



HAL
open science

Multi-Agent Navigation using Convex Lifting on Dynamic Environment

Turan Konyalıođlu, Sorin Olaru, Silviu-Iulian Niculescu, Iris Ballesteros-Tolosana, Carlos Flores

► **To cite this version:**

Turan Konyalıođlu, Sorin Olaru, Silviu-Iulian Niculescu, Iris Ballesteros-Tolosana, Carlos Flores. Multi-Agent Navigation using Convex Lifting on Dynamic Environment. MECC 2024 - 4th Modeling, Estimation and Control Conference, Oct 2024, Chicago, United States. hal-04701406

HAL Id: hal-04701406

<https://hal.science/hal-04701406v1>

Submitted on 18 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Multi-Agent Navigation using Convex Lifting on Dynamic Environment

Turan Konyalıođlu^{*,**} Sorin Olaru^{*} Silviu-Iulian Niculescu^{*}
Iris Ballesteros-Tolosana^{**} Carlos Flores^{**}

^{*} *Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, Gif-sur-Yvette 91190, France*

{turan.konyalioglu, sorin.olaru, silviu.niculescu}@centralesupelec.fr

^{**} *Ampere Software Technology, DEA-OCDT1, 1 Av. du Golf, 78280 Guyancourt, France*

{turan.konyalioglu, iris.ballesteros-tolosana, carlos.flores}@ampere.cars

Abstract: This paper revisits the conventional convex lifting method for space partition in static topologies by adding a series of guarantees for obstacle avoidance in the framework of multi-agent navigation strategies. Further, the present work proposes a new methodology that adapts the convex lifting method in such a dynamic environment. To complete the developments towards navigation, the paper introduces a Model Predictive Control method for trajectory planning and navigation for an agent and presents examples of integrator-like multi-agent systems in a cluttered environment prioritizing the safety and control objectives of each agent.

Keywords: Path Planning, Motion Control, Multi-agent Dynamical Systems, Optimal Control.

1. INTRODUCTION

A multi-agent system (MAS) comprises multiple autonomous entities, like processes, robots, or autonomous vehicles, operating within a specific environment and engaging in actions governed by preset rules. Each agent possesses autonomy, enabling independent decision-making and action-taking. MAS focuses on two primary concerns: enabling agents to achieve control objectives autonomously and facilitating cooperation and negotiation among agents for task completion (Wooldridge, 2009). Ongoing MAS research explores diverse domains, including military systems, traffic management, supply chain optimization, and robotics (Murray, 2007).

In tackling MAS challenges, researchers have proposed various solutions. Notably, Ames et al. (2019) integrates control barrier techniques to ensure agent safety in a MAS coverage scenario. Rezaee and Abdollahi (2011) employs potential field methods for formation conservation with obstacle avoidance. Voronoi partitions are also employed, for instance Chevet et al. (2019) which defines safety sets for agents using guaranteed partitions, controlled in a decentralized manner via robust model predictive control. Another study (Hatleskog et al., 2018) suggests Voronoi partitioning, where agents are internally controlled by tracking their respective cell's Chebyshev center.

On the other hand, several research groups are employing model predictive control (MPC) as a tool for the MAS field. Earl and D'Andrea (2007) demonstrates the utility of mixed-integer linear programming (MILP) in modeling complex decision-making processes, obstacle avoidance, and trajectory planning. De Oliveira and Camponogara (2010) applies MPC to manage traffic networks using a

multi-agent MPC model. Wang and Ding (2014) presents a distributed MPC study where control actions are synchronized. Afonso et al. (2020) contributes a study on task allocations and trajectory planning, addressing challenges such as obstacle avoidance and connectivity constraints.

The present paper aims to tackle the MAS deployment issue in a decentralized way with collaborative assumptions but a minimal collection of information with respect to the environment. Convex lifting techniques Ioan et al. (2019) are used to form polyhedral space partitions for agents and obstacles, establishing paths and safety sets. While the previous framework suits static topologies, we further extend their constructions to fit dynamic environments. This approach is applied to a MAS environment where agents deploy to target points in a decentralized manner with a limited information on the environment and the decision-making on the neighbor agents. Also, we propose a reachability analysis able to certify a set of target objective and an MPC strategy for navigation.

The paper is organized as follows: Section 2 briefly defines the multi-agent problem addressed. Section 3 discusses the convex lifting method and provides a methodology for dynamic environment cases. Section 4 proposes a decentralized control algorithm to be used within the proposed framework. Section 5 discusses the numerical results before the conclusions.

Notation: \mathbb{R}^n , $\mathbb{R}_{>}$ denote the set of real numbers in n -dimensional space, the set of positive real numbers and the set of non-negative integers up to any given variable, respectively. \mathcal{I}_N is a set of indices which represents ordered subsets of \mathbb{N} where N denotes the cardinality of the set, \mathbf{I}_n refers to $n \times n$ identity matrix. In particular, $\mathcal{I}_N^2 :=$

$\{(i, j) : i \in \mathcal{I}_N, j \in \mathcal{I}_N, i \neq j\}$. Besides, $\mathcal{V}(S)$, $\text{int}(S)$ denote the set of vertices, the interior of a polytope S , respectively. $\text{Proj}(S, \mathbb{S})$ represents the orthogonal projection of S onto the space \mathbb{S} . Given two sets $S_1, S_2 \in \mathbb{R}^d$, the Minkowski sum, distance and complement are $S_1 \oplus S_2 = \{x_1 + x_2 : x_1 \in S_1, x_2 \in S_2\}$, $d(S_1, S_2) = \min_{s_1 \in S_1, s_2 \in S_2} \|s_1 - s_2\|$, and $S_1/S_2 = \{x \in S_1 : x \notin S_2\}$ respectively.

2. PROBLEM FORMULATION

A cluttered environment denotes a space defined by a compact set $X \subset \mathbb{R}^n$ but densely populated with objects, whether obstacles or agents, resulting in a chaotic and obstructed navigation setting. In order to establish a generic representation of the obstacles and agents, a polytopic representation will be adopted. In other words, we will consider these regions to be represented by polyhedra. Two reasons for this choice: i) polyhedra are universal approximators for the convex set; ii) any nonconvex set can be represented as a union of convex sets. Based on those principles, any cluttered environment can be described in terms of a collection of polyhedral sets.

Definition 1. A polyhedron is defined as the intersection of a finite number of halfspaces.

$$P = \{x \in \mathbb{R}^n | Hx \leq w\} \quad \square$$

We will be interested in dealing with arbitrary polyhedral sets in a given domain, and the next definition is useful for notational purposes:

Definition 2. Given a bounded polyhedral set $X \subset \mathbb{R}^n$, the family of polyhedral subsets of X is defined as:

$$\mathbb{P}(X) = \{P \subseteq X | \exists (H, w) \text{ s.t. } x \in P \Leftrightarrow Hx \leq w\} \quad \square$$

Starting from this family of sets, one can describe a collection of N polyhedra in X as $\{P_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}^N(X)$. Obviously, overlaps of these objects should be avoided when describing a navigation scene. Thus, all the valid spatial distributions of obstacles and agents within a scene X can be described as follows.

Definition 3. Given a bounded polyhedral set $X \subset \mathbb{R}^n$, the family of valid spatial distribution (disjoint) of $N \in \mathbb{R}$ objects in X is denoted as:

$$\mathbb{P}_D^N(X) = \{\{P_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}^N(X) \text{ such that} \\ \text{int}(P_i) \cap \text{int}(P_j) = \emptyset, (i, j) \in \mathcal{I}_N^2\} \quad \square$$

Addressing navigation challenges in such environments involves establishing safety regions that adapt to changes in the environment and provide clear pathways from the origin to the destination.

A key component is the design of the safe regions, which can be done effectively by exploiting the space partitions.

Definition 4. Given a set of disjoint polyhedral objects, $\{P_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}_D^N(X)$ in a cluttered environment X , a family of space partitions induced by these objects is defined as:

$$\mathbb{W}^N(\{P_i\}_{i \in \mathcal{I}_N}, X) = \{\{X_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}_D^N(X) | X = \bigcup_{i \in \mathcal{I}_N} X_i, \\ P_i \subseteq X_i, \forall i \in \mathcal{I}_N\} \quad \square \quad (1)$$

As a shorthand notation, a collection of polyhedral objects will be denoted as $\mathcal{P} \triangleq \{P_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}_D^N(X)$, and the

space partitions $\mathcal{X} \triangleq \{X_i\}_{i \in \mathcal{I}_N} \subset \mathbb{W}^N(\mathcal{P}, X)$, whenever the cardinality of the respective collection of sets is clear from the context.

The collection of all conceivable positions or waypoints accessible to a robot or agent within its environment is known as its configuration space. This element holds significant importance in motion planning, especially when analyzing feasible paths and obstacle avoidance schemes. In terms of navigation, the feasible region of the spatial configuration space can be expressed compactly as in (2),

$$C_{\mathcal{X}}(\mathcal{P}) \triangleq X/\mathcal{P} \text{ or equivalently } C_{\mathcal{X}}(\mathcal{P}) \triangleq \mathcal{X}/\mathcal{P} \quad (2)$$

Assumption 1. The agents are deployed in a common space to perform given separate tasks, which will avert conflict and convergence to their respective target positions. Furthermore, it is assumed that during the realization of their ego objectives, all agents obey safety rules cooperatively. \square

Definition 5. A collection of safety sets, $\mathcal{S} = \{S_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}_D^N$, corresponds to bounded regions (subsets of S). It is defined such that each agent has the capability to navigate while ensuring that its trajectory remains confined within the respective set S_i . \square

A multi-agent system denoted by Σ where an agent, $\alpha \in \mathcal{I}_\Sigma$ will be considered to be described by means of a discrete linear time-invariant dynamics,

$$x_\alpha(k+1) = A_\alpha x_\alpha(k) + B_\alpha u_\alpha(k) \quad (3)$$

with $u_\alpha \in \mathbb{R}^m$ the control input (generally the speed) and $x_\alpha \in \mathbb{R}^n$ (generally the position).

Assumption 2. The set of polyhedral objects, which induce the space partitions, consist of agents and obstacles $\mathcal{P} = \{W_i\}_{i \in \mathcal{I}_\Sigma} \cup \{O_i\}_{i \in \mathcal{I}_o} = \{P_i\}_{i \in \mathcal{I}_N} \subset \mathbb{P}_D^N$ where $|\mathcal{I}_N| = |\mathcal{I}_\Sigma| + |\mathcal{I}_o|$ with $|\mathcal{I}_o|$ is the cardinality of the set of obstacles and $|\mathcal{I}_\Sigma|$ is cardinality of the set of agents and $\mathcal{I}_N = \mathcal{I}_\Sigma \cup \mathcal{I}_o$. Each agent has their own polyhedral geometry, $W_\alpha^0 \subset \mathbb{P}$ as a compact set, and they occupy a space parameterized as $W_\alpha^k = W_\alpha^0 \oplus x_\alpha(k)$ where $x_\alpha(k)$ is the position of the center of the agent at time instant k . \square

Assumption 3. For each agent, the information-sharing and decision-making criteria are made according to the following principles:

- Each agent sends its state information, $x_\alpha(x)$ and observed obstacles position to central unit.
- The central unit is responsible for collecting the state measurements to compute and broadcast the space partition information, \mathcal{X} , to each agent. This partition can be updated at each time instant, thus will be denoted $\mathcal{X}(k)$
- *Safety:* each agent obeys collision avoidance constraints and chooses the control actions to fulfill this restriction as a first priority.
- *Privacy:* Each agent manages its own control objective associated with their target point \bar{x}_α . Neither the target point nor the control actions $u_\alpha(k)$ need to be shared.
- The local feedback control actions are exclusively based on agents' control actions and the space partition that they belong to $u_\alpha(x_\alpha(k), \bar{x}_\alpha, \mathcal{X}(k))$ \square

Definition 6. During the mission, safety guarantees are provided for all agents in such an environment if

- (1) $S_i(k) \cap S_j(k) = \emptyset, (i, j) \in \mathcal{I}_N$
- (2) $W_\alpha^k \subset S_\alpha(k), \alpha \in \Sigma$
- (3) $W_\alpha^{k+l|k} \subset S_\alpha(k), \text{ for } l = 1, \dots, N_p$

are established with N_p number of predicted steps. \square

The control objective is to provide a decentralized control law ensuring safety for all agents and prioritizing convergence to their final location as a part of their mission.

$$\lim_{k \rightarrow \infty} \|x_\alpha(k) - \bar{x}_\alpha\| = 0, \alpha \in \mathcal{I}_\Sigma \quad (4)$$

If this goal comes against the safety requirements, then the decentralized control policy will guarantee:

$$\lim_{k \rightarrow \infty} \|x_\alpha(k) - \hat{x}_\alpha\| = 0, \alpha \in \mathcal{I}_\Sigma \quad (5)$$

with \hat{x}_α (local target) such that a path exists between this point and \bar{x}_α .

3. SAFETY BY TIME-VARYING PARTITIONING OF A DYNAMIC ENVIRONMENT

3.1 Space Partitioning via Convex Lifting

Previous studies built on convex lifting Ioan et al. (2019), a versatile design framework able to decompose the space in order to generate a connectivity graph enabling path planning and obstacle avoidance. The idea is to generate space partitions based on lift and projection operations.

Definition 7. Given a collection of polyhedral sets $\mathcal{P} = \{P_i\}_{\mathcal{I}_N} \subset \mathbb{P}_D^N(X)$, a piecewise affine *lifting function* $z : X \rightarrow \mathbb{R}$ defined as

$$z(x) = \max_{i \in \mathcal{I}_N} a_i^T x + b_i$$

is admissible for \mathcal{P} if it satisfies the following property:

$$z(x) > a_j^T x + b_j, \forall x \in \text{int}(P_i), (i, j) \in \mathcal{I}_N^2 \quad \square$$

The lifting functions can be obtained by solving the following optimization problem:

$$\min_{a_i, b_i} J = \sum_{i=1}^{|\mathcal{I}_N|} \| [a_i^T \ b_i] \|_2^2 \quad (6a)$$

$$\text{s.t. } a_i^T v + b_i \leq M, \forall v \in \mathcal{V}(P_i), \forall i \in \mathcal{I}_N, \quad (6b)$$

$$a_i^T v + b_i \geq a_j^T v + b_j + \epsilon, \forall v \in \mathcal{V}(P_i), (i, j) \in \mathcal{I}_N^2 \quad (6c)$$

with $M, \epsilon > 0$ respectively for boundedness condition in (6b) and convexity condition in (6c). The traditional method involves lifting and projection procedures depicted in Fig. 1a. Lifting functions are computed for each polyhedral object shown in red, solving (6). The lifted polyhedron in blue in Fig. 1a is constructed by computing the epigraph of the lifting functions. The facets of the lifted polyhedron represent the epigraph of each lifting function and their projections satisfy the constraints in (6b) and (6c). This is demonstrated in green by lifting the polyhedral objects (in red) onto the lifted polyhedron. Space partition in purple can be obtained by projecting the facets of the lifted polyhedron into the original space. Besides, the properties of the lifting functions in the definition lead to the same polyhedral space partition which can be computed in (7) that surrounds obstacles such that $\mathcal{X} \subset \mathbb{W}_D^N(\mathcal{P}, X)$.

$$X_i = \{x | a_i^T x + b_i \geq a_j^T x + b_j, (i, j) \in \mathcal{I}_N^2\}. \quad (7)$$

Proposition 1. The polyhedral partition, $\mathcal{X} = \{X_i\}_{i \in \mathcal{I}_N} \subset \mathbb{W}^N$, resulted from (7) has the following properties:

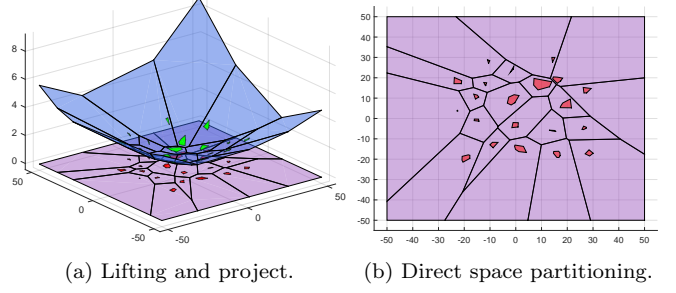


Fig. 1. Convex lifting procedures.

- (1) $P_i \subseteq X_i, \forall i$
- (2) $X_i \cap P_j = \emptyset, (i, j) \in \mathcal{I}_N^2 \quad \square$

It can be highlighted that whenever the current position of an agent and the target location \bar{x} are located in the same region of the partition, i.e. $(x_\alpha(k), \bar{x}_\alpha) \in X_i$, then as a local navigation problem one can define safety regions, $S_i \leftarrow X_i$ and use a constrained control to reach to the objective by guaranteeing the safety policies.

3.2 Path Planning in Static Environment

To solve the navigation problem in a cluttered environment globally, an interconnected graph is created using the facets of the partitions generated via convex lifting.

Definition 8. The interconnected graph of paths is denoted as $\Gamma(\mathcal{N}, \mathcal{E}, f)$ and is defined by the tuple $(\mathcal{N}, \mathcal{E}, f)$, where \mathcal{N} represents the set of nodes corresponding to the vertices of the graph, \mathcal{E} represents the set of edges, and $f : \mathcal{E} \rightarrow \mathbb{R}$ is a function that denotes the weights associated with each edge of the graph. \square

The shortest path, determined by given weights, can be found using graph search techniques such as the Dijkstra algorithm (Dijkstra, 1959) which subsequently yields the path denoted by $\text{Path}(x_0, \bar{x}) = (\bar{x}^0, \bar{x}^1, \dots, \bar{x}^p = \bar{x})$. Definition 9 characterizes a continuous path, path size, and corridor functions based on the piecewise linear connection of these waypoints.

Definition 9. Given a collection of polyhedral set \mathcal{P} and the related partition generated via convex-lifting, \mathcal{X} , a corridor between two nodes $(x_0, \bar{x}) \in \text{int}(C_{\mathcal{X}}(\mathcal{P}))$ is characterized by the existence of two functions: $\gamma : [0, 1] \rightarrow C_{\mathcal{X}}(\mathcal{P})$ and $\rho : [0, 1] \rightarrow \mathbb{R} > 0$ satisfying $\gamma(0) = x_0$ and $\gamma(1) = \bar{x}$, $\gamma(\theta) \oplus \mathbb{B}_{0, \rho(\theta)}^2 \subset C_{\mathcal{X}}(\mathcal{P}), \forall \theta \in [0, 1]$.

Next, a series of corridor segments, safe from any obstacles in the configuration space, can be specified as follows:

$$\Pi = \{x \in \mathbb{R}^d : \exists \theta \in [0, 1] \text{ s.t. } x \in \gamma(\theta) \oplus \mathbb{B}_{0, \rho(\theta)}^2\} \quad (8)$$

Computing corridors by considering the union of convex sets can enhance navigation tasks, simplifying the representation. Specifically, we define the corridor as the combination of convex sets calculated for each segment of the piecewise linear path, $\Pi = \bigcup_{i=1}^{N_c} \Pi_i$ where,

$$\Pi_i = \{x \in \mathbb{R}^d : \exists \tilde{\theta} \in [0, 1] \text{ s.t. } x \in \gamma_i(\tilde{\theta}) \oplus \mathbb{B}_{0, \rho_i(\tilde{\theta})}\} \quad (9)$$

and,

$$\gamma_i(0) = x_i, \gamma_i(1) = x_{i+1} \quad (10)$$

Also, the radius defining the ball in (9) can be found by the minimum distance along a path segment.

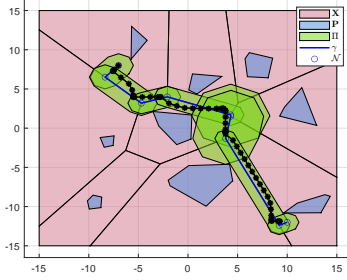


Fig. 2. Trajectory generation for static environment.

$$\rho_i(\theta) = \min_{P_j \in \mathcal{P}} d(P_j, \gamma_i), \forall \theta \in [0, 1] \quad (11)$$

Every corridor segment is formed by combining the Minkowski sum of a path segment with a ball determined by the minimum distance obtained in equation (11). The navigation along the generated trajectory from x_0 to \bar{x} can be achieved using MPC simply for each corridor segment, Π_i as detailed in Section 4. A complete trajectory resulting from the navigation from the initial corridor to the final corridor is illustrated in Fig. 2.¹

3.3 Time-varying Convex Lifting Framework

We describe next one of the contributions of the present work which targets a time-varying update of convex lifting based on the real-time position of the obstacles $\mathcal{P}(k)$.

Remark 1. Considering moving obstacles, the space partition generated by convex lifting in the static framework becomes invalid when one of the following statements related to dynamic environments holds.

- (1) Each polyhedral object is located in the corresponding partition, $P_i(k) \subset X_i(k)$, but the new agent/obstacle position intersects with any corridor $\Pi_i(k) \cap P_i(k+1) \neq \emptyset$
- (2) One or more polyhedral objects does intersect with the corresponding partition, $P_i(k+1) \cap X_j(k) \neq \emptyset$ for some $j \neq i \in \mathcal{I}_N$ \square

As described in Proposition 1, the partition of the environment provides inclusion property for each pair of polyhedral objects, $P_i \subseteq X_i$. This enables the usage of space partitions as a safety set for each agent. The partition update is implicitly dependent on time, but it explicitly relies on the position of the agents. Therefore, it can be triggered by state updates, and agents can realize their mission by obeying the safety policy in Definition 6.

Scaling approach To guarantee safety, each region of the partition can be scaled by solving the linear program,

$$\min_{\lambda} \lambda \text{ s.t. } \lambda X_i \supset P_i, \lambda_0 < \lambda < 1 \quad (12)$$

with λ_0 is a user-defined minimal scaling factor. Based on the optimal scaling, a safety set construction can be obtained.

$$S_i = \lambda_i X_i \quad (13)$$

Proposition 2. Iterative lifting along the trajectories of the moving agents provides a recursive safe set for each agent by means of the time-varying state partition. \square

¹ YALMIP (Löfberg, 2004) and MPT toolboxes (Herceg et al., 2013) are utilized in the construction of algorithms and solving optimization problems.

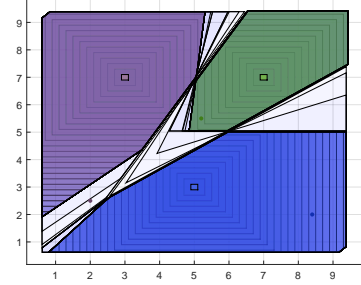


Fig. 3. Reachability analysis.

Proof: If the optimization problem (6) is feasible, it provides $W_\alpha(k) \subset X_i(k)$ at time instance k and considering the safety control policy, $W_\alpha(k+l) \subset S_i(k) \subset X_i$ for $l = 1, \dots, N_p$, thus the solution, $z^k(x) = a_i^T x + b_i$ for $x \in P_i^k, \forall i$, is feasible at time instance, $k+1$. \square

Reachability analysis Guaranteeing the ability to achieve the mission has considerable importance for each agent. The condition $\bar{x}_\alpha \in S_\alpha$ doesn't guarantee mission completion due to the evolution of $\mathcal{X}(k)$ as a function of the position of the neighboring agents. These can occupy space and render a previous target point infeasible. The reachability analysis can be used to find a safe set for each agent as a collection of target points that can be recursively tracked regardless of the other agents' trajectory as long as they obey the safety requirements.

Given a collection of initial polyhedral sets \mathcal{P} , we aim to construct iteratively collections of polyhedral sets \mathcal{R}^j where j is the iteration index. and each particular set within the collection is given by:

$$R_i^{j+1} = (A_i R_i^j \oplus B_i U) \bigcap_{l=1}^j \mathcal{L}(\mathcal{R}^j)_l \quad (14)$$

As part of the construction, the convex lifting function $\mathcal{X} = \mathcal{L}(\{R_i\}_{i \in \mathcal{I}_N})$ is employed with the notational convention $\mathcal{L}(\mathcal{R}^j)_i$ to denote the i -th region of the partition at iteration j . Using an initialization $R_i^0 = P_i(k)$, it can be asserted that a navigation mission can be completed for each corresponding agent if the target positions are such that $\bar{x}_i \oplus P_i \subset R_i^\infty$, for $i \in \mathcal{I}_N$.

Proposition 3. For an integrator-like dynamics ($A_i = I$), given the current positions of the polyhedral objects $\mathcal{P}(k)$, it exists a corresponding polyhedral collection $\mathcal{R}(k) = \{R_i^\infty\}_{i \in \mathcal{I}_N}$. Moreover, for any $\bar{x}_i \in R_i^\infty$ there exists a control policy such that $\lim_{k \rightarrow \infty} \|x_i(k) - \bar{x}_i\| = 0$ for all $i \in \mathcal{I}_N$.

Proof: The polyhedra resulting from the lifting function will belong to the class \mathbb{W}^N which is a collection of bounded polyhedra in space, \mathbb{R}^n . The first term with Minkowski sum will iteratively enlarge the reachable set on the principle $R_i \subset (R_i \oplus B_i U_i)$ as long as this is viable with respect to the limits of each region's boundaries. The iterative procedure is expansive and upper-bounded and, as such, will converge to a collection of sets that represent the reachable regions.

In Fig. 3, reachability analysis is illustrated for three agents. Their reachability sets, polyhedral shapes, and desired positions are given in corresponding colors, and black lines show the evolution of facets of the partitions during the computation. To conclude, a mapping based on

reachability analysis,

$$\begin{aligned} \mathcal{M} : \mathbb{P}_D^N(X) &\rightarrow \mathbb{P}_D^N(X) \\ \mathcal{M}(\mathcal{P}) &= \{R_i^\infty\}_{\mathcal{I}_\Sigma} \end{aligned} \quad (15)$$

can be constructed to characterize the collection of sets that gather for each agent the reachable set within the clutter environment. The underlying convex-lifting partition guarantees the the joint evolution of the neighbours' position by guaranteeing the safety requirements.

4. MPC-BASED NAVIGATION

As is described in subsection 3.2, the facets of the space partitioning can be used as edges in the interconnected graphs, $\Gamma(\mathcal{N}, \mathcal{E}, f)$, providing connectivity with the initial and target positions for each agent. Then, by computing the shortest path in the graph, a set of waypoints can be defined as $\text{Path}(x_k, \bar{x}) = (\bar{x}_k^0, \bar{x}_k^1, \dots, \bar{x}_k^p = \bar{x})$ at any time instant and agents can accomplish their mission by following the paths' first waypoint, \bar{x}_k^1 . Furthermore, to ensure reachability, the point can be projected into reachable sets obtained from $\mathcal{R} = \mathcal{M}(\mathcal{P})$, as in (15), and used as a waypoint.

$$\hat{x}_\alpha = \min_{\tilde{x}} \|\bar{x}_k^1 - \tilde{x}\|_2^2 \quad (16a)$$

$$\text{s.t. } \tilde{x} \in R_\alpha \quad (16b)$$

As a drawback, even small changes in the path at each iteration can affect the tracking control drastically. Thus, agents update their paths after they reach the waypoint retrieved from the previous path solution.

The navigation mechanism completes the real-time deployment by obeying safety policies, which include explicit constraints imposing the trajectory to stay in the safe set. The ideas of safety and control policies make the MPC a suitable candidate for trajectory tracking during the deployment. Considering each LTI dynamics of the agents as in (3) to be integrator type $A = I$, one can formalize the quadratic cost for the MPC problem:

$$\begin{aligned} \mathcal{J}(\bar{x}, x_k, U) &= \|x_{k+N_p|k} - \hat{x}_\alpha\|_P^2 + \\ &\sum_{l=1}^{N_p-1} \|x_{k+l|k} - \hat{x}_\alpha\|_Q^2 + \sum_{l=1}^{N_p-1} \|\Delta u_{k+l|k}\|_R^2 \end{aligned} \quad (17)$$

where N_p is prediction horizon, \hat{x}_α is reference point in where the agent should deploy, Q is state penalty matrix, R is control increment penalty matrix, and P is the terminal cost penalty matrix. The vector $U = [u_{k|k} \dots u_{k+N_p-1|k}]^T$ is the optimization argument:

$$\mathcal{T}(N_p, \Omega_\alpha, \hat{x}_\alpha) : \min_U \mathcal{J}(N_p, \hat{x}_\alpha, x_k, U) \quad (18a)$$

$$\text{s.t. } x_{k+l+1|k} = x_{k+l|k} + B u_{k+l|k}, \quad (18b)$$

$$u_{k+l|k} \in \mathcal{U}, \quad \forall l = 1 : N_p - 1, \quad (18c)$$

$$W_i^{k+N_p} \subset \Omega_\alpha \quad (18d)$$

with state-space dynamics (18b), input constraints (18c) and state constraints (18d) and the problem is solved at each time step k to produce trajectories. To guarantee the system will remain inside the safety set, state constraints are defined by a controlled invariant set Ω_α , such that $(A + K)\Omega_\alpha \subseteq \Omega_\alpha$, assuming K is a feedback control gain that stabilizes the system (i.e. $A + K$ is Schur). Then, the set, Ω_α , can be computed for each agent as follows:

$$\mathcal{O}^j = \mathcal{O}^{j-1} \cap \{H A_c^j \leq w\} \quad (19)$$

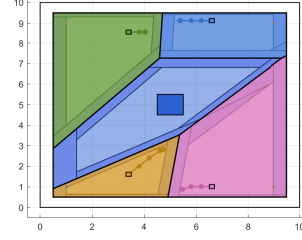


Fig. 4. Safe navigation in a MAS environment.

where $A_c = A + K$ and the iterative procedure is initialized by the set

$\mathcal{O}^0 = \{S_i \oplus -x_\alpha\} \cap \{z \in \mathbb{R}^n | Kz \in \mathcal{U}\} \triangleq \{z \in \mathbb{R}^n | Hz \leq w\}$ such that for each region of the partition, a particular change of variable $z = x - x_\alpha$ takes place. The set iteration will be stopping when $\tilde{\Omega}_\alpha = \mathcal{O}^j = \mathcal{O}^{j-1}$ and the terminal set for problem (18) is found as, $\Omega_\alpha = \tilde{\Omega}_\alpha \oplus x_\alpha$.

Blocking Case: During the mission, it can be unavoidable that agents may block each other, in particular when their paths intersect. In this case, for the sake of decentralized control, agents are expected to solve the blocking situation by themselves. In the 2D case, the problem can be approached by proposing an optimization problem to be solved for each agent when the situation occurs. The optimization problem defined in (20) is solved for each agent with n consecutive point minimizing (20a).

$$\tilde{x}_i = \min_{\tilde{x}_i} \|\tilde{x}_i - \bar{x}_\alpha\|_2^2 \quad (20a)$$

$$\text{s.t. } \tilde{x}_i \in R_\alpha \quad (20b)$$

$$-\beta \begin{bmatrix} 1 \\ 1 \end{bmatrix} \leq \tilde{x}_{i-1} - \tilde{x}_i \leq \beta \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (20c)$$

for $i = 1, 2, \dots, n$ with \bar{x}_α is the set point, β is the user-defined variable as defined in (20c), it allows the maximum amount of step by remaining in the region (20b). If the condition, $\|\tilde{x}_i - \tilde{x}_{i-2}\| \leq \|\tilde{x}_{i-1} - \tilde{x}_{i-2}\|$ holds true, in other words, cost function doesn't improve, the last results rotated by an angle defined by θ as follows:

$$\tilde{x}_i = \bar{x}_\alpha + T^{-1}(\theta)(\tilde{x}_i - \bar{x}_\alpha) \quad (21)$$

where $T(\theta)$ is the planar rotation matrix,

$$T(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (22)$$

Then, \tilde{x}_i found in (21) is projected onto the set, R_α by solving optimization problem in (23).

$$\tilde{x}_i = \min_x \|x - \tilde{x}_i\|_2^2 \quad (23a)$$

$$\text{s.t. } x \in R_\alpha \quad (23b)$$

As a result, the solution can be used as a reference, \hat{x}_α , for the MPC problem defined in (18), which forces each blocked agent to perform a heuristic maneuver (commonly accepted – e.g., avoid by the right-hand side).

5. SIMULATION RESULTS

The proposed framework is employed within a 2D environment featuring a single static obstacle positioned at coordinates (5,5), with four agents initialized at locations (1,1), (9,9), (1,9), and (9,1), in the meantime,

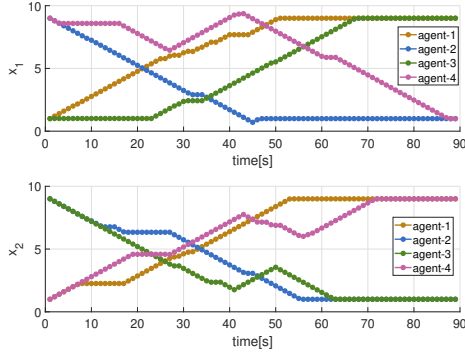


Fig. 5. Agents' trajectories.

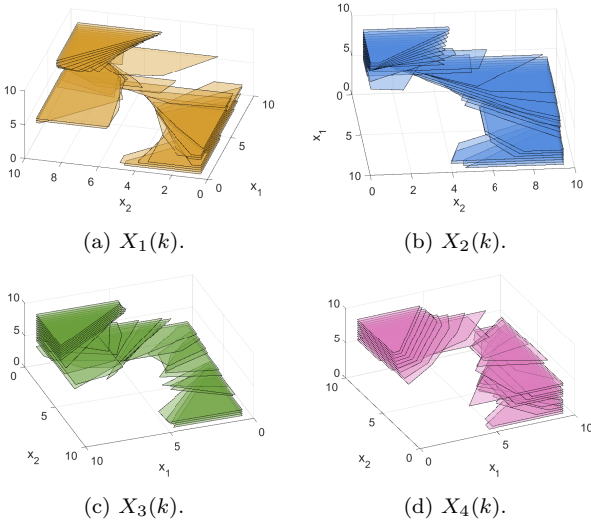


Fig. 6. Evolution of space partition during deployment.

target positions for the agents are designated as (9,9), (1,1), (9,1), and (1,9), respectively to the numbering. The environment is depicted in the Fig. 4. The agents have linear dynamics as their states include position. The space partitions are shown in corresponding colors, whereas the brighter regions are the associated safety sets, which can be obtained via the scaling approach. The space partition evolution during the simulation is shown in Fig. 6 for each agent with an associated color whereas the partition of the obstacle is omitted. The z-axis represents some time instants. Parameters for convex lifting optimization are set as $\epsilon = 0.1$ and $M = 0.01$. The MPC parameter $Q = 10\mathbf{I}_2$, $R = 0.1\mathbf{I}_2$, $u_{max} = -u_{min} = 0.4$, and N_p is set to 10. The resultant trajectories are shown in Fig. 5, and it can be inferred that the control policy in (5) does not always exhibit negative decay. This is not only because of the information constraints for one agent on another but also because there is no explicit decision-making mechanism on which path is to be selected concerning other agents' positions and trajectories. However, the proposed approach demonstrates its capability to handle obstacle avoidance and global control objectives in this environment.

6. CONCLUSION

This paper presents an approach to addressing the deployment problem towards target points within MAS by utiliz-

ing convex lifting, which facilitates space partitioning and path planning. A reachability analysis integrated with the convex lifting method is proposed to ensure partial mission completion. Also, a set-based MPC-based decentralized control method is introduced to ensure safety and enable efficient navigation with limited environmental information. Our findings show the capability of our approach to handle complex MAS deployment scenarios in dynamic environments. Future work will focus on developing safe navigation algorithms, particularly addressing uncertainty and enhancing agents' decision-making.

REFERENCES

- Afonso, R.J., Maximo, M.R., and Galvao, R.K. (2020). Task allocation and trajectory planning for multiple agents in the presence of obstacle and connectivity constraints with mixed-integer linear programming. *International Journal of Robust and Nonlinear Control*, 30(14), 5464–5491.
- Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, 3420–3431. IEEE.
- Chevet, T., Maniu, C.S., Vlad, C., and Zhang, Y. (2019). Guaranteed voronoi-based deployment for multi-agent systems under uncertain measurements. In *2019 18th European Control Conference (ECC)*, 4016–4021. IEEE.
- De Oliveira, L.B. and Camponogara, E. (2010). Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1), 120–139.
- Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.
- Earl, M.G. and D'Andrea, R. (2007). Multi-vehicle cooperative control using mixed integer linear programming. *Cooperative Control of Distributed Multi-Agent Systems*, 231–259.
- Hatleskog, J., Oлару, S., and Hovd, M. (2018). Voronoi-based deployment of multi-agent systems. In *2018 IEEE Conference on Decision and Control (CDC)*, 5403–5408. IEEE.
- Herceg, M., Kvasnica, M., Jones, C.N., and Morari, M. (2013). Multi-parametric toolbox 3.0. In *2013 European control conference (ECC)*, 502–510. IEEE.
- Ioan, D., Oлару, S., Prodan, I., Stoican, F., and Niculescu, S.I. (2019). From obstacle-based space partitioning to corridors and path planning. A convex lifting approach. *IEEE Control Systems Letters*, 4(1), 79–84.
- Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Murray, R.M. (2007). Recent research in cooperative control of multivehicle systems.
- Rezaee, H. and Abdollahi, F. (2011). Mobile robots cooperative control and obstacle avoidance using potential field. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 61–66. IEEE.
- Wang, P. and Ding, B. (2014). Distributed rhc for tracking and formation of nonholonomic multi-vehicle systems. *IEEE Transactions on Automatic Control*, 59(6), 1439–1453.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & sons.