperative manipulation with intermittent contact, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2008) 1483–1488.
- [65] Z. Wang, M. Schwager, Kinematic multi-robot manipulation with no communication using force feedback, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2016) 427–432.
- [66] H. Bai, J. T. Wen, Cooperative load transport: A formation-control perspective, *IEEE Transactions on Robotics* 26 (4) (2010) 742–750.
- [67] D. Sieber, S. Hirche, Human-guided multirobot cooperative manipulation, *IEEE Transactions on Control Systems Technology* 27 (4) (2018) 1492–1509.
- [68] M. Guo, D. V. Dimarogonas, Multi-Agent Plan Reconfiguration Under Local LTL Specifications, *The International Journal of Robotics Research (IJRR)* 34 (2) (2015) 218–235.

- [69] A. Ulusoy, S. Smith, X. Ding, C. Belta, D. Rus, Optimal Multi-Robot Path Planning with Temporal Logic Constraints, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2011) 3087–3092.
- [70] I. Filippidis, D. V. Dimarogonas, K. J. Kyriakopoulos, Decentralized Multi-Agent Control from Local LTL Specifications, 51st IEEE Conference on Decision and Control (CDC), Maui, Hawaii, USA (2012) 6235–6240.
- [71] Z. Zhang, R. V. Cowlagi, Motion-planning with global temporal logic specifications for multiple nonholonomic robotic vehicles, Proceedings of the IEEE American Control Conference (ACC), Boston, MA, USA (2016) 7098–7103.
- [72] Y. Diaz-Mercado, A. Jones, C. Belta, M. Egerstedt, Correct-by-construction control synthesis for multi-robot mixing, Proceedings of the IEEE Conference on Decision and Control (CDC), Osaka, Japan (2015) 221–226.
- [73] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, D. Rus, Optimality and robustness in multi-robot path planning with temporal logic constraints, *The International Journal of Robotics Research* 32 (8) (2013) 889–911.
- [74] A. Nikou, D. Boskos, J. Tumova, D. V. Dimarogonas, On the Timed Temporal Logic Planning of Coupled Multi-Agent Systems, *Automatica* (2017, (Under Review)).
- [75] S. Karaman, E. Frazzoli, Linear temporal logic vehicle routing with applications to multi-uav mission planning, *International Journal of Robust and Nonlinear Control* 21 (12) (2011) 1372–1395.
- [76] L. Lindemann, D. V. Dimarogonas, Decentralized robust control of coupled multi-agent systems under local signal temporal logic tasks, 2018 Annual American Control Conference (ACC) (2018) 1567–1573.
- [77] D. Sun, J. Chen, S. Mitra, C. Fan, Multi-agent motion planning from signal temporal logic specifications, arXiv preprint arXiv:2201.05247 (2022).
- [78] A. T. Buyukkocak, D. Aksaray, Y. Yazıcıoğlu, Planning of heterogeneous multi-agent systems under signal temporal logic specifications with integral predicates, *IEEE Robotics and Automation Letters* 6 (2) (2021) 1375–1382.
- [79] C. Baier, J. Katoen, K. G. Larsen, Principles of model checking, MIT Press (2008).
- [80] J. Ouaknine, J. Worrell, On the Decidability of Metric Temporal Logic, 20th Annual IEEE Symposium on Logic in Computer Science (LICS) (2005) 188–197.

- [81] D. Souza, P. Prabhakar, On the expressiveness of mtl in the pointwise and continuous semantics, *International Journal on Software Tools for Technology Transfer* 9 (1) (2007) 1–4.
- [82] C. K. Verginis, D. V. Dimarogonas, Closed-form barrier functions for multi-agent ellipsoidal systems with uncertain lagrangian dynamics, *IEEE Control Systems Letters (LCSS)* (2019) 727–732.
- [83] C. K. Verginis, D. V. Dimarogonas, Adaptive leader-follower coordination of lagrangian multi-agentsystems under transient constraints, *IEEE Conference on Decision and Control (CDC)* (2020) 3833–3838.
- [84] C. K. Verginis, D. V. Dimarogonas, Adaptive robot navigation with collision avoidance subject to 2nd-order uncertain dynamics, *Automatica* 123 (2021) 109303.
- [85] E. Rimon, D. Koditschek, Exact robot navigation using artificial potential functions, *IEEE Transactions on Robotics and Automation (TRA)* 8 (5) (1992) 501–518.
- [86] T. Pan, C. K. Verginis, A. M. Wells, D. V. Dimarogonas, L. E. Kavraki, Augmenting control policies with motion planning for robust and safe multi-robot navigation, submitted to the *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020).
- [87] N. P. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator., *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 4 (2004).
- [88] C. K. Verginis, W. S. Cortez, D. V. Dimarogonas, Adaptive cooperative manipulation with rolling contacts, To appear in the *American Control Conference (ACC)*, Denver, Colorado, USA (2020).
- [89] W. S. Cortez, C. K. Verginis, D. V. Dimarogonas, A distributed, event-triggered, adaptive controller for cooperative manipulation with rolling contacts, Under Review (2022).
- [90] B. S. L. Sciavicco, L. Villani, G. Oriolo, *Robotics: modelling, planning and control*, Springer Science & Business Media (2010).
- [91] C. Bechlioulis, G. Rovithakis, Robust Adaptive Control of Feedback Linearizable MIMO Nonlinear Systems with Prescribed Performance, *IEEE Transactions on Automatic Control (TAC)* 53 (9) (2008) 2090–2099.
- [92] C. K. Verginis, M. Mastellaro, D. V. Dimarogonas, Robust cooperative manipulation without force/torque measurements: Control design and experiments, *IEEE Transactions on Control Systems Technology* 28 (3) (2020) 713–729.

- [93] C. K. Verginis, M. Mastellaro, D. V. Dimarogonas, Robust quaternion-based cooperative manipulation without force/torque information, *IFAC-PapersOnLine* 50 (1) (2017) 1754–1759.
- [94] C. K. Verginis, A. Nikou, D. V. Dimarogonas, Robust formation control in $se(3)$ for tree-graph structures with prescribed transient and steady state performance, *Automatica* 103 (2019) 538–548.
- [95] C. K. Verginis, D. V. Dimarogonas, Asymptotic stability of uncertain lagrangian systems with prescribed transient response, *IEEE Conference on Decision and Control (CDC)* (2019) 7037–7042.
- [96] C. K. Verginis, D. V. Dimarogonas, Asymptotic tracking of second-order nonsmooth feedback stabilizable unknown systems with prescribed transient response, under review (2019).
- [97] A. Nikou, C. K. Verginis, S. Heshmati-alamdari, D. V. Dimarogonas, A nonlinear model predictive control scheme for cooperative manipulation with singularity and collision avoidance, *Proceedings of the 25th IEEE Mediterranean Conference on Control and Automation (MED)*, Valletta, Malata (2017) 707–712.
- [98] C. K. Verginis, A. Nikou, D. V. Dimarogonas, Communication-based decentralized cooperative object transportation using nonlinear model predictive control, *European control conference (ECC)* (2018) 733–738.
- [99] N. Lissandrini, C. K. Verginis, P. Roque, A. Cenedese, D. V. Dimarogonas, Decentralized nonlinear mpc for robust cooperative manipulation by heterogeneous aerial-ground robots, submitted to the *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020).
- [100] A. Nikou, D. V. Dimarogonas, Decentralized tube-based model predictive control of uncertain nonlinear multiagent systems, *International Journal of Robust and Nonlinear Control* 29 (10) (2019) 2799–2818.
- [101] S. Yu, C. Maier, H. Chen, F. Allgöwer, Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems, *Systems & Control Letters* 62 (2) (2013) 194–200.
- [102] C. K. Verginis, D. Zelazo, D. V. Dimarogonas, Cooperative manipulation via internal force regulation: A rigidity theory perspective, To appear in the *IEEE Transactions on Control of Network Systems* (2022).
- [103] C. K. Verginis, D. V. Dimarogonas, Energy-optimal cooperative manipulation via provable internal-force regulation, *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France (2020).
- [104] L. Grüne, J. Pannek, *Nonlinear Model Predictive Control*, Springer London (2011).

- [105] R. Eric, P. N. S. S. F., Marc, V-rep: a versatile and scalable robot simulation framework, Proceedings of The International Conference on Intelligent Robots and Systems (IROS) (2013).
- [106] C. K. Verginis, Z. Xu, D. V. Dimarogonas, Decentralized motion planning with collision avoidance for a team of uavs under high level goals, Proceedings of the International Conference on Robotics and Automation (ICRA) (2017) 781–787.
- [107] C. K. Verginis, D. V. Dimarogonas, Robust decentralized abstractions for multiple mobile manipulators, Proceedings of the Conference on Decision and Control (CDC), Melbourne, Australia (2017) 2222–2227.
- [108] C. K. Verginis, D. V. Dimarogonas, Multi-agent motion planning and object transportation under high level goals, IFAC-PapersOnLine 50 (1) (2017) 15816–15821.
- [109] C. K. Verginis, Y. Kantaros, D. V. Dimarogonas, Planning and control of multi-robot-object systems under temporal logic tasks and uncertain dynamics, Arxiv. Available online: <https://arxiv.org/pdf/2204.11783.pdf> (2022).
- [110] D. Z. Chen, Sphere packing problem, Encyclopedia of Algorithms (2008) 1–99.
- [111] M. R. Cutkosky, Robotic grasping and fine manipulation, Vol. 6, Springer Science & Business Media, 2012.
- [112] C. K. Verginis, D. V. Dimarogonas, Distributed cooperative manipulation under timed temporal specifications, American Control Conference (ACC), Seattle, WA, USA (2017) 1358–1363.
- [113] C. K. Verginis, D. V. Dimarogonas, Timed abstractions for distributed cooperative manipulation, Autonomous Robots 42 (4) (2018) 781–799.
- [114] R. Alur, D. L. Dill, A theory of timed automata, Theoretical computer science 126 (2) (1994) 183–235.
- [115] A. Nikou, J. Tumova, D. V. Dimarogonas, Cooperative task planning of multi-agent systems under timed temporal specifications, Proceedings of the IEEE American Control Conference (ACC), Boston, MA, USA (2016) 7104–7109.
- [116] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, K. J. Kyriakopoulos, Robot navigation in complex workspaces using harmonic maps, 2018 IEEE International Conference on Robotics and Automation (ICRA) (2018) 1726–1731.
- [117] Y. Kantaros, M. M. Zavlanos, Stylus*: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems, The International Journal of Robotics Research 39 (7) (2020) 812–836.

- [118] J.-M. Alliot, R. Demolombe, M. Diéguez, L. Fariñas del Cerro, G. Favre, J.-C. Faye, N. Obeid, O. Sordet, Temporal logic modeling of biological systems, in: *Towards paraconsistent engineering*, Springer, 2016, pp. 205–226.
- [119] Z. Xu, B. Wu, U. Topcu, Control strategies for covid-19 epidemic with vaccination, shield immunity and quarantine: A metric temporal logic approach, *PloS one* 16 (3) (2021) e0247660.
- [120] C. K. Verginis, C. Vrohidis, C. P. Bechlioulis, K. J. Kyriakopoulos, D. V. Dimarogonas, Reconfigurable motion planning and control in obstacle cluttered environments under timed temporal tasks, *IEEE International Conference on Robotics and Automation (ICRA)* (2019) 951–957.
- [121] C. K. Verginis, D. V. Dimarogonas, L. E. Kavraki, Sampling-based motion planning for uncertain high-dimensional systems via adaptive control, *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Oulu, Finland (2020).
- [122] C. K. Verginis, D. V. Dimarogonas, L. E. Kavraki, Kdf: Kinodynamic motion planning via geometric sampling-based algorithms and funnel control, Under Review. Arxiv Link: <https://arxiv.org/pdf/2104.11917.pdf> (2022).