Robotics and Autonomous Systems 174 (2024) 104646

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Planning and control of multi-robot-object systems under temporal logic tasks and uncertain dynamics

Christos K. Verginis [a,*], Yiannis Kantaros [b], Dimos V. Dimarogonas [c]

[a] Uppsala University, Uppsala, Sweden
[b] Washington University, St. Louis, MO, USA
[c] KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

*Keywords:*
Multi-robot systems
Temporal logics
Motion planning
Action planning
Adaptive control

ABSTRACT

We develop an algorithm for the motion and task planning of a system composed of multiple robots and unactuated objects under tasks expressed as Linear Temporal Logic (LTL) constraints. The robots and objects evolve subject to uncertain dynamics in an obstacle-cluttered environment. The key part of the proposed solution is the intelligent construction of a coupled transition system that encodes the motion and tasks of the robots *and* the objects. We achieve such a construction by designing appropriate adaptive control protocols in the lower level, which guarantee the safe robot navigation/object transportation in the environment while compensating for the dynamic uncertainties. The transition system is efficiently interfaced with the temporal logic specification via a sampling-based algorithm to output a discrete path as a sequence of synchronized actions of the robots; such actions satisfy the robots' as well as the objects' specifications. The robots execute this discrete path by using the derived low level control protocol. Numerical experiments verify the proposed framework.

## 1. Introduction

Temporal-logic-based motion planning has gained significant amount of attention over the last decade, as it provides a fully automated correct-by-design controller synthesis approach for autonomous robots. Temporal logics, such as linear temporal logic (LTL), provide formal high-level languages that can describe planning objectives more complex than the well-studied navigation algorithms, and have been used extensively both in single- as well as in multi-robot setups (see, indicatively, [1–10]). The task is given as a temporal logic formula, which is coupled with a discrete representation of the underlying system, abstracted from the underlying continuous dynamics, in order to derive an appropriate high-level discrete path.

There exist numerous works that consider temporal-logic-based tasks, both for single- and multi-robot systems [1–3,5,7,9,11]. Nevertheless, the related works consider temporal logic-based motion planning for fully actuated, autonomous robotic agents. Consider, however, cases where some unactuated objects must undergo a series of processes in a workspace with autonomous robots (e.g., car factories or automated warehouses). In such cases, the robots, except for satisfying their own motion specifications, are also responsible for coordinating with each other in order to transport the objects around the workspace. When the unactuated objects' specifications are expressed using temporal logics, then the abstraction of the robots' behavior becomes much more complex, since it has to take into account the objects' goals.

In addition, the spatial discretization of a multi-agent system to an abstracted higher level system necessitates the design of appropriate continuous-time controllers for the transition of the agents among the states of discrete system. Many works in the related literature, however, either assume that there *exist* such continuous controllers or adopt non-realistic assumptions. For instance, many works either do not take into account continuous agent dynamics or consider single or double integrators [1,3,8–10], which can deviate from the actual dynamics of the agents, leading thus to poor performance in real-life scenarios. Discretized abstractions, including design of the discrete state space and/or continuous-time controllers, can be found in [7,12–15] for general systems and [16,17] for multi-agent systems. Moreover, many works adopt dimensionless point-mass agents and therefore do not consider inter-agent collision avoidance [7,9,10], which can be a crucial safety issue in applications involving autonomous robots.

Since we aim at incorporating the unactuated objects' specifications in our framework, the robots have to perform (cooperative) transportation of the objects around the workspace, while avoiding collisions with obstacles. Cooperative transportation/manipulation has

* Corresponding author.
*E-mail addresses:* christos.verginis@angstrom.uu.se (C.K. Verginis), ioannisk@wustl.edu (Y. Kantaros), dimos@kth.se (D.V. Dimarogonas).

https://doi.org/10.1016/j.robot.2024.104646
Received 25 April 2022; Received in revised form 15 August 2023; Accepted 25 January 2024
Available online 1 February 2024

been extensively studied in the literature (see, for instance, [18–28]), with collision avoidance specifications being incorporated in [25,26,29]; [25,26] rely on MPC-based optimization algorithms, which might not converge to optimal solutions due to the potentially non-convex workspace and might further hinder real-time execution due to computational complexity. Further, MPC usually requires knowledge of bounds of dynamic uncertainties, which it uses to inflate the obstacles and force the system to remain in a restricted version of the workspace, possibly yielding conservative solutions. Similarly, [29] adopts the unrealistic assumption of perfectly known models, without addressing dynamic uncertainties. Cooperative object transportation with dynamic uncertainties and temporal logics has also been considered in our previous work [30], restricted however to trajectory tracking without taking into account collisions with obstacles.

The contribution of this paper is the development of an end-to-end task and motion planning algorithm that guarantees the satisfaction of local tasks expressed as LTL formulas by multiple heterogeneous robotic agents and unactuated objects that suffer from dynamic model uncertainties and operate in obstacle-cluttered environments.

The proposed algorithm receives the robots' and objects' LTL tasks as inputs and automatically derives the tasks that the agents need to execute as well as the respective control inputs that successfully execute these tasks, solving both planning and control problems simultaneously. More specifically, we propose a novel abstraction design of the multi-robot-object system into a finite-state transition system that encodes the motion of the robots and the objects as well as the interactions among them. To that end, we introduce grasping variables that encode the robots-objects interaction and we extend our previous work on safe robotic navigation [31] to design feedback-control protocols that guarantee (i) the navigation of the robots and (ii) the cooperative transportation of the objects by the robots in the environment. These protocols combine barrier functions and point-world transformations to guarantee collision-avoidance properties as well as adaptive-control methodologies to compensate for the dynamic uncertainties in real time. They are further decentralized in the sense that each robot calculates its own control action. The robot navigation and cooperative object transportation constitute robot action primitives that enable the transitions in the aforementioned transition system. Subsequently, we use previous results on formal-verification-based planning to derive a high-level path, as a sequence of robot action primitives, that satisfies the given tasks. In particular, we compose the resulting transition system with an automaton-based representation of the underlying LTL tasks in a centralized fashion. Then, we employ an efficient sampling-based procedure to search the composition of the transition system and the task's automaton for the derivation of the aforementioned high-level plan. According to the authors' best knowledge, this is the first work that addresses multi-robot-object LTL-based planning and control with model uncertainties in obstacle-cluttered environments. Finally, we stress that, although we consider *local* LTL tasks, the robots are responsible for both their as well as the objects' tasks; the proposed algorithm automatically allocates these tasks to the robots, resembling the notion of "global" LTL tasks [32].

This paper extends our previous work [33] in the following directions: firstly, we consider transportation of the objects by multiple robots (as opposed to [33], which consider single-robot object transportation). Secondly, we consider more complex, obstacle-cluttered environments. Thirdly, we consider dynamic uncertainty in the dynamics of the robots and objects. Finally, instead of model-checking tools based on graph search algorithms, we use a sampling-based procedure to derive an optimal task-satisfying plan, yielding significantly lower complexity in terms of runtime and memory.

The rest of the paper is organized as follows. Section 2 provides necessary background and formulates the problem, while Section 3 presents the proposed solution. Section 4 provides simulation results and, finally, Section 5 concludes the paper.

## 2. Preliminaries and problem formulation

### 2.1. Task specification in LTL

We focus on the task specification $\phi$ given as a Linear Temporal Logic (LTL) formula. The basic ingredients of a LTL formula are a set of atomic propositions $\Psi$ and several boolean and temporal operators. LTL formulas are formed according to the following grammar [34]: $\phi ::= \texttt{true} \mid a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \circ\phi \mid \phi_1 \mathcal{U} \phi_2$, where $a \in \Psi$, $\phi_1$ and $\phi_2$ are LTL formulas and $\circ$, $\mathcal{U}$ are the next and until operators, respectively. Definitions of other useful operators like $\square$ (always), $\lozenge$ (eventually) and $\Rightarrow$ (implication) are omitted and can be found at [34]. The semantics of LTL are defined over infinite words over $2^{\Psi}$. Intuitively, an atomic proposition $\psi \in \Psi$ is satisfied on a word $w = w_1 w_2 \ldots$ if it holds at its first position $w_1$, i.e. $\psi \in w_1$. Formula $\phi$ holds true if $\phi$ is satisfied on the word suffix that begins in the next position $w_2$, whereas $\phi_1 \mathcal{U} \phi_2$ states that $\phi_1$ has to be true until $\phi_2$ becomes true. Finally, $\lozenge\phi$ and $\square\phi$ holds on $w$ eventually and always, respectively. For a full definition of the LTL semantics, we refer the reader to [34].

### 2.2. Problem formulation

Consider $N > 1$ robotic agents operating in a compact 2D workspace $\mathcal{W} \subset \mathbb{R}^2$ with an outer boundary with $M > 0$ objects, and $\mathcal{N} := \{1, \ldots, N\}$, $\mathcal{M} := \{1, \ldots, M\}$; The objects are represented by rigid bodies whereas the robotic agents are fully actuated and holonomic, equipped with a transportation tool, such as a robotic arm. In addition, the workspace is populated with $J \in \mathbb{N}$ connected, closed sets $\{\mathfrak{O}_j\}_{j \in \mathcal{J}}$, with $\mathcal{J} := \{1, \ldots, J\}$, representing obstacles, and we define the free space as $\mathcal{W}_{\rm f} := \mathcal{W} \backslash \bigcup_{j \in \mathcal{J}} \mathfrak{O}_j$.

We further assume that there exist $K > 1$ points within $\mathcal{W}_{\rm f}$, denoted by $p_{\pi_k}$, corresponding to certain properties of interest (e.g., gas station, repairing area, etc.), with $\mathcal{K} := \{1, \ldots, K\}$. Since, in practise, these properties are naturally inherited to some neighborhood of the respective point of interest, we define for each $k \in \mathcal{K}$ the *region of interest* $\pi_k$, corresponding to $p_{\pi_k}$, as the closed ball $\pi_k := \bar{\mathcal{B}}(p_{\pi_k}, r_{\pi_k}) \subset \mathcal{W}_{\rm f}$, with $r_{\pi_k} > 0$ positive radii, $\forall k \in \mathcal{K}$. These properties of interest are expressed as boolean variables via finite sets of atomic propositions. In particular, we introduce disjoint sets of atomic propositions $\Psi_i, \Psi^o_j$, expressed as boolean variables, that represent services provided to agent $i \in \mathcal{N}$ and object $j \in \mathcal{M}$ in $\Pi := \{\pi_1, \ldots, \pi_K\}$. The services provided at each region $\pi_k$ are given by the labeling functions $\mathcal{L}_i : \Pi \to 2^{\Psi_i}, \mathcal{L}^o_j : \Pi \to 2^{\Psi^o_j}$, which assign to each region $\pi_k, k \in \mathcal{K}$, the subset of services $\Psi_i$ and $\Psi^o_j$, respectively, that can be provided in that region to agent $i \in \mathcal{N}$ and object $j \in \mathcal{M}$, respectively. In addition, we consider that the agents and the object are initially ($t = 0$) in the regions of interest $\pi_{init(i)}, \pi_{init_o(j)}$, where the functions $init : \mathcal{N} \to \mathcal{K}$, $init_o : \mathcal{M} \to \mathcal{K}$ specify the initial region indices.

**Example.** Consider the workspace shown in Fig. 1. There exist four obstacles, four regions of interest $\pi_1, \ldots, \pi_4$, four robots, and three objects, in an initial configuration determined by $\pi_{init}(1) = \pi_{init_o}(1) = 1$, $\pi_{init}(2) = \pi_{init_o}(2) = 2$, $\pi_{init}(3) = \pi_{init}(4) = \pi_{init_o}(3) = 4$. The services provided in the respective regions can be encoded with the atomic propositions $\Psi_i = \{$"$i$-$\pi_1$","$i$-$\pi_2$","$i$-$\pi_3$","$i$-$\pi_4$"$\}$ and $\Psi^o_i = \{$"$O_j$-$\pi_1$","$O_j$-$\pi_2$","$O_j$-$\pi_3$","$O_j$-$\pi_4$"$\}$, for all $i \in \mathcal{N}$, $j \in \mathcal{M}$, indicating the presence of robot $i$ or object $j$ in the regions, respectively. The corresponding labeling functions are defined as $\mathcal{L}_i(\pi_k) = \{$"$i$-$\pi_k$"$\}$, $\mathcal{L}^o_i(\pi_k) = \{$"$O_j$-$\pi_k$"$\}$ for all $i \in \mathcal{N}$, $j \in \mathcal{M}$, $k \in \mathcal{K}$ and $\emptyset$ otherwise.

In the following, we present the modeling of the coupled dynamics of the objects and the robots. We denote by $x_i \in \mathbb{R}^2$ the position of robot $i$'s center of mass. In this work we explicitly consider the actions of robot navigation, as well as (cooperative) transportation of an object via $x_i$, where the joint variables of the potential mounted robotic arm

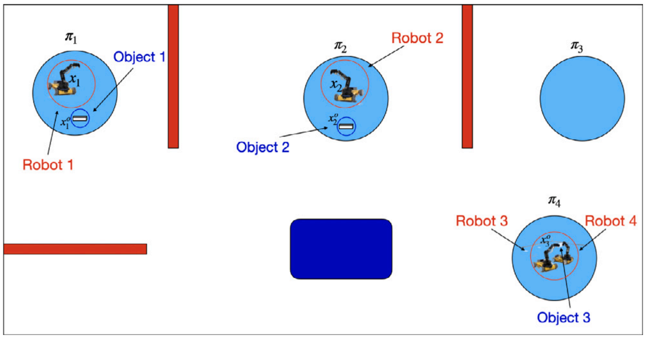


**Fig. 1.** Multi-robot-object system example.

are assumed to be *fixed*. We consider that the robotic arm joints are used only for grasping/releasing an object (when the respective mobile base is fixed), actions that we do not explicitly model. The dynamics describing the motion of robot $i$'s center of mass are given by

$$m_i \ddot{x}_i + f_i(x_i, \dot{x}_i) = u_i - h_i, \tag{1}$$

where $m_i$ is the *unknown* mass of robot $i$, $f_i : \mathbb{R}^4 \to \mathbb{R}^2$ are *unknown* friction-like functions, $u_i \in \mathbb{R}^2$ is the control input of robot $i$, and $h_i \in \mathbb{R}^2$ is the force exerted by robot $i$ at the grasping point with object $j$, in case of contact; in such a case, $h_i$ is a function of the respective robot and object dynamics [35]. The overall robot configuration is denoted by $x := [x_1^\top, \dots, x_N^\top]^\top \in \mathbb{R}^{2N}$. The aforementioned dynamics concern the cases when (i) robot $i$ is navigating to some pre-defined point, and (ii) robot $i$ is transporting, possibly collaboratively with other robots, an object $j$. In both of these cases, the joint variables of the mounted robotic arm are assumed to be constant. The procedures of grasping/releasing an object, where the robotic arm would have to be activated, are not considered here.

We consider that each robot $i$, for a given $x_i$, covers a spherical region of constant radius $r_i \in \mathbb{R}_{>0}$ that bounds its volume, i.e., $\bar{\mathcal{B}}(x_i, r_i)$, $\forall i \in \mathcal{N}$. Furthermore, the robots are heterogeneous in the sense that $m_i$, $f_i()$ can be different. Similarly, the robots exhibit different *power capabilities*, i.e., the maximum value for the control input $u_i$ differs among the robots. By using a simple normalization procedure, we map these maximum values to positive constants $\zeta_i > 0$, $i \in \mathcal{N}$; such constants represent the ability of the robot to transport an object and are proportional to the maximum control-input value the robots can exert. A larger maximum value indicates a more powerful robotic agent, which in turn implies a larger constant $\zeta_i$.

Regarding the objects, we denote by $x_j^o \in \mathbb{R}^2$ the position of the $j$th object's center of mass, $\forall j \in \mathcal{M}$. We consider the following second-order Newton-Euler dynamics:

$$m_j^o \ddot{x}_j^o + f_j^o(x_j^o, \dot{x}_j^o) = h_j^o, \tag{2}$$

where $m_j^o$ is the *unknown* mass of object $j$, $f_j^o : \mathbb{R}^4 \to \mathbb{R}^2$ are *unknown* friction terms, and $h_j^o \in \mathbb{R}^2$ are the forces exerted to the $j$th object's center of mass, in case of contact with a robot. The overall object configuration is denoted by $x^o := [x_1^o, \dots, x_M^o]^\top$, and $\bar{x} := [x^\top, (x^o)^\top]^\top$. Similarly to the robots, note that the objects can be heterogeneous in the sense of different masses $m_j^o$ and functions $f_j^o()$.

The functions $f_i$ and $f_j^o$ are assumed to satisfy the following assumption:

**Assumption 1.** The functions $f_i, f_j^o : \mathbb{R}^4 \to \mathbb{R}^2$ are analytic and satisfy

$$\|f_i(\mathsf{x}, \mathsf{y})\| \le \alpha_i \|\mathsf{y}\|, \tag{3}$$

$$\|f_j^o(\mathsf{x}, \mathsf{y})\| \le \alpha_j^o \|\mathsf{y}\|, \tag{4}$$

$\forall \mathsf{x}, \mathsf{y} \in \mathbb{R}^2$, where $\alpha_i$, $\alpha_j^o$ are unknown positive constants, $\forall i \in \mathcal{N}, j \in \mathcal{M}$.

The aforementioned assumption is inspired by standard friction-like terms, which can be approximated by continuously differentiable velocity functions [36].

Similarly to the robots, each object's volume is represented by the spherical set of constant radius $r_j^o \in \mathbb{R}_{>0}$, i.e., $\bar{\mathcal{B}}(x_j^o, r_j^o)$, $\forall j \in \mathcal{M}$.

Next, we provide the coupled dynamics between an object $j \in \mathcal{M}$ and a subset $\mathcal{A} \subseteq \mathcal{N}$ of robots that grasp it rigidly. In the case of such a grasp, the forces $h_i$, $i \in \mathcal{A}$, at the grasping contacts with the object depend on the dynamic terms $m_i$, $f_i(x_i, \dot{x}_i)$ and $m_j^o$, $f_j^o(x_j^o, \dot{x}_j^o)$ that appear in (1) and (2), respectively, and the robots' control inputs $u_i$. An analytic expression for $h_i$ can be derived using Gauss' minimization principle [35]. Nevertheless, since the robots-object grasp is assumed to be rigid, it holds that $h_j^o = \sum_{i \in \mathcal{A}} h_i$, and since the joint variables of the robotic arms are fixed, $x_j^o = x_i + d_{ij}^o$, $\dot{x}_j^o = \dot{x}_i$ and $\ddot{x}_j^o = \ddot{x}_i$, where $d_{ij}^o \in \mathbb{R}^2$ is the constant offset between $x_j^o$ and $x_i$, $\forall i \in \mathcal{A}$. Therefore, one obtains that

$$m_{\mathcal{A},j} \ddot{x}_j^o + f_{\mathcal{A},j}(x_j^o, \dot{x}_j^o) = \sum_{i \in \mathcal{A}} u_i, \tag{5}$$

where $m_{\mathcal{A},j} := m_j^o + \sum_{i \in \mathcal{A},j} m_i$, $f_{\mathcal{A},j} := f_j^o(x_j^o, \dot{x}_j^o) + \sum_{i \in \mathcal{A},j} f_i(x_j^o - d_{ij}^o, \dot{x}_j^o)$. Note that Assumption 1 implies that

$$\|f_{\mathcal{A},j}(x_j^o, \dot{x}_j^o)\| \le \alpha_{\mathcal{A},j} \|\dot{x}_j^o\|, \tag{6}$$

for an unknown positive constant $\alpha_{\mathcal{A},j}$.

Regarding the volume of the coupled robots-object system, we consider that it is bounded by a sphere of radius centered at $x_j^o$ with constant radius $r_{\mathcal{A},j} \in \mathbb{R}_{>0}$, i.e., $\mathcal{B}(x_j^o, r_{\mathcal{A},j})$, which is large enough to cover the volume of the coupled system.

Moreover, in order to take into account the introduced robots' power capabilities $\zeta_i$, $i \in \mathcal{N}$, we consider a set-valued function $\Lambda \in \{\mathsf{True}, \mathsf{False}\}$ that outputs whether the robots that grasp an object are able to transport the object, based on their power capabilities. For instance, $\Lambda(m_j^o, \zeta_{\mathcal{A}}) = \mathsf{True}$, where $m_j^o \in \mathbb{R}_{>0}$ is the mass of object $j$ and $\zeta_{\mathcal{A}} := [\zeta_i]_{i \in \mathcal{A}}^\top$, implies that the robots $\mathcal{A}$ have sufficient power capabilities to cooperatively transport object $j$. Although, as noted before, the masses of the objects are considered to be *unknown*, we assume the existence of a mechanism, used in the high-level planner of the next section, that is able to determine which robots are powerful enough to transport each object. For instance, the high-level planner might have access to rough estimates of the masses of the objects, sufficient to determine which robots are capable of transporting which objects. However, such estimates are not enough to be used for continuous feedback control design in (2).

Next, we define the functions $\mathcal{AG}_j : \mathbb{R}_{\ge 0} \to 2^{\mathcal{N}_0}$, with $\mathcal{N}_0 := \mathcal{N} \cup \{0\}$, to denote which robots grasp an object $j \in \mathcal{M}$; $\mathcal{AG}_j = \emptyset$ means that no robots grasp object $j$. Note also that $i \in \mathcal{AG}_j \Leftrightarrow i \notin \mathcal{AG}_{j'}, \forall j' \in \mathcal{M} \backslash \{j\}$, i.e., robot $i$ can grasp at most one object at a time. We further denote $\mathcal{AG} := [\mathcal{AG}_1, \dots, \mathcal{AG}_M]^\top \in (2^{\mathcal{N}_0})^M$.

In the following, we use the term "entity" to refer to single robots, objects as well as systems composed of robots that grasp an object (robots-object systems). The number of these systems depends on the variables $\mathcal{AG}$. Given a grasping configuration $\mathcal{AG} \in (2^{\mathcal{N}_0})^M$, consider $\bar{T}(\mathcal{AG})$ number of entities, indexed by the set $\mathcal{T}(\mathcal{AG}) := \{1, \dots, \bar{T}(\mathcal{AG})\}$. Each entity (robot, object, or robots-object system) is characterized by the respective configuration ($x_i$, $x_j^o$, $x_{j'}^o$) and radius ($r_i$, $r_j^o$, $r_{\mathcal{A},j'}$), respectively, which we denote for simplicity by the generic variables $x_i^e$, $r_i^e$, for all $i \in \mathcal{T}(\mathcal{AG})$.

We further define the free space for each entity

$$\mathcal{F}_i(\mathcal{AG}) := \Big(\mathcal{W}_f \backslash \bigcup_{j \in \mathcal{T}(\mathcal{AG}) \backslash \{i\}} \bar{\mathcal{B}}(x_j^e, r_j^e)\Big) \ominus r_i^e,$$

where the incorporation of $\ominus r_i^e$ enlarges the obstacles and the other entities with the radius $r_i^e$.

**Example** (*Continued*)**.** Consider the workspace shown in Fig. 1. Robots 3, 4, and object 3 form a coupled robots-object system and there are a total of 5 entities. The volumes of the robots, objects, and coupled system $\bar{\mathcal{B}}(x_i, r_i)$, $i \in \{1,2\}$, $\bar{\mathcal{B}}(x_j^o, r_j^o)$, $j \in \{1,2\}$ and $\bar{\mathcal{B}}(x_3^o, r_{\mathcal{A},3})$ are depicted with red and blue circles. At the configuration shown, the grasping variables are $\mathcal{AG}_1 = \mathcal{AG}_2 = \emptyset$, and $\mathcal{AG}_3 = \{3,4\}$.

We give now the definitions for the transitions of the robots and the objects between the regions of interest.

**Definition 1** (*Navigation*). Let $\mathcal{AG}(t_0) \in (2^{\mathcal{N}_0})^M$ such that

$$\left.\begin{array}{l} x_i^e(t_0) \in \mathcal{F}_i(\mathcal{AG}(t_0)) \\ \bar{\mathcal{B}}_i(x_i^e(t_0), r_i^e) \subset \pi_{i_k} \end{array}\right\} i \in \mathcal{T}(\mathcal{AG}(t_0))$$

where $i_k \in \mathcal{K}$, for all $i \in \mathcal{T}(\mathcal{AG})$ and some $t_0 \in \mathbb{R}_{\geq 0}$. Then, entity $j \in \mathcal{T}(\mathcal{AG})$ executes a transition from $\pi_{j_k}$ to $\pi_{j_{k'}}$, with $j_{k'} \in \mathcal{K}\backslash\{j_k\}$, if there exists a finite $t_f \geq t_0$ such that

$$\bar{\mathcal{B}}_j(x_j^e(t_f), r_j^e) \subset \pi_{j_{k'}}$$
$$x_j(t) \in \mathcal{F}_j(\mathcal{AG})$$

for all $t \in [t_0, t_f]$.

**Definition 2** (*Grasping*). Let $\mathcal{AG}(t_0) \in (2^{\mathcal{N}_0})^M$ such that

$$\left.\begin{array}{l} x_i^e(t_0) \in \mathcal{F}_i(\mathcal{AG}(t_0)) \\ \bar{\mathcal{B}}_i(x_i^e(t_0), r_i^e) \subset \pi_{i_k} \end{array}\right\} i \in \mathcal{T}(\mathcal{AG}(t_0))$$
$$j \notin \mathcal{AG}_\ell(t_0)$$
$$x_j(t_0), x_\ell^o(t_0) \in \pi_\ell$$

for some $j \in \mathcal{N}$, $\ell \in \mathcal{M}$, $\ell \in \mathcal{K}$, $t_0 \in \mathbb{R}_{\geq 0}$ where $i_k \in \mathcal{K}$, for all $i \in \mathcal{T}(\mathcal{AG})$. Then, agent $j$ *grasps* object $\ell$, denoted by $j \xrightarrow{g} \ell$, if there exists a finite $t_f \geq t_0$ such that $j \in \mathcal{AG}_\ell(t_f)$.

Similarly, we can define the *releasing* action $j \xrightarrow{r} \ell$ for an agent $j \in \mathcal{N}$ and object $\ell \in \mathcal{M}$. Loosely speaking, the aforementioned definitions correspond to specific action primitives of the robots, namely robot navigation or object transportation, which define the *navigation* transition, and grasping and releasing actions. When the navigation transition from $\pi_{j_k}$ to $\pi_{j'_k}$ corresponds to a robot navigation for robot $j \in \mathcal{N}$, we denote it by $\pi_{j_k} \rightarrow_j \pi_{j'_k}$; when it corresponds to a cooperative transportation of object $j \in \mathcal{M}$ by a subset of robots $\mathcal{A}$, we denote it by $\pi_{j_k} \xrightarrow{T}_{\mathcal{A},j} \pi_{j'_k}$.

We also assume the existence of a procedure $\mathcal{P}_s$ that outputs whether or not a set of non-intersecting spheres fits in a larger sphere as well as possible positions of the spheres in the case they fit. More specifically, given a region of interest $\pi_k$ and a number $\widetilde{N} \in \mathbb{N}$ of sphere radii (of robots, objects, or object-robots systems) the procedure can be seen as a function $\mathcal{P}_s := [\mathcal{P}_{s,0}, \mathcal{P}_{s,1}^\top]^\top$, where $\mathcal{P}_{s,0} : \mathbb{R}_{\geq 0}^{\widetilde{N}+1} \rightarrow \{\mathsf{True}, \mathsf{False}\}$ outputs whether the spheres fit in the region $\pi_k$ whereas $\mathcal{P}_{s,1}$ provides possible configurations of the robots and the objects or 0 in case the spheres do not fit. For instance, $P_{s,0}(r_{\pi_2}, r_1, r_3, r_1^o, r_5^o)$ determines whether the robots 1,3 and the objects 1,5 fit in region $\pi_2$, without colliding with each other; $(x_1^e, x_2^e, x_3^e, x_4^e) = (x_1, x_3, x_1^o, x_5^o) = P_{s,1}(r_{\pi_2}, r_1^e, r_2^e, r_3^e, r_4^e) = P_{s,1}(r_{\pi_2}, r_1, r_3, r_1^o, r_5^o)$ provides a set of configurations such that $\bar{\mathcal{B}}_i(x_i^e, r_i^e)$ $i \in \{1, \ldots, 4\}$, and the pairwise intersections of these sets are empty. The problem of finding an algorithm $\mathcal{P}_s$ is a special case of the sphere packing problem [37]. Note, however, that we are not interested in finding the maximum number of spheres that can be packed in a larger sphere but, rather, in the simpler problem of determining whether a set of spheres can be packed in a larger sphere. Moreover, it should be noted that $\mathcal{P}_s$ provides only collision-free configurations in a region for the robots to navigate to, if such configurations exist. For grasping an object, we assume the existence of a control protocol that the robots can use to safely grasp an object, once they navigate to the corresponding region.

Our goal is to control the multi-robot-object system defined above such that the robots and the objects obey a given specification over their atomic propositions $\Psi_i, \Psi_j^o, \forall i \in \mathcal{N}, j \in \mathcal{M}$. Given the trajectories $x_i(t), x_j^o(t), t \in \mathbb{R}_{\geq 0}$, of robot $i$ and object $j$, respectively, their corresponding *behaviors* are given by the infinite sequences

$$b_i := (x_i(t), \sigma_i) := (x_i(t_{i,1}), \sigma_{i,1})(x_i(t_{i,2}), \sigma_{i,2}) \ldots,$$
$$b_j^o := (x_j^o(t), \sigma_j^o) := (x_j^o(t_{j,1}^o), \sigma_{j,1}^o)(x_j^o(t_{j,2}^o), \sigma_{j,2}^o) \ldots,$$

with $t_{i,\ell+1} > t_{i,\ell} \geq 0, t_{j,\ell+1}^o > t_{j,\ell}^o \geq 0, \forall \ell \in \mathbb{N}$, representing specific time stamps. The sequences $\sigma_i, \sigma_j^o$ are the services provided to the agent and the object, respectively, over their trajectories, i.e., $\sigma_{i,\ell} \in 2^{\Psi_i}, \sigma_{j,l}^o \in 2^{\Psi_j^o}$ with $\mathcal{A}_i(q_i(t_{i,\ell})) \subset \pi_{k_{i,\ell}}, \sigma_{i,\ell} \in \mathcal{L}_i(\pi_{k_{i,\ell}})$ and $\mathcal{O}_j(x_j^o(t_{j,l}^o)) \subset \pi_{k_{j,l}^o}, \sigma_{j,l}^o \in \mathcal{L}_j^o(\pi_{k_{j,l}^o}), k_{i,\ell}, k_{j,l}^o \in \mathcal{K}, \forall \ell, l \in \mathbb{N}, i \in \mathcal{N}, j \in \mathcal{M}$, where $\mathcal{L}_i$ and $\mathcal{L}_j^o$ are the previously defined labeling functions. The following Lemma then follows:

**Lemma 1.** *The behaviors $b_i, b_j^o$ satisfy formulas $\phi_i, \phi_{o_j}$ if $\sigma_i \models \phi_i$ and $\sigma_j^o \models \phi_j^o$, respectively.*

The control objectives are given as LTL formulas $\phi_i, \phi_j^o$ over $\Psi_i, \Psi_j^o$, respectively, $\forall i \in \mathcal{N}, j \in \mathcal{M}$. The LTL formulas $\phi_i, \phi_j^o$ are satisfied if there exist behaviors $b_i, b_j^o$ of agent $i$ and object $j$ that satisfy $\phi_i, \phi_j^o$. We are now ready to give a formal problem statement:

**Problem 1.** Consider $N$ robotic agents and $M$ objects in $\mathcal{W}$ satisfying

$$\bar{\mathcal{B}}_i(x_i(0), r_i) \subset \pi_{\mathrm{init}(i)}, \bar{\mathcal{B}}_j(x_j^o(0), r_j^o) \subset \pi_{\mathrm{init}_o(j)}, \dot{x}_i(0) = 0,$$
$$\dot{x}_j^o = 0, \forall i \in \mathcal{N}, j \in \mathcal{M},$$

in collision-free initial configurations. Given the disjoint sets $\Psi_i, \Psi_j^o$, $N$ LTL formulas $\phi_i$ over $\Psi_i$ and $M$ LTL formulas $\phi_j^o$ over $\Psi_j^o$, develop a control strategy that achieves behaviors $b_i, b_j^o$ which yield the satisfaction of $\phi_i, \phi_j^o, \forall i \in \mathcal{N}, j \in \mathcal{M}$.

Note that it is implicit in the problem statement the fact that the agents/objects starting in the same region can actually fit without colliding with each other. Technically, it holds that $\mathcal{P}_{s,0}(r_{\pi_k}, [r_i]_{i \in \{i \in \mathcal{N}: \mathrm{init}(i)=k\}}, [r_j^o]_{j \in \{j \in \mathcal{M}: \mathrm{init}_o(j)=k\}}) = \mathsf{True}, \forall k \in \mathcal{K}$.

To solve Problem 1, we need the following feasibility assumption regarding the free space:

**Assumption 2.** For every entity $i$ with bounding radius $x_i^e$ (robot, object, or coupled robots-object system) and every pair of regions $\pi_{k_1}$, $\pi_{k_2}$, $k_1, k_2 \in \mathcal{K}$, $k_1 \neq k_2$, there exists a path $x_\mathsf{p} : [0, \sigma] \rightarrow \mathcal{W}_\mathsf{f}$ such that $\bar{\mathcal{B}}(x_\mathsf{p}(0), r_i^e) \in \pi_{k_1}$, $\bar{\mathcal{B}}(x_\mathsf{p}(\sigma), r_i^e) \in \pi_{k_2}$, and $\bar{\mathcal{B}}(x_\mathsf{p}(t), r_i^e) \in \mathcal{F}_i(\mathcal{AG}(t)) \backslash \{\cup_{k \in \mathcal{K}\backslash\{k_1,k_2\}} \pi_k\}$, for all $t \in [0, \sigma]$.

The aforementioned assumption establishes the capability of designing a control algorithm to achieve navigation among the regions of interest for the robots and the coupled robots-object systems.

## 3. Main results

This section presents the main results of our work. Section 3.1 develops decentralized control protocols that guarantee robot navigation and cooperative object transportation. Section 3.2 uses the aforementioned protocols to abstract the multi-robot-object system to a finite transition system and develops a centralized high-level plan that satisfies the given LTL tasks.

### 3.1. Continuous control design

The first ingredient of our solution is the development of feedback control laws that establish the navigation transition of Definition 1, i.e., robot navigation and cooperative object transportations. As noted before, we do not focus on the grasping/releasing actions (see Definition 2) and we assume control methodologies that can safely accomplish these actions in the region of interest; existing methodologies cam be found, e.g., in [38,39].

The control design is based on the integration of the adaptive control scheme and point world transformation proposed in [31,40], respectively. The former accommodates the uncertain robot and object dynamics and the latter deals with the complex environment obstacles. Since [40] consider single-robot navigation in complex environments, we consider here a *sequential* execution of the navigation and object-transportation transitions.[1] That is, only one entity is allowed to navigate from one RoI to another, viewing the rest of the entities as fixed obstacles.

**(1) Robot Navigation**

Assume that the conditions of Problem 1 hold for some $t_0 \in \mathbb{R}_{\geq 0}$, i.e., all robots and objects are located in some RoI with zero velocity. We first consider the control problem of safe navigation for a robot $i \in \mathcal{N}$ satisfying $i \notin \mathcal{AG}_j(t_0)$ for all $j \in \mathcal{M}$, i.e., not grasping any objects, from region $i_k$ to $i'_k$ (transition $\pi_{i_k} \rightarrow_i \pi_{i'_k}$), with $\bar{\mathcal{B}}_i(x_i(t_0), r_i) \subset \pi_{i_k}$ and $i_k$, $i'_k \in \mathcal{K}$. Let $x^e_i = x_i$. As stated before, the other entities are fixed and viewed as static obstacles by robot $i$. Moreover, to account for safety specifications, we wish the robot to avoid entering the RoI $\pi_k$, $k \in \mathcal{K}\backslash\{i_k, i'_k\}$. Therefore, the free space of robot $i$ becomes $\mathcal{F}_i = (\mathcal{W}_\mathrm{f}\backslash\mathcal{O}_i) \ominus r_i$, where

$$\mathcal{O}_i := \left(\bigcup_{\ell\in\mathcal{T}(\mathcal{AG})\backslash\{i\}} \bar{\mathcal{B}}(x^e_\ell, r^e_\ell)\right) \bigcup \left(\bigcup_{k\in\mathcal{K}\backslash\{i_k,i'_k\}} \pi_k\right)$$

Let the desired navigation goal of the robot be $x_{\mathrm{d,i}} \in \pi_{i'_k}$, which will be provided by the procedure $\mathcal{P}_s$, as explained in the previous section. Next, we use the workspace transformation $\chi = \mathcal{H}(x)$ from [40] that converts the environment to the open unit disk $\mathcal{D} := \mathcal{B}([0,0]^\top, 1)$, and the obstacles (environment obstacles, entities other than robot $i$, and RoI other than $\pi_{i_k}$, $\pi_{i'_k}$) to $\bar{J}$ points $b_\ell \in \mathcal{D}$, $\ell \in \bar{\mathcal{J}} := \{1, \ldots, \bar{J}\}$, robot $i$ to the point $\chi_i := \mathcal{H}(x_i)$, and the goal of robot $i$ to $\chi_\mathrm{d} := \mathcal{H}(x_{\mathrm{d,i}})$. The transformation $\mathcal{H}$ is computed numerically using the Boundary Element Method. Following [40], one can provide a homeomorphism $\mathcal{H}^\star$ that maps $\partial\mathcal{W}$ to the boundary of the unit disk $\partial\mathcal{D}$. Then, the transformation $\mathcal{H}$ satisfies $\mathcal{H}(x) = \mathcal{H}^\star(x)$ for $x \in \partial\mathcal{W}$ and is computed by solving the corresponding boundary value problem subject to the constraints $\int_{\mathcal{O}_i} \frac{\partial\mathcal{H}}{\partial n}\mathrm{d}s = 0$, for all obstacles $i \in \{1, \ldots, J\}$. More details can be found in [40]. In the subsequent analysis we omit the robot index $i$ for ease of presentation.

Let now a constant $\bar{r} > 0$ satisfying

$$\|b_i - b_j\| > 2\bar{r}, \forall i, j \in \bar{\mathcal{J}}, i \neq j$$
$$1 - \|b_i\| > 2\bar{r}, \forall i \in \bar{\mathcal{J}}.$$

The transformed free space of the robot can be defined as $\mathcal{F}_\mathcal{H} := \mathcal{D}\backslash\{b_1, \ldots, b_{\bar{J}}\}$. We define next the set $\bar{\mathcal{J}}_0 := \{0\} \cup \bar{\mathcal{J}}$ as well as the distances $d_\ell : \mathcal{F}_\mathcal{H} \to \mathbb{R}_{\geq 0}$, $\ell \in \bar{\mathcal{J}}_0$, with $d_\ell(\chi) := \|\chi - b_\ell\|^2$, $\forall \ell \in \bar{\mathcal{J}}$, and $d_0(x) := 1 - \|\chi\|^2$. Note that, by keeping $d_\ell(\chi) > 0$, $d_0(\chi) > 0$, we guarantee that $\chi \in \mathcal{F}_\mathcal{H}$ and hence the safety of the robot. We also define the constant

$$\bar{r}_\mathrm{d} := \min\left\{1 - \|\chi_\mathrm{d}\|^2, \min_{\ell\in\bar{\mathcal{J}}}\left\{d_\ell(\chi_\mathrm{d})\right\}\right\} \tag{7}$$

as the minimum distance of the goal to the transformed obstacles/workspace boundary. We revisit now the notion of the 2*nd-order navigation function* from [31]:

**Definition 3.** A 2*nd-order navigation function* is a function $\phi : \mathcal{F}_\mathcal{H} \to \mathbb{R}_{\geq 0}$ of the form

$$\varphi(\chi) := k_1\|\chi - \chi_\mathrm{d}\|^2 + k_2 \sum_{\ell\in\bar{\mathcal{J}}_0} \beta(d_\ell(\chi)),$$

where $\beta : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ is a (at least) twice contin. differentiable function and $k_1, k_2$ are positive constants, with the followings properties:

1. $\beta((0, \tau])$ is strictly decreasing, $\lim_{\mathrm{x}\to 0}\beta(\mathrm{x}) = \infty$, and $\beta(\mathrm{x}) = \beta(\tau)$, $\forall \mathrm{x} \geq \tau$, for some $\tau > 0$,
2. $\varphi(\chi)$ has a global minimum at $\chi = \chi_\mathrm{d} \in \mathrm{int}(\mathcal{F}_\mathcal{H})$ where $\varphi(\chi_\mathrm{d}) = 0$,
3. if $\beta'(d_k(\chi)) \neq 0$ and $\beta''(d_k(\chi)) \neq 0$ for some $k \in \bar{\mathcal{J}}_0$, then $\beta'(d_\ell(\chi)) = \beta''(d_\ell(\chi)) = 0$, for all $\ell \in \bar{\mathcal{J}}_0\backslash\{k\}$.
4. The function $\widetilde{\beta} : (0, \tau) \to \mathbb{R}_{\geq 0}$, with $\widetilde{\beta}(\mathrm{x}) := \beta''(\mathrm{x})\mathrm{x}\sqrt{\mathrm{x}}$ is strictly decreasing.

An example for $\beta$ that satisfies properties (1) and (4), is

$$\beta(\mathrm{x}) := \begin{cases} (6\mathrm{x}^5 - 15\mathrm{x}^4 + 10\mathrm{x}^3)^{-1}, & \mathrm{x} \leq 1 \\ 1, & \mathrm{x} \geq 1, \end{cases} \tag{8}$$

By appropriately choosing $\tau$, only one $\beta(d_\ell(\chi))$, $\ell \in \bar{\mathcal{J}}_0$ affects the robotic agent for each $\chi \in \mathcal{F}_\mathcal{H}$, with $\beta'(d_\ell(\chi_\mathrm{d})) = \beta''(d_\ell(\chi_\mathrm{d})) = 0$. Hence, properties (2) and (3) of Definition 3 are satisfied.

**Proposition 1** ([31]). *By choosing $\tau$ as $\tau \in (0, \min\{\bar{r}^2, \bar{r}_\mathrm{d}\})$, we guarantee that at each $\chi \in \mathcal{F}_\mathcal{H}$ there is at most one $\ell \in \bar{\mathcal{J}}_0$ such that $d_\ell(\chi) \leq \tau$, implying that $\beta'(d_\ell(\chi))$ and $\beta''(d_\ell(\chi))$ are non-zero.*

Intuitively, the obstacles and the workspace boundary have a local region of influence defined by the constant $\tau$.

To compensate for the unknown mass and friction terms of the dynamics (1), we define the estimates $\hat{m}$, $\hat{\alpha}$ of $m$ and $\alpha$ (see Assumption 1), respectively, to be used in the control design.

Given the aforementioned definition, we design a reference signal $v_\mathrm{d} : \mathcal{F}_\mathcal{H} \to \mathbb{R}^2$ for the robot velocity $\dot{x}$ as

$$v_\mathrm{d}(\chi) = -J_\mathcal{H}(x)^{-1}\nabla_{\mathcal{H}(x)}\varphi(\mathcal{H}(x)), \tag{9}$$

where $J_\mathcal{H}(x) := \frac{\partial\mathcal{H}(x)}{\partial x}$ is the nonsingular Jacobian matrix of $\mathcal{H}$. Next, we define the respective velocity error $e_v := \dot{x} - v_\mathrm{d}(x)$, and design the control law as $u : \mathcal{F} \times \mathbb{R}^4 \to \mathbb{R}^2$, with

$$u := u(x, v, \hat{m}, \hat{\alpha}) := -k_\varphi J_\mathcal{H}(x)^\top\nabla_{\mathcal{H}(x)}\varphi(\mathcal{H}(x)) + \hat{m}\dot{v}_\mathrm{d} - \left(k_v + \frac{3}{2}\hat{\alpha}\right)e_v, \tag{10}$$

where $k_\varphi$ is a positive gain, and $\hat{m}$, $\hat{\alpha}$ evolve according to

$$\dot{\hat{m}} := -k_m e_v^\top \dot{v}_\mathrm{d} \tag{11a}$$
$$\dot{\hat{\alpha}} := k_\alpha\|e_v\|^2, \tag{11b}$$

with $k_m$, $k_\alpha$ positive constants. The aforementioned control protocol guarantees safe asymptotic convergence of the robot to its goal from almost all collision-free initial conditions, i.e., except for a set of measure zero, provided that $\tau$ is sufficiently small and $k_\varphi > \frac{\alpha}{2}$ (see Theorem 2 in [31]). Therefore, since the convergence is asymptotic, we conclude that there exists a finite time instant $t_i > t_0$ such that $\bar{\mathcal{B}}(x_i(t_i), r_i) \subset \pi_{i'_k}$, achieving thus the transition $\pi_{i_k} \rightarrow_i \pi_{i'_k}$.

**(2) Cooperative Object Transportation**

We deal now with the control design for the cooperative object transportation problem. Consider an object $j \in \mathcal{M}$ grasped by a team of robots $\mathcal{A}$ at $t_0 \in \mathbb{R}_{\geq 0}$, i.e., $\mathcal{AG}_j(t_0) = \mathcal{A}$, evolving subject to the dynamics (5) and satisfying $\bar{\mathcal{B}}(x^o_j(t_0), r_{\mathcal{A},j}) \subset \pi_{j_k}$, for some $j_k \in \mathcal{K}$. The goal is the transportation of the object to some $\pi_{j'_k}$, $j'_k \in \mathcal{K}$. As in the robot navigation case, we let $x^e_j = x_j$, $r^e_j = r^o_{\mathcal{A},j}$, denoting the entity consisting of object $j$ and the robots $\mathcal{A}$, viewing the rest of the entities as static obstacles. By also aiming to avoid entering the RoI $\pi_k$, for $k \in \mathcal{K}\backslash\{j_k, j'_k\}$, the free space of the entity $\mathcal{F}_j = (\mathcal{W}_\mathrm{f}\backslash\mathcal{O}_j) \ominus r^o_{\mathcal{A},j}$, where

$$\mathcal{O}_i := \left(\bigcup_{\ell\in\mathcal{T}(\mathcal{AG})\backslash\{j\}} \bar{\mathcal{B}}(x^e_\ell, r^e_\ell)\right) \bigcup \left(\bigcup_{k\in\mathcal{K}\backslash\{j_k,j'_k\}} \pi_k\right)$$

Let the desired navigation goal of the object be $x_{\mathrm{d},j}$, provided by the procedure $\mathcal{P}_s$. Similarly to the robot navigation case, we use the transformation $\chi = \mathcal{H}(x)$ to transform the environment to the open unit disk $\mathcal{D}$, the obstacles to points $b_\ell$, the object-robots entity to the point $\chi$, and the object goal to $\chi_\mathrm{d} = \mathcal{H}(x_{\mathrm{d},j})$. Next, by employing the function

[1] The correctness of the multi-robot adaptive control scheme of [31] has only been proven for simple, spherical environments.

$\beta()$, we design a reference signal $v_{\rm d} : \mathcal{F}_{\mathcal{H}} \to \mathbb{R}^2$ for the object velocity that is identical to (9). In addition, we define the adaptation variables $\hat{m}_{\mathcal{A},j}$ and $\hat{\alpha}_{\mathcal{A},j}$ as the estimates of the unknown coupled mass and friction coefficients $m_{\mathcal{A},j}$ and $\alpha_{\mathcal{A},j}$ (see Assumption 1), respectively, and design the control law for the robots $\mathcal{A}$ as $u : \mathcal{F} \times \mathbb{R}^4 \to \mathbb{R}^2$, with

$$\begin{aligned} u_\ell :=& u_\ell(x, v, \hat{m}_{\mathcal{A},j}, \hat{\alpha}_{\mathcal{A},j}) \\ :=& -\mathsf{cf}_\ell \left\{ k_\varphi J_{\mathcal{H}}(x)^\top \nabla_{\mathcal{H}(x)} \varphi(\mathcal{H}(x)) + \hat{m}_{\mathcal{A},j} \dot{v}_{\rm d} - \left( k_v + \frac{3}{2}\hat{\alpha}_{\mathcal{A},j} \right) e_v \right\}, \end{aligned} \tag{12}$$

where $\mathsf{cf}_\ell$ are *load sharing coefficients* satisfying $\mathsf{cf}_\ell \geq 0$, for all $\ell \in \mathcal{A}$, and $\sum_{\ell \in \mathcal{A}} \mathsf{cf}_\ell = 1$; $k_\varphi$ is a positive gain, and $\hat{m}_{\mathcal{A},j}$, $\hat{\alpha}_{\mathcal{A},j}$ evolve according to (11). Note that (12) is decentralized; each robot in $\mathcal{A}$ computes its own control signal by measuring its own and the object's state without communicating with the rest of the robots; the load-sharing coefficients $\mathsf{cf}_\ell$ can be transmitted off-line.

By invoking the property $\sum_{\ell \in \mathcal{A}} \mathsf{cf}_\ell = 1$ and the fact that the object-robots system is converted to a point by the transformation $\mathcal{H}$, we guarantee, similar to the robot navigation case, the safe, asymptotic convergence of the object-robots entity to $\pi_{j'_k}$ from almost all collision-free initial conditions, provided that $k_\varphi > \frac{\alpha}{2}$. Hence, there exists a finite time instant $t_j > t_0$ such that $\bar{\mathcal{B}}(x_j(t_j), r^o_{\mathcal{T},j}) \subset \pi_{j'_k}$, achieving thus the transition $\pi_{i_k} \xrightarrow{T}_{\mathcal{A},j} \pi_{j'_k}$.

As opposed to many previous works, such as [30], where the transitions are restricted among neighboring regions in fully partitioned workspaces, the proposed robot navigation and cooperative object transportation are achieved among any pair of regions of interest.

**Remark 1.** The designed control algorithms guarantee collision-free robot navigation and object transportation from almost all initial conditions. That is, there exists a set of initial conditions that can lead to local minima configurations, where the robots are trapped in a deadlock. However, this set has zero measure and it is hence practically impossible for the robots to start in such conditions.[2] The aforementioned guarantees rely on Assumption 2, i.e., the existence of a collision-free path among any pairs of regions of interest, and the fact that only one entity (robot or coupled robots-object system) navigates at a time. We note that the control algorithms can be adjusted for simultaneous robot navigation/object transportation, but convergence guarantees would require additional assumptions on the distance between the obstacles and the regions of interest. In particular, one could apply the leader-follower prioritization scheme of [31]; in this setting, pre-assigned follower robots give priority to a leader, which treats them as static obstacles. However, convergence guarantees require the regions of interest and obstacles to be sufficiently distant from each other, which creates conservative restrictions on the type of workspaces that the algorithms can be applied to.

### 3.2. High-level plan generation

The second part of the solution is the derivation of a high-level plan that satisfies the given LTL formulas $\phi_i$ and $\phi^o_j$. Thanks to (i) the proposed control laws that allow robot transitions and object transportations $\pi_k \to_i \pi_{k'}$ and $\pi_k \xrightarrow{T}_{\mathcal{A},j} \pi_{k'}$, respectively, and (ii) the off-the-self control laws that guarantee grasp and release actions $i \xrightarrow{g} j$ and $i \xrightarrow{r} j$, we can abstract the behavior of the multi-robot-object system using a finite transition system as presented in the sequel.

**Definition 4.** The coupled behavior of the overall system of all the $N$ agents and $M$ objects is modeled by the transition system

$$\mathcal{TS} = (\Pi_s, \Pi_s^{\rm init}, \to_s, \mathcal{AG}, \Psi, \mathcal{L}, \Lambda, P_s, \chi)$$

where

(i) $\Pi_s \subset \bar{\Pi} \times \bar{\Pi}^o \times (2^{\mathcal{N}_0})^M$ is the set of states; $\bar{\Pi} := \Pi_1 \times \cdots \times \Pi_N$ and $\bar{\Pi}^o := \Pi^o_1 \times \cdots \times \Pi^o_M$ are the set of states-regions that the agents and the objects can be at, with $\Pi_i = \Pi^o_j = \Pi, \forall i \in \mathcal{N}, j \in \mathcal{M}$;

By defining $\bar{\pi} := \left(\pi_{k_1}, \ldots, \pi_{k_N}\right), \bar{\pi}_o := (\pi_{k^o_1}, \ldots, \pi_{k^o_M})$, then the coupled state $\pi_s := (\bar{\pi}, \bar{\pi}_o, \mathcal{AG})$ belongs to $\Pi_s$, i.e., $(\bar{\pi}, \bar{\pi}_o, \mathcal{AG}) \in \Pi_s$ if

(a) $\mathcal{P}_{s,0}\left(r_{\pi_k}, [r_i]_{i \in \{i \in \mathcal{N}: k_i = k\}}, [r^o_j]_{j \in \{j \in \mathcal{M}: k^o_j = k\}}\right) = \top$, i.e., the respective robots and objects fit in the region, $\forall k \in \mathcal{K}$,
(b) $k_i = k^o_j$ for all $i \in \mathcal{N}, j \in \mathcal{M}$ such that $i \in \mathcal{AG}_j$, i.e., a robot must be in the same region with the object it grasps,

(ii) $\Pi_s^{\rm init} \subset \Pi_s$ is the initial set of states at $t = 0$, which, owing to (i), satisfies the conditions of Problem 1,
(iii) $\to_s \subset \Pi_s \times \Pi_s$ is a transition relation defined as follows: given the states $\pi_s, \widetilde{\pi}_s \in \Pi$, with

$$\begin{aligned} \pi_s :=& (\bar{\pi}, \bar{\pi}_o, \mathcal{AG}) \\ :=& (\pi_{k_1}, \ldots, \pi_{k_N}, \pi_{k^o_1}, \ldots, \pi_{k^o_M}, \mathcal{AG}_1, \ldots, \mathcal{AG}_M), \\ \widetilde{\pi}_s :=& (\widetilde{\bar{\pi}}, \widetilde{\bar{\pi}}_o, \widetilde{\mathcal{AG}}) \\ :=& (\pi_{\widetilde{k}_1}, \ldots, \pi_{\widetilde{k}_N}, \pi_{\widetilde{k}^o_1}, \ldots, \pi_{\widetilde{k}^o_1}, \widetilde{\mathcal{AG}}_1, \ldots, \widetilde{\mathcal{AG}}_M), \end{aligned} \tag{13}$$

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

(a) $\nexists i \in \mathcal{N}, j \in \mathcal{M}$ such that $i \in \mathcal{AG}_j$, $i \notin \widetilde{\mathcal{AG}}_j$ (or $i \notin \mathcal{AG}_j$, $i \in \widetilde{\mathcal{AG}}_j$), and $k_i \neq \widetilde{k}_i$, i.e., there are no simultaneous grasp/release and navigation actions,
(b) $\nexists i \in \mathcal{N}, j \in \mathcal{M}$ such that $i \in \mathcal{AG}_j$, $i \notin \widetilde{\mathcal{AG}}_j$ (or $i \notin \mathcal{AG}_j$, $i \in \widetilde{\mathcal{AG}}_j$), and $k_i = k^o_j \neq \widetilde{k}_i = \widetilde{k}^o_j$, i.e., there are no simultaneous grasp/release and transportation actions,
(c) $\nexists i \in \mathcal{N}, j, j' \in \mathcal{M}$, with $j \neq j'$, such that $i \in \mathcal{AG}_j$ and $i \in \widetilde{\mathcal{AG}}_{j'}$, i.e., there are no simultaneous grasp and release actions,
(d) $\nexists j \in \mathcal{M}$ such that $k^o_j \neq \widetilde{k}^o_j$ and $i \notin \mathcal{AG}_j, \forall i \in \mathcal{N}$ (or $i \notin \widetilde{\mathcal{AG}}_j, \forall i \in \mathcal{N}$), i.e., there is no transportation of a non-grasped object,
(e) $\nexists j \in \mathcal{M}, \mathcal{T} \subseteq \mathcal{N}$ such that $k^o_j \neq \widetilde{k}^o_j$ and $\Lambda(m^o_j, \zeta_{\mathcal{T}}) = \bot$, with $(i \in \mathcal{AG}_j, i \in \widetilde{\mathcal{AG}}_j) \Leftrightarrow i \in \mathcal{T}$, i.e., the agents grasping an object are powerful enough to transfer it,

(iv) $\Psi := \bar{\Psi} \cup \bar{\Psi}^o$ with $\bar{\Psi} = \bigcup_{i \in \mathcal{N}} \Psi_i$ and $\bar{\Psi}^o = \bigcup_{j \in \mathcal{M}} \Psi^o_j$, are the atomic propositions of the agents and objects, respectively, as defined in Section 2.
(v) $\mathcal{L} : \Pi_s \to 2^{\Psi}$ is a labeling function defined as follows: Given a state $\pi_s$ as in (13) and $\psi_s := \left(\bigcup_{i \in \mathcal{N}} \psi_i\right) \bigcup \left(\bigcup_{j \in \mathcal{M}} \psi^o_j\right)$ with $\psi_i \in 2^{\Psi_i}, \psi^o_j \in 2^{\Psi^o_j}$, then $\psi_s \in \mathcal{L}(\pi_s)$ if $\psi_i \in \mathcal{L}_i(\pi_{k_i})$ and $\psi^o_j \in \mathcal{L}^o_j(\pi_{k^o_j}), \forall i \in \mathcal{N}, j \in \mathcal{M}$.
(vi) $\Lambda$ and $P_s$ as defined in Section 2.
(vii) $\chi : (\to_s) \to \mathbb{R}_{\geq 0}$ is a function that assigns a cost to each transition $\pi_s \to_s \widetilde{\pi}_s$. This cost might be related to the distance of the robots' regions in $\pi_s$ to the ones in $\widetilde{\pi}_s$, combined with the cost efficiency of the robots involved in transport tasks (according to $\zeta_i, i \in \mathcal{N}$).

We build algorithmically the transition system $\mathcal{TS}$ by the following the conditions of Definition 4. In particular, the proposed algorithm constructs all possible states $\pi_s$ and checks whether they satisfy the conditions (i)(a) and (i)(b) of Definition 4; the ones that do, form the states of the transition system. Subsequently, it constructs all possible edges (transitions) among these states and checks whether they satisfy conditions (iii)(a)–(e), keeping the ones that do. Note that, as mentioned before, although the object masses are not known accurately and cannot hence be used for control-design purposes, we implicitly assume the existence of a mechanism that can determine which robots can transport which objects. Hence, the proposed algorithm can check whether an edge satisfies condition (iii)(e). Finally, we conclude that all conditions of Definition 4 are guaranteed for the resulted transition system.

[2] If the initial conditions are chosen randomly, the probability of resulting in a set of zero measure is zero.

Next, we form the global LTL formula $\phi := (\wedge_{i\in\mathcal{N}}\phi_i)\wedge(\wedge_{j\in\mathcal{M}}\phi_j^o)$ over the set $\Psi$. Given $\phi$, we define the *language* $\texttt{Words}(\phi) = \{\sigma \in (2^{\Psi})^{\omega} | \sigma \vDash \phi\}$, where $\vDash \subseteq (2^{\Psi})^{\omega} \times \phi$ is the satisfaction relation, as the set of infinite words $\sigma \in (2^{\Psi})^{\omega}$ that satisfy the LTL formula $\phi$. Then, we translate $\phi$ to a Büchi Automaton $\mathcal{BA}$ defined over $(2^{\Psi})^{\omega}$. The Büchi Automaton is defined as follows [34]:

**Definition 5** (*NBA*). A *Nondeterministic Büchi Automaton* (NBA) $\mathcal{BA}$ is defined as a tuple $B = (\Pi_B, \Pi_B^0, \Sigma, \rightarrow_B, \Pi_B^F)$, where $\Pi_B$ is the set of states, $\Pi_B^0 \subseteq \Pi_B$ is a set of initial states, $\Sigma = 2^{\Psi}$ is the alphabet, $\rightarrow_B \subseteq \Pi_B \times \Psi \times \Pi_B$ is the transition relation, and $\Pi_B^F \subseteq \Pi_B$ is a set of accepting/final states.

Given the $\mathcal{TS}$ and the NBA $\mathcal{BA}$ that corresponds to the LTL $\phi$, we can now define the *Product Büchi Automaton* (PBA) $\mathcal{PBA} := \mathcal{TS} \otimes \mathcal{BA}$, as follows [34]:

**Definition 6** (*PBA*). Given the product transition system $\mathcal{TS} = (\Pi_s, \Pi_s^{\text{init}}, \rightarrow_s, \mathcal{AG}, \Psi, \mathcal{L}, \Lambda, P_s, \chi)$ and the NBA $B = (\Pi_B, \Pi_B^0, \Psi, \rightarrow_B, \Pi_B^F)$, we can define the *Product Büchi Automaton* $\mathcal{PBA} := \mathcal{PTS} \otimes \mathcal{BA}$ as a tuple $\mathcal{PBA} = (\Pi_P, \Pi_P^0, \longrightarrow_P, \Pi_P^F)$ where (a) $\Pi_P = \Pi_s \times \Pi_B$ is the set of states; (b) $\Pi_P^0 = \Pi_s^{\text{init}} \times \Pi_B^0$ is a set of initial states; (c) $\longrightarrow_P \subseteq \Pi_P \times \Sigma \times \Pi_P$ is the transition relation defined by the rule: $\frac{(\pi_s \rightarrow_s \pi_s')\wedge(\pi_B \xrightarrow{\mathcal{L}(\pi_s)} \pi_B')}{\pi_P=(\pi_s,\pi_B)\longrightarrow_P \pi_P'=(\pi_s',\pi_B')}$. Transition from state $\pi_P \in \Pi_P$ to $\pi_P' \in \Pi_P$, is denoted by $(\pi_P, \pi_P') \in \longrightarrow_P$, or $\pi_P \longrightarrow_P \pi_P'$; (d) $\chi_P(\pi_{\mathrm{P}}, \pi_{\mathrm{p}}') = \chi(\pi_s, \pi_s')$, where $\pi_P = (\pi_s, \pi_B)$ and $\pi_{\mathrm{p}}' = (\pi_s', \pi_B')$; and (e) $\Pi_P^F = \Pi_s \times \Pi_B^F$ is a set of accepting/final states.

Given $\phi$ and the PBA, an infinite path $\pi_{\text{pl}} := \pi_{s,1}\pi_{s,2}\ldots$ of a $\mathcal{TS}$ satisfies $\phi$ if and only if $\texttt{trace}(\pi_{\text{pl}}) \in \texttt{Words}(\phi)$, which is equivalently denoted by $\pi_{\text{pl}} \vDash \phi$, where $\texttt{trace}(\pi_{\text{pl}}) \in (2^{\Psi})^{\omega}$ is defined as $\texttt{trace}(\pi_{\text{pl}}) = \mathcal{L}(\pi_{s,1})\mathcal{L}(\pi_{s,2})\ldots$. Specifically, if there is a path satisfying $\phi$, then there exists a path $\pi_{\text{pl}} \vDash \phi$ that can be written in a finite representation, called prefix-suffix structure, i.e., $\pi_{\text{pl}} = \pi_{\text{pl}}^{\text{pre}}[\pi_{\text{pl}}^{\text{suf}}]^{\omega}$, where the prefix part $\pi_{\text{pl}}^{\text{pre}}$ is executed only once followed by the indefinite execution of the suffix part $\pi_{\text{pl}}^{\text{suf}}$. The prefix part $\pi_{\text{pl}}^{\text{pre}}$ is the projection of a finite path $p^{\text{pre}}$ that lives in $\Pi_P$ onto $\Pi_s$.

Computing a plan $\pi_{\text{pl}}$ is typically accomplished by applying graph-search methods to the PBA. Specifically, to generate a motion plan $\pi_{\text{pl}}$ that satisfies $\phi$, the PBA is viewed as a weighted directed graph $\mathcal{G}_P = \{\mathcal{V}_P, \mathcal{E}_P, w_P\}$, where the set of nodes $\mathcal{V}_P$ is indexed by the set of states $\Pi_P$, the set of edges $\mathcal{E}_P$ is determined by the transition relation $\longrightarrow_P$, and the weights assigned to each edge are determined by the function $\chi_P$. Then, to find the optimal plan $\tau \vDash \phi$, shortest paths towards final states and shortest cycles around them are computed. More details about this approach can be found in [41,42] and the references therein. While any of the aforementioned methodologies could be used, in this work we employ STyLuS*, an algorithm that is designed to solve complex temporal planning problems in large-scale multi-robot systems and has been shown to achieve significantly lower complexity, in terms of running time and memory, than standard graph-search methods [43]. Specifically, STyLuS* is a sampling-based method that builds incrementally trees that approximate the state-space and transitions of the product automaton and does not require sophisticated graph-search techniques. Technically, STyLuS* builds a tree $\mathcal{G}_T = \{\mathcal{V}_T, \mathcal{E}_T, \texttt{Cost}\}$ first, where $\mathcal{V}_T \subseteq \Pi_P$ is the set of nodes, $\mathcal{E}_T$ is the set of edges, and $\texttt{Cost}$ is defined as per $\chi_P$, determining the cost of reaching a tree node from its root. This tree is rooted at the initial state $\pi_P^0$ and is used for the synthesis of the prefix part. The tree is constructed incrementally, in a sampling-based fashion, and its construction terminates after a user-specified number of iterations $n_{\max}$. Then, we compute paths in the constructed tree structure that connect the root to the detected, if any, final states. These paths correspond to prefix parts of candidate feasible paths. To construct the corresponding suffix paths, new trees are built, similarly, rooted at the previously detected final states aiming to compute cycles around the tree roots. Among all the detected prefix-suffix paths, STyLuS* returns the one with the minimum cost. As it was shown in [43], STyLuS* is probabilistically complete and asymptotically optimal; that is, the probability of finding a feasible and the optimal solution converges to 1 as $n_{\max} \rightarrow \infty$.

Finally, note that the constructed trees explore the state space of the product automaton and, therefore, the designed prefix-suffix path is defined as an infinite sequence of product automaton states. By projecting it onto the state-space of the transition system $\mathcal{TS}$, we obtain a high-level prefix-suffix plan defined as a sequence of states $\pi_{\text{pl}} := \pi_{s,1}\pi_{s,2}\cdots \vDash \phi$. The corresponding sequence of atomic propositions is $\psi_{\text{pl}} = \texttt{trace}(\pi_{\text{pl}}) = \psi_{s,1}\psi_{s,2}\ldots$, with

$$\pi_{s,\ell} := (\bar{\pi}_\ell, \bar{\pi}_{o,\ell}, \bar{w}_\ell) \in \Pi_s, \forall \ell \in \mathbb{N},$$
$$\psi_{s,\ell} := \Big(\bigcup_{i\in\mathcal{N}} \psi_{i,\ell}\Big) \bigcup \Big(\bigcup_{j\in\mathcal{M}} \psi_{j,\ell}^o\Big) \in 2^{\Psi}, \mathcal{L}(\pi_{s,\ell}), \forall \ell \in \mathbb{N},$$

where

- $\bar{\pi}_\ell := \pi_{k_{1,\ell}}, \pi_{k_{2,\ell}}, \ldots$ with $k_{i,\ell} \in \mathcal{K}, \forall i \in \mathcal{N}$,
- $\bar{\pi}_{o,\ell} := \pi_{k_{1,\ell}^o}, \pi_{k_{2,\ell}^o}, \ldots$ with $k_{j,\ell}^o \in \mathcal{K}, \forall j \in \mathcal{M}$,
- $\bar{w}_\ell := w_{1,\ell}, w_{2,\ell}, \ldots$ with $w_{i,\ell} \in \mathcal{AG}_i, \forall i \in \mathcal{N}$,
- $\psi_{i,\ell} \in 2^{\Psi_i}, \mathcal{L}_i(\pi_{k_{i,\ell}}), \forall i \in \mathcal{N}$,
- $\psi_{j,\ell}^o \in 2^{\Psi_j^o}, \mathcal{L}_j^o(\pi_{k_{j,\ell}^o}), \forall j \in \mathcal{M}$.

The path $\pi_{\text{pl}}$ is then projected to the individual sequences of the regions $\pi_{k_{j,1}^o}\pi_{k_{j,2}^o}\ldots$ for each object $j \in \mathcal{M}$, as well as to the individual sequences of the regions $\pi_{k_{i,1}}\pi_{k_{i,2}}\ldots$ and the boolean grasping variables $w_{i,1}w_{i,2}\ldots$ for each robot $i \in \mathcal{N}$. The aforementioned sequences determine the behavior of robot $i \in \mathcal{N}$, i.e., the sequence of actions (transition, transportation, grasp, release or stay idle) it must take.

By the definition of $\mathcal{L}$ in Definition 4, we obtain that $\psi_{i,\ell} \in \mathcal{L}_i(\pi_{k_{i,\ell}}), \psi_{j,\ell}^o \in \mathcal{L}_j^o(\pi_{k_{j,\ell}^o}), \forall i \in \mathcal{N}, j \in \mathcal{M}, \ell \in \mathbb{N}$. Therefore, since $\phi = (\wedge_{i\in\mathcal{N}}\phi_i) \wedge (\wedge_{j\in\mathcal{M}}\phi_{o_j})$ is satisfied by $\psi$, we conclude that $\psi_{i,1}\psi_{i,2}\cdots \vDash \phi_i$ and $\psi_{j,1}^o\psi_{j,2}^o\cdots \vDash \phi_j^o, \forall i \in \mathcal{N}, j \in \mathcal{M}$.

The sequences $\pi_{k_{i,1}}\pi_{k_{i,2}}\ldots$, $\psi_{i,1}\psi_{i,2}\ldots$ and $\pi_{k_{j,1}^o}\pi_{k_{j,2}^o}\ldots, \psi_{j,1}^o\psi_{j,2}^o\ldots$ over $\Pi, 2^{\Psi_i}$ and $\Pi, 2^{\Psi_j^o}$, respectively, produce the trajectories $q_i(t)$ and $x_j^o(t), \forall i \in \mathcal{N}, j \in \mathcal{M}$. The corresponding behaviors are

$$\beta_i = (q_i(t), \sigma_i) = (q_i(t_{i,1}), \sigma_{i,1})(q_i(t_{i,2}), \sigma_{i,2})\ldots$$
$$\beta_j^o = (x_j^o(t), \sigma_j^o) = (x_j^o(t_{j,1}^o), \sigma_{j,1}^o)(x_j^o(t_{j,2}^o), \sigma_{j,2}^o)\ldots$$

according to Section 2.2, with $\mathcal{A}_i(q_i(t_{i,\ell})) \subset \pi_{k_{i,\ell}}, \sigma_{i,\ell} \in \mathcal{L}_i(\pi_{k_{i,\ell}})$ and $\mathcal{O}_j(x_{o_j}(t_{o_{j,m}})) \in \pi_{k_{j,\ell}^o}, \sigma_{j,\ell}^o \in \mathcal{L}_j^o(\pi_{k_{j,\ell}^o})$. Thus, it is guaranteed that $\sigma_i \vDash \phi_i, \sigma_j^o \vDash \phi_j^o$ and consequently, the behaviors $\beta_i$ and $\beta_j^o$ satisfy the formulas $\phi_i$ and $\phi_j^o$, respectively, $\forall i \in \mathcal{N}, j \in \mathcal{M}$. The aforementioned reasoning is summarized in the next theorem:

**Theorem 1.** *The execution of the path $(\pi_{\text{pl}}, \psi_{\text{pl}})$ of $\mathcal{TS}$ guarantees behaviors $\beta_i, \beta_j^o$ that yield the satisfaction of $\phi_i$ and $\phi_j^o$, respectively, $\forall i \in \mathcal{N}, j \in \mathcal{M}$, providing, therefore, a solution to Problem 1.*

**Remark 2.** Note that the high-level planner and low-level controllers of Section 3.1 are only implicitly coupled. In particular, the abstracted transition system $\mathcal{TS}$ of Definition 4 for the coupled multi-robot-object system is well-defined because of the theoretical guarantees provided by the designed low-level controllers for robot navigation and object transportation, as well as the employed assumption on the grasping/releasing actions. Consequently, the actions provided by the planner are guaranteed to be executed successfully by the low-level control, leading to the satisfaction of the LTL specification.

### 3.3. Time complexity of high-level plan

We elaborate now on the time complexity of the generation of the high-level plan, including the time to construct the transition system as well as the time required to derive the required plan.

The transition system is built by constructing the nodes and edges. Regarding the nodes, all possible combinations of regions for robots, objects and grasping variables are checked for satisfaction of the conditions of Definition 4. Following Definition 4, there are $Q_n = 2^{M(N+1)}NM$ such nodes. Subsequently, all possible edges (transitions) among the constructed nodes $|\Pi_s| \leq Q_n$ are checked for satisfaction of the respective conditions of Definition 4, which constitutes $|\Pi_s|^2$ edges. However, self-edges always satisfy the required conditions, and further note that $\pi_s \to \pi_{s'}$ automatically implies that $\pi_{s'} \to \pi_s$. Therefore, the number of edges that have to be checked is reduced to $Q_e = \frac{|\Pi_s|(|\Pi_s|-1)}{2}$. Therefore, we conclude that the time complexity of the transition-system construction is $O(Q_n + Q_e) = O\left(2^{M(N+1)}NM + \frac{|\Pi_s|(|\Pi_s|-1)}{2}\right) \leq O\left(Q_n + \frac{Q_n(Q_n-1)}{2}\right) = O\left(\frac{Q_n(Q_n+1)}{2}\right)$.

The time complexity of the high-level planner, STyLuS*, is reported in our prior work [43]. Specifically, the memory needed to store the tree $\mathcal{G}_T$, constructed by the planner is $O(|\mathcal{V}_T|)$, where $|\mathcal{V}_T|$ is the number of the tree nodes. Since STyLuS* is a sampling-based approach with asymptotic completeness/optimality guarantees, we report its time complexity, to construct $\mathcal{G}_T$, per iteration. First, it samples a TS state at every iteration. The time complexity of the sampling strategy per iteration is $O(|\Pi_B|+|\Pi_s|+|\to_s|+|\Pi_s|\log(|\Pi_s|))$, where $|\Pi_B|, |\Pi_s|$, and $|\to_s|$ capture the number of NBA states, TS states, and TS edges. Observe that the time complexity increases as the task complexity increase and as the TS state space and the number of transitions increase. Second, STyLuS* examines if the tree can be extended towards the sampled TS state. The time complexity of this step is $O(|\mathcal{V}_T|)$. Third, if the tree is extended towards the sampled state, the rewiring step follows. In this step, STyLuS* examines if existing tree nodes can be rewired, using the new node, to optimize the tree structure. This step is critical to compute optimal paths and its time complexity is $O(|\mathcal{V}_T|)$.

## 4. Numerical experiments

In this section, we provide two case studies in an obstacle-cluttered office environment as well as a scalability analysis of the high-level planner. We choose the atomic propositions for the robots and objects as $\Psi_i = \{``i\text{-}\pi_1", \ldots, ``i\text{-}\pi_K"\}$ and $\Psi^o_j = \{``O_j\text{-}\pi_1", \ldots, ``O_j\text{-}\pi_K"\}$, respectively, for $i \in \mathcal{N}$, $j \in \mathcal{M}$, indicating their presence in the regions of interest. For the constructed transition systems, we set the cost $\chi$ as the Euclidean distance among the RoI contained in the nodes of the transitions.

For the continuous control design, we choose the robot and object dynamics of the form (1) and (2), respectively, with $f_i(\cdot) = \mathsf{f}_{i,1}\sin(\mathsf{f}_{i,2}(x_{i_1}+x_{i_2}))F(\dot{x}_i)\dot{x}_i$, with $F(\dot{x}_i) = \text{diag}\{[\exp(-\text{sgn}(\dot{x}_i)\dot{x}_i)+1]_{i\in\{1,2\}}\}$, $f^o_j(\cdot) = \mathsf{f}_{j,1}\sin(\mathsf{f}_{j,2}(x^o_{j_1}+x^o_{j_2}))F(\dot{x}^o_j)\dot{x}^o_j$, with $F(\dot{x}^o_j) = \text{diag}\{[\exp(-\text{sgn}(\dot{x}^o_j)\dot{x}^o_j)+1]_{i\in\{1,2\}}\}$, where we denote $(x_{i_1}, x_{i_2}) = x_i$, $(x^o_{j_1}, x^o_{j_2}) = x^o_j$, $i \in \mathcal{N}$, $j \in \mathcal{M}$. The constants $\mathsf{f}_{i,1}$ and $\mathsf{f}_{i,2}$, as well as the robot masses $m_i$ are randomly chosen in the interval $[0.5, 2]$, for $i \in \mathcal{N}$. Similarly, the constants $\mathsf{f}^o_{j,1}$, $\mathsf{f}^o_{j,2}$, and object masses $m^o_j$ are randomly chosen in the interval $[0.1, 0.5]$, for $j \in \mathcal{M}$. We further choose $\bar{r} = 0.1$ and a variation of (8) for $\beta$ with $\tau = \bar{r}^2$. Finally, we set the control gains to $k_1 = 0.01$, $k_2 = 5$, $k_\phi = 1$, and $k_m = k_\alpha = 0.01$.

### 4.1. Case study I

In this case study, we consider $N = 3$ robots, $K = 4$ regions and $M = 2$ objects. The regions of interest are centered at $(88, -280)$m, $(100, -160)$m, $(200, -130)$m, and $(250, -285)$m, with radius equal to $r_{\pi_k} = 4$m, $k \in \{1, \ldots, 4\}$. The robots' and object' spherical volume's radii are chosen as $r_i = 0.75$m, $i \in \{1, 2, 3\}$ and $r^o_j = 0.2$m, $j \in \{1, 2\}$ respectively. The power capabilities of the robots are $(\zeta_1, \zeta_2, \zeta_3) = (2, 3, 4)$, respectively, and the power required for each object is 5,6, respectively. Initially, the robots are located in regions $\pi_{init(1)} = \pi_1$, $\pi_{init(2)} = \pi_3$, and $\pi_{init(3)} = \pi_4$, respectively, whereas the objects are located in regions $\pi_{init_o(1)} = \pi_2$, $\pi_{init_o(2)} = \pi_1$, respectively. Fig. 2 depicts the considered environment and the initial configuration of the multi-robot-object system.

The resulting transition system consists of 3112 reachable states and 154,960 transitions and it was created within 102.6 s. We set the following formula for the multi-robot-object system:

$$\phi = \Box\Diamond\bar{\psi}_1 \wedge \Box\Diamond\bar{\psi}_2 \wedge \Box\Diamond\bar{\psi}_2 \wedge \Diamond\bar{\psi}_4 \wedge (\neg\bar{\psi}_4 \mathcal{U} \bar{\psi}_5) \tag{14}$$

with

- $\bar{\psi}_1 = \{``1\text{-}\pi_1", ``2\text{-}\pi_1", ``3\text{-}\pi_1", ``O_1\text{-}\pi_1", ``O_2\text{-}\pi_4"\}$
- $\bar{\psi}_2 = \{``1\text{-}\pi_1", ``2\text{-}\pi_3", ``3\text{-}\pi_1", ``O_1\text{-}\pi_1", ``O_2\text{-}\pi_3"\}$
- $\bar{\psi}_3 = \{``1\text{-}\pi_3", ``2\text{-}\pi_3", ``3\text{-}\pi_3", ``O_1\text{-}\pi_4", ``O_2\text{-}\pi_3"\}$
- $\bar{\psi}_4 = \{``1\text{-}\pi_2", ``2\text{-}\pi_4", ``3\text{-}\pi_1", ``O_1\text{-}\pi_4", ``O_2\text{-}\pi_2"\}$
- $\bar{\psi}_5 = \{``1\text{-}\pi_4", ``2\text{-}\pi_4", ``3\text{-}\pi_2", ``O_1\text{-}\pi_3", ``O_2\text{-}\pi_4"\}$

In words, the mission specification in (14) requires the robots and objects to satisfy $\bar{\psi}_1$, $\bar{\psi}_2$, and $\bar{\psi}_3$ infinitely often, satisfy $\bar{\psi}_4$ eventually, and satisfy $\bar{\psi}_5$ before satisfying $\bar{\psi}_4$; $\bar{\psi}_1$ indicates the visit of all robots and object 1 to $\pi_1$ and of object 2 to $\pi_4$; $\bar{\psi}_2$ indicates the visit of robots 1, 3 and object 1 to $\pi_1$, and of robot 2 and object 2 to $\pi_3$; $\bar{\psi}_3$ indicates the visit of all robots and object 2 to $\pi_3$, and of object 1 to $\pi_4$; $\bar{\psi}_4$ indicates the visit of robot 1 and object 2 to $\pi_2$, of robot 2 and object 1 to $\pi_4$, and of robot 3 to $\pi_1$; finally, $\bar{\psi}_5$ indicates the visit of robots 1, 2 and object 2 to $\pi_4$, of robot 3 to $\pi_2$, and of object 1 to $\pi_3$.

The LTL formula in (14) corresponds to a NBA with 6 states – among which one is a final state – and 18 transitions. STyLuS* found the first feasible prefix and suffix path within 73.8 s and 38.4 s on average. The action path of the robots is depicted in Tables 1–2, starting from $\pi_{s,1}$ and satisfying $\{``1\text{-}\pi_1", ``2\text{-}\pi_2", ``3\text{-}\pi_4", ``O_1\text{-}\pi_2", ``O_2\text{-}\pi_1"\}$; the suffix states are indicated with a superscript " $\star$ ".

We further illustrate the continuous control design. In particular, we consider the actions of robot navigation and object transportation. Firstly, we examine the navigation of robot 1 from $\pi_1$ to $\pi_3$. The results are depicted in Figs. 3, 4. The left part of Fig. 3 shows the trajectory of robot 1 in the environment, where the obstacles and boundary have absorbed the spherical volume of the robot; the right part of Fig. 3 shows the trajectory of robot 1 in the transformed point world, where the obstacles are represented by points. Fig. 4 depicts the control input $u_1(t)$ (left) and the evolution of the adaptation signals $\hat{m}_1(t)$, $\hat{\alpha}_1(t)$ (right).

We further examine the transportation of object 1 by robots 1 and 2 from $\pi_2$ to $\pi_4$, where we chose the load-sharing coefficients $\mathsf{cf}_1 = \mathsf{cf}_2 = 0.5$. The results are depicted in Figs. 5, 6. The left part of Fig. 5 shows the trajectory of the coupled object-robots system in the environment, where the obstacles and boundary have absorbed the its spherical volume; the right part of Fig. 3 shows the trajectory of the coupled system in the transformed point world, where the obstacles are represented by points. Fig. 4 depicts the control input $u_1(t)$ (left) and the evolution of the adaptation signals $\hat{m}_1(t)$, $\hat{\alpha}_1(t)$ (right) of the robot 1, which are identical to the ones of robot 2.

### 4.2. Case study II

In this case study, we consider $N = 4$ robots, $K = 3$ regions and $M = 3$ objects. The regions of interest are centered at $(88, -280)$m, $(100, -160)$m, and $(200, -130)$m. The radii of the regions and robots' and objects' bounding volume are set as in the previous case study. The power capabilities of the robots are set as $(\zeta_1, \zeta_2, \zeta_3, \zeta_4) = (2, 3, 4, 4)$, respectively, and the power required for each object is 5, 8, 10.5, respectively. Initially, the robots are located in regions $\pi_{init(1)} = \pi_1$, $\pi_{init(2)} = \pi_3$, $\pi_{init(3)} = \pi_2$, and $\pi_{init(4)} = \pi_2$, respectively, whereas the objects are located in regions $\pi_{init_o(1)} = \pi_2$, $\pi_{init_o(2)} = \pi_1$, and $\pi_{init_o(3)} = \pi_3$, respectively.

The resulting transition system consists of 24,012 reachable states and $1,805,202$ transitions and it was created within 1.52 h. We set the following formula for the multi-robot-object system:

$$\begin{aligned}\phi =& \Box\Diamond(\bar{\psi}_1 \vee \bar{\psi}_2) \wedge \Box\Diamond(\bar{\psi}_3) \wedge \Box\Diamond(\bar{\psi}_4) \wedge (\neg\bar{\psi}_1 \mathcal{U} \bar{\psi}_5) \\ &\wedge \Box(\neg\bar{\psi}_6) \wedge \Diamond(\bar{\psi}_7 \vee \bar{\psi}_8) \wedge \Box\Diamond\bar{\psi}_5\end{aligned} \tag{15}$$

with

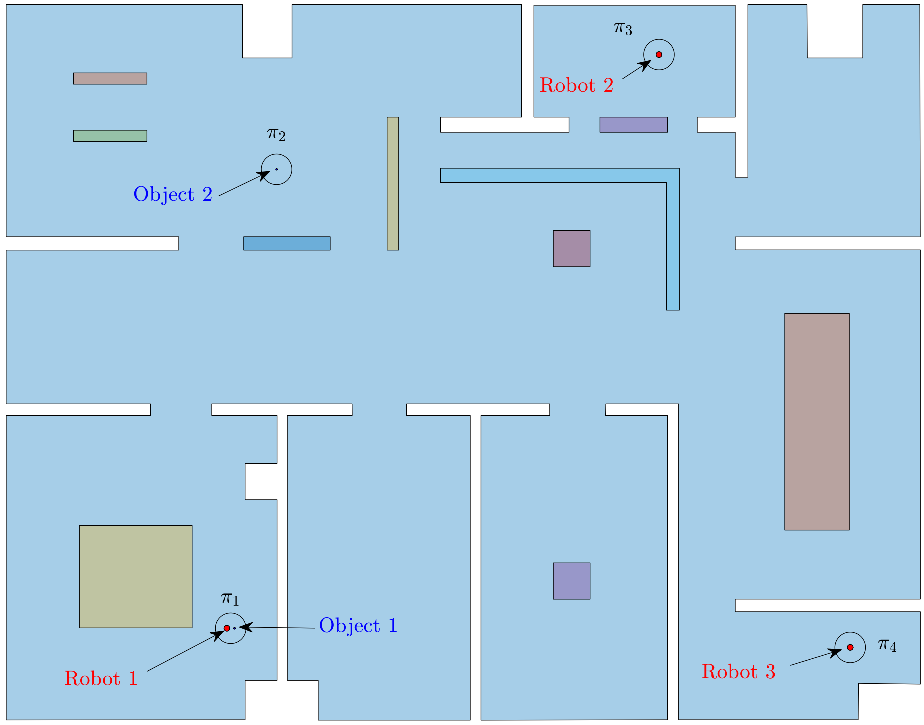


**Fig. 2.** The multi-robot-object initial configuration in the first case study.

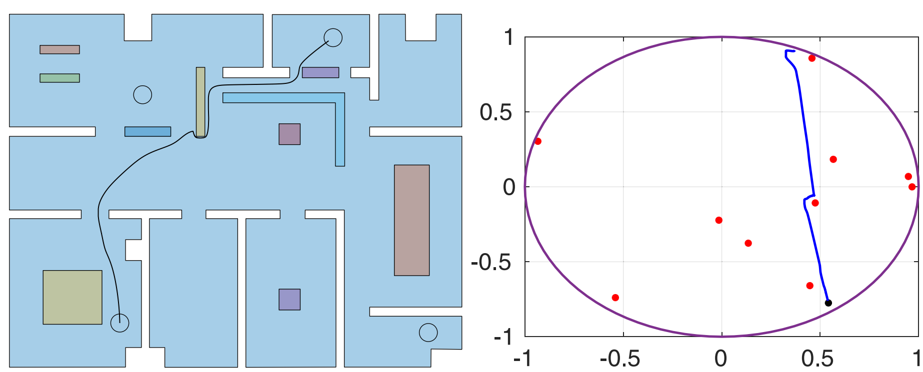


**Fig. 3.** Navigation of robot 1 from $\pi_1$ to $\pi_3$ in the original (left) and the transformed point-world (right) environment. In the original environment, the robot's circular volume has been transferred to the obstacles and workspace boundary.

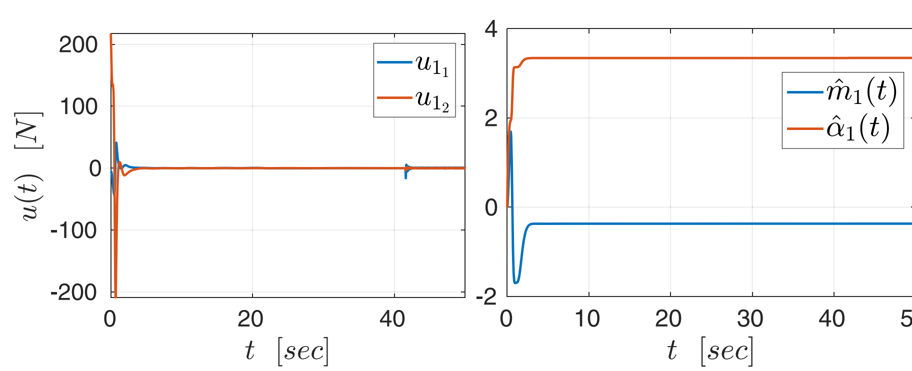


**Fig. 4.** The evolution of the control inputs $u_1(t)$ (left) and adaptation signals $\hat{m}_1(t)$, $\hat{\alpha}_1(t)$ (right) of robot's 1 navigation.

- $\bar{\psi}_1 = \{$ "1-$\pi_1$", "2-$\pi_1$", "3-$\pi_1$", "4-$\pi_1$", "$O_1$-$\pi_1$", "$O_2$-$\pi_1$", "$O_3$-$\pi_1$" $\}$
- $\bar{\psi}_2 = \{$ "1-$\pi_1$", "2-$\pi_4$", "3-$\pi_3$", "4-$\pi_2$", "$O_1$-$\pi_1$", "$O_2$-$\pi_1$", "$O_3$-$\pi_2$" $\}$
- $\bar{\psi}_3 = \{$ "1-$\pi_2$", "2-$\pi_2$", "3-$\pi_2$", "4-$\pi_3$", "$O_1$-$\pi_1$", "$O_2$-$\pi_1$", "$O_3$-$\pi_3$" $\}$
- $\bar{\psi}_4 = \{$ "1-$\pi_1$", "2-$\pi_1$", "3-$\pi_2$", "4-$\pi_3$", "$O_1$-$\pi_3$", "$O_2$-$\pi_3$", "$O_3$-$\pi_1$" $\}$
- $\bar{\psi}_5 = \{$ "1-$\pi_2$", "2-$\pi_2$", "3-$\pi_3$", "4-$\pi_3$", "$O_1$-$\pi_2$", "$O_2$-$\pi_3$", "$O_3$-$\pi_3$" $\}$
- $\bar{\psi}_6 = \{$ "1-$\pi_2$", "2-$\pi_2$", "3-$\pi_2$", "4-$\pi_2$", "$O_1$-$\pi_3$", "$O_2$-$\pi_3$", "$O_3$-$\pi_3$" $\}$

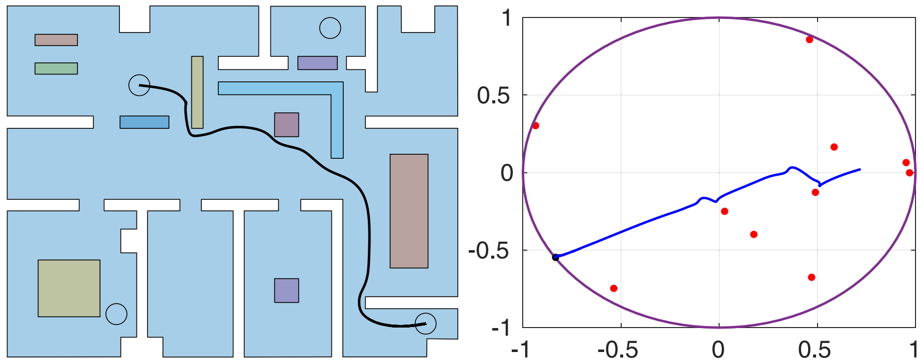


**Fig. 5.** Transportation of object 1 from $\pi_2$ to $\pi_4$ by robots 1 and 2 in the original (left) and the transformed point-world (right) environment. In the original environment, the circular volume of the coupled object-robots system has been transferred to the obstacles and workspace boundary.

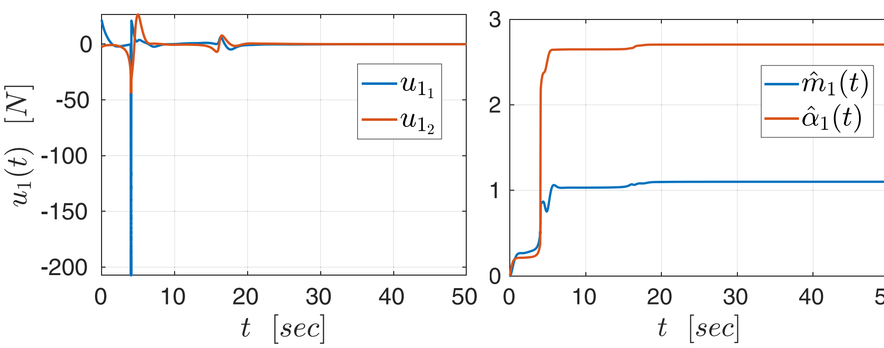


**Fig. 6.** The evolution of the control inputs $u_1(t)$ (left) and adaptation signals $\hat{m}_1(t)$, $\hat{\alpha}_1(t)$ (right) of robot 1 in the cooperative object transportation.

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

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

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

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

- $\bar{\psi}_7 = \{$ "1-$\pi_3$","2-$\pi_3$","3-$\pi_2$","4-$\pi_1$","$O_1$-$\pi_3$","$O_2$-$\pi_1$","$O_3$-$\pi_2$" $\}$
- $\bar{\psi}_8 = \{$ "1-$\pi_4$","2-$\pi_4$","3-$\pi_3$","4-$\pi_2$","$O_1$-$\pi_3$","$O_2$-$\pi_3$","$O_3$-$\pi_3$" $\}$

In words, the mission specification in (15) requires the robots and objects to satisfy $\bar{\psi}_1$ or $\bar{\psi}_2$, $\bar{\psi}_3$, $\bar{\psi}_4$, and $\bar{\psi}_5$ infinitely often, eventually satisfy $\bar{\psi}_7$ or $\bar{\psi}_8$, satisfy $\bar{\psi}_5$ before satisfying $\bar{\psi}_1$, and always avoid $\bar{\psi}_6$; $\bar{\psi}_1$ indicates the visit of all robots and all objects to $\pi_1$; $\bar{\psi}_2$ indicates the visit of robot 1, objects 1 and 2 to $\pi_1$, of robot 2 to $\pi_4$, of robot 3 to $\pi_3$, and of robot 4 and object 3 to $\pi_2$; $\bar{\psi}_3$ indicates the visit of robots 1, 2, and 3 to $\pi_2$, of robot 4 and object 3 to $\pi_3$, and of objects 1 and 2 to $\pi_1$; $\bar{\psi}_4$ indicates the visit of robots 1, 2, and object 3 to $\pi_1$, of robot 3 to $\pi_2$, and of robot 4 and objects 1 and 2 to $\pi_3$; $\bar{\psi}_5$ indicates the visit of robots 1, 2, and object 1 to $\pi_2$, and of robots 3, 4, and objects 2, 3 to $\pi_3$; $\bar{\psi}_6$ indicates the visit of all robots to $\pi_2$ and all objects to $\pi_3$; $\bar{\psi}_7$ indicates the visit of robots 1, 2, and object 1 to $\pi_3$, of robot 3 and object 3 to $\pi_2$, and of robot 4 and object 2 to $\pi_1$; finally, $\bar{\psi}_8$ indicates the visit of robots 1 and 2 to $\pi_4$, of robot 3 and all objects to $\pi_3$, and of robot 4 to $\pi_2$.

The LTL formula in (15) corresponds to a NBA with 8 states – among which one is a final state – and 27 transitions. STyLuS* found the first feasible prefix and suffix path within within 223.8 s and 85.8 s on average. We omit the action path of the robots for ease of exposition.

**Table 3**
Scalability analysis of the construction of $\mathcal{TS}$.

| (N,M,K) | $\lvert\Pi_s\rvert$ | $\lvert\rightarrow_s\rvert$ | $t_n$ (s) | $t_e$ (s) |
|---|---|---|---|---|
| (2,2,2) | 64 | 392 | 0.0435 | 0.036 |
| (5,2,2) | 1268 | 41 796 | 2.89 | 13.19 |
| (3,2,3) | 1017 | 16 443 | 0.4671 | 6.41 |
| (3,2,4) | 3288 | 67 816 | 1.279 | 63 |
| (2,4,3) | 3813 | 32 313 | 1.77 | 99.49 |
| (2,2,7) | 3969 | 62 377 | 0.788 | 81.5 |
| (3,2,5) | 8335 | 205 975 | 2.955 | 399.66 |
| (2,5,3) | 14 683 | 126 745 | 8.09 | 1617.32 |
| (4,2,4) | 17 816 | 687 704 | 11.06 | 1997.97 |

**Table 4**
Scalability analysis of the high-level planner.

| (N,M,K) | $\lvert\Pi_s\rvert$ | $\phi_1$ | | $\phi_2$ | |
|---|---|---|---|---|---|
| | | Prefix (s) | Suffix (s) | Prefix (s) | Suffix (s) |
| (2,2,2) | 56 | 0.07 ± 0.1 | 0.08 ± 0.11 | 0.06 ± 0.08 | 0.17 ± 0.12 |
| (5,2,2) | 845 | 0.06 ± 0.06 | 0.21 ± 0.12 | 0.15 ± 0.1 | 0.28 ± 0.14 |
| (3,2,3) | 927 | 0.3 ± 0.21 | 0.23 ± 0.05 | 3.97 ± 0.8 | 0.28 ± 0.09 |
| (3,2,4) | 3112 | 0.26 ± 0.24 | 1.64 ± 0.7 | 6.55 ± 3.62 | 2.46 ± 0.5 |
| (2,4,3) | 3417 | 0.07 ± 0.07 | 0.91 ± 0.36 | 0.24 ± 0.07 | 0.87 ± 0.25 |
| (2,2,7) | 3941 | 1.6 ± 0.63 | 1.30 ± 0.22 | 23.3 ± 10.21 | 2.33 ± 0.42 |
| (3,2,5) | 8045 | 2.49 ± 0.92 | 5.68 ± 3.35 | 76.16 ± 31.8 | 5.23 ± 1.85 |
| (2,5,3) | 12 936 | 3.48 ± 1.99 | 0.48 ± 0.17 | 94.06 ± 26.9 | 9.43 ± 2.51 |
| (4,2,4) | 16 392 | 5.08 ± 2.41 | 6.13 ± 1.83 | 86.3 ± 42.39 | 8.14 ± 2.06 |

### *4.3. Scalability analysis*

We further perform additional numerical experiments that investigate the scalability of the high-level planner. More specifically, we have considered 9 different transition systems with various numbers of robots, objects, and regions of interest. We have further considered two LTL formulas for each transition system, namely $\phi_{i,1}$ and $\phi_{i,2}$, $i \in \{1,\dots,9\}$; these formulas have the same length (same number of predicates) for the different transition systems, i.e., $|\phi_{i,1}| = |\phi_{i',1}|$ and $|\phi_{i,2}| = |\phi_{i',2}|$ for all $i,i' \in \{1,\dots,9\}$, $i \neq i'$. The first LTL formulas $\phi_{i,1}$ are defined as $\phi_{i,1} = \Box\Diamond(\pi_{i,A} \wedge (\Diamond\pi_{i,B}))$, with $\pi_{i,A}, \pi_{i,B} \in \Pi_s$, requiring to satisfy first $\pi_{i,A}$ and then $\pi_{i,B}$ infinitely often. This formulas correspond to an NBA with 4 states and 12 transitions. The second LTL formulas are similarly defined as $\phi_{i,2} = \Box\Diamond(\pi_{i,A} \wedge (\Diamond\pi_{i,B} \wedge (\Diamond\pi_{i,C})))$, with $\pi_{i,A}, \pi_{i,B}, \pi_{i,C} \in \Pi_s$, corresponding to an NBA with 9 states and 43 transitions. For instance, in the first case with 2 robots, 2 objects, and 2 regions of interest, $\pi_{1,A}$ is the state entailing "1-$\pi_1$", "2-$\pi_2$", "$O_1$-$\pi_1$" and "$O_2$-$\pi_2$", i.e., robot 1 and object 1 in region 1 and robot 2 and object 2 in region 2, while $\pi_{1,B}$ is the state entailing "1-$\pi_1$", "2-$\pi_1$", "$O_1$-$\pi_1$" and "$O_2$-$\pi_2$", i.e., robots 1, 2, and object 1 in region 1 and object 2 in region 2. For ease of presentation, we omit the detailed description of the formulas for the rest of the cases. In all cases, we considered regions with radius $r_{\pi_k} = 0.4$m, and robots and objects with bounding volume's radii as $r_i = 0.75$m and $r_j^o = 0.2$m, respectively. We further considered all robots with equal power capabilities $\zeta_i = 3$, and objects with required transportation power of 4.

The results are summarized in Tables 3 and 4; Table 3 shows the number of feasible nodes $|\Pi_s|$ and transitions $|\rightarrow_s|$, along with the time to construct them ($t_n$ and $t_e$, respectively). The table demonstrates that the runtime to construct the transition system increases with the number of entities (robots/objects) and the number of regions of interest, verifying the theoretical analysis. Table 4 shows the average runtime over ten runs of STyLuS* for each case of transition system. Similarly to the construction of the transition system, the runtime of STyLuS* required to compute the first feasible solution tends to increase as the size of the TS and/or the task complexity (i.e., the number of NBA states) increase (see Tables 3 and 4).

## 5. Conclusion

We propose an algorithm for the control and planning of multi-robot-object systems subject LTL tasks. We develop adaptive feedback-control protocols for robot navigation and cooperative object transportation, which enable the abstraction of the underlying continuous dynamics to a finite transition system. We compose the transition system with an automaton that represents the LTL task and use a sampling-based planner to derive an optimal task-satisfying plan for the robots.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## References

[1] G.E. Fainekos, A. Girard, H. Kress-Gazit, G.J. Pappas, Temporal logic motion planning for dynamic robots, Automatica 45 (2) (2009) 343–352.
[2] M. Lahijanian, M.R. Maly, D. Fried, L.E. Kavraki, H. Kress-Gazit, M.Y. Vardi, Iterative temporal planning in uncertain environments with partial satisfaction guarantees, IEEE Trans. Robot. 32 (3) (2016) 583–599.
[3] S.G. Loizou, K.J. Kyriakopoulos, Automatic synthesis of multi-agent motion tasks based on ltl specifications, in: IEEE Conference on Decision and Control, CDC, Vol. 1, IEEE, 2004, pp. 153–158.
[4] Y. Diaz-Mercado, A. Jones, C. Belta, M. Egerstedt, Correct-by-construction control synthesis for multi-robot mixing, in: IEEE Conference on Decision and Control, CDC, IEEE, 2015, pp. 221–226.
[5] Y. Chen, X.C. Ding, A. Stefanescu, C. Belta, Formal approach to the deployment of distributed robotic teams, IEEE Trans. Robot. 28 (1) (2012) 158–171.
[6] R.V. Cowlagi, Z. Zhang, Motion-planning with linear temporal logic specifications for a nonholonomic vehicle kinematic model, in: American Control Conference, ACC, American Automatic Control Council (AACC), 2016, pp. 6411–6416.
[7] C. Belta, V. Isler, G.J. Pappas, Discrete abstractions for robot motion planning and control in polygonal environments, IEEE Trans. Robot. 21 (5) (2005) 864–874.
[8] A. Bhatia, M.R. Maly, L.E. Kavraki, M.Y. Vardi, Motion planning with complex goals, IEEE Robot. Autom. Mag. 18 (3) (2011) 55–64.
[9] I. Filippidis, D.V. Dimarogonas, K.J. Kyriakopoulos, Decentralized multi-agent control from local LTL specifications, in: IEEE Conference on Decision and Control, CDC, IEEE, 2012, pp. 6235–6240.
[10] M. Guo, D.V. Dimarogonas, Multi-agent plan reconfiguration under local LTL specifications, Int. J. Robot. Res. 34 (2) (2015) 218–235.
[11] C.-I. Vasile, J. Tumova, S. Karaman, C. Belta, D. Rus, Minimum-violation scLTL motion planning for mobility-on-demand, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 1481–1488.
[12] C. Belta, L.C. Habets, Controlling a class of nonlinear systems on rectangles, IEEE Trans. Automat. Control 51 (11) (2006) 1749–1759.
[13] G. Reißig, Computing abstractions of nonlinear systems, IEEE Trans. Automat. Control 56 (11) (2011) 2583–2598.
[14] A. Tiwari, Abstractions for hybrid systems, Form. Methods Syst. Des. 32 (1) (2008) 57–83.
[15] M. Rungger, A. Weber, G. Reissig, State space grids for low complexity abstractions, in: IEEE Conference on Decision and Control, CDC, IEEE, 2015, pp. 6139–6146.
[16] D. Boskos, D.V. Dimarogonas, Decentralized abstractions for feedback interconnected multi-agent systems, in: IEEE Conference on Decision and Control, CDC, IEEE, 2015, pp. 282–287.
[17] C. Belta, V. Kumar, Abstraction and control for groups of robots, IEEE Trans. Robot. 20 (5) (2004) 865–875.
[18] T.G. Sugar, V. Kumar, Control of cooperating mobile manipulators, IEEE Trans. Robot. Autom. 18 (1) (2002) 94–103.
[19] D. Heck, D. Kostic, A. Denasi, H. Nijmeijer, Internal and external force-based impedance control for cooperative manipulation, 2013, pp. 2299–2304.
[20] Y. Kume, Y. Hirata, K. Kosuge, Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2007, pp. 4077–4082.
[21] A. Tsiamis, C.K. Verginis, C.P. Bechlioulis, K.J. Kyriakopoulos, Cooperative manipulation exploiting only implicit communication, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2015, pp. 864–869.

[22] F. Ficuciello, A. Romano, L. Villani, B. Siciliano, Cartesian impedance control of redundant manipulators for human-robot co-manipulation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2014, pp. 2120–2125.
[23] A.-N. Ponce-Hinestroza, J.-A. Castro-Castro, H.-I. Guerrero-Reyes, V. Parra-Vega, E. Olguín-Díaz, Cooperative redundant omnidirectional mobile manipulators: Model-free decentralized integral sliding modes and passive velocity fields, in: IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2016, pp. 2375–2380.
[24] A. Marino, Distributed adaptive control of networked cooperative mobile manipulators, IEEE Trans. Control Syst. Technol. (2017).
[25] A. Nikou, C. Verginis, S. Heshmati-Alamdari, D.V. Dimarogonas, A nonlinear model predictive control scheme for cooperative manipulation with singularity and collision avoidance, in: Mediterranean Conference on Control and Automation, MED, IEEE, 2017, pp. 707–712.
[26] C.K. Verginis, A. Nikou, D.V. Dimarogonas, Communication-based decentralized cooperative object transportation using nonlinear model predictive control, in: European Control Conference, ECC, IEEE, 2018, pp. 733–738.
[27] C.K. Verginis, M. Mastellaro, D.V. Dimarogonas, Robust cooperative manipulation without force/torque measurements: Control design and experiments, IEEE Trans. Control Syst. Technol. 28 (3) (2019) 713–729.
[28] S. Erhart, S. Hirche, Model and analysis of the interaction dynamics in cooperative manipulation tasks, IEEE Trans. Robot. 32 (3) (2016) 672–683.
[29] H.G. Tanner, S.G. Loizou, K.J. Kyriakopoulos, Nonholonomic navigation and control of cooperating mobile manipulators, IEEE Trans. Robot. Autom. 19 (1) (2003) 53–64.
[30] C.K. Verginis, D.V. Dimarogonas, Timed abstractions for distributed cooperative manipulation, Auton. Robots 42 (2018) 781–799.
[31] C.K. Verginis, D.V. Dimarogonas, Adaptive robot navigation with collision avoidance subject to 2nd-order uncertain dynamics, Automatica 123 (2021) 109303.
[32] P. Schillinger, M. Bürger, D.V. Dimarogonas, Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems, Int. J. Robot. Res. 37 (7) (2018) 818–838.
[33] C.K. Verginis, D.V. Dimarogonas, Multi-agent motion planning and object transportation under high level goals, IFAC-PapersOnLine 50 (1) (2017) 15816–15821.
[34] C. Baier, J.-P. Katoen, K.G. Larsen, Principles of Model Checking, MIT Press, 2008.
[35] C.K. Verginis, D. Zelazo, D.V. Dimarogonas, Cooperative manipulation via internal force regulation: A rigidity theory perspective, IEEE Trans. Control Netw. Syst. (2022).
[36] C. Makkar, W. Dixon, W. Sawyer, G. Hu, A new continuously differentiable friction model for control systems design, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE, 2005, pp. 600–605.
[37] D.Z. Chen, Sphere packing problem, Encycl. Algorithms (2008) 1–99.
[38] M.R. Cutkosky, Robotic Grasping and Fine Manipulation, vol. 6, Springer Science & Business Media, 2012.
[39] M.F. Reis, A.C. Leite, F. Lizarralde, Modeling and control of a multifingered robot hand for object grasping and manipulation tasks, in: IEEE Conference on Decision and Control, CDC, IEEE, 2015, pp. 159–164.
[40] P. Vlantis, C. Vrohidis, C.P. Bechlioulis, K.J. Kyriakopoulos, Robot navigation in complex workspaces using harmonic maps, in: IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 1726–1731.
[41] S.L. Smith, J. Tumova, C. Belta, D. Rus, Optimal path planning for surveillance with temporal-logic constraints, Int. J. Robot. Res. 30 (14) (2011) 1695–1708.
[42] A. Ulusoy, S.L. Smith, C. Belta, Optimal multi-robot path planning with LTL constraints: guaranteeing correctness through synchronization, in: Distributed Autonomous Robotic Systems, Springer, 2014, pp. 337–351.
[43] Y. Kantaros, M.M. Zavlanos, Stylus: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems, Int. J. Robot. Res. 39 (7) (2020) 812–836.


**Christos K. Verginis** is an Assistant Professor at the Division of Signals and Systems, Department of Electrical Engineering, Uppsala University. He received the Diploma in electrical and computer engineering and M.Sc. degree in automatic control systems and robotics from the National Technical University of Athens (NTUA), Greece, in 2013 and 2015, respectively, and the Ph.D. in electrical engineering from KTH Royal Institute of Technology in 2020. His Ph.D. thesis received the EECI Ph.D. award on control for complex and heterogeneous systems, 2020. Prior to joining Uppsala University, he was a Postdoctoral Researcher at the University of Texas at Austin. His research interests include nonlinear and adaptive control, hybrid and safety-critical systems, multi-robot systems, data-driven control and reinforcement learning. He has authored more than 40 papers in scientific journals and conference proceedings.

**Yiannis Kantaros** is an Assistant Professor in the Department of Electrical and Systems Engineering, Washington University in St. Louis (WashU), St. Louis, MO, USA. He received the Diploma in Electrical and Computer Engineering in 2012 from the University of Patras, Patras, Greece. He also received the M.Sc. and the Ph.D. degrees in mechanical engineering from Duke University, Durham, NC, in 2017 and 2018, respectively. Prior to joining WashU, he was a postdoctoral associate in the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA. His current research interests include machine learning, distributed control and optimization, and formal methods with applications in robotics. He received the Best Student Paper Award at the 2nd IEEE Global Conference on Signal and Information Processing in 2014 and the 2017-18 Outstanding Dissertation Research Award from the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC.

**Dimos V. Dimarogonas** received the Diploma in Electrical and Computer Engineering in 2001 and the Ph.D. in Mechanical Engineering in 2007, both from the National Technical University of Athens (NTUA), Greece. From May 2007 to February 2009, he was a Postdoctoral Researcher at the Automatic Control Laboratory, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, and a Postdoctoral Associate at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He is currently a Professor in Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. His current research interests include multiagent systems, hybrid systems, robot navigation, networked control and eventtriggered control. Dr. Dimarogonas was awarded a Docent in Automatic Control from KTH in 2012. He serves on the Editorial Board of Automatica, the IEEE Transactions on Automation Science and Engineering and the IET Control Theory and Applications, and is a member of the Technical Chamber of Greece.