

Article

# Mutli-Robot Cooperative Object Transportation with Guaranteed Safety and Convergence in Planar Obstacle Cluttered Workspaces via Configuration Space Decomposition

Panagiotis Vlantis <sup>1</sup>, Charalampos P. Bechlioulis <sup>1,\*</sup>  and Kostas J. Kyriakopoulos <sup>2</sup> <sup>1</sup> Department of Electrical and Computer Engineering, University of Patras, 26504 Rio, Greece<sup>2</sup> School of Mechanical Engineering, National Technical University of Athens, 15772 Athens, Greece\* Correspondence: [chmpechl@upatras.gr](mailto:chmpechl@upatras.gr)

**Abstract:** In this work, we consider the autonomous object transportation problem employing a team of mobile manipulators within a compact planar workspace with obstacles. As the object is allowed to translate and rotate and each robot is equipped with a manipulator consisting of one or more moving links, the overall system (object and mobile manipulators) should adapt its shape in a flexible way so that it fulfills the transportation task with safety. To this end, we built a sequence of configuration space cells, each of which defines an allowable set of configurations of the object, as well as explicit intervals for each manipulator's states. Furthermore, appropriately designed under- and over-approximations of the free configuration space are used in an innovative way to guide the configuration space's exploration without loss of completeness. In addition, we coupled methodologies based on Reference Governors and Prescribed Performance Control with harmonic maps, in order to design a distributed control law for implementing the transitions specified by the high-level planner, which possesses guaranteed invariance and global convergence properties, thus avoiding the requirement for synchronized motion as inherently dictated by the majority of the related works. Furthermore, the proposed low-level control law does not require continuous information exchange between the robots, which rely only on measurements of the object's configuration and their own states. Finally, a transportation scenario within a complex warehouse workspace demonstrates the proposed approach and verifies its efficiency.

**Keywords:** object transportation; cooperative control; mobile manipulator

**Citation:** Vlantis, P.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Mutli-Robot Cooperative Object Transportation with Guaranteed Safety and Convergence in Planar Obstacle Cluttered Workspaces via Configuration Space Decomposition. *Robotics* **2022**, *11*, 148. <https://doi.org/10.3390/robotics11060148>

Academic Editor: António Paulo Moreira

Received: 10 October 2022

Accepted: 7 December 2022

Published: 9 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Until recently, the most common type of robot used on a global scale was single industrial-style arms operating on automated production lines and in well-defined and protected obstacle-free environments. Their main advantages were robust and reliable operation, high precision, and repeatability in their movement; consequently, they were employed in repetitive and high-load manual tasks in order to reduce production costs and increase the productivity. In recent years, however, the research activity shifted towards pushing robots out of a “sterile” workplace and enabling them to co-exist with humans and unknown obstacles in a fully functional industrial environment. Additionally, efforts have been put to create multi-robot systems that exhibit collective and cooperative behavior, since many practical problems are impossible to be solved with a single robot, either due to physical limitations and/or due to limited resources or information. These reasons highlight the importance of a cooperative behavior, where each member contributes to the achievement of a final goal.

The present work focuses on the object transportation problem from an initial point to an end point-goal within an arbitrary obstacle-cluttered indoor environment employing multiple collaborative mobile manipulator systems. The problem at hand applies to a variety of real-world applications in industry, such as moving objects on a production line

or storing objects in a warehouse, particularly when the object to be transported is bulky or heavy. Towards this direction, a cooperative multi-robot system is designed, which involves a high-level planner that builds a sequence of intermediate configurations that are implemented by local controllers based on the reference governor and prescribed performance control techniques. In this way, the specifications of the problem are achieved, as each robot is tasked with fulfilling the object trajectory plan by acting completely individually.

### 1.1. Related Work

Since 1960, significant progress has been made in the field of robotics and new functionalities and capabilities have been added, with the direct consequence of ever expanding the range of applications in which a robot can take part. One of them is the transportation of an object from an initial point to an end-point within a workspace, which is the subject of this work. This problem was originally defined in the framework of the multi-cooperative robot system [1], with research oriented to the distribution of tasks and the cooperation of the individual robotic systems. A problem of this kind has innumerable practical applications; the most widespread are the transport of dangerous objects, automation in an industrial environment, the transport of very bulky objects, the rescue of injured people, etc. Since then, various strategies have been proposed to solve the considered problem that involves various sub-tasks such as trajectory planning, obstacle avoidance, robustness, etc. The communication between the robots of the system has also been thoroughly studied. Yamada states that, regardless of system configuration (centralized or decentralized), communication needs may be omitted and replaced by specific behavioral mechanisms based on local information [2]. However, such strategies have the disadvantage of requiring a set of predefined behaviors to handle new challenges and obstacles. On the other hand, Munoz claims that communication can significantly improve the performance of multi-robot systems, positively influencing coordination, cooperation, and conflict resolution [3]. In particular, the object transportation problem has been addressed by the well-known object closure strategy, in which a team of robots enclose the object so that the position of the object can be controlled by reference to the position of each robot [4–8]. Another widespread technique is that of leader–follower, where a robot determines the movement of the object and defines the behavior of the others [9–13]. In addition, more recent strategies make use of swarm intelligence, a variation of which is followed in this work, in which homogeneous robots are used, which are based on decentralized and collective behaviors [14]. Finally, techniques based on machine learning and artificial intelligence have also appeared, which understandably require high computing power [15].

The problem of cooperation is distinguished by increased complexity and requires the combination of many individual research processes in order to achieve it [16]. The most critical issue to be solved, without even the slightest discount, is that of safety, as the robots' operating space is now accessible by humans and other valuable equipment. It is therefore necessary to develop reliable safety systems [17], either on the design via robots with elastic and low-inertia moving parts or alongside algorithmic active solutions to avoid collisions. Additionally, collaboration with physical contact involves the development of forces between the robots and the commonly grasped, which affect both the safety as well as the smooth operation of the robots. Until recently, industrial robots were in a protected environment without obstacles. Therefore, only position control was effectively used for their movement. However, when robots are introduced in unstructured and uncertain environments, there are immediate effects on the safety and stability of the overall control system. For example, a potential force applied to the robot's body, depending on its magnitude and direction, will create a position error that the controller will try to compensate for with unknown results. In general, the greater the stiffness of the environment is, the greater forces and moments will be developed. Possible results of such an unavoidable scenario include, among others, the breakdown of the robot and even injury of a human or damage within the environment [18].

Cooperative manipulation particularly has been well-studied in the literature, especially the centralized scheme. In [19], a hybrid position/force control was presented. In [20], the overall closed-chain system is treated as an augmented object, with its inertial properties expressed via a single inertia matrix. The authors in [21] propose a centralized motion planning methodology based on dipolar navigation functions for nonholonomic mobile manipulators. The concept of object impedance control is also presented in [22]. Nevertheless, despite its efficacy, centralized control is less robust, since all robots depend on a central system and its complexity rises sharply and heavy inter-robot communication is required, as the number of team-robots increases. On the other hand, decentralized control usually depends on either explicit communication or off-line knowledge of the desired object trajectory, e.g., [23,24]. Furthermore, position–force hybrid control schemes, where the position of the object is controlled towards a given direction in the workspace and the internal forces on the object are controlled close to the origin are presented in [25–27]. Moreover, in other leader–follower schemes, e.g., [28,29], the leader has to transmit on-line the desired object trajectory to the follower.

Alternatively, implicit communication has been adopted in various decentralized studies on mobile manipulators. Kosuge et al. [30], presented a leader–follower scheme for holonomic manipulators in free space, where the leader implements a reference trajectory through an impedance scheme, while the follower estimates it through the motion of the object. However, the estimation error is kept small only for fixed velocity profiles. Regarding non-holonomic mobile robots, the follower’s passive caster behavior was adopted in [31]. Although, the stability of the follower’s contact is established, it is not mentioned how the object’s trajectory can be controlled. Alternatively, the authors in [32] designed a motion coordination controller with no explicit communication for a group of physically connected robots using only interaction force measurements. In a similar direction but following a pushing-only strategy, refs. [33,34] employed a visual occlusion notion to guide the robot swarm to the goal position without exchanging any information. Finally, in [35] the robots coordinate their actions by sensing the motion of the object itself.

## 1.2. Contribution

In this work, we generalize our previous effort [36] by presenting a methodology for coordinating the transportation of an object that is rigidly grasped by a team of mobile manipulators, which operate within a compact planar workspace with obstacles of arbitrary shape. Owing to the object rotation and the manipulators’ motion, our scheme takes into consideration the varying configuration of the robotic system, as opposed to [36], in order to build a plan that can safely drive the robotic system to the goal configuration. More specifically, we devise a high-level planner which is tasked with building a sequence of adjacent configuration space cells of the overall system (i.e., robots and object) that connect the system’s initial and desired configurations, each of which defines an allowable set of configurations for the object, as well as explicit intervals for each manipulator’s states. The main contributions of this work are summarized as follows:

- **Completeness:** We innovatively introduce appropriately designed under- and over-approximations of the free configuration space in order to guide the configuration space’s exploration by selecting the cells that need further subdivision, thus establishing the completeness of our approach (i.e., if there exists a feasible path to go to the goal configuration, our algorithm will discover it, otherwise it halts when the problem cannot be solved when the initial and the goal configurations belong to disconnected parts of the workspace).
- **Safety and Convergence:** We combine methodologies based on Reference Governors and Prescribed Performance Control with our recently proposed harmonic maps approach [37] in order to design a distributed control law that implements specified cell transitions with guaranteed invariance and convergence properties.
- **Lean Communication:** Contrary to majority of the related literature, the proposed low-level control law does not require continuous information exchange between the

robots (e.g., via a local network), thus rendering the expected latency negligible, since it relies exclusively on measurements of the object’s current configuration and the state of the corresponding robot in order to compute the respective control inputs. Regarding potential delays in the local measurements since our approach is a feedback control approach certain levels of robustness against measurement delays are expected.

To the best of our knowledge, there is no other approach that addresses the coupled path and motion planning problems with guaranteed safety. For instance, probabilistic methods can be employed for obtaining a trajectory of the states of the augmented robotic system but the low-level controllers employed for realizing it would require exact coordination/synchronization of the independent units for tracking, otherwise the robots would risk to collide with the obstacles and among them, because of the transient behavior of its individual robot’s controller. On the contrary, our approach adopts a decentralized control scheme that executes the transitions with provable guarantees of safety and convergence.

### 1.3. Outline

The outline of this work is given as follows. First, we present some preliminary notation and definitions in Section 2. Next, the problem at hand is formulated in Section 3. In Section 4, we present the control scheme that drives the robotic system to the specified goal configuration while ensuring collision avoidance with the workspace boundary, and we elaborate on the closed-loop system’s properties in Section 5. Finally, we provide simulation results verifying the efficacy of our approach in Section 6.

## 2. Preliminaries

Throughout this chapter, we shall use  $\mathbb{R}$  to denote the set of real numbers and  $\mathbb{N}$  to denote the set of natural numbers starting from zero. Moreover, we shall use  $\mathcal{I}_N \triangleq \{1, 2, \dots, N\}$  (resp.  $\mathcal{I}_N^* \triangleq \{0, 1, 2, \dots, N\}$ ) to denote the set consisting of all natural numbers up to  $N$ , starting from 1 (resp. 0). Additionally, given sets  $A$  and  $B$ , we use  $\partial A$ ,  $\text{int}(A)$ ,  $\text{cl}(A)$  to denote the boundary, interior, closure respectively, and  $A \setminus B$  to denote the complement of  $B$  with respect to  $A$ .

Given a coordinate frame  $\mathcal{F}_O$  in  $\mathbb{R}^2$  and two points  $P_A, P_B \in \mathbb{R}^2$ , we will use  ${}^{O}P_B$  to denote the position of point  $P_B$  relative to point  $P_A$ , expressed with respect to  $\mathcal{F}_O$ . Given frames  $\mathcal{F}_A, \mathcal{F}_B, \mathcal{F}_C$ , we will use  ${}^A P_C \in \mathbb{R}^2$  to denote the position of the origin of frame  $\mathcal{F}_C$  relative to the origin of frame  $\mathcal{F}_B$ , expressed with respect to  $\mathcal{F}_A$ . Accordingly, given frames  $\mathcal{F}_A, \mathcal{F}_B$ , we will use  ${}^A R \in \mathbb{R}^{2 \times 2}$  to denote the rotation matrix corresponding to the relative orientation of  $\mathcal{F}_B$  with respect to  $\mathcal{F}_A$ .

Given a rotation angle  $\theta$ , let  $R(\theta)$  be the rotation matrix defined as

$$R(\theta) \triangleq \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

For two given coordinate frames  $\mathcal{F}_A, \mathcal{F}_B$ , we define  ${}^A \mathcal{T}$  as the homogeneous transformation from frame  $\mathcal{F}_B$  to  $\mathcal{F}_A$ , defined as

$${}^A \mathcal{T} \triangleq \begin{bmatrix} {}^A R & {}^A P_B \\ 0 & 1 \end{bmatrix}.$$

We recall that the following equation holds for any given point  $P$ :

$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = {}^A \mathcal{T} \cdot \begin{bmatrix} {}^B P \\ 1 \end{bmatrix}$$

where  ${}^A P$  is the position of  $P$  with respect to frame  $\mathcal{F}_A$  and  ${}^B P$  is the position of  $P$  with respect to frame  $\mathcal{F}_B$ . For brevity’s sake, we shall abuse notation slightly and write  ${}^A P = {}^A \mathcal{T} \cdot {}^B P$  instead of the above when convenient.

### 3. Problem Formulation

We consider a compact workspace  $\mathcal{W} \subseteq \mathbb{R}^2$  enclosed by a static outer boundary  $\partial\mathcal{W}_0$  and  $N_o$  inner static boundaries  $\partial\mathcal{W}_i$ ,  $i \in \mathcal{I}_{N_o}$ , with  $N_o \in \mathbb{N}$ . More specifically, we assume that  $\mathcal{W}$  can be written as follows:

$$\mathcal{W} \triangleq \overline{\mathcal{O}_0} \setminus \bigcup_{i \in \mathcal{I}_{N_o}} \mathcal{O}_i. \tag{1}$$

where  $\mathcal{O}_0$  denotes the area that lies outside of  $\partial\mathcal{W}_0$  with  $\overline{\mathcal{O}_0} \triangleq \mathbb{R}^2 \setminus \mathcal{O}_0$ , and  $\mathcal{O}_i$  denotes the area enclosed by  $\partial\mathcal{W}_i$ , for all  $i \in \mathcal{I}_{N_o}$  (see Figure 1). We shall also use  $\overline{\mathcal{W}}$  to denote the complement of  $\mathcal{W}$  with respect to  $\mathbb{R}^2$ , i.e.,  $\overline{\mathcal{W}} \triangleq \mathbb{R}^2 \setminus \mathcal{W}$ , which is assumed to be closed. In addition, the workspace outer boundary  $\partial\mathcal{W}_0 \triangleq \partial\mathcal{O}_0$  and its inner boundaries  $\partial\mathcal{W}_i \triangleq \partial\mathcal{O}_i$ ,  $i \in \mathcal{I}_{N_o}$  are considered to be disjoint Jordan curves. Without loss of generality, we assume that  $\mathcal{W}$  is embedded with the arbitrarily positioned and oriented inertial frame  $\mathcal{F}_{\mathcal{W}}$ .

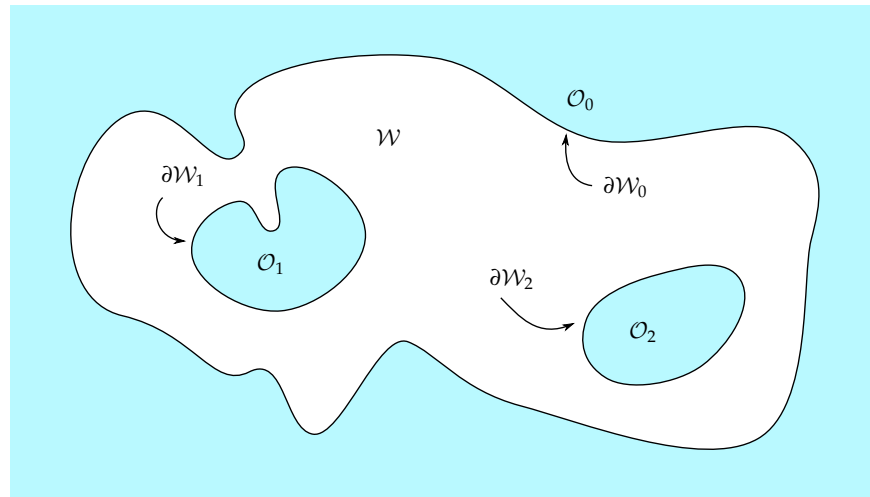


Figure 1. Typical workspace.

We now consider an object  $\mathcal{L} \subset \mathbb{R}^2$  whose body is a compact, closed, polygonal 2-manifold, able to translate and rotate freely within  $\mathcal{W}$  as long as it is not in contact with the workspace boundary. Let  $\mathcal{F}_{\mathcal{L}}$  be a fixed coordinate frame arbitrarily embedded in  $\mathcal{L}$ . We shall use  $p_{\mathcal{L}}$  and  $\theta_{\mathcal{L}}$  to denote the current position and orientation of  $\mathcal{L}$  with respect to  $\mathcal{F}_{\mathcal{W}}$ , i.e.:

$$p_{\mathcal{L}} \triangleq \begin{matrix} \{\mathcal{W}\} \\ \{\mathcal{L}\} \end{matrix} P_{\{\mathcal{L}\}} \quad R(\theta_{\mathcal{L}}) \triangleq \begin{matrix} \{\mathcal{W}\} \\ \{\mathcal{L}\} \end{matrix} R.$$

Object  $\mathcal{L}$  is considered a rigid body and let  $M_{\mathcal{L}}$ ,  $P_{\mathcal{L},com}$ ,  $I_{\mathcal{L}}$  denote the object’s mass, its center of mass, and its moment of inertia about  $P_{\mathcal{L},com}$ , respectively, expressed with respect to frame  $\mathcal{F}_{\mathcal{L}}$ . Assuming that  $P_{\mathcal{L},com}$  coincides with the origin of  $\mathcal{F}_{\mathcal{L}}$ , the dynamics of  $\mathcal{L}$  is given by:

$$\begin{aligned} M_{\mathcal{L}} \cdot \ddot{p}_{\mathcal{L}} &= \tau_{\mathcal{L},p} \\ I_{\mathcal{L}} \cdot \ddot{\theta}_{\mathcal{L}} &= \tau_{\mathcal{L},\theta} \end{aligned}$$

where  $\tau_{\mathcal{L},p} \in \mathbb{R}^2$  and  $\tau_{\mathcal{L},\theta} \in \mathbb{R}$  are the force and torque applied externally to the object. Lastly, we define  $\mathcal{L}(p, \theta)$  as the footprint of  $\mathcal{L}$ , i.e., the space of  $\mathcal{W}$  that the body of  $\mathcal{L}$  occupies when  $p_{\mathcal{L}} = p$  and  $\theta_{\mathcal{L}} = \theta$ .

In order to transport object  $\mathcal{L}$  from an initial configuration to a desired one, a team of  $N_{\mathcal{R}} \geq 2$  cooperating mobile manipulators is employed. More specifically, each robot  $\mathcal{R}_i$ ,  $i \in \mathcal{I}_{N_{\mathcal{R}}}$  consists of a holonomic base platform  $\mathcal{B}_i$  and a manipulator  $\mathcal{A}_i$  which is attached to the base and is equipped with an end-effector  $\mathcal{A}_{i,E}$  that rigidly grasps object  $\mathcal{L}$  at a specified point, and is thus able to exert a wrench onto it. The kinematics and dynamics of each mobile manipulator  $\mathcal{R}_i$ ,  $i \in \mathcal{I}_{N_{\mathcal{R}}}$  is described in detail in Section 3.1 and Section 3.2,

respectively. It is also assumed that the bodies of  $\mathcal{B}_i$  and  $\mathcal{A}_i$  can be described by compact, closed and connected 2-manifolds, for all  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ .

Thus, given an initial configuration  $q_{\mathcal{L},\text{init}} = [p_{\mathcal{L},\text{init}}^T, \theta_{\mathcal{L},\text{init}}]^T$  and a desired configuration  $q_{\mathcal{L},\text{goal}} = [p_{\mathcal{L},\text{goal}}^T, \theta_{\mathcal{L},\text{goal}}]^T$  for the object  $\mathcal{L}$ , our goal is to design a control scheme for the mobile manipulators  $\mathcal{R}_i$ ,  $i \in \mathcal{I}_{N_{\mathcal{R}}}$  which can drive the object to its destination, if a path between the two configurations exists, while ensuring that neither the object nor the robots will collide with the workspace boundary  $\partial\mathcal{W}$ . In addition, if the given problem is infeasible (i.e., no collision-free path connecting the given configurations exists) our control scheme should be able to conclude so in finite time.

### 3.1. Mobile Manipulator Kinematics

For each  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ , let  $\mathcal{F}_{\mathcal{B}_i}$  be a body-fixed frame arbitrarily embedded in  $\mathcal{B}_i$ . Without loss of generality, we assume that the origin of  $\mathcal{F}_{\mathcal{B}_i}$  coincides with the center of rotation of the base platform  $\mathcal{B}_i$ . For brevity's sake, let  $p_i$  and  $\theta_i$  denote the current position and orientation of  $\mathcal{F}_{\mathcal{B}_i}$  with respect to  $\mathcal{F}_{\mathcal{W}}$ , i.e.:

$$p_i \triangleq {}^{\{\mathcal{W}\}}P_{\{\mathcal{B}_i\}} \quad R(\theta_i) \triangleq {}^{\{\mathcal{W}\}}R_{\{\mathcal{B}_i\}}$$

Furthermore, we will use  $\mathcal{B}_i(p, \theta)$  to denote the footprint of the base platform of robot  $\mathcal{R}_i$  when it is centered at  $p$  with orientation  $\theta$ .

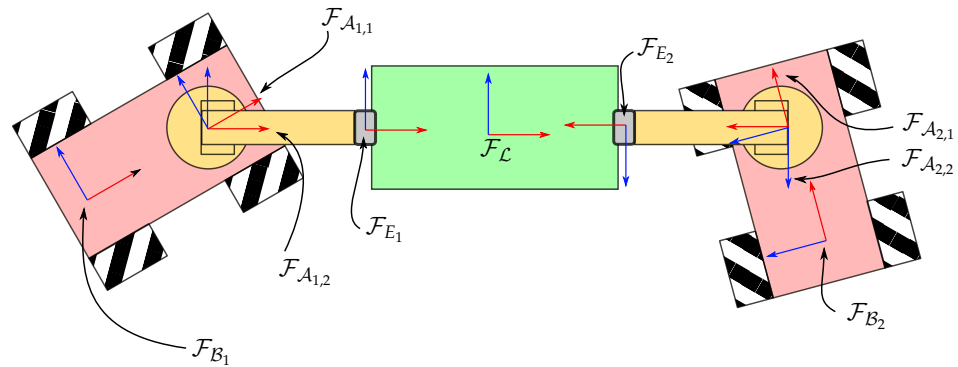
Regarding the manipulator  $\mathcal{A}_i$  affixed to robot  $\mathcal{R}_i$ , we assume that it consists of one or more links  $\mathcal{A}_{i,j}$ ,  $j \in N_{\mathcal{A}_i}$  which are connected such that they form an open chain. Furthermore, the first link  $\mathcal{A}_{i,1}$  is rigidly affixed to the base platform  $\mathcal{B}_i$ , whereas the end-effector is rigidly affixed to the last link  $\mathcal{A}_{i,N_{\mathcal{A}_i}}$ , for all  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ . The indexing of the remaining links of each manipulator is such that the body of link  $\mathcal{A}_{i,j+1}$  is able to either rotate or slide about the joint it shares with link  $\mathcal{A}_{i,j}$ . For each manipulator  $\mathcal{A}_i$ , we shall use  $q_{i,j}$  and  $\mathcal{D}_{q_{i,j}}$  to denote the state and domain, respectively, of the  $j$ -th degree of freedom, corresponding to the joint between links  $\mathcal{A}_{i,j}$  and  $\mathcal{A}_{i,j+1}$ , for all  $j \in \mathcal{I}_{N_{\mathcal{A}_i}-1}$  and  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ . We remark that each domain  $\mathcal{D}_{q_{i,j}}$  is a subset of either  $\mathbb{R}$  or  $\mathbb{S}^1$  depending on whether the joint is prismatic or revolute, respectively. The augmented state vector  $z_i$  of robot  $\mathcal{R}_i$  as follows:

$$z_i \triangleq [p_i^T, \theta_i, q_i^T]^T$$

where  $q_i$  is the stacked vector of joint states of manipulator  $\mathcal{A}_i$ , for all  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ . Similarly, for each  $i \in \mathcal{I}_{N_{\mathcal{R}}}$  and  $j \in \mathcal{I}_{N_{\mathcal{A}_i}}$ , let  $\mathcal{F}_{\mathcal{A}_{i,j}}$  be a body-fixed frame arbitrarily embedded in  $\mathcal{A}_{i,j}$ . Additionally, we affix an arbitrary coordinate frame  $\mathcal{F}_{E_i}$  at the point of contact between the end-effector of manipulator  $\mathcal{A}_i$  and the object  $\mathcal{L}$ . For the sake of simplicity and without harming generality, we assume herein that a) the origin of frame  $\mathcal{F}_{\mathcal{A}_{i,j+1}}$  lies on the axis of rotation or sliding of the  $j$ -th joint, and b) the origin of frame  $\mathcal{F}_{E_i}$  coincides with the corresponding contact point (see Figure 2).

Regarding each robot's forward kinematics, we shall use  $\mathcal{T}_{\mathcal{B}_i}(p, \theta)$  to denote the rigid transformation from  $\mathcal{F}_{\mathcal{B}_i}$  to  $\mathcal{F}_{\mathcal{W}}$  when the robot is placed at  $p$  with orientation  $\theta$ , i.e.,  ${}^{\{\mathcal{W}\}}P_{\{\mathcal{B}_i\}} = p$  and  ${}^{\{\mathcal{W}\}}R_{\{\mathcal{B}_i\}} = R(\theta)$ . Additionally, let  $\mathcal{T}_{\mathcal{A}_i}(q_i)$  be the forward kinematics of manipulator  $\mathcal{A}_i$ , i.e.,  $\mathcal{T}_{\mathcal{A}_i}(q_i) \triangleq {}^{\{\mathcal{A}_{i,1}\}}\mathcal{T}_{\{\mathcal{E}_i\}}$ . Since the manipulator of each robot is rigidly attached to its base, there exists a fixed homogeneous transformation, denoted by  $\mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i}$ , between the base  $\mathcal{B}_i$  and the manipulator's first link  $\mathcal{A}_{i,1}$ , i.e.:  $\mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i} \triangleq {}^{\{\mathcal{B}_i\}}\mathcal{T}_{\{\mathcal{A}_{i,1}\}}$ . The forward kinematics  $\mathcal{T}_{\mathcal{R}_i}(p, \theta, q)$  of robot  $\mathcal{R}_i$  is given by:

$$\mathcal{T}_{\mathcal{R}_i}(p, \theta, q) \triangleq \mathcal{T}_{\mathcal{B}_i}(p, \theta) \cdot \mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i} \cdot \mathcal{T}_{\mathcal{A}_i}(q).$$



**Figure 2.** Example of robotic system consisting of two mobile manipulators carrying a rectangular object. Each platform is equipped with a 2-link manipulator, which is able to rotate about the joint with the base.

Lastly, for each  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ , we shall use  $\mathcal{J}_{\mathcal{R}_i}$  and  $\mathcal{J}_{\mathcal{A}_i}$  to denote the Jacobian matrices of robot  $\mathcal{R}_i$  and its manipulator  $\mathcal{A}_i$ , i.e.,:

$$\begin{bmatrix} \{{W}\} \dot{p}_{\{E_i\}} \\ \{{W}\} \Omega_{\{E_i\}} \end{bmatrix} = \mathcal{J}_{\mathcal{R}_i}(z_i) \cdot z_i$$

and

$$\begin{bmatrix} \{\mathcal{B}_i\} \dot{p}_{\{E_i\}} \\ \{\mathcal{B}_i\} \Omega_{\{E_i\}} \end{bmatrix} = \mathcal{J}_{\mathcal{A}_i}(q_i) \cdot \dot{q}_i$$

where  $\Omega_{\{E_i\}}$  is the angular velocity of end-effector  $\mathcal{A}_{i,E}$ .

### 3.2. Mobile Manipulator Dynamics

The dynamics of each robot  $\mathcal{R}_i$ ,  $i \in \mathcal{I}_{N_{\mathcal{R}}}$  is assumed to obey the standard Euler–Lagrange model, i.e.,:

$$M_{\mathcal{R}_i}(z_i) \cdot \ddot{z}_i + C_{\mathcal{R}_i}(z_i, \dot{z}_i) \cdot \dot{z}_i + G_{\mathcal{R}_i}(z_i) = \tau_{m,i} - (\mathcal{J}_{\mathcal{R}_i}(z_i))^T \cdot \tau_{e,i} \tag{2}$$

where  $M_{\mathcal{R}_i}$ ,  $C_{\mathcal{R}_i}$ ,  $G_{\mathcal{R}_i} \in \mathbb{R}^{(3+N_{\mathcal{A}_i}) \times (3+N_{\mathcal{A}_i})}$  are the corresponding inertia, Coriolis and gravity matrices,  $\tau_{m,i} \in \mathbb{R}^{(3+N_{\mathcal{A}_i})}$  is the wrench applied by the robot’s actuators to the robot, and  $\tau_{e,i} \in \mathbb{R}^3$  is the wrench applied by the robot to the object  $\mathcal{L}$  via its end-effector.

## 4. Control Design

To address the aforementioned problem, we design a hybrid control scheme which consists of:

- (a) a high-level controller that given an initial configuration  $q_{\mathcal{L},\text{init}}$  and a final configuration  $q_{\mathcal{L},\text{goal}}$  configuration, can compute a sequence of reachable intermediate goals for the robotic system, if a solution to the above problem exists, or determine its infeasibility otherwise (completeness), and
- (b) a low-level controller which utilizes appropriate workspace transformations in order to drive the object and the mobile manipulators from each goal to the next while avoiding collisions with the workspace boundary (safety and convergence).

More specifically, the high-level controller, presented in Section 4.1, constructs a partitioning of the system’s configuration space into cells by adaptively subdividing the domain of the robotic system’s degrees of freedom until a sequence of connected cells containing  $q_{\mathcal{L},\text{init}}$  and  $q_{\mathcal{L},\text{goal}}$  is found (if one exists). Then, for each cell, intermediate goals for the object’s position  $p_{\mathcal{L}}$  and orientation  $\theta_{\mathcal{L}}$  are computed, as described in Section 4.2, and a suitable low-level control law is employed for driving the system to the corresponding goal configuration while ensuring forward invariance of the current configuration space cell.

#### 4.1. Configuration Space Decomposition

In this subsection, we present the hierarchical cell decomposition scheme that shall be employed for designing a sequence of high-level, feasible instructions that define a “path” leading to the desired configuration. Before doing so, we shall first take a closer look at the configuration space  $\mathcal{C}$  of the aforementioned robotic system. Throughout this subsection, we shall model this system as one virtual robot  $\mathfrak{R}$  consisting of  $N_{\mathfrak{R}} = 1 + \sum_{i \in \mathcal{I}_{N_{\mathfrak{R}}}} N_{\mathcal{A}_i}$  connected components, which correspond to the object  $\mathcal{L}$ , the base platform  $\mathcal{B}_i$  and the links  $\mathcal{A}_{i,j}$  of each mobile manipulator  $\mathcal{R}_i$ , for all  $j \in \mathcal{I}_{N_{\mathcal{A}_i}}$  and  $i \in \mathcal{I}_{N_{\mathfrak{R}}}$ .

One can readily see that the components of  $\mathfrak{R}$  form an undirected tree  $\mathfrak{T}(\mathfrak{n}, \mathfrak{e})$ , where  $\mathfrak{n}$  is the set of components and  $\mathfrak{e} \subset \mathfrak{n} \times \mathfrak{n}$  is the set of connections between the nodes. We shall use  $\mathfrak{R}_i$  to denote the  $i$ -th component of  $\mathfrak{R}$ . A connection  $(i, j) \in \mathfrak{e}$  implies that the  $j$ -th component is able to move (rotate, translate, slice) relative to the  $i$ -th component about a pivot point  $\mathfrak{P}_{i,j}$ . Furthermore, given  $i \in \mathcal{I}_{N_{\mathfrak{R}}}$ , we will use  $\mathfrak{n}_c^i$  to denote the children of component  $\mathfrak{R}_i$ , i.e., the set of components  $\mathfrak{R}_j$  such that  $(i, j) \in \mathfrak{e}$ , for all  $j \in \mathcal{I}_{N_{\mathfrak{R}}}$ . Moreover,  $\mathfrak{n}_p^i$  will be used to denote the parent  $\mathfrak{R}_j$  of component  $\mathfrak{R}_i$ , i.e., the sole component such that  $(j, i) \in \mathfrak{e}$ , if one exists. Accordingly, we define  $\mathfrak{n}_d^i$  and  $\mathfrak{n}_a^i$  as the set of descendants and ancestors, respectively, of component  $\mathfrak{R}_i$ . Without loss of generality, we can choose the indexing of the components such that the first component of  $\mathfrak{R}$  is the root of  $\mathfrak{T}$ , corresponding to the object  $\mathcal{L}$ . For simplicity's sake, we will use  $\mathcal{F}_{\mathfrak{R}_i}$ ,  $i \in \mathcal{I}_{N_{\mathfrak{R}}}$  to denote the coordinate frames embedded in each component of  $\mathfrak{R}$  and we shall refer to their origins as the reference point of the corresponding component, respectively. Furthermore, let  $p_{\mathfrak{R}} \triangleq [x_{\mathfrak{R}}, y_{\mathfrak{R}}]^T \in \mathbb{R}^2$  and  $\theta_{\mathfrak{R}} \in \mathcal{D}_{\theta} \subseteq \mathbb{S}^1$  denote the relative position of the robotic system's reference point and the relative orientation of its coordinate frame  $\mathcal{F}_{\mathfrak{R}_0}$  with respect to the workspace's coordinate frame  $\mathcal{F}_{\mathcal{W}}$ , respectively.

Regarding the coupling between components, we will refer to the joint between two connected components as prismatic (resp. revolutive) if the child is able to slide (resp. rotate) about the corresponding pivot point. We will use  $q_i$  and  $\mathcal{D}_{q_i}$ ,  $i \in \mathcal{I}_{N_q}$ , to denote the degree of freedom and its domain, respectively, corresponding to the joint between the  $i$ -th component and its parent, where  $N_q \triangleq \sum_{j \in \mathcal{I}_{N_{\mathfrak{R}}}} (N_{\mathcal{A}_j} - 1)$ . Without loss of generality, since each component other than the root has exactly one parent, we assume that each pivot  $\mathfrak{P}_{i,j}$  coincides with the origin of frame  $\mathcal{F}_{\mathfrak{R}_j}$ . Furthermore, by treating the orientation  $\theta_{\mathcal{L}}$  of the object as a virtual joint state, the state  $z$  of the virtual robotic system  $\mathfrak{R}$  is defined as follows:

$$z \triangleq [p_{\mathfrak{R}}^T, \theta_{\mathfrak{R}}, \mathfrak{q}^T]^T = [p_{\mathfrak{R}}^T, \mathfrak{q}^T]^T$$

where  $\mathfrak{q} \triangleq [q_i]_{i \in \mathcal{I}_{N_q}^*}$ , is the stacked vector of virtual joint parameters with  $q_0 \triangleq \theta_{\mathfrak{R}}$  and  $q_i \triangleq q_i$  for all  $i \in \mathcal{I}_{N_q}$ .

Let us now consider the footprint of the robotic system while it moves within the workspace. We notice that, for each  $i \in \mathcal{I}_{N_{\mathfrak{R}}}$ , the footprint of the individual component  $\mathfrak{R}_i$ , i.e., the area occupied by it at a given configuration, is defined by the position of its pivot point and the current value of its (virtual) joint parameter. We shall use  $\mathfrak{R}_i(p, q)$  to denote the footprint when the pivot point is placed at  $p$  and the joint parameter value is  $q$ . We also remark that, although each component may move freely with respect to its pivot point, any motion of theirs propagates directly to their children, thus potentially inducing a translation and/or rotation onto every one of its descendants  $\mathfrak{n}_d^i$ . Thus, the footprint of component  $\mathfrak{R}_i$  can also be defined in terms of the current position  $p_{\mathfrak{R}}$  of the robotic system and the (virtual) joint parameters of every component  $\mathfrak{R}_j$  belonging to  $\mathfrak{n}_d^i$ . By remarking that the footprint  $\mathfrak{R}(z)$  of the robotic system at a given configuration  $z$  is simply the union of the footprints of its individual components, i.e.,:

$$\mathfrak{R}(p_{\mathfrak{R}}, \mathfrak{q}) \triangleq \bigcup_{i \in \mathcal{I}_{N_{\mathfrak{R}}}} \mathfrak{R}_i \left( p_{\mathfrak{R}}, [q_j]_{j \in \mathfrak{n}_d^i}^T \right) \tag{3}$$



we are now ready to formally define the set of admissible configurations to our problem. For brevity,  $\mathfrak{R}(q)$  will be used instead of  $\mathfrak{R}(0, q)$  where is deemed preferable. By noticing that the configuration space  $\mathcal{C}$  of this robotic system is a manifold diffeomorphic to  $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathcal{D}_{q_1} \times \dots \times \mathcal{D}_{q_{N_q}}$  and recalling that neither the object  $\mathcal{L}$  nor any of the robots  $\mathcal{R}_i, i \in \mathcal{I}_{N_R}$  are allowed to collide with the workspace boundary  $\partial\mathcal{W}$ , the set  $\mathcal{C}_f$  of collision free configurations of  $\mathfrak{R}$  is given by:

$$\mathcal{C}_f = \{z \mid \mathcal{W} \cap \mathfrak{R}(z) = \emptyset \text{ and } z \in \mathcal{C}\}. \tag{4}$$

Finally, let  $\mathcal{C}_o \triangleq \mathcal{C} \setminus \mathcal{C}_f$ .

Now, in order to design a continuous “path” inside  $\mathcal{C}_f$  connecting the two given configurations  $q_{\mathcal{L},\text{init}}$  and  $q_{\mathcal{L},\text{goal}}$ , we extend the methodology presented in [38]. More specifically, by designing a suitable cover of the free configuration space via recursive subdivision of the domain of  $q$ , our goal is to obtain a hierarchical partitioning of  $\mathcal{C}_f$ . For each implicitly defined cell, we compute over suitable over- and under-approximations, whose shape is much simpler than the shape of the corresponding exact cell, which are used for both guiding the configuration space’s exploration, as well as designing a high-level plan will drive the robotic system to its goal. To do so, we first consider the domain  $\mathcal{D}_{q_i}$  of the joint state  $q_i$ , for  $i \in \mathcal{I}_{N_q}^*$ . Furthermore, we shall refer to a set of the form  $\mathcal{S}_{[q_{i1}, q_{i2}]}^{q_i}$  as a simple slice of the parameter  $q_i$ , where  $q_{i1}, q_{i2} \in \mathcal{D}_{q_i}$ . Furthermore, a set  $\mathfrak{S}_i = \{\mathcal{S}_j^{q_i} \mid j \in \mathcal{I}_{N_{\mathfrak{S}_i}}\}$  consisting of  $N_{\mathfrak{S}_i}$  simple slices of  $q_i$  shall be called a cover of  $\mathcal{D}_{q_i}$  if

$$\mathcal{D}_{q_i} = \bigcup_{j \in \mathcal{I}_{N_{\mathfrak{S}_i}}} \mathcal{S}_j^{q_i}$$

for all  $k, \ell \in \mathcal{I}_{N_{\mathfrak{S}_i}}$  with  $k \neq \ell$ . A compound slice  $\widehat{\mathcal{S}}$  is defined as a set of simple slices of the form  $\widehat{\mathcal{S}} = \{\mathcal{S}^{q_i} \mid i \in \mathcal{I}_{N_q}^*\}$ . Respectively, a set  $\widehat{\mathfrak{S}} = \{\mathfrak{S}_i \mid i \in \mathcal{I}_{N_q}^*\}$  is called a cover of the free configuration space  $\mathcal{C}_f$  if each  $\mathfrak{S}_i$  is a cover of  $\mathcal{D}_{q_i}$ . We note that a cover  $\widehat{\mathfrak{S}}$  induces a partitioning of  $\mathcal{C}_f$  into regions

$$\mathcal{C}_{\widehat{\mathfrak{S}}} = \{[p_{\mathfrak{R}}^T, q^T]^T \mid p_{\mathfrak{R}} \in \mathcal{W} \text{ and } q \in \widehat{\mathcal{S}}\} \cap \mathcal{C}_f, \quad \widehat{\mathcal{S}} \in \widehat{\mathfrak{S}} \tag{5}$$

each of which consists of  $N_{\widehat{\mathcal{S}}} \geq 0$  individually connected but pairwise disjoint subsets  $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}, i \in \mathcal{I}_{N_{\mathcal{C}_{\widehat{\mathfrak{S}}}}}$ . Taking a closer look at the connectedness of these cells, one can readily see that two configuration space cells  $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}$  and  $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},j}$  are connected iff  $\widehat{\mathcal{S}}_i, \widehat{\mathcal{S}}_j$  are adjacent and the projections of  $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}, \widehat{\mathcal{C}}_{\widehat{\mathcal{S}},j}$  onto the plane intersect. We recall that two distinct simple slices  $\mathcal{S}_i^{q_k}$  and  $\mathcal{S}_j^{q_k}$  are called adjacent if their intersection  $\mathcal{S}_i^{q_k} \cap \mathcal{S}_j^{q_k}$  is not empty, whereas two compound slices  $\widehat{\mathcal{S}}_i = \{\mathcal{S}_i^{q_k} \mid k \in \mathcal{I}_{N_q}^*\}$ , and  $\widehat{\mathcal{S}}_j = \{\mathcal{S}_j^{q_k} \mid k \in \mathcal{I}_{N_q}^*\}$ , are called adjacent if  $\mathcal{S}_i^{q_k}, \mathcal{S}_j^{q_k}$  are adjacent, for all  $k \in \mathcal{I}_{N_q}^*$ .

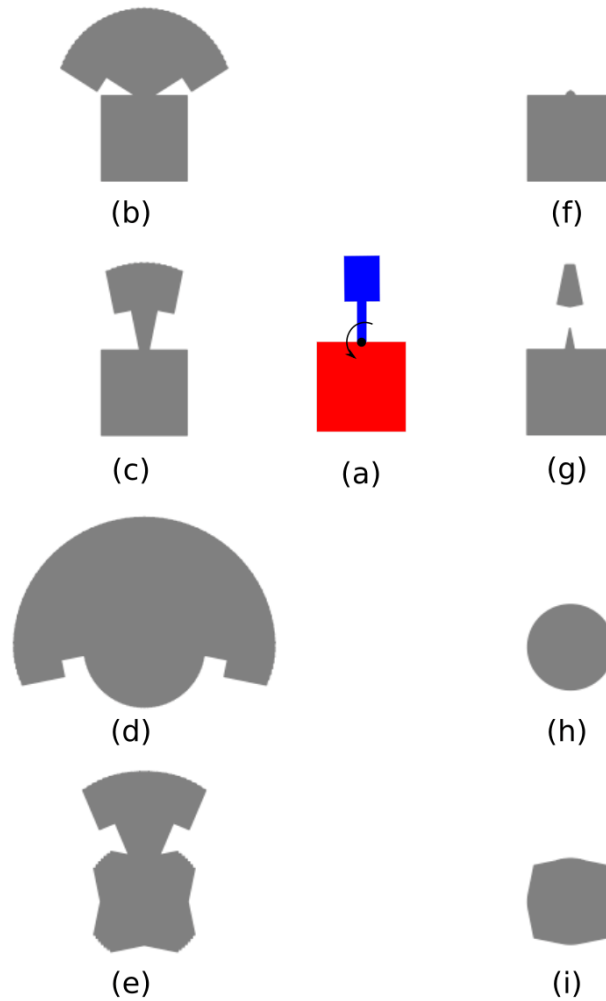
Similarly to the method employed in [38], in order to avoid explicitly computing the shape of a given configuration space cell, we shall define suitable over- and under-approximations of it, which, in addition, shall be used for guiding the configuration space exploration in a similar manner. In order to build these approximations of the set of free configurations corresponding to the compound slice  $\widehat{\mathcal{S}} = \{\mathcal{S}^{q_i} \mid i \in \mathcal{I}_{N_q}^*\}$ , we first compute an over-approximation  $\overline{\mathfrak{R}}(\widehat{\mathcal{S}})$  and an under-approximation  $\underline{\mathfrak{R}}(\widehat{\mathcal{S}})$  of the robotic system’s footprint as follows:

$$\begin{aligned} \overline{\mathfrak{R}}(\widehat{\mathcal{S}}) &= \bigcup_{q \in \mathcal{I}_{\widehat{\mathcal{S}}}} \mathfrak{R}(q) \\ \underline{\mathfrak{R}}(\widehat{\mathcal{S}}) &= \bigcap_{q \in \mathcal{I}_{\widehat{\mathcal{S}}}} \mathfrak{R}(q) \end{aligned} \tag{6}$$

where

$$\mathcal{I}_{\hat{\mathcal{S}}} = \mathcal{S}^{q_0} \times \mathcal{S}^{q_1} \times \dots \times \mathcal{S}^q. \tag{7}$$

We remark that, although seemingly daunting at first sight, the computation of  $\overline{\mathfrak{A}}(\hat{\mathcal{S}})$  and  $\underline{\mathfrak{A}}(\hat{\mathcal{S}})$  can be significantly simplified by recalling that the footprint of each component does not necessarily depends on every component of  $q$  but only on those of its ancestors, by virtue of the robotic system's tree-like structure. An example of such over- and under-approximation for a robotic system consisting of two connected components can be seen in Figure 3.



**Figure 3.** Over- and under-approximations of a robotic system (a) consisting of two components corresponding to slices:  $\{0\} \times [2\pi - 1, 1]$  (b,f),  $\{0\} \times [2\pi - 0.2, 0.2]$  (c,g),  $[3\pi/2, \pi/2] \times [2\pi - 0.2, 0.2]$  (d,h),  $[2\pi - 0.2, 0.2] \times [2\pi - 0.2, 0.2]$  (e,i).

Following these definitions, the over-approximation  $\overline{\mathfrak{C}}_{\hat{\mathcal{S}}}$  and the under-approximation  $\underline{\mathfrak{C}}_{\hat{\mathcal{S}}}$  of a given partition  $\mathfrak{C}_{\hat{\mathcal{S}}}$ , can be computed as follows:

$$\begin{aligned} \overline{\mathfrak{C}}_{\hat{\mathcal{S}}} &\triangleq \{ [p^T, q^T]^T \mid p \in \overline{\mathcal{W}}_{\hat{\mathcal{S}}} \text{ and } q \in \mathcal{I}_{\hat{\mathcal{S}}} \} \\ \underline{\mathfrak{C}}_{\hat{\mathcal{S}}} &\triangleq \{ [p^T, q^T]^T \mid p \in \underline{\mathcal{W}}_{\hat{\mathcal{S}}} \text{ and } q \in \mathcal{I}_{\hat{\mathcal{S}}} \} \end{aligned} \tag{8}$$

where

$$\begin{aligned} \overline{\mathcal{W}}_{\hat{\mathcal{S}}} &\triangleq \mathcal{W} \ominus \underline{\mathfrak{A}}(\hat{\mathcal{S}}) \\ \underline{\mathcal{W}}_{\hat{\mathcal{S}}} &\triangleq \mathcal{W} \ominus \overline{\mathfrak{A}}(\hat{\mathcal{S}}) \end{aligned} \tag{9}$$

with  $A \ominus B$  denoting the Minkowski difference of sets  $A$  and  $B$ , and  $\mathcal{I}_{\hat{\mathcal{S}}} = \mathcal{S}^{q_0} \times \mathcal{S}^{q_1} \times \dots \times \mathcal{S}^q$ . Obviously, each of  $\overline{\mathcal{C}}_{\hat{\mathcal{S}}}$  and  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$  consists of individually connected but pairwise disjoint cells  $\overline{\mathcal{C}}_{\hat{\mathcal{S}},i}, i \in \mathcal{I}_{N_{\overline{\mathcal{C}}_{\hat{\mathcal{S}}}}}$  and  $\underline{\mathcal{C}}_{\hat{\mathcal{S}},i}, i \in \mathcal{I}_{N_{\underline{\mathcal{C}}_{\hat{\mathcal{S}}}}}$ , respectively, which enclose or are enclosed by the cells of  $\mathcal{C}_{\hat{\mathcal{S}}}$ .

At this point, we remark that the approximation of  $\mathcal{C}_f$  improves as one subdivides the configuration space into more and finer slices. Thus, choosing a sufficiently fine partitioning of  $\mathcal{C}$ , a sequence of adjacent under-approximation cells connecting  $q_{\mathcal{L},\text{init}}$  and  $q_{\mathcal{L},\text{goal}}$  will appear, as long as one exists in the first place. Instead of choosing such a fine partitioning arbitrary, we design an adaptive subdivision scheme which makes also use of the space's over-approximations for choosing which slice to subdivide at each iteration. More specifically, we design an algorithm which given compound cover  $\hat{\mathcal{C}}$ , it tries to find a sequence  $\overline{\Pi}$  of adjacent under-approximation cells connecting the initial and goal configurations. If no such path can be found, then our algorithm tries to connect the two given configurations with a sequence  $\overline{\Pi}$  made of adjacent over-approximation cells instead. If such a path exists, then a slice corresponding to a cell of  $\overline{\Pi}$  is selected according to a suitable heuristic and becomes subdivided, producing a new partitioning of  $\mathcal{C}_f$ . Otherwise, if no such path can be found, then this obviously indicates that the problem at hand is infeasible (i.e., the two given configurations exist in disjoint components of the robotic system's configuration space) and our algorithm terminates. In short, one can readily verify that the following statements hold:

1. If there exists a path of adjacent under-approximation cells for a given cover  $\hat{\mathcal{C}}$  containing  $q_{\mathcal{L},\text{init}}$  and  $q_{\mathcal{L},\text{goal}}$ , then a solution to our problem exists.
2. If there exists a path of adjacent over-approximation cells for a given cover  $\hat{\mathcal{C}}$  containing  $q_{\mathcal{L},\text{init}}$  and  $q_{\mathcal{L},\text{goal}}$ , then whether our problem has a solution is unknown and further expansion of  $\hat{\mathcal{C}}$  is in order.
3. If there is no path of adjacent over-approximation cells for a given cover  $\hat{\mathcal{C}}$  containing  $q_{\mathcal{L},\text{init}}$  and  $q_{\mathcal{L},\text{goal}}$ , then our problem is infeasible.

The proposed algorithm can be seen in Algorithm 1. More specifically, we begin the configuration space exploration with a rough partitioning of  $\mathcal{C}$  induced by a compound slice covering the entire domain of virtual joint parameters  $q$ . Then, we search for cells  $\underline{\mathcal{C}}_{\text{init}}$  and  $\underline{\mathcal{C}}_{\text{goal}}$  containing the robot's initial and final configurations, respectively, by subdividing  $\hat{\mathcal{C}}$ . If no such pair of cells exists, our initial problem is obviously infeasible and the algorithm terminates. Otherwise, we try to connect  $\underline{\mathcal{C}}_{\text{init}}$  and  $\underline{\mathcal{C}}_{\text{goal}}$  using the available under-approximation cells corresponding to  $\hat{\mathcal{C}}$ . If this attempt fails, then we try instead to find a path of over-approximation cells connecting  $\overline{\mathcal{C}}_{\text{init}}$  and  $\overline{\mathcal{C}}_{\text{goal}}$ . If such a path cannot be found, this also implies that no solution exists and the algorithm terminates. Otherwise, a heuristic is utilized for selecting a compound slice in  $\hat{\mathcal{C}}$  to be expanded and the process starts anew. The heuristic used, which can be seen in Algorithm 2, selects which slice of  $\hat{\mathcal{C}}$  to expand as follows. Given a path  $\overline{\Pi}$  of over-approximation cells, it essentially tries to construct a path made of the under-approximation cells that belong in the same compound slices as the elements of  $\overline{\Pi}$ . Failing to connect under-approximation cells belonging in two adjacent compound slices  $\hat{\mathcal{S}}_i$  and  $\hat{\mathcal{S}}_j$  indicates that the connectedness of the over- and under-approximation cells in this slices is not the same, which means that these slices need to be further expanded. Thus, the largest simple slice of these compound slices becomes subdivided and the function returns. Finally, we remark that the functions CONNECTUACELLS and CONNECTOACELLS employ standard graph search algorithms for constructing the corresponding paths based on a heuristic that penalizes cells with smaller slices (i.e., cells corresponding to larger slices are preferred).

**Algorithm 1** Configuration space exploration algorithm.

---

```

function CONNECTCONFIGS( $q_{\mathcal{L},\text{init}}, q_{\mathcal{L},\text{goal}}$ )
   $\widehat{\mathcal{S}} \leftarrow \{ \mathcal{D}_{q_i} \mid i \in \mathcal{I}_{N_q}^* \}$ 
  loop
     $\widehat{\mathcal{S}}, \underline{\mathcal{C}}_{\text{init}}, \overline{\mathcal{C}}_{\text{init}} \leftarrow \text{FINDENCLOSINGCELLS}(q_{\mathcal{L},\text{init}}, \widehat{\mathcal{S}})$ 
     $\widehat{\mathcal{S}}, \underline{\mathcal{C}}_{\text{goal}}, \overline{\mathcal{C}}_{\text{goal}} \leftarrow \text{FINDENCLOSINGCELLS}(q_{\mathcal{L},\text{goal}}, \widehat{\mathcal{S}})$ 

    if  $\overline{\mathcal{C}}_{\text{init}}$  is Nil or  $\overline{\mathcal{C}}_{\text{goal}}$  is Nil then
      return Nil
    end if
     $\overline{\Pi} \leftarrow \text{CONNECTUACELLS}(\underline{\mathcal{C}}_{\text{init}}, \underline{\mathcal{C}}_{\text{goal}})$ 
    if  $\overline{\Pi}$  is Nil then
       $\overline{\Pi} \leftarrow \text{CONNECTOACELLS}(\overline{\mathcal{C}}_{\text{init}}, \overline{\mathcal{C}}_{\text{goal}})$ 
      if  $\overline{\Pi}$  is empty then
        return Nil
      else
         $\widehat{\mathcal{S}} \leftarrow \text{REFINE}(\widehat{\mathcal{S}}, \overline{\Pi}, \{\underline{\mathcal{C}}_{\text{init}}\}, \{\underline{\mathcal{C}}_{\text{goal}}\})$ 
      end if
    else if
      return  $\overline{\Pi}$ 
    end if
  end loop
end function

```

---

**Algorithm 2** Heuristic choosing next simple slice for subdivision.

---

```

function REFINE( $\widehat{\mathcal{S}}, \overline{\Pi}, \text{src}, \text{dst}$ )
  if  $\text{len}(\overline{\Pi}) = 1$  then
    return SUBDIVIDE( $\widehat{\mathcal{S}}, \overline{\Pi}[0]$ )
  else
     $\text{cells} \leftarrow \{ \}$ 
     $\text{connected} \leftarrow \{ \}$ 
    if  $\text{len}(\overline{\Pi}) == 2$  then
       $\text{cells} \leftarrow \text{dst}$ 
    else
       $\overline{\mathcal{C}}_{\widehat{\mathcal{S}}_j} \leftarrow \overline{\Pi}[1]$ 
      for all  $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$  in  $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$  do
         $\text{cells} \leftarrow \text{cells} \cup \{ \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j} \}$ 
      end for
    end if
    for all  $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}$  in  $\text{src}$  do
      for all  $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$  in  $\text{cells}$  do
        if  $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i} \cap \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j} \neq \emptyset$  then
           $\text{connected} \leftarrow \text{connected} \cup \{ \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j} \}$ 
        end if
      end for
    end for
    if  $\text{len}(\text{connected}) > 0$  then
      return REFINE( $\widehat{\mathcal{S}}, \overline{\Pi}[1 : ], \text{connected}, \text{dst}$ )
    else
      return SUBDIVIDELONGEST( $\widehat{\mathcal{S}}, \widehat{\mathcal{S}}_i, \widehat{\mathcal{S}}_j$ )
    end if
  end if
end function

```

---

#### 4.2. Distributed Control Law

Given now a path  $\Pi$  of cells obtained by the high-level planner described in the previous sub-section, we shall now design a distributed control scheme for the mobile manipulators that ensures safe transitions from one cell to the next until the goal configuration  $q_{\mathcal{L},\text{goal}}$  is reached. Let  $\mathcal{C}_{\hat{\mathcal{S}}}$  be a cell in  $\Pi$  and let  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$  denote its the projection on the plane. We recall that  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$  is an under-approximation of the actual free configuration space, constructed by extruding  $\underline{\mathcal{W}}_{\hat{\mathcal{S}}}$ , which implies that, as long as  $q \in \hat{\mathcal{S}}$ , then  $p_{\mathcal{L}}$  can safely occupy any position of  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$ . We also note that  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$  is a non-empty, compact region of  $\mathbb{R}^2$  with arbitrary connectedness and shape. Exploiting this fact, we can decouple the low-level control laws for: (a) the object's position  $p_{\mathcal{L}}$ , (b) the object's orientation  $\theta_{\mathcal{L}}$ , and (c) the joints  $q_i$  of each manipulator  $\mathcal{R}_i$ ,  $i \in \mathcal{I}_{N_{\mathcal{R}}}$ , as explained in the following.

For each intermediate cell of  $\Pi$ , we can obtain goal sets corresponding to  $p_{\mathcal{L}}$ ,  $\theta_{\mathcal{L}}$  and  $q$  separately, by computing its intersection with the next one (which is non-empty by construction of  $\Pi$ ). Let us consider a pair of consecutive cells  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$  and  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}$  in  $\Pi$ . Regarding the object's position, in order to safely traverse from  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$  to  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}$ , it is sufficient that  $p_{\mathcal{L}}$  reaches the set

$$\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \triangleq \underline{\mathcal{C}}_{\hat{\mathcal{S}}_i} \cap \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}. \quad (10)$$

We also note that  $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  is generally made of one or more disjoint subsets of arbitrary connectedness and that, as long as the object's position reaches either of these, the system can cross to the next cell. Respectively, a goal set corresponding to the object's orientation can be obtained by computing the intersection of the corresponding simple slices of  $\hat{\mathcal{S}}_i$  and  $\hat{\mathcal{S}}_j$ , i.e.,:

$$\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \triangleq \mathcal{S}_i^{q_0} \cap \mathcal{S}_j^{q_0}. \quad (11)$$

Goal sets for the joints of each mobile manipulator can be computed in a similar manner. Particularly, let  $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}})$  denote the projection  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$  along the dimensions corresponding to the degrees of freedom of  $\mathcal{A}_k$ . Obviously,  $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}})$  is equal to the product of the simple slices of  $\hat{\mathcal{S}}$  corresponding to  $q_k$ . Then, the corresponding goal set of  $q_k$  is given by

$$\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}) \cap \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}), \quad \forall k \in \mathcal{I}_{N_{\mathcal{R}}}. \quad (12)$$

Thus, for successfully driving the robotic system from  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$  to  $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}$ , we need to design decoupled control laws for the mobile manipulators which:

- ensure invariance of the current cell, i.e.,  $p_{\mathcal{L}} \in \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ ,  $\theta_{\mathcal{L}} \in \mathcal{S}_i^{q_0}$  and  $q_k \in \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ ,  $\forall k \in \mathcal{I}_{N_{\mathcal{R}}}$ , until the transition is complete, and
- ensure converge to the system's states to the corresponding goals sets  $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ ,  $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  and  $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ ,  $k \in \mathcal{I}_{N_{\mathcal{R}}}$ .

Finally, the transition is considered complete after all states have reached the corresponding goal sets. We remark that, regarding the last cell of  $\Pi$ , the goal sets corresponding to the object's position and orientation can taken equal to  $\{p_{\mathcal{L},\text{goal}}\}$  and  $\{\theta_{\mathcal{L},\text{goal}}\}$ , respectively, while the joints of the manipulators need only to remain within the bounds imposed by the last cell.

Before we proceed with formulating the corresponding control laws, we must first formally state the following assumptions about our system.

**Assumption 1.** Each robot  $\mathcal{R}_k$ ,  $k \in \mathcal{I}_{N_{\mathcal{R}}}$  has exact knowledge of the object's and its own dynamic model, i.e.,  $M_{\mathcal{L}}$ ,  $I_{\mathcal{L}}$ ,  $P_{\mathcal{L},\text{com}}$  and  $M_{\mathcal{R}_k}$ ,  $C_{\mathcal{R}_k}$ ,  $D_{\mathcal{R}_k}$ ,  $G_{\mathcal{R}_k}$  are known.

**Assumption 2.** Each robot  $\mathcal{R}_k$ ,  $k \in \mathcal{I}_{N_{\mathcal{R}}}$  has full knowledge of its own state  $z_k$  and the current configuration  $q_{\mathcal{L}}$  of the object  $\mathcal{L}$ .

**Assumption 3.** The plan generated by the high-level planner is available to all robots. Furthermore, each robot  $\mathcal{R}_k$ ,  $k \in \mathcal{I}_{N_{\mathcal{R}}}$  is able to communicate with the others only for announcing that it is ready to transition to the next cell, i.e., that  $p_{\mathcal{L}} \in \mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ ,  $\theta_{\mathcal{L}} \in \mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ , and  $q_k \in \mathcal{G}_{A_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  (lean communication).

**Assumption 4.** Each mobile manipulator  $\mathcal{R}_k$ ,  $k \in \mathcal{I}_{N_{\mathcal{R}}}$  is sufficiently redundant, i.e., it can independently apply a desired wrench to its end-effector while keeping  $q_k$  in  $\mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ . Additionally, the lower diagonal  $N_{A_k} - 1 \times N_{A_k} - 1$  block of  $(M_{\mathcal{R}_k})^{-1} (I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^{\dagger} \cdot \mathcal{J}_{\mathcal{R}_k}(z_k))$  is non-singular.

#### 4.2.1. Object’s Position

First, we shall design a suitable vector field for safely driving the object’s position  $p_{\mathcal{L}}$  to  $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ . To do so, we construct a transformation  $T_i$  of  $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}) \setminus \mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  to the unit disk and collapse the selected component of  $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  to a point, using the procedure described in [37]. By recalling that  $T_i$  is a diffeomorphism that collapses all inner obstacles of to isolated points, one can readily verify that the chance of a line connecting the image  $q_{\mathcal{L}}^{[i]} \triangleq T_i(p_{\mathcal{L}})$  of the object’s current position to the image  $q_{\mathcal{L},d}^{[i]}$  of the current cell’s goal is zero [39,40]. Therefore, the following velocity control law would safely drive the object’s position to the goal set for almost all initial configurations:

$$v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}) \triangleq (J_{T_i}(p_{\mathcal{L}}))^{-1} \cdot (q_{\mathcal{L},d}^{[i]} - q_{\mathcal{L}}^{[i]}) \tag{13}$$

where  $J_{T_i}$  is the Jacobian matrix of  $T_i$ . In order to design a law for the desired force to be applied to the object  $\mathcal{L}$  by the robots, we employ a novel methodology presented in [41] which allows us to extend the vector field from Equation (13) to second-order dynamics. The corresponding control law for the desired force applied to the object is formed by a term proportional to the error with respect to the reference governor’s state (to keep it small) plus a damping term to avoid oscillations, as follows:

$$\begin{aligned} \tau_{\mathcal{L},p}^{des} &= -M_{\mathcal{L}} \cdot (K_{p_{\mathcal{L}}}^{[i]} \cdot (p_{\mathcal{L}} - p_{\mathcal{L},G}^{[i]}) + \zeta_{p_{\mathcal{L}}}^{[i]} \cdot \dot{p}_{\mathcal{L}}) \\ \dot{p}_{\mathcal{L},G}^{[i]} &= K_{p_{\mathcal{L},G}}^{[i]} \cdot \frac{v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})}{\|v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})\|} \\ &\min \left( \|v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})\|, \sqrt{\Delta E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) / K_{p_{\mathcal{L}}}^{[i]}} \right) \end{aligned} \tag{14}$$

where

$$\Delta E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \triangleq K_{p_{\mathcal{L}}}^{[i]} \cdot d(p_{\mathcal{L},G}^{[i]}, \partial \mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})) - E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \tag{15}$$

$$E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \triangleq \frac{1}{2} \cdot (\|\dot{p}_{\mathcal{L}}\|^2 + K_{p_{\mathcal{L}}}^{[i]} \cdot \|p_{\mathcal{L}} - p_{\mathcal{L},G}^{[i]}\|^2) \tag{16}$$

$p_{\mathcal{L},G}^{[i]}$  is the (virtual) state of the reference governor,  $d(x, \mathcal{X})$  is the distance of  $x$  from the set  $\mathcal{X}$ ,  $K_{p_{\mathcal{L}}}^{[i]}$ ,  $K_{p_{\mathcal{L},G}}^{[i]}$  are fixed, positive gains and  $\zeta_{p_{\mathcal{L}}}^{[i]}$  is a virtual damping.

#### 4.2.2. Object’s Orientation

To drive the orientation  $\theta_{\mathcal{L}}$  of the object to the specified goal set  $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  while ensuring that it remains within  $\mathcal{S}_i^{q_0} = [\theta_i^{[i]}, \theta_i^{[i]}]$ , we design the desired torque  $\tau_{\mathcal{L},\theta}$  applied to the object based on the Prescribe Performance Control (PPC) methodology. We assume

that  $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$  is of the form  $[\theta_{l,G}^{[i]}, \theta_{u,G}^{[i]}]$ , which can be ensured by designing the partitioning scheme of the configuration space planner described in Section 4.1 such that the compound slices which form a valid cover are overlapping. We now define the following two performance functions:

$$\begin{aligned} \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}(t) &\triangleq \theta_{l,G}^{[i]} + \left(\theta_l^{[i]} - \theta_{l,G}^{[i]}\right) \cdot e^{-\lambda_{\theta_{\mathcal{L}}} \cdot t} \\ \bar{\rho}_{\theta_{\mathcal{L}}}^{[i]}(t) &\triangleq \theta_{u,G}^{[i]} + \left(\theta_u^{[i]} - \theta_{u,G}^{[i]}\right) \cdot e^{-\lambda_{\theta_{\mathcal{L}}} \cdot t} \end{aligned} \tag{17}$$

where  $t$  denotes the time and  $\lambda_{\theta_{\mathcal{L}}}$  is a positive constant. The corresponding control law is given by

$$\tau_{\mathcal{L},\theta}^{des} = I_{\mathcal{L}} \cdot \left( -K_{\theta_{\mathcal{L}},2}^{[i]} \cdot \left(\dot{\theta}_{\mathcal{L}} - v_{\theta_{\mathcal{L}}}^{[i]}\right) + \dot{v}_{\theta_{\mathcal{L}}}^{[i]} - a_{\theta_{\mathcal{L}}}^{[i]} \cdot \ln\left(\frac{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}}\right) \right) \tag{18}$$

$$v_{\theta_{\mathcal{L}}}^{[i]} \triangleq \frac{b_{\theta_{\mathcal{L}}}^{[i]} - K_{\theta_{\mathcal{L}},1}^{[i]} \cdot \ln\left(\frac{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}}\right)}{a_{\theta_{\mathcal{L}}}^{[i]}} \tag{19}$$

$$a_{\theta_{\mathcal{L}}}^{[i]} \triangleq \frac{1}{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}} + \frac{1}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \tag{20}$$

$$b_{\theta_{\mathcal{L}}}^{[i]} \triangleq \frac{\dot{\underline{\rho}}_{\theta_{\mathcal{L}}}^{[i]}}{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}} + \frac{\dot{\bar{\rho}}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}}$$

with  $K_{\theta_{\mathcal{L}},1}^{[i]}$  and  $K_{\theta_{\mathcal{L}},2}^{[i]}$  being positive gains. Notice that the logarithmic term attains a high positive or negative value, when the orientation of the object approaches the upper or lower performance function defined in (17), thus confining it strictly within them. Hence, the orientation never escapes the set of viable orientations of the cell and moreover converges to the set of orientations requested by the planner.

### 4.2.3. Manipulators

Considering now the control scheme for the mobile manipulators, we remark that, by virtue of Assumption 1 and Assumption 2 and assuming a common initialization policy for the virtual states of the reference governors corresponding to the object’s position and orientation, respectively, each robot is able to compute the desired total force  $\tau_{\mathcal{L},p}$  and torque  $\tau_{\mathcal{L},\theta}$  that should be applied to the object. Thus, the wrench  $\tau_{e,k}$  that each robot  $\mathcal{R}_k$ ,  $k \in \mathcal{I}_{N_{\mathcal{R}}}$  should apply to the object via its end-effector is given by

$$\tau_{e,k} = \frac{1}{N_{\mathcal{R}}} \cdot \left[ \begin{array}{c} \tau_{\mathcal{L},p} \\ \tau_{\mathcal{L},\theta} - (\tau_{\mathcal{L},p})^T \cdot {}_{\{E_k\}}P_{\mathcal{L},com}^{\perp} \end{array} \right] \tag{21}$$

where  ${}_{\{E_k\}}P_{\mathcal{L},com}^{\perp} = R\left(\frac{\pi}{2}\right) \cdot R(\theta_{\mathcal{L}}) \cdot {}_{\{E_k\}}P_{\mathcal{L},com}$  with  ${}_{\{E_k\}}P_{\mathcal{L},com}$  being the position of the object’s center of mass relative to the contact point of manipulator  $\mathcal{A}_k$ . Furthermore, each robot must also ensure that  $q_k \in \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$  while driving  $q_k$  to  $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ . To do so, we shall exploit the redundancy of each robot to design a force in the null-space of  $\mathcal{J}_{\mathcal{R}_k}$  which can ensure that the aforementioned specifications are met without affecting the force applied to the object. We now recall the dynamics of mobile manipulator  $\mathcal{R}_k$ :

$$M_{\mathcal{R}_k}(z_k) \cdot \ddot{z}_k + C_{\mathcal{R}_k}(z_k, \dot{z}_k) \cdot \dot{z}_k + G_{\mathcal{R}_k}(z_k) = \tau_{m,k} - (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \tag{22}$$

Assuming known dynamic parameters and state, we can design

$$\tau_{m,k} = C_{\mathcal{R}_k}(z_k, \dot{z}_k) \cdot \dot{z}_k + G_{\mathcal{R}_k}(z_k) + \tau_{m,k,1} + \left( I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) \cdot \tau_{m,k,2} \quad (23)$$

where  $\tau_{m,k,1}$  and  $\tau_{m,k,2}$  are new virtual inputs to be defined later and  $(\mathcal{J}_{\mathcal{R}_k})^\dagger$  denotes the pseudo-inverse of  $\mathcal{J}_{\mathcal{R}_k}$ . Substituting the above in Equation (22) yields:

$$M_{\mathcal{R}_k}(z_k) \cdot \ddot{z}_k = \tau_{m,k,1} + \left( I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) \cdot \tau_{m,k,2} - (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \quad (24)$$

We now consider the above dynamical model in the robot's task-space:

$$M'_{\mathcal{R}_k} \cdot \begin{bmatrix} \ddot{p}_{E_k} \\ \ddot{\theta}_{E_k} \end{bmatrix} + C'_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} = \left( \mathcal{J}_{\mathcal{R}_k}^T \right)^\dagger \cdot \tau_{m,k,1} - \tau_{e,k} \quad (25)$$

where  $\ddot{p}_{E_k}$  and  $\ddot{\theta}_{E_k}$  are the position and orientation of the corresponding end-effector's contact point and

$$\begin{aligned} M'_{\mathcal{R}_k} &= \left( \mathcal{J}_{\mathcal{R}_k}^T \right)^\dagger \cdot M_{\mathcal{R}_k} \cdot \left( \mathcal{J}_{\mathcal{R}_k} \right)^\dagger \\ C'_{\mathcal{R}_k} &= - \left( \mathcal{J}_{\mathcal{R}_k}^T \right)^\dagger \cdot M_{\mathcal{R}_k} \cdot \left( \mathcal{J}_{\mathcal{R}_k} \right)^\dagger \cdot \dot{\mathcal{J}}_{\mathcal{R}_k} \cdot \left( \mathcal{J}_{\mathcal{R}_k} \right)^\dagger. \end{aligned} \quad (26)$$

Let  $\begin{bmatrix} p_{E_k}^T & \theta_{E_k} \end{bmatrix}^T = \mathcal{T}_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}})$  be the rigid transformation between the positions and orientations of the corresponding points. It holds that

$$\begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} = J_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (27)$$

$$\begin{bmatrix} \ddot{p}_{E_k} \\ \ddot{\theta}_{E_k} \end{bmatrix} = J_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + \dot{J}_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (28)$$

with  $J_{\mathcal{L},E_k}$  denoting the Jacobian matrix of this rigid transformation. Therefore, Equation (25) can be re-written with respect to to the object's state as follows

$$M''_{\mathcal{R}_k} \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} = \left( \mathcal{J}_{\mathcal{R}_k}^T \right)^\dagger \cdot \tau_{m,k,1} - \tau_{e,k} \quad (29)$$

with

$$\begin{aligned} M''_{\mathcal{R}_k} &= M'_{\mathcal{R}_k} \cdot J_{\mathcal{L},E_k} \\ C''_{\mathcal{R}_k} &= C'_{\mathcal{R}_k} \cdot J_{\mathcal{L},E_k} + M'_{\mathcal{R}_k} \cdot \dot{J}_{\mathcal{L},E_k}. \end{aligned} \quad (30)$$

We notice that achieving our indented behavior, i.e., the object obeying the dynamics imposed by Equations (14) and (18) while distributing the load equally between the robots, is equivalent to

$$\tau_{e,k} = \frac{1}{N_{\mathcal{R}}} \left[ I_{\mathcal{L}} \cdot \ddot{\theta}_{\mathcal{L}} - M_{\mathcal{L}} \cdot (\dot{p}_{\mathcal{L}})^T \cdot {}_{\{E_k\}} P_{\mathcal{L},com}^\perp \right] = \frac{1}{N_{\mathcal{R}}} M_{\mathcal{L},E_k} \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (31)$$

where  $M_{\mathcal{L},E_k}$  is the fragment of the object's inertia, as perceived by the manipulator  $\mathcal{R}_k$ . Substituting the above into Equation (25) yields

$$\left( M''_{\mathcal{R}_k} + \frac{1}{N_{\mathcal{R}}} \cdot M_{\mathcal{L},E_k} \right) \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} = \left( \mathcal{J}_{\mathcal{R}_k}^T \right)^\dagger \cdot \tau_{m,k,1}. \quad (32)$$



As such, we can see that selecting

$$\tau_{m,k,1} \triangleq \mathcal{J}_{\mathcal{R}_k} \cdot \left( \left( M''_{\mathcal{R}_k} + \frac{1}{N_{\mathcal{R}}} \cdot M_{\mathcal{L},E_k} \right) \cdot \begin{bmatrix} \tau_{\mathcal{L},p}^{des} / M_{\mathcal{L}} \\ \tau_{\mathcal{L},\theta}^{des} / I_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \right) \quad (33)$$

will achieve the desired behavior, assuming all  $N_{\mathcal{R}}$  robots execute the same control law.

Considering again Equation (24), we shall now design  $\tau_{m,k,2}$  appropriately in order to satisfy the manipulator joint limit specifications. We recall that the projection of  $\tau_{m,k,2}$  with respect to  $I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^{\dagger} \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)$  has no effect on the wrench applied to the attached object. Now, let  $L_{A,k}, L_{B,k}, L_{C,k}, L_{D,k}$  be matrices such that  $L_{A,k} \in \mathbb{R}^{3 \times 3}$  and

$$L_k \triangleq M_{\mathcal{R}_k}^{-1} \cdot \left( I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^{\dagger} \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) = \begin{bmatrix} L_{A,k} & L_{B,k} \\ L_{C,k} & L_{D,k} \end{bmatrix}. \quad (34)$$

By recalling that  $L_{D,k}$  is assumed to be invertible according to Assumption 4, we employ the Prescribed Performance Control method along with back-stepping to design  $\tau_{m,k,2}$  as follows:

$$\tau_{m,k,2} \triangleq \begin{bmatrix} 0 \\ 0 \\ 0 \\ (L_{D,k})^{-1} \cdot v_{k,B} \end{bmatrix} \quad (35)$$

where

$$v_{k,B} \triangleq \begin{bmatrix} -(\dot{q}_{k,1} - v_{k,b,1}) + \dot{v}_{k,b,1} - a_{q_{k,1}} \cdot \ln \left( \frac{q_{k,1} - \underline{\rho}_{q_{k,1}}}{\bar{\rho}_{q_{k,1}} - q_{k,1}} \right) \\ -(\dot{q}_{k,1} - v_{k,b,1}) + \dot{v}_{k,b,2} - a_{q_{k,2}} \cdot \ln \left( \frac{q_{k,2} - \underline{\rho}_{q_{k,2}}}{\bar{\rho}_{q_{k,2}} - q_{k,2}} \right) \\ \vdots \\ -(\dot{q}_{k,N_{A_k}-1} - v_{k,b,N_{A_k}-1}) + \dot{v}_{k,b,N_{A_k}-1} - a_{q_{k,N_{A_k}-1}} \cdot \ln \left( \frac{q_{k,N_{A_k}-1} - \underline{\rho}_{q_{k,N_{A_k}-1}}}{\bar{\rho}_{q_{k,N_{A_k}-1}} - q_{k,N_{A_k}-1}} \right) \end{bmatrix} \quad (36)$$

$$v_{k,b,\ell} \triangleq \frac{b_{q_{k,\ell}} - \ln \left( \frac{q_{k,\ell} - \underline{\rho}_{q_{k,\ell}}}{\bar{\rho}_{q_{k,\ell}} - q_{k,\ell}} \right)}{a_{q_{k,\ell}}} \quad (37)$$

$$a_{q_{k,l}} \triangleq \frac{1}{q_{k,l} - \underline{\rho}_{q_{k,l}}} + \frac{1}{\bar{\rho}_{q_{k,l}} - q_{k,l}} \quad (38)$$

$$b_{q_{k,l}} \triangleq \frac{\dot{\underline{\rho}}_{q_{k,l}}}{q_{k,l} - \underline{\rho}_{q_{k,l}}} + \frac{\dot{\bar{\rho}}_{q_{k,l}}}{\bar{\rho}_{q_{k,l}} - q_{k,l}}$$

is the reference velocity control law,  $\underline{\rho}_{q_{k,\ell}}$  and  $\bar{\rho}_{q_{k,\ell}}$  are performance functions which smoothly “shrink”  $\mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\hat{S}_i})$  to  $\mathcal{G}_{A_k}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j})$ , given by

$$\begin{aligned} \underline{\rho}_{q_{k,\ell}}(t) &\triangleq \bar{q}_{G,k,\ell} + (\bar{q}_{k,\ell} - \bar{q}_{G,k,\ell}) \cdot e^{-\lambda_q \cdot t} \\ \bar{\rho}_{q_{k,\ell}}(t) &\triangleq \underline{q}_{G,k,\ell} + (\underline{q}_{k,\ell} - \underline{q}_{G,k,\ell}) \cdot e^{-\lambda_q \cdot t} \end{aligned} \quad (39)$$

with  $\bar{q}_{k,\ell}, \underline{q}_{k,\ell}$  and  $\bar{q}_{G,k,\ell}, \underline{q}_{G,k,\ell}$  being the lower and upper bounds of the joint parameters of  $A_k$  corresponding to  $\mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\hat{S}_i})$  and  $\mathcal{G}_{A_k}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j})$ , respectively. Similar to the orientation control design, the input control signal  $\tau_{m,k,2}$  was designed to constrain the evolution of the manipulators state within the corresponding upper and lower performance functions to enforce the necessary safety and convergence properties.

### 5. Stability Analysis

In this section, we provide an analysis of the robotic system’s stability properties under the proposed control scheme.

**Proposition 1.** (Safety). *Given two adjacent under-approximation cells  $\underline{\mathcal{C}}_{\hat{S}_i}$  and  $\underline{\mathcal{C}}_{\hat{S}_j}$ , the object’s configuration will asymptotically converge to  $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j}) \times \mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j})$  for almost all initial configurations under control laws in Equations (14) and (18). Furthermore, the set  $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j}) \times \mathcal{S}_i^{q_0}$  is invariant.*

**Proof.** We begin this proof by first recalling that, as long as object’s orientation and robot joints remain within the bounds imposed by  $\underline{\mathcal{C}}_{\hat{S}_i}$ , control of the object’s position and orientation can be safely decoupled. Regarding the object’s position, one can readily verify that since  $T_i$  is a diffeomorphism in  $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j})$  (see [37]), the reference velocity control law  $i$  is Lipschitz, has exactly one critical point which is located at the transformed goal configuration and is inward pointing at the outer boundary of  $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j})$ . Then, by invocation of Theorem 2 in [41], the control law in Equation (14) ensures invariance of cell and convergence to the goal set of  $p_{\mathcal{L}}$  for almost all initial configurations.

Regarding the object’s orientation, we define the following coordinate transformation:

$$\begin{aligned} z_{1,i} &= \ln \left( \frac{\theta_{\mathcal{L}} - \rho_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \right) \\ z_{2,i} &= \dot{\theta}_{\mathcal{L}} - v_{\theta_{\mathcal{L}}}^{[i]} \end{aligned} \tag{40}$$

and consider the following Lyapunov candidate:

$$V = \frac{1}{2} \cdot z_{1,i}^2 + \frac{1}{2} \cdot z_{2,i}^2. \tag{41}$$

The time derivatives of  $z_{1,i}$  and  $z_{2,i}$  are given by

$$\begin{aligned} \dot{z}_{1,i} &= a_{\theta_{\mathcal{L}}}^{[i]} \cdot z_{2,i} + a_{\theta_{\mathcal{L}}}^{[i]} \cdot v_{\theta_{\mathcal{L}}}^{[i]} - b_{\theta_{\mathcal{L}}}^{[i]} \\ \dot{z}_{2,i} &= \frac{\tau_{\mathcal{L},\theta}}{I_{\mathcal{L}}} - \dot{v}_{\theta_{\mathcal{L}}}^{[i]} \end{aligned} \tag{42}$$

Thus, computing the derivative of  $V$  with respect to time yields

$$\begin{aligned} \dot{V} &= z_{1,i} \cdot \dot{z}_{1,i} + z_{2,i} \cdot \dot{z}_{2,i} \\ &= a_{\theta_{\mathcal{L}}}^{[i]} \cdot z_{1,i} \cdot z_{2,i} + z_{1,i} \cdot \left( a_{\theta_{\mathcal{L}}}^{[i]} \cdot v_{\theta_{\mathcal{L}}}^{[i]} - b_{\theta_{\mathcal{L}}}^{[i]} \right) + z_{2,i} \cdot \left( \frac{\tau_{\mathcal{L},\theta}}{I_{\mathcal{L}}} - \dot{v}_{\theta_{\mathcal{L}}}^{[i]} \right) \end{aligned} \tag{43}$$

Noting that  $a_{\theta_{\mathcal{L}}}^{[i]} \cdot v_{\theta_{\mathcal{L}}}^{[i]} - b_{\theta_{\mathcal{L}}}^{[i]} = -K_{\theta_{\mathcal{L},1}}^{[i]} z_{1,i}^2$  and substituting the control law for  $\tau_{\mathcal{L},\theta}$  to the above, we obtain

$$\dot{V} = - \left( K_{\theta_{\mathcal{L},1}}^{[i]} \cdot z_{1,i}^2 + K_{\theta_{\mathcal{L},2}}^{[i]} \cdot z_{2,i}^2 \right). \tag{44}$$

Since  $\dot{V}$  is negative definite, assuming that the initial value of  $\theta_{\mathcal{L}}$  lies within the specified bounds, the proposed control law ensures that  $\mathcal{S}_i^{q_0}$  remains invariant and that  $\theta_{\mathcal{L}}$  will asymptotically converge to  $(\theta_{l,G}^{[i]} + \theta_{u,G}^{[i]})/2$ . □

**Proposition 2.** (Safety). *Given two adjacent under-approximation cells  $\underline{\mathcal{C}}_{\hat{S}_i}$  and  $\underline{\mathcal{C}}_{\hat{S}_j}$ , under the control law Equation (23), the joint states  $q_k$  of mobile manipulator  $\mathcal{R}_k$  will converge to  $\mathcal{G}_{A_k}(\underline{\mathcal{C}}_{\hat{S}_i}, \underline{\mathcal{C}}_{\hat{S}_j})$ . Furthermore, the set  $\mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\hat{S}_i})$  is invariant.*

**Proof.** We consider once again Equation (24). Since  $M_{\mathcal{R}_k}$  is an inertia matrix, we know that its inverse exists, thus multiplying both sides with  $(M_{\mathcal{R}_k})^{-1}$  and substituting Equation (35) yields:

$$\begin{aligned} \dot{z}_k &= (M_{\mathcal{R}_k}(z_k))^{-1} \cdot \left( I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) \cdot \tau_{m,k,2} + \\ &\quad (M_{\mathcal{R}_k}(z_k))^{-1} \cdot \tau_{m,k,1} - (M_{\mathcal{R}_k}(z_k))^{-1} \cdot (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \\ &= \begin{bmatrix} L_{B,k} \cdot (L_{D,k})^{-1} \\ I \end{bmatrix} \cdot v_{k,B} + (M_{\mathcal{R}_k}(z_k))^{-1} \cdot \tau_{m,k,1} - (M_{\mathcal{R}_k}(z_k))^{-1} \cdot (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \end{aligned} \tag{45}$$

We note that the term  $\tau_{m,k,2}$ , which is designed to ensure satisfaction of joint parameter specifications, has no effect on the stability properties involving the object’s position  $p_{\mathcal{L}}$  and orientation  $\theta_{\mathcal{L}}$  by virtue of  $I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)$ . Therefore, the last two r.h.s. terms of Equation (45) are bounded by design and vanish as the object approaches the specified configuration corresponding to the current cell. As such, the dynamics of the joint parameters can be written as:

$$\dot{q}_k = v_{k,B} + w_B \tag{46}$$

where the term  $w_B$  corresponding to  $\tau_{m,k,1}$  and  $\tau_{e,k}$  and can be viewed as a bounded and vanishing disturbance. We now define the following coordinate transformation for each joint value  $q_{k,\ell}$ ,  $\ell \in \mathcal{J}_{N_{A_k}-1}$ :

$$\begin{aligned} z_{q,1,\ell} &= \ln \left( \frac{q_{k,\ell} - \underline{\rho}_{q_{k,\ell}}}{\bar{\rho}_{q_{k,\ell}} - q_{k,\ell}} \right) \\ z_{q,2,\ell} &= \dot{q}_{k,\ell} - v_{k,b,\ell} \end{aligned} \tag{47}$$

and consider the Lyapunov candidate

$$V_\ell = \frac{1}{2} \cdot z_{q,1,\ell}^2 + \frac{1}{2} \cdot z_{q,2,\ell}^2. \tag{48}$$

Following the same procedure as above, we derive that

$$\begin{aligned} \dot{V}_\ell &= - \left( z_{q,1,\ell}^2 + z_{q,2,\ell}^2 \right) + z_{q,2,\ell} \cdot w_{B,\ell} \\ &\leq -z_{q,1,\ell}^2 - z_{q,2,\ell}^2 + |z_{q,2,\ell}| \cdot |w_{B,\ell}| \\ &\leq -z_{q,1,\ell}^2 - |z_{q,2,\ell}| \cdot |w_{B,\ell}| + (w_{B,\ell})^2 \end{aligned} \tag{49}$$

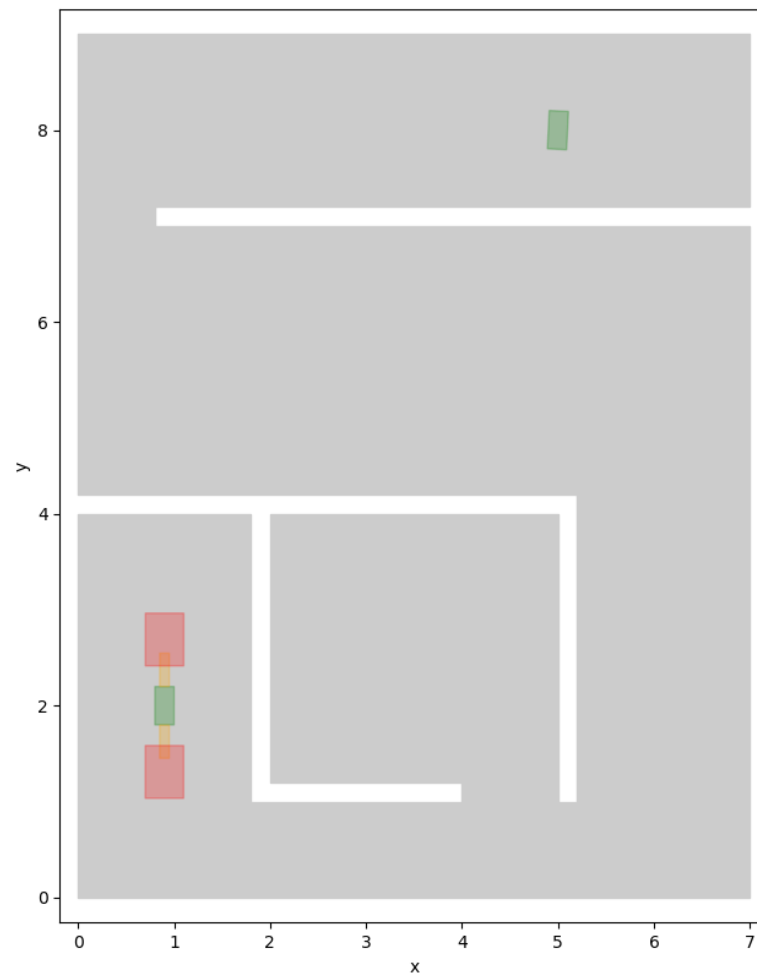
which implies that  $z_{q,1,\ell}, z_{q,2,\ell}$  and the control law are globally uniformly bounded (Lemma 2.28 [42]), and, thus, concludes the proof.  $\square$

**Theorem 1.** (Convergence). *The robotic system under the distributed control law in Equation (23) will successfully drive the object  $\mathcal{L}$  to its goal configuration  $q_{\mathcal{L},\text{init}}$ , from almost all initial configurations.*

**Proof.** First, we note that, by virtue of Equation (4) and the design of Equation (23), the total force and torque applied to the object’s center of mass by the robotic system is equal to the desired ones specified by Equation (14) and Equation (18), since the remaining terms either cancel the robot’s dynamics or are projected along the kernel space of  $\mathcal{J}_{\mathcal{R}_k}$ , respectively. As such, according to Proposition 1, the object is guaranteed to traverse from one cell to another till it arrives to the desired configuration  $q_{\mathcal{L},\text{goal}}$ , starting from almost any initial configuration  $q_{\mathcal{L},\text{init}}$ , as long as the robots do not collide with the workspace boundary. However, according to Proposition 2, the configurations of the mobile manipulators remain within the bounds specified by the under-approximation cell  $\underline{\mathcal{C}}_{\hat{S}_i}$ , which, by design of the high-level planner, implies that the robotic system’s footprint cannot intersect the workspace’s boundary.  $\square$

## 6. Simulation Results

To demonstrate the efficacy of the proposed control scheme, we consider a robotic system consisting of two mobile manipulators holding a rectangular object within the workspace depicted in Figure 4. The robotic system was initialized at  $q_{\mathcal{L},\text{init}} = [0.9, 2, 1.57]^T$  and  $q_{1,1} = q_{2,1} = 0$  whereas the desired configuration of the object was set to  $q_{\mathcal{L},\text{goal}} = [5, 8, 4.663]^T$ . The intervals for the object's orientation and robot joints generated by the high-level planner can be seen in Table 1, whereas the control parameters selected during this simulation are given in Table 2. Notice that our planner extracted a viable sequence of configuration cells despite the fact that the feasible configuration space becomes very narrow particularly when the robotic systems has to transverse a corner, thus verifying the completeness of our approach. Figure 5 shows the trajectory executed by the robotic system under the proposed control law, whereas plots of the object's position, orientation and corresponding rates can be seen in Figure 6, Figure 7, Figure 8, and Figure 9, respectively. It should be noted that the transition between successive cells is executed by the proposed low level control algorithms without harming either the safety or the convergence properties. Accordingly, Figures 10 and 11 show the evolution of each manipulator's state, as well as the computed lower and upper bounds corresponding to each cell. The total force and torque applied to the object is also displayed in Figure 12 and Figure 13, respectively. As one can verify from the aforementioned figures, the robotic system successfully reaches the goal configuration while satisfying the specifications corresponding to  $\theta_{\mathcal{L}}, q_{1,1}, q_{2,1}$ . A video of the aforementioned transportation task can be found in the following url: [https://youtu.be/AQ\\_8z3tysRo](https://youtu.be/AQ_8z3tysRo) (accessed on 6 December 2022).



**Figure 4.** Initial and goal configuration of the robotic system and object, respectively.

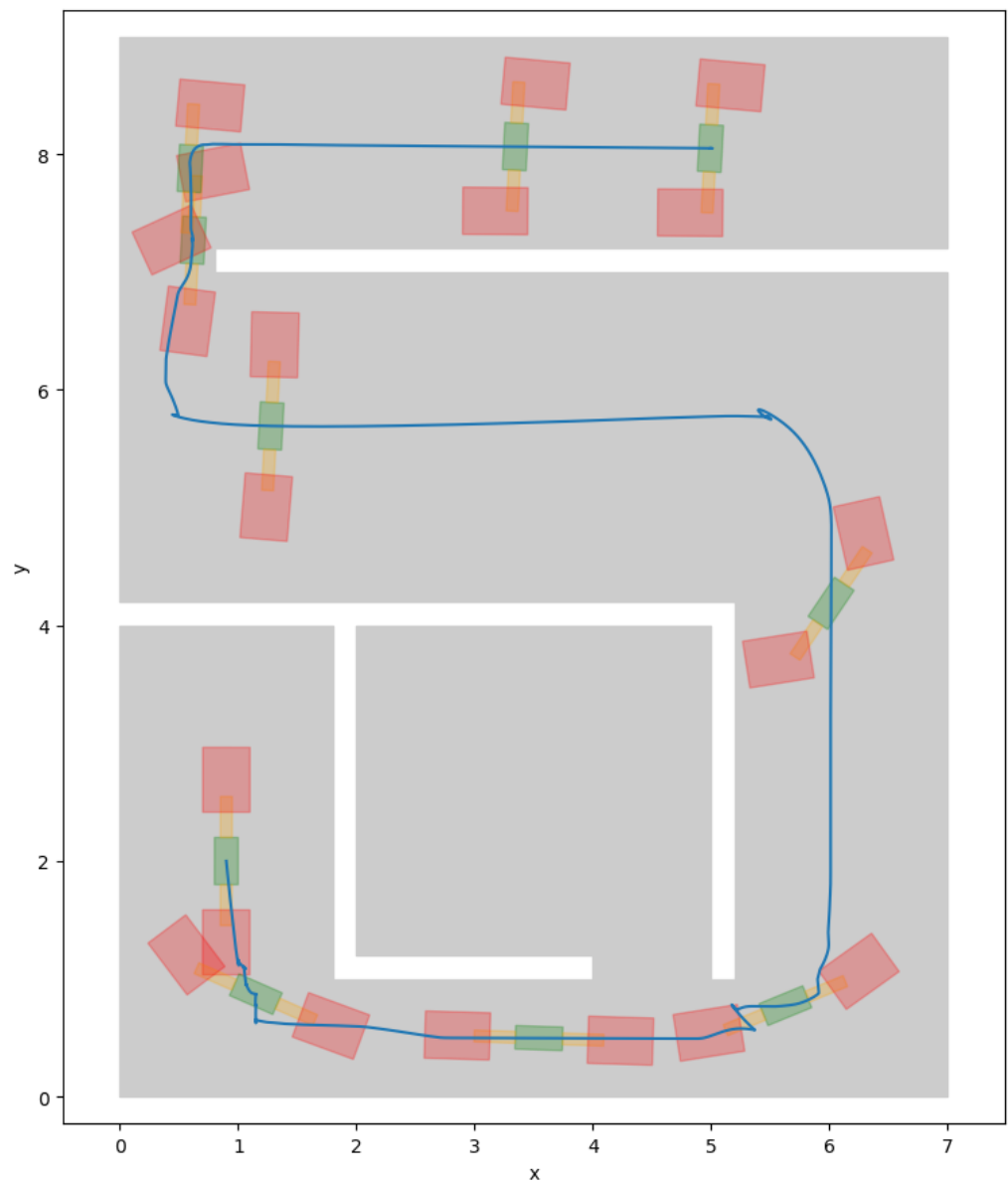
Notice that the simulated workspace consists of both narrow and wide areas, which results in both situations that the robotic system needs to undergo major reconfiguration and situations where the system can navigate without the need to alter its configuration, demonstrating the adaptive nature of the algorithm. Adding more than two robots in this example would render the overall space around the object overcrowded by the robots carrying it, thus limiting its flexibility and not demonstrating the searching capabilities of the proposed algorithm with respect to the shape of the robotic system that leads to feasible paths. In other words, when multiple robots, grasping the object uniformly at its boundary, are adopted, the expected motion of each robot with respect to the object would be very constrained in order to avoid collisions with neighboring robots, i.e., the robotic system would travel as a rigid formation. Consequently, we selected to demonstrate the most reconfigurable case in order to show how the proposed algorithm seeks and finds viable configurations in the workspace that avoid inter-robot collisions and collisions with the environment and fulfill the transportation task.

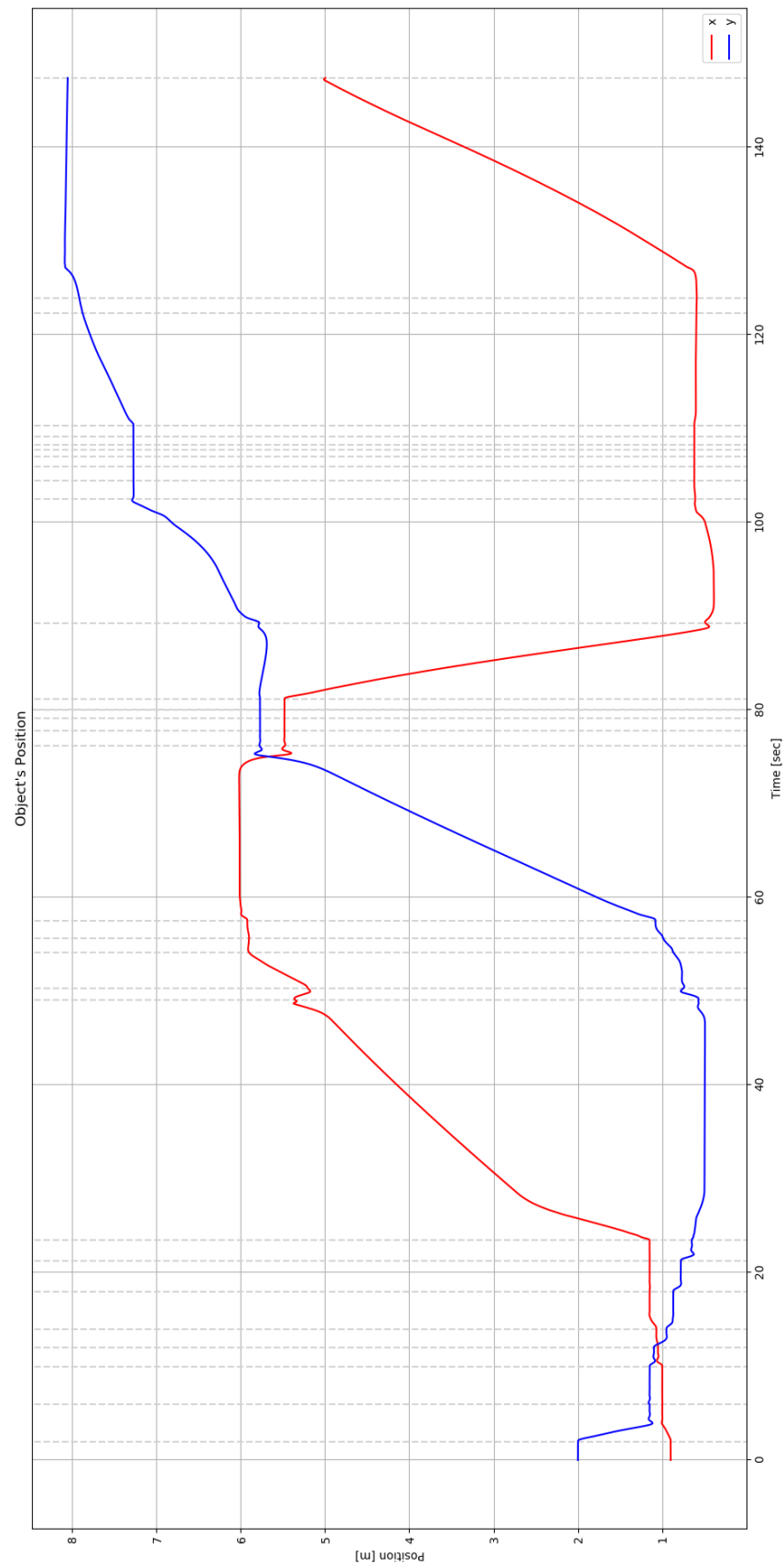
**Table 1.** Lower and upper bounds of the intervals corresponding to each cell, as generated by the planner.

Cell ID	$\theta_l$	$\theta_u$	$\bar{q}_{1,1}$	$\underline{q}_{1,1}$	$\bar{q}_{2,1}$	$\underline{q}_{2,1}$
1	1.470	1.867	-0.100	0.567	-0.100	0.567
2	1.667	2.065	-0.100	0.567	-0.567	0.100
3	1.863	2.456	-0.100	0.567	-1.035	0.367
4	1.863	2.456	-0.100	0.333	-1.870	-0.835
5	2.256	2.652	-0.100	0.333	-1.503	-0.835
6	2.452	2.849	-0.100	0.333	-1.035	-0.368
7	2.649	2.947	-0.100	0.217	-0.567	0.100
8	2.747	3.045	-0.050	0.108	-0.050	0.108
9	2.895	3.191	-0.050	0.108	-0.050	0.108
10	3.091	3.388	-0.050	0.158	-0.158	0.050
11	3.287	3.584	0.067	0.284	-0.284	-0.067
12	3.484	3.781	0.184	0.518	-0.518	-0.184
13	3.681	3.977	0.418	0.985	-0.985	-0.418
14	3.877	4.172	0.651	0.984	-0.984	-0.651
15	4.072	4.370	0.418	0.985	-0.985	-0.418
16	4.270	4.576	0.184	0.518	-0.518	-0.184
17	4.476	4.762	0.067	0.284	-0.284	-0.067
18	4.564	4.762	0.067	0.284	-0.284	-0.067
19	4.564	4.762	-0.050	0.108	-0.108	0.050
20	4.564	4.762	-0.050	0.108	-0.108	0.050
21	4.564	4.762	-0.985	0.050	-0.108	0.050
22	4.564	4.762	-1.453	-0.885	-0.108	0.050
23	4.564	4.762	-1.687	-1.353	-0.108	0.050
24	4.564	4.762	-1.687	-1.528	-0.108	-0.050
25	4.564	4.762	-1.687	-1.528	-0.168	-0.008
26	4.564	4.762	-1.687	-1.528	-0.284	-0.067
27	4.564	4.762	-1.687	-1.528	-0.518	-0.184
28	4.564	4.762	-1.687	-1.528	-0.985	-0.418
29	4.564	4.762	-1.687	-1.528	-1.453	-0.885
30	4.564	4.762	-1.687	-1.528	-1.687	-1.353

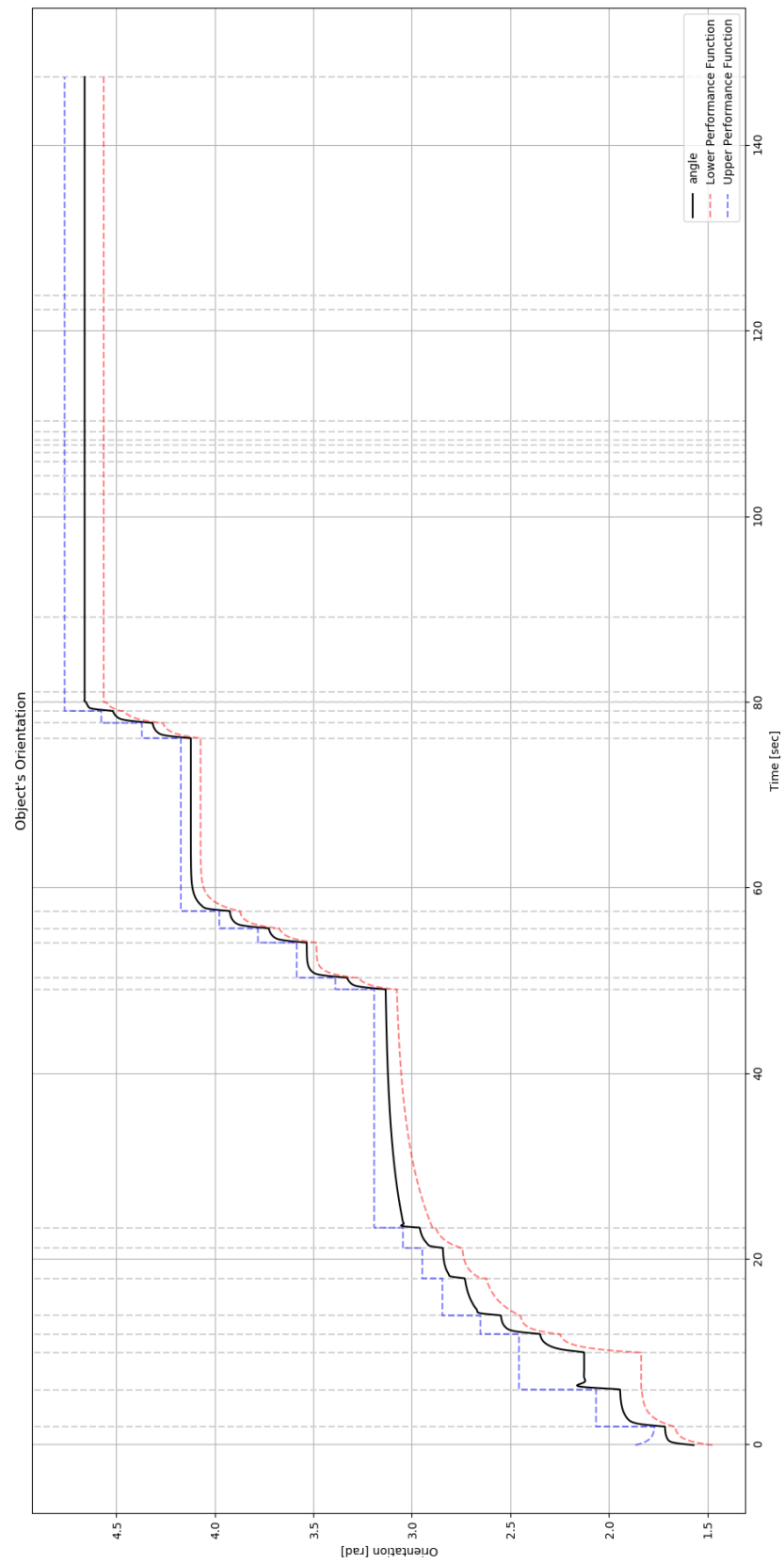
**Table 2.** Simulation parameters.

Parameter	Value
$M_{\mathcal{L}}$	1 kg
$I_{\mathcal{L}}$	1 kg m <sup>2</sup>
$K_{p_{\mathcal{L}}}$	50
$\zeta_{p_{\mathcal{L}}}$	5
$K_{p_{\mathcal{L},G}}$	5
$\lambda_{\theta_{\mathcal{L}}}$	2
$K_{\theta_{\mathcal{L},1}}$	1
$K_{\theta_{\mathcal{L},2}}$	1
$\lambda_q$	2

**Figure 5.** Path executed by the robotic system during the simulations (blue line), as well as the footprint of the robotic system at various instants.



**Figure 6.** Evolution of the object's position  $p_{\mathcal{L}}$  over time. The vertical dashed lines indicate transitions between consecutive cells.



**Figure 7.** Evolution of the object's orientation  $\theta_L$  over time (solid line), as well as the corresponding performance functions  $\underline{\rho}_{\theta_L}$  and  $\bar{\rho}_{\theta_L}$ .



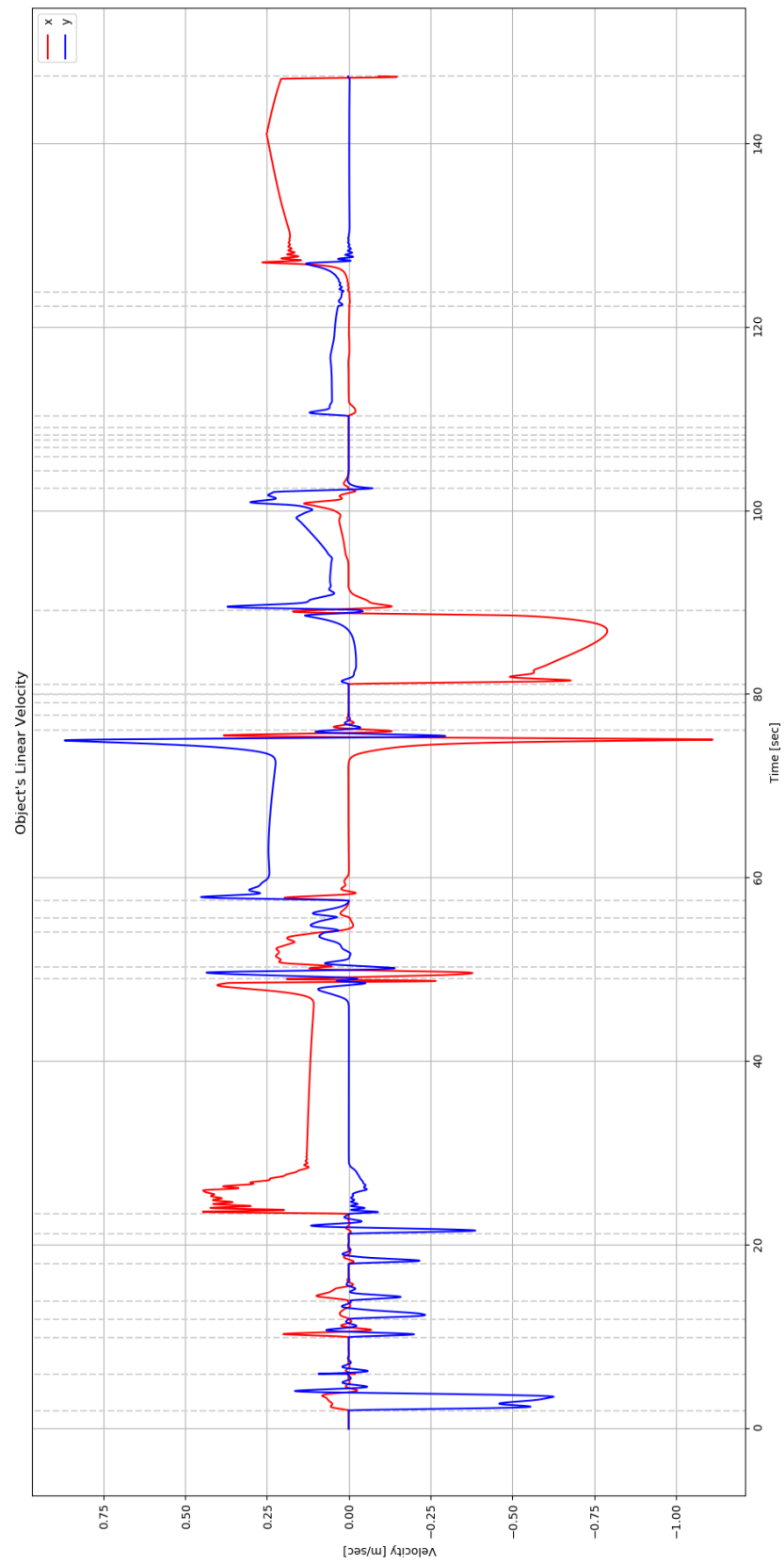


Figure 8. Object's linear velocity  $\dot{p}_{\mathcal{L}}$ .

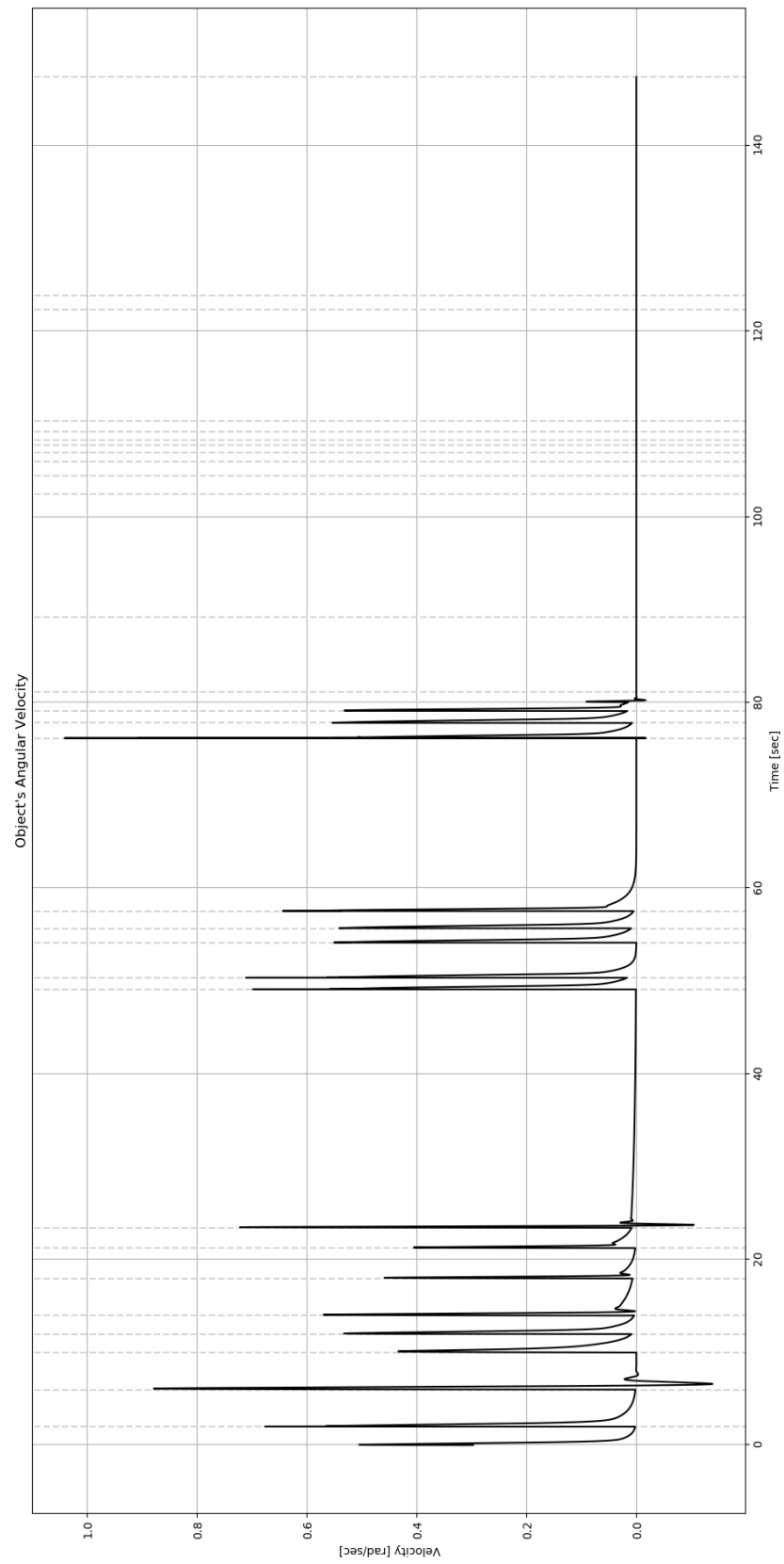


Figure 9. Object's angular velocity  $\dot{\theta}_{\mathcal{L}}$ .

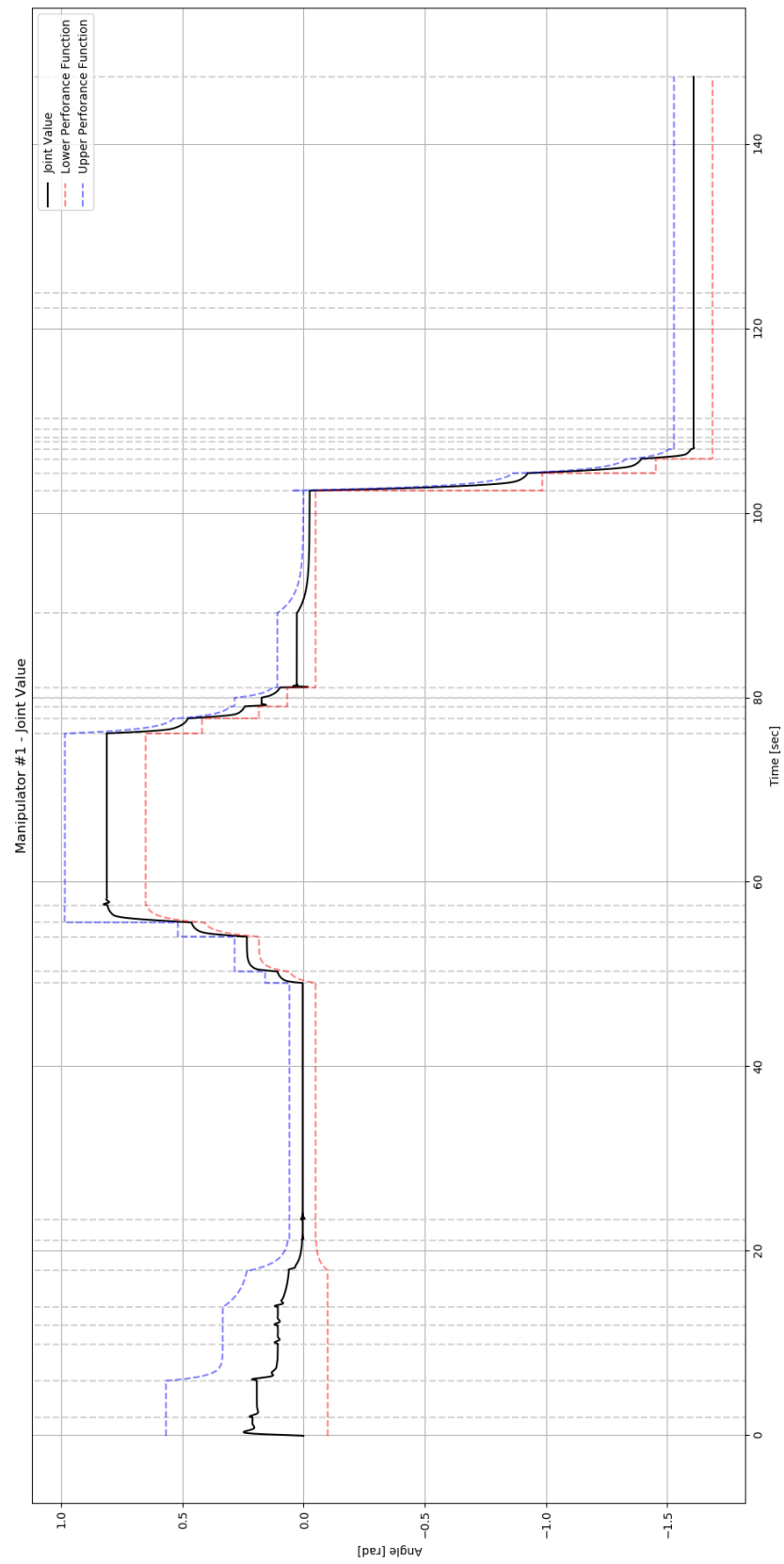


Figure 10. Evolution of joint value  $q_{1,1}$  with corresponding lower and upper bounds.

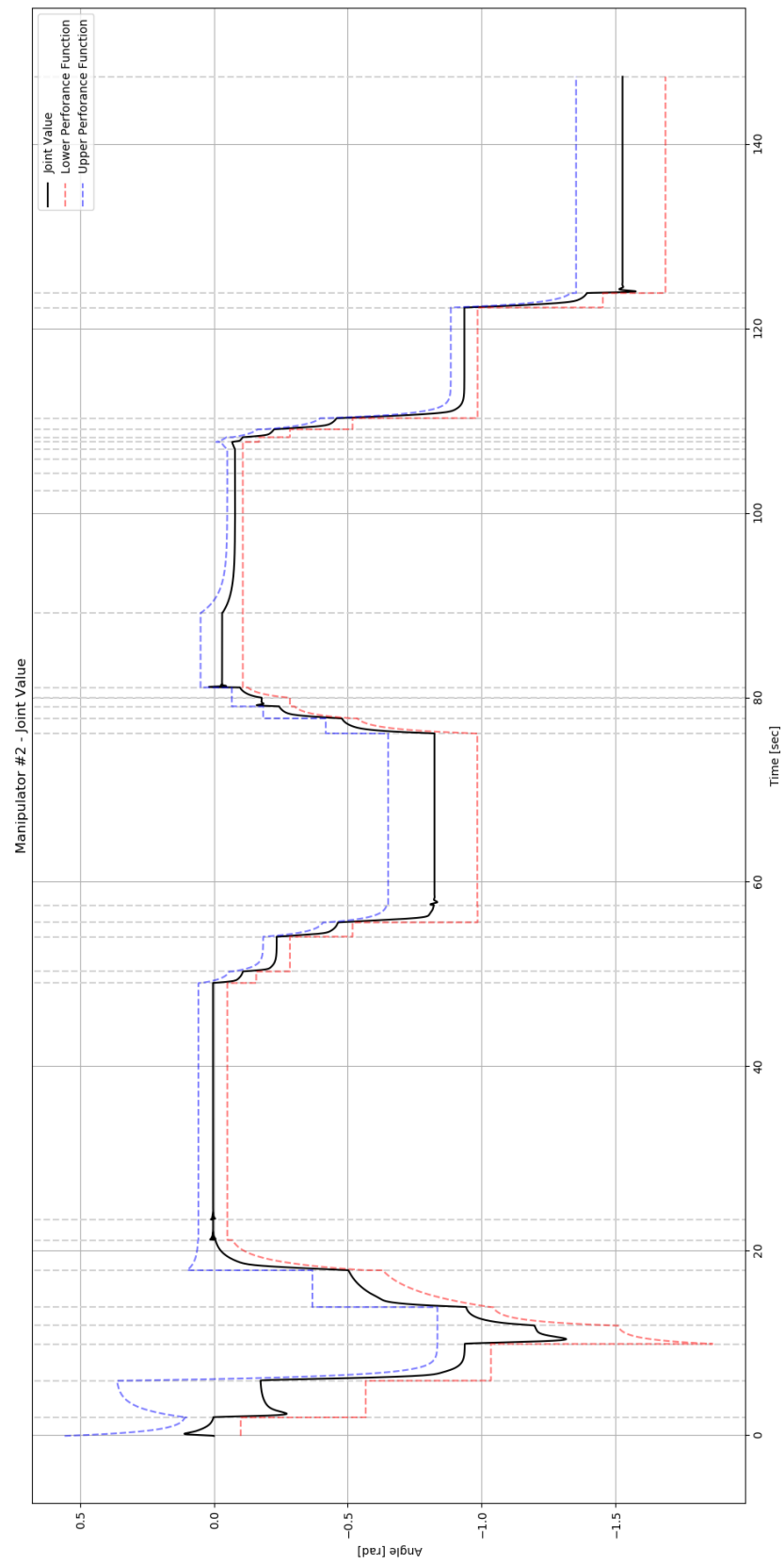


Figure 11. Evolution of joint value  $q_{2,1}$  with corresponding lower and upper bounds.

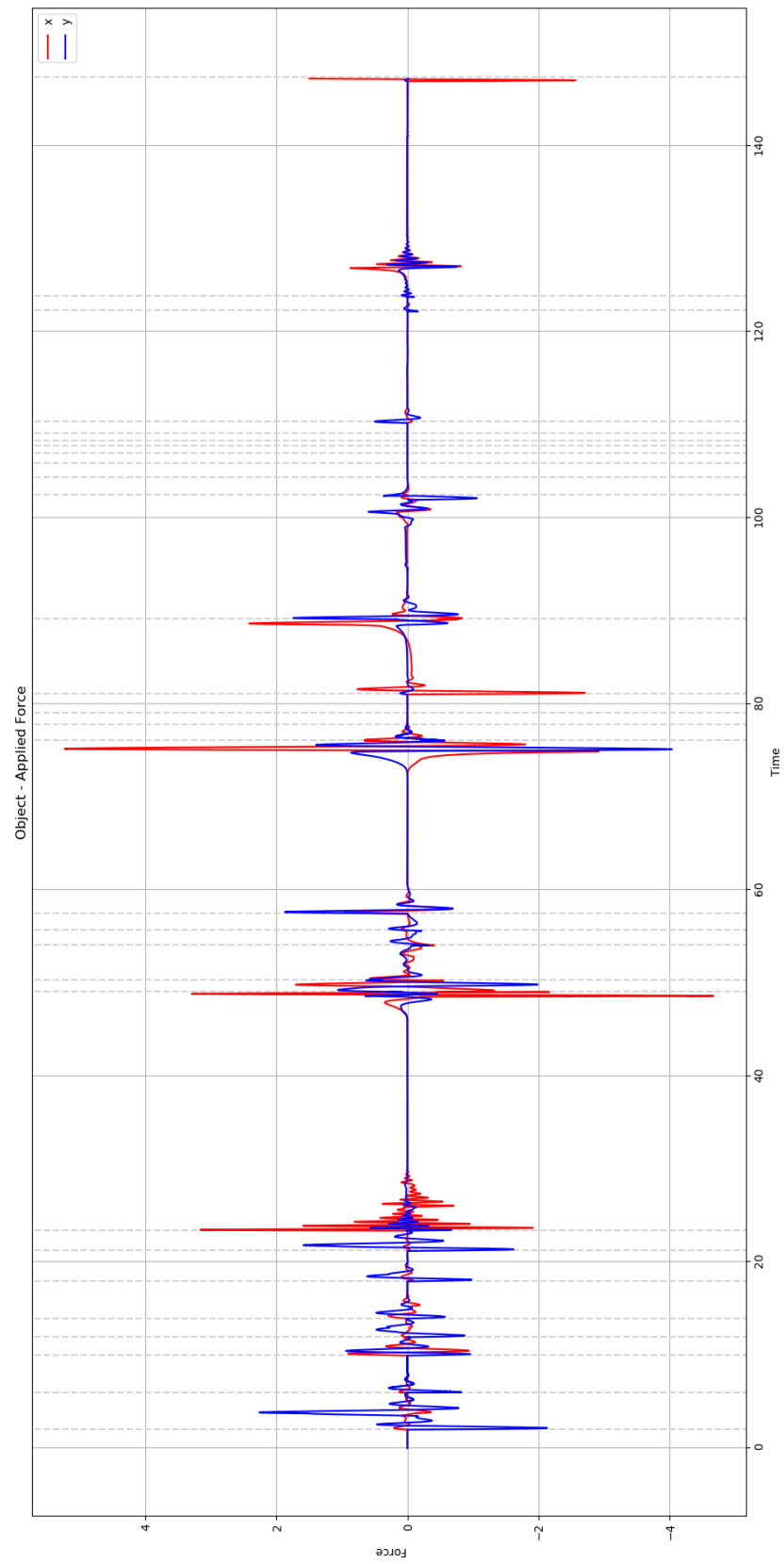


Figure 12. Total force  $\tau_{\mathcal{L},p}^{des}$  applied to the object by the robots.

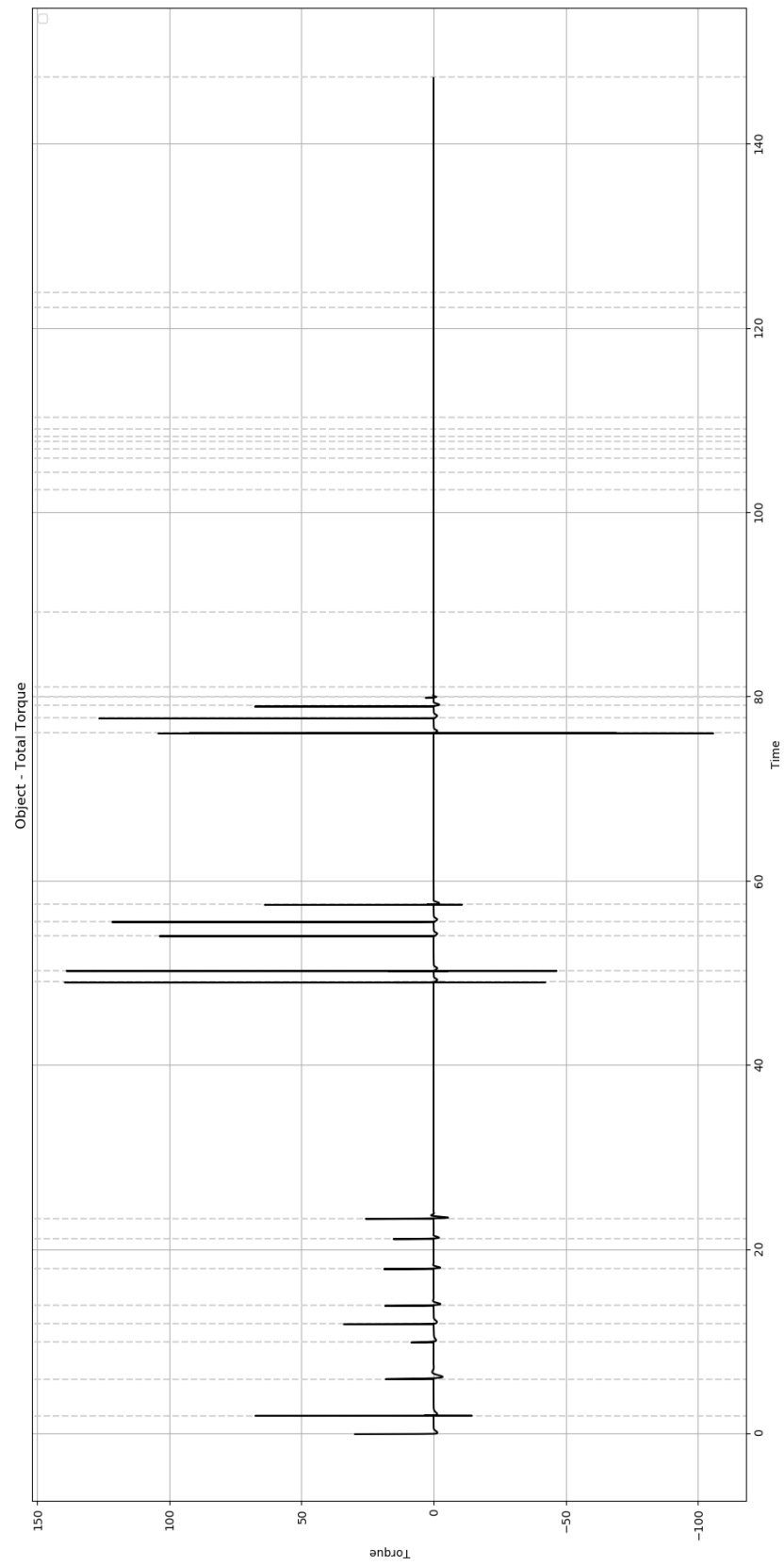


Figure 13. Total torque  $\tau_{\mathcal{L},\theta}^{des}$  applied to the object by the robots.

## 7. Conclusions and Future Directions

In this work, we presented a hybrid control scheme for addressing the cooperative transportation problem for a team of mobile manipulators carrying an object within a planar workspace. Particularly, a high-level planner was designed for computing a sequence of feasible cells by adaptively subdividing the system's configuration space using a hierarchical cell decomposition scheme. In addition, a distributed low-level control law was employed for realizing the given plan with guaranteed collision avoidance and convergence properties. Finally, simulation results validating the proposed scheme's efficacy were provided. Future research efforts will be devoted towards extending the proposed framework for human–robot collaborative transportation tasks within obstacle cluttered workspaces, where the robots are in charge of taking over the load while avoiding collisions and the human performs only high-level planning. More work is also needed towards devising optimal performance criteria to quantify the achieved response and guide appropriately the selection of the control parameters as well as to evaluate the robustness level against actuation limitations, disturbances, measurement delays, and noise.

**Author Contributions:** Conceptualization, P.V. and C.P.B. and K.J.K.; methodology, P.V. and C.P.B.; software, P.V.; validation, P.V.; formal analysis, P.V. and C.P.B.; investigation, P.V. and C.P.B. and K.J.K.; resources, K.J.K.; data curation, P.V.; writing—original draft preparation, P.V.; writing—review and editing, C.P.B. and K.J.K.; visualization, P.V.; supervision, C.P.B. and K.J.K.; project administration, C.P.B. and K.J.K.; funding acquisition, K.J.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was carried out with an SSF scholarship funded by the Act “Enhancement of human research potential through doctoral research” from the resources of the “Human Resources Development, Education and Lifelong Learning” Program, 2014–2020 with the co-financing of the European Social Fund (ESF) and the Greek State.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gerkey, B.P.; Mataric, M.J. Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 1, pp. 464–469.
2. Yamada, S.; Saito, J. Adaptive action selection without explicit communication for multi-robot box-pushing. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Phoenix, AZ, USA, 15–19 March 1999; Volume 3, pp. 1444–1449.
3. Munoz Melendez, A.; Drogoul, A. *Analyzing Multi-Robot Box-Pushing*; Technical Report; Universidad de Colima, Colima, Mexico, 2004.
4. Wang, Z.D.; Kumar, V. Object closure and manipulation by multiple cooperating mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 1, pp. 394–399.
5. Fink, J.; Michael, N.; Kumar, V. Composition of vector fields for multi-robot manipulation via caging. In Proceedings of the International Conference on Robotics Science and Systems, Orlando, FL, USA, 7–10 July 2008; Volume 3, pp. 25–32.
6. Eoh, G.; Lee, S.; Lee, T.H.; Lee, B. Distributed Object Transportation Using Virtual Object. *J. Ind. Intell. Inf.* **2014**, *2*, 20–25. [[CrossRef](#)]
7. Dai, Y.; Kim, Y.; Wee, S.; Lee, D.; Lee, S. Symmetric caging formation for convex polygonal object transportation by multiple mobile robots based on fuzzy sliding mode control. *ISA Trans.* **2016**, *60*, 321–332. [[CrossRef](#)] [[PubMed](#)]
8. Wan, W.; Shi, B.; Wang, Z.; Fukui, R. Multirobot Object Transport via Robust Caging. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *50*, 270–280. [[CrossRef](#)]
9. Ferrante, E.; Brambilla, M.; Birattari, M.; Dorigo, M. Socially-mediated negotiation for obstacle avoidance in collective transport. In *Distributed Autonomous Robotic Systems*; Springer Tracts in Advanced Robotics; Springer: Berlin/Heidelberg, Germany, 2012; Volume 83, pp. 571–583.
10. Habibi, G.; Kingston, Z.; Xie, W.; Jellins, M.; McLurkin, J. Distributed centroid estimation and motion controllers for collective transport by multi-robot systems. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 1282–1288.
11. Machado, T.; Malheiro, T.; Monteiro, S.; Erlhagen, W.; Bicho, E. Multi-constrained joint transportation tasks by teams of autonomous mobile robots using a dynamical systems approach. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016, pp. 3111–3117.
12. Wang, Z.; Schwager, M. Kinematic multi-robot manipulation with no communication using force feedback. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016, pp. 427–432.

13. Wang, Z.; Yang, G.; Su, X.; Schwager, M. OujiaBots: Omnidirectional Robots for Cooperative Object Transport with Rotation Control Using No Communication. In *Distributed Autonomous Robotic Systems*; Springer Proceedings in Advanced Robotics; Springer: Cham, Switzerland, 2018; Volume 6, pp. 117–131.
14. Hekmatfar, T.; Masehian, E.; Mousavi, S.J. Cooperative object transportation by multiple mobile manipulators through a hierarchical planning architecture. In Proceedings of the 2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014, Tehran, Iran, 15–17 October 2014; pp. 503–508.
15. Wang, Y.; De Silva, C.W. Multi-robot box-pushing: Single-agent Q-learning vs. team Q-learning. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 9–13 October 2006; pp. 3694–3699.
16. Goldman, C.V.; Zilberstein, S. Decentralized control of cooperative systems: Categorization and complexity analysis. *J. Artif. Intell. Res.* **2004**, *22*, 143–174. [[CrossRef](#)]
17. Haddadin, S.; De Luca, A.; Albu-Schäffer, A. Robot collisions: A survey on detection, isolation, and identification. *IEEE Trans. Robot.* **2017**, *33*, 1292–1312. [[CrossRef](#)]
18. Haddadin, S.; Houry, A.; Rokahr, T.; Parusel, S.; Burgkart, R.; Bicchi, A.; Albu-Schäffer, A. On making robots understand safety: Embedding injury knowledge into control. *Int. J. Robot. Res.* **2012**, *31*, 1578–1602. [[CrossRef](#)]
19. Uchiyama, M.; Dauchez, P. A symmetric hybrid position/force control scheme for the coordination of two robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 24–29 April 1988; pp. 350–356.
20. Khatib, O. Object Manipulation in a Multi-effector Robot System. In *Proceedings of the 4th International Symposium on Robotics Research*; MIT Press: Cambridge, MA, USA, 1988; Volume 4, pp. 137–144.
21. Tanner, H.; Loizou, S.; Kyriakopoulos, K. Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Trans. Robot. Autom.* **2003**, *19*, 53–64. [[CrossRef](#)]
22. Erhart, S.; Hirche, S. Model and Analysis of the Interaction Dynamics in Cooperative Manipulation Tasks. *IEEE Trans. Robot.* **2016**, *32*, 672–683. [[CrossRef](#)]
23. Khatib, O.; Yokoi, K.; Chang, K.; Ruspini, D.; Holmberg, R.; Casal, A. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Osaka, Japan, 4–8 November 1996; Volume 2, pp. 546–553.
24. Verginis, C.K.; Mastellaro, M.; Dimarogonas, D.V. Robust cooperative manipulation without force/torque measurements: Control design and experiments. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 713–729. [[CrossRef](#)]
25. Zhang, Y.; Zou, M.; Xiao, H.; Wen, J.; Wang, Y. Cooperative-manipulation scheme of routh-hurwitz type for simultaneous repetitive motion planning of two-manipulator robotic systems. In Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016, Yinchuan, China, 28–30 May 2016; pp. 4409–4414.
26. Petitti, A.; Franchi, A.; Di Paola, D.; Rizzo, A. Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 441–446.
27. Noohi, E.; Zefran, M. Modeling the interaction force during a haptically-coupled cooperative manipulation. In Proceedings of the 25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016, New York, NY, USA, 26–31 August 2016; pp. 119–124.
28. Pajak, I. Real-Time Trajectory Generation Methods for Cooperating Mobile Manipulators Subject to State and Control Constraints. *J. Intell. Robot. Syst. Theory Appl.* **2019**, *93*, 649–668. [[CrossRef](#)]
29. He, Y.; Wu, M.; Liu, S. An optimisation-based distributed cooperative control for multi-robot manipulation with obstacle avoidance. *IFAC-PapersOnLine* **2020**, *53*, 9859–9864. [[CrossRef](#)]
30. Kosuge, K.; Oosumi, T. Decentralized control of multiple robots handling an object. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Osaka, Japan, 4–8 November 1996; Volume 1, pp. 318–323.
31. Stilwell, D.J.; Bay, J.S. Toward the development of a material transport system using swarms of ant-like robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2–6 May 1993; Volume 1, pp. 766–771.
32. Gross, R.; Dorigo, M. Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adapt. Behav.* **2008**, *16*, 285–305.
33. Chen, J.; Gauci, M.; Gross, R. A strategy for transporting tall objects with a swarm of miniature mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 863–869.
34. Chen, J.; Gauci, M.; Li, W.; Kolling, A.; Gross, R. Occlusion-Based Cooperative Transport with a Swarm of Miniature Mobile Robots. *IEEE Trans. Robot.* **2015**, *31*, 307–321. [[CrossRef](#)]
35. Wang, Z.; Schwager, M. Force-Amplifying N-robot Transport System (Force-ANTS) for cooperative planar manipulation without communication. *Int. J. Robot. Res.* **2016**, *35*, 1564–1586. [[CrossRef](#)]
36. Bechlioulis, C.P.; Kyriakopoulos, K.J. Collaborative multi-robot transportation in obstacle-cluttered environments via implicit communication. *Front. Robot. AI* **2018**, *5*, 90. [[CrossRef](#)] [[PubMed](#)]
37. Vlantis, P.; Vrohidis, C.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Robot Navigation in Complex Workspaces Using Harmonic Maps. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 21–25 May 2018; pp. 1726–1731.



38. Vlantis, P.; Vrohidis, C.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Orientation-Aware Motion Planning in Complex Workspaces using Adaptive Harmonic Potential Fields. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8592–8598. [[CrossRef](#)]
39. Loizou, S.G. The navigation transformation: Point worlds, time abstractions and towards tuning-free navigation. In Proceedings of the 2011 19th Mediterranean Conference on Control & Automation (MED), Corfu, Greece, 20–23 June 2011; pp. 303–308.
40. Loizou, S.G. The Navigation Transformation. *IEEE Trans. Robot.* **2017**, *33*, 1516–1523. [[CrossRef](#)]
41. Arslan, O.; Koditschek, D.E. Smooth extensions of feedback motion planners via reference governors. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4414–4421. [[CrossRef](#)]
42. Krstic, M.; Kokotovic, P.V.; Kanellakopoulos, I. *Nonlinear and Adaptive Control Design*, 1st ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1995.