On corridor enlargement for MPC-based navigation in cluttered environments

Turan Konyalıoğlu, Sorin Olaru, Silviu-Iulian Niculescu, Iris Ballesteros-Tolosana, Simon Mustaki

On corridor enlargement for MPC-based navigation in cluttered environments

T. Konyalioglu [*,**,***] **S. Olaru** [*] **S. I. Niculescu** [*,***]
I. Ballesteros-Tolosana [**] **S. Mustaki** [**]

[*] *Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, France*
{*turan.konyalioglu, sorin.olaru, silviu.niculescu*}*@centralesupelec.fr*
[**] *Ampere Software Technology, DEA-OCDT1, Technocentre, France*
{*turan.konyalioglu, iris.ballesteros-tolosana, simon.mustaki@*}*renault.com*
[***] *Inria-Saclay (team "DISCO")*

Abstract: The construction of a space partition in a cluttered environment allows for the creation of graph-based paths, establishing safe navigation corridors for agents. Then, it exploits them according to the available control degrees of freedom and dynamical constraints. Evaluating corridor safety relies on the distance between the path points and the nearest obstacles, influencing the real-time performance and robustness of navigation. This paper revisits the convex lifting method for space partition, emphasizing the generation and enlargement of safe corridors. The iterative enlargement algorithm pursues an increase in average corridor width while ensuring a monotonic increase in the minimum corridor width.

Keywords: Robotics, Motion control, Optimization and model predictive control.

1. INTRODUCTION

Path planning is a fundamental concept in robotics, which can be resumed as the process of choosing waypoints, ensuring the connectivity between a starting point and a destination point, and maximizing the maneuverability of the robot or vehicle in a particular environment. Finding a practical and effective route while considering obstacles, dynamic and kinematic constraints, and possible uncertainties is a complex problem recognized as such in the control design literature (Paden et al., 2016).

Space representation techniques for path planning can be categorized as follows: sampling-based, connected cells partition, and lattice representation (Claussmann et al., 2019). The challenge lies in determining suitable parameters to provide sufficient information for local mobility and global environment description. Sampling-based partitioning, widely used in robotics and autonomous systems, efficiently navigates complex environments by generating a discrete set of random or strategic points within the configuration space, avoiding exhaustive searches. PRM (Probabilistic Roadmaps) Hsu et al. (2007) is a popular sampling-based algorithm constructing obstacle-free roadmaps, with points connected by pathfinding algorithms like Dijkstra (Dijkstra, 1959) or A* (Hart et al., 1968).

Connected cell decomposition methods like Voronoi diagrams, visibility graphs, and grid-based methods are used to depict the navigation space. A pathfinding algorithm operates on these graphs to determine paths based on specified cost metrics. Voronoi decomposition (Sugihara, 1993) suits cluttered environments, utilizing predetermined points often linked to obstacle vertices or the Chebyshev center. The drawbacks are related to complex adjacent cells and the need for replanning in dynamic environments. The visibility decomposition partitions space by generating segments between obstacle vertices (Lozano-Pérez and Wesley, 1979). Obstacle enlargement algorithms are used to adjust the graph and to shift paths away from obstacles.

The convex lifting approach (Ioan et al., 2020) offers a distinct method for representing the state space by partitioning the environment through convexity. Unlike traditional path planning techniques, it produces partitions independent of obstacle topology, thus simplifying the complexity. It readily ensures path existence, finite-time selection, and collision-free guarantees. However, it lacks constraints or performance indices for directly enhancing corridor width. Thus, there's a need for an algorithm to improve corridor robustness and width. Building upon prior work, this study proposes an enlargement algorithm aiming for a globally monotonic corridor enhancement and a non-decreasing minimum corridor width.

The paper is organized as follows. Section 2 provides a brief explanation of path planning construction via convex lifting. In Section 3, previous research on safe corridors and their enhancements are presented. Section 4 proposes the novel algorithms developed in this work. Finally, Section 5 presents a conclusion summarizing the proposed approach.

Notation: $\mathbb{R}^n$, $\mathbb{R}_>$, $\mathcal{I}_N$ denote the set of real numbers in n-dimensional space, the set of positive real num- bers and the set of non-negative real numbers, $\{1, 2, \ldots, N\}$, respectively. Especially, $\mathcal{I}_N^2 := \{(i, j) : i \in \mathcal{I}_N, j \in \mathcal{I}_N, i \neq j\}$. Besides, $\mathcal{V}(\mathcal{S})$, $\text{int}(\mathcal{S})$ and $\mathcal{A}(\mathcal{S})$ denote the set of vertices, the interior and polyhedral volume of polytope $\mathcal{S}$, respectively. $\text{Proj}(\mathcal{S}, \mathbb{S})$ represents the orthogonal projection of $\mathcal{S}$ onto the space $\mathbb{S}$. Given two sets $\mathcal{S}_1, \mathcal{S}_2 \in \mathbb{R}^d$, we denote the Minkowski sum of two sets, denoted by $\mathcal{S}_1 \oplus \mathcal{S}_2$, is defined as $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{x_1 + x_2 : x_1 \in \mathcal{S}_1, x_2 \in \mathcal{S}_2\}$. The ball and distance function are defined as $\mathbb{B}_{p,r}^n = \{x \in \mathbb{R}^d : ||x - p||_n \leq r\}$ and $d(\mathcal{S}_1, \mathcal{S}_2) = \min_{s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2} ||s_1 - s_2||$.

2. PATH PLANNING USING CONVEX LIFTING

A cluttered environment refers to a finite-dimensional space with a high concentration of obstacles, creating a congested and disorganized navigation environment. The navigation problem in a cluttered environment can be alleviated by constructing safety corridors, which offer a secure path from the starting point to the destination point. A key component is the design of the corridors, which can be done effectively by exploiting space partitions. The space partitioning in a cluttered environment can be defined formally as follows:

Definition 1. Let a finite set of disjoint obstacles $\mathcal{P} = \bigcup_{i\in\mathcal{I}} \mathcal{P}_i$ in a finite-dimensional space $\mathcal{X}$. The sets $\{\mathcal{X}_i\}_{i\in\mathcal{I}}$ satisfying,

(1) $\mathcal{X} = \bigcup_{i\in\mathcal{I}} \mathcal{X}_i$
(2) $\text{int}(\mathcal{X}_i) \cap \text{int}(\mathcal{X}_j) = \emptyset,\ (i,j) \in \mathcal{I}^2$
(3) $\mathcal{P}_i \subset \text{int}(\mathcal{X}_i), \forall i \in \mathcal{I}$ □

is called a partition of $\mathcal{X}$ induced by the obstacles $\mathcal{P}$. Furthermore, if sets, $\mathcal{X}$ and $\mathcal{X}_i$ are polyhedral, then $\mathcal{X}$ is called a polyhedral partition.

The configuration space or feasible region, expressed as

$$C_{\mathcal{X}}(\mathcal{P}) = \mathcal{X}/\mathcal{P} \tag{1}$$

is the set of all possible poses/waypoints that a robot/agent can have in its surroundings.

The essence of the convex lifting approach seeks to derive favorable properties through projection after lifting.

Definition 2. Given a polyhedral partition of a finite-dimensional space $\mathcal{X} = \bigcup_{i\in\mathcal{I}} \mathcal{X}_i$, a convex lifting is defined as a piecewise affine function $z : \mathcal{X} \to \mathbb{R}$ satisfying the following properties:

$$z(x) = a_i^T x + b_i \text{ for } x \in \mathcal{X}_i$$

and

$$z(x) > a_j^T x + b_j,\ \forall x \in \text{int}(\mathcal{X}_i),\ (i,j) \in \mathcal{I}^2, \forall i \neq j \qquad \square$$

The convex lifting approach in the context of path planning involves constructing a polyhedral space partition originating from the obstacles to encapsulate them. Practically, it is implemented by the construction of surfaces in a higher-dimensional space $\mathbb{R}^{d+1}$, such that their projection on $\mathbb{R}^d$ contains one-to-one some given collection of polyhedra (obstacles).

Starting from the provided obstacles, the convex lifting generates a lifted polyhedron. The facets of the lifted polyhedra are projected back onto the original space. The facets of the projections are used as edges and the vertices as nodes to construct an interconnection graph. Dijkstra graph search algorithm (Dijkstra, 1959) is eventually used for path generation. To establish a partition $\mathcal{X}$ as outlined in Definition 1, it is essential to construct the lifting function as in Definition 2 with particular inclusion specifications for each obstacle P_i in the collection of disjoint polyhedral obstacles $\mathcal{P} = \bigcup_{i\in\mathcal{I}} \mathcal{P}_i$. This construction can be tackled by means of a convex optimization as follows:

$$\min_{a_i,b_i} J = \sum_{i=1}^{N_0} \| \begin{bmatrix} a_i^T & b_i \end{bmatrix} \|_2^2 \tag{2a}$$

$$\text{s.t.} \quad a_i^T v + b_i \leq M, \forall v \in \mathcal{V}(\mathcal{P}_i), \forall i \in \mathcal{I}, \tag{2b}$$

$$a_j^T v + b_j \geq a_i^T v + b_i + \epsilon, \forall v \in \mathcal{V}(\mathcal{P}_j), \forall i \neq j \tag{2c}$$

where $M, \epsilon > 0$ are pre-defined boundeness and convexity parameters. Contrary to the generic convex lifting problem in Nguyen et al. (2017), boundedness constraints are imposed instead of continuity conditions to obtain expansions outside of the obstacles on the lifted cells. The epigraph of the computed lifting function represents a polyhedral set

$$\mathcal{L} = \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{d+1} : \begin{bmatrix} a_i^T & -1 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \leq -b_i, i \in \mathcal{I} \right\} \tag{3}$$

Each cell $\mathcal{X}_i$ in Definition 1 is obtained by projecting the facets of the lifted polyhedron $\mathcal{L}$, back into the space, $\mathbb{R}^d$.

$$\mathcal{X}_i = \text{proj}(\mathcal{F}_i^{d-1}(\mathcal{L}), \mathcal{X}), i \in \mathcal{I} \tag{4}$$

Fig. 1 provides an intuitive visual representation of the lifted polyhedron, $\mathcal{L}$, obstacles, $\mathcal{P}_i$, and projected partitions, $\mathcal{X}_i$. In addition, the properties underlined in Proposition 1 result from the nature of the constraints imposed in (2a)-(2c).

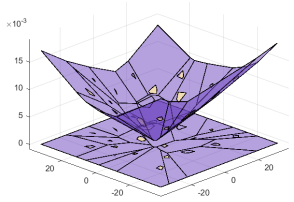


Fig. 1. Space partitioning via convex lifting.

Proposition 1. The polyhedral partition resulted from (4) $\{\mathcal{X}_i\}_{i\in\mathcal{I}}$ has the following properties:

(1) $\mathcal{P}_i \subset \text{int}(\mathcal{X}_i), \forall i$
(2) $\mathcal{X}_i \cap \mathcal{P}_j = \emptyset, \forall j \neq i$ □

3. FROM PATH PLANNING TO NAVIGATION CORRIDORS

3.1 Corridor generation

While the generation of a path addresses the feasibility issue in view of navigation, the corridors allow planning issues to be managed in cluttered environments when considering dynamic constraints and uncertainties. They serve as hard constraints on the trajectory for the agent's navigation by delivering a representation of the collision-free space between obstacles. Corridors are represented by sets to be included in optimization-based controllers to navigate the environment's configuration space by providing a feasible path and a (convex) constrained domain.

Considering the convex lifting approach and its associated interconnection graph, the beginning and ending nodes is required to have a complete path solution (Ioan, 2021) and can be completed by the Definitions 3-4,

Definition 3. 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 $\mathcal{N}$ is set of nodes represented by the vertices of the graph, $\mathcal{E}$ the set of edges, and $f : \mathcal{E} \to \mathbb{R}$, a weights function associated with each edge of the graph. □

The shortest path in terms of given weights may then be found by using the Dijkstra algorithm (Dijkstra, 1959), which yields the path in terms of $\text{Path}(x_0, x_f) = (\bar{x}_0, \bar{x}_1, ..., \bar{x}_n, \bar{x}_{n+1} = \bar{x}_f)$ which can be seen as waypoints. Definition 4 describes a continuous path, path size, and corridor functions.

Definition 4. With given obstacles $\mathcal{P}$, a corridor between two nodes $(x_0, x_f) \in \text{int}(C_{\mathcal{X}}(\mathcal{P}))$, is characterized by the existence of two functions, $\gamma : [0,1] \to C_{\mathcal{X}}(\mathcal{P})$ and

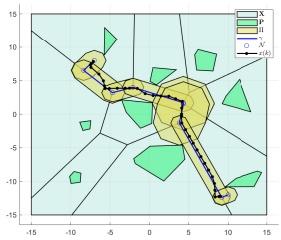


Fig. 2. Safe corridors and MPC-based navigation.

$\rho : [0,1] \to \mathbb{R}_{>0}$ while satisfying the conditions, $\gamma(0) = x_0$, $\gamma(1) = x_f$ and $\gamma(\theta) \oplus \mathbb{B}^2_{0,\rho(\theta)} \subset C_{\mathcal{X}}(\mathcal{P}),\ \forall \theta \in [0,1]$. Then, a corridor in configuration space can be defined as,

$$\Pi = \{x \in \mathbb{R}^d : \exists \theta \in [0,1] \,\text{s.t.}\, x \in \gamma(\theta) \oplus \mathbb{B}^2_{0,\rho(\theta)}\} \qquad \square \quad (5)$$

In particular, the corridor can be defined as the union of convex sets computed for each segment of the piecewise linear path: $\Pi = \bigcup_{i=1}^{N_c} \Pi_i$ with,

$$\Pi_i = \{\ x \in \mathbb{R}^d : \exists \tilde{\theta} \in [0,1] \,\text{s.t.}\ \ x \in \gamma_i(\tilde{\theta}) \oplus \mathbb{B}_{0,\rho_i(\tilde{\theta})}\} \quad (6)$$

where, $\gamma_i(0) = x_i$ and $\gamma_i(1) = x_{i+1}$. Also, the radius that defines the ball in (6) can be found by searching the minimum distance along a path segment.

$$\rho_i(0) = \min_{\mathcal{P}_j \in \mathcal{P}} d(\mathcal{P}_j, \gamma_i), \forall \tilde{\theta} \in [0,1] \quad (7)$$

Each corridor segment is created by applying the Minkowski sum of a path segment and a ball defined by the minimum distance found in (7) as illustrated in Fig. 2.

3.2 MPC-based trajectory generation

In the preceding subsection, a space partitioning-based geometrical path is created and the associated corridor can be used as an explicit constraint for an MPC-based navigation. This method has two advantages: it replaces inherited non-convex constraints with piecewise convex ones, and (as long as linear dynamics and convex constraints are considered) the complexity of the convex MPC technique can be exploited. In order to fix the ideas, let us consider a MPC formulation with LTI dynamics of an agent (e.g. position-speed-acceleration) as follows:

$$x_{k+1} = Ax_k + Bu_k, \quad (8)$$

where $x_k \in \mathbb{R}^d$ is the state vector and $u_k \in \mathbb{R}^m$ is the input vector. The physical limitation of the system induces are represented by compact convex sets $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{U} \subset \mathbb{R}^m$. Each corridor segment represents a subset in the feasible state space domain $\Pi_i \subset \mathcal{X} \subset \mathbb{R}^d$. Using a quadratic cost:

$$\begin{aligned} \mathcal{J}(N_p, \bar{x}_i, x_k, U) &= \|x_{k+N_p|k} - \bar{x}_i\|^2_P \\ &+ \sum_{l=1}^{N_p-1} \|x_{k+l|k} - \bar{x}_i\|^2_Q + \sum_{l=1}^{N_p-1} \|\Delta u_{k+l|k}\|^2_R \end{aligned} \quad (9)$$

where N_p is prediction horizon, $\bar{x}_i$ is reference point such that $\bar{x}_i \in \Pi_i$, Q is state penalty matrix, R is control increment penalty matrix, and P is the terminal cost penalty matix. The vector $U = \begin{bmatrix} u_{k|k} & \dots & u_{k+N_p-1|k} \end{bmatrix}^{\mathrm{T}}$ is the optimization argument:

$$\mathcal{T}(\Pi_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X}) : \min_U \mathcal{J}(N_p, \bar{x}_i, x_k, U) \quad (10a)$$

$$\text{s.t.} \quad x_{k+l+1|k} = Ax_{k+l|k} + Bu_{k+l|k}, \quad (10b)$$

$$u_{k+l|k} \in \mathcal{U},\ \forall l = 1 : N_p - 1, \quad (10c)$$

$$x_{k+l|k} \in \Pi_i, \quad (10d)$$

$$x_{k+N_p|k} \in \mathcal{X}_f(\bar{x}_i) \quad (10e)$$

Regarding the non-convexity of the corridor, Π, the MPC analysis cannot be performed from the starting point to the endpoint and should be divided into each corridor segment Π_i. Consequently, by integrating state-space dynamics (10b), input constraints (10c), and state constraints (10d) induced by the generated corridor, the MPC problem in (10) can be solved at each time step throughout a convex corridor segment. On the other hand, during the transition to another corridor, we have to ensure that the agent will end up in a set defined by terminal constraints, defining a control invariant set for the agents dynamics.

Another issue that should be pointed out regarding the tuning of the MPC is that the prediction horizon should be chosen such that an agent can reach the terminal set, $\mathcal{X}_f(\bar{x}_i)$ from any initial point within the corridor. Using the BRS (backward reachable set) construction, one can calculate a prediction horizon that guarantees recursive feasibility. This step includes iteratively advancing backward to a set such that $S_i \subset \mathcal{X}(\bar{x}_i)$ from the terminal set of the segment i by taking into account the dynamics and constraint sets. The related control invariance set design step is recalled in the Algorithm 1 for completeness.

Algorithm 1 Controlled Invariance and BRS computation.

Input: $\bar{x}, A, B, U, \Pi_i$
Output: $\mathcal{X}_f, N_p$

1: Obtain a stabilizing feedback gain, K, by solving the Riccati equation for the system (8).
2: Find the closed loop system matrix, $A_c = A + BK$.
3: Define a $\mathcal{X}_f = O_0 = \{Hx \leq w\}$, such that $\mathcal{X}_f \subseteq \Pi_i \cap \Pi_{i+1}$
4: **for** $j = 1 : N$ **do**
5: $\quad O_j = O_{j-1} \cap \{HA_c^j \leq w\}$
6: **end for** $X_f^0 = O_j$
7: Solve the following LP:

$$\max_{\lambda} \lambda \,\text{s.t.}\, \bar{x}_i \oplus \lambda \mathcal{X}_f^0 \subset \Pi_i,\ \lambda < 1 \quad (11)$$

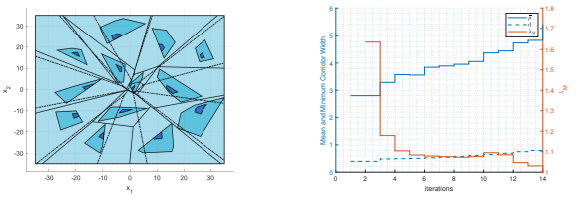
8: $\tilde{R}_0^i = \bar{x}_i \oplus \lambda \mathcal{X}_f,\ j = 0$
9: **while** $\tilde{R}_j^i \subset \Pi_i \mid \tilde{R}_{j+1}^i \neq \tilde{R}_j^i$ **do**
10: $\quad \tilde{R}_j^i = A^{-1}(\tilde{R}_{j-1}^i \oplus (-BU)) \cap \Pi_i$
11: $\quad N_p += 1$
12: **end while**

Remark 1. Reachability analysis underlying Algorithm 1 is affected by the size of the terminal set, which is related to the size of the corridors. BRS computation will better perform (resulting in a smaller number of prediction steps in MPC) if the terminal set and the corridors are larger.

Remark 1 points to the necessity of a corridor enlargement. Once the prediction horizon and terminal set are constructed, one can solve the MPC problem for the sequence of the corridors iteratively. For each corridor, the parameters N_p, $\mathcal{X}_f$ and $\bar{x}_i$ are updated, and the terminal state is considered as the initial state of the next iteration. Fig. 2 depicts the results of Algorithm 1 for an integrator-like dynamics.

3.3 Enlargement with obstacle scaling

Starting from the need of large navigation corridors, the improvement of the paths obtained through convex lifting



(a) Illustration. (b) Results.

Fig. 3. Enlargement with obstacle scaling

was investigated in Mirabilio et al. (2022). The concept involves constructing partitions while iteratively enlarging the obstacles as a subset of polyhedral space partition.

Let Q and H be two non-empty convex obstacles, respectively defined as in the following:

$$Q = \{x \in \mathbb{R}^d : A_q x \leq b_q\}, A_q \in \mathbb{R}^{q \times d}, b_q \in \mathbb{R}^q \quad (12)$$

$$H = \{x \in \mathbb{R}^d : A_h x \leq b_h\}, A_h \in \mathbb{R}^{h \times d}, b_h \in \mathbb{R}^h \quad (13)$$

where Q and H represent obstacles and space partitions respectively, such that $Q \subset H$, with $q, h \in \mathbb{N}$. Recalling the Extended Farkas' Lemma in Hennet (1989):

Theorem 1. Considering two non-empty polytopes, H, defined by $A_h x \leq b_h$ and Q, defined by $A_q x \leq b_q$ such that $Q \subset H$, H is satisfied by any point by Q, if and only if there exist a matrix $U \in \mathbb{R}^{h \times q}$ satisfying conditions $UA_q = A_h$ and $Ub_q \leq b_h$.

$$Q^\lambda = \{x \in \mathbb{R}^n : A_q x \leq \lambda b_q + (1-\lambda) A_q c_q \quad (14)$$

Then, the enlarged set can be defined as in (14). Enlargement is achieved regarding the linear optimization problem shown in equation (15).

$$\min_{c_q, \mu_1, \mu_2, \tilde{U}} \quad \mu_1 \quad (15a)$$

$$\text{s.t.} \quad \tilde{U} A_q = \mu_1 A_h, \ \tilde{U} \succ 0, \quad (15b)$$

$$\tilde{U} b_q - A_h c_q \leq \mu_2 b_h, \quad (15c)$$

$$A_q c_q \leq b_q \quad (15d)$$

$$\mu_1 - \mu_2 = 1, \ \mu_1 \geq 1, \mu_2 \geq 0, \quad (15e)$$

where μ_1 and μ_2 are the auxiliary variables used to transform nonlinear optimization problem (22) in Mirabilio et al. (2022) into linear one and $\tilde{U}$ is the resulting matrix corresponds to *Theorem*1. Constraint in (15d) guarantees the center of the enlargement will be inside of the obstacles, $c_q \in Q$. After solving the linear optimization problem in (15), λ_M, the enlargement coefficient can be calculated as $\lambda_M = 1 + 1/\mu_2^*$. In the optimization problem, the maximum enlargement of an obstacle, $\mathcal{P}_i$, constrained with the encapsulating partition, $\mathcal{X}_i$, is solved for each obstacle. Next, the minimum of the achievable enlargement is selected, and the enlargement operation is performed for all obstacles. The algorithm is repeated until the enlargement coefficient, λ_M, is converged to a certain value. Since enlarged obstacles in each iteration, k, ensures $P_i^0 \subset P_i^1 ... \subset P_i^k$, the algorithm doesn't make any concessions with regards to feasibility. An analysis related to the Algorithm 2 is performed, and enlargement results are illustrated in Fig. 3a for comparison. The dotted line shows the boundaries of the initial space partition. As is seen, one can achieve better performance regarding the performance index computed based on (16).

$$\bar{p} = \frac{1}{N_{\tilde{\mathcal{E}}}} \sum_{\tilde{\mathcal{E}}_i \in \tilde{\mathcal{E}}} \min_{\mathcal{P}_i \in \mathcal{P}} d(\mathcal{P}_j, \tilde{\mathcal{E}}_i) \quad (16)$$

Algorithm 2 Enlargement with polytope scaling.

Input: $\mathcal{X}, \mathcal{P} = \bigcup_{i=1}^N \mathcal{P}_i, \epsilon, M > 0$, and $\bar{\lambda}_M$.
Output: $\{X_i\}_{i \in \mathcal{I}}$.
1: $\mathcal{P}^0 = \mathcal{P}$.
2: **while** $\lambda_M^k > \bar{\lambda}_M$ **do**
3: Find $\{\mathcal{X}_i\}_{i \in \mathcal{I}}$ by solving (2) with respect to $\mathcal{P}^k$ for $\mathcal{L}^k$ and project the facets of $\mathcal{L}^k$ into $\mathcal{X}$. as in (4)
4: For each $(\mathcal{P}_i^k, \mathcal{X}_i^k)$, solve the optimization problem in (15) for $\lambda_{M,i}^k$ and $c_{q,i}^k$, then find $\lambda_M^k = \min_{i \in \mathcal{I}} \lambda_{M,i}^k$.
5: Compute set $\mathcal{P}^{k+1}$ by scaling each $\mathcal{P}_i^k$ using (14).
6: **end while**

with $\tilde{\mathcal{E}}$ is set of edges for all possible paths and $N_{\tilde{\mathcal{E}}} = |\tilde{\mathcal{E}}|$ number of edges. The performance index corresponds to the average width of all possible paths. However, the minimum width of the corridor,

$$\mathrm{d} = \min_{\mathcal{P}_j \in \mathcal{P}} d(\mathcal{P}_j, \mathcal{E}_i), \text{ for } i = 1, ..., |\tilde{\mathcal{E}}| \quad (17)$$

directly affects the global corridor performance and this is independent of the average corridor width. Furthermore, opting for the center of enlargement on obstacle facets via optimization problem (15) can lead to significant reductions in minimum corridor width. Hence, based on these findings, we suggest incorporating the second performance index outlined in (17) into the enlargement analysis.

In order to illustrate the potential issues in the obstacle scaling, the result of the existing algorithm is tested [1] as shown in Fig. 3a and Fig. 3b. The dotted black lines in Fig. 3a shows the space partition, $\mathcal{X}^0$. In Fig. 3b, performance indices are shown in blue, and the enlargement coefficient is shown in red line. The convex lifting optimization constants are selected as follows: $\epsilon = 10^{-4}$, $M = 10^5$ and $\bar{\lambda}_M = 1.005$. The average corridor width reaches approximately 5.4. However, the second performance index, d, increases only marginally, and this issue will represent the focus of the developments in the next section.

4. NOVEL SOLUTIONS FOR CORRIDOR ENLARGEMENT

In this section, several methods to overcome the issues of corridor enlargements are presented with the aim of obtaining a homogeneous enlargement. The procedures will proceed iteratively, providing an increase in the mean corridor width while preserving the feasibility.

4.1 Chebyshev-centered enlargement approach

As a first technique, one can use the enlargement method mentioned previously by ensuring the center of each enlargement is inside the respective obstacle (polytope). An intuitive candidate in this respect is the Chebyshev center.

$$c_q = \arg\min_{x_c} \quad -r \quad (18a)$$

$$\text{s.t.} \quad \{x \in \mathbb{R}^d : ||x - x_c||_2 \leq r\} \subseteq \mathcal{X}, \ x_c \in \mathcal{X}, \quad (18b)$$

Computing the Chebyshev center is actually an effective linear program Boyd and Vandenberghe (2004).

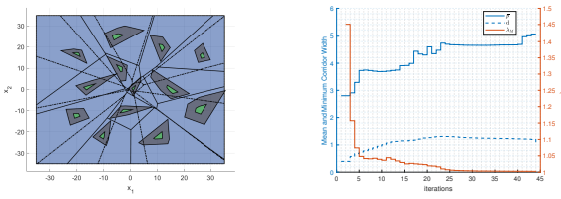
$$\min_{\mu_1, \mu_2, \tilde{U}} \quad \mu_1 \quad (19a)$$

$$\text{s.t.} \quad \tilde{U} A_q = \mu_1 A_h, \ \tilde{U} \succ 0, \quad (19b)$$

$$\tilde{U} b_q - A_h c_q \leq \mu_2 b_h, \quad (19c)$$

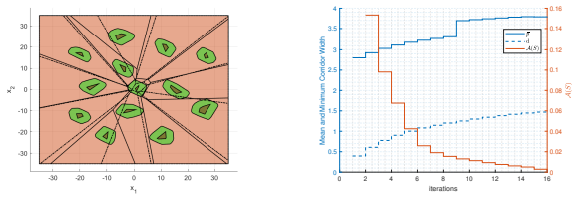
$$\mu_1 - \mu_2 = 1, \ \mu_1 \geq 1, \mu_2 \geq 0, \quad (19d)$$

[1] YALMIP (Löfberg, 2004) and MPT toolboxes (Herceg et al., 2013) are used for the construction.



(a) Illustration. (b) Results.

Fig. 4. Enlargement with Chebyshev-centered scaling.



(a) Illustration. (b) Results.

Fig. 5. Enlargement based on Minkowski-sum.

Consequently, the framework (15) can be adapted by removing (15d), leading to the optimization problem (19). For the sake of completeness, the whole procedure is described in Algorithm 3, and the result of the procedure is illustrated in Fig. 4a for the same numerical example.

Remark 2. The Chebychev-centered method provides an iterative enlargement for each obstacle ensuring the performance index in (17) is non-decreasing. □

The numerical results of the Chebyshev-centered enlargement method are depicted in Fig. 4b with the same convex lifting optimization parameters. The maximum enlargement for the polytope cannot be achieved under such a strong structural constraint; thus, a low convergence rate can be considered normal. As discussed in Remark 2, the method performs better with respect to the minimum corridor performance index, which converges to a value of approximately 1.2. Also, the improvement in the minimum corridor width is done to the detriment of the mean corridor width, which is found to be 5.1.

Proposition 2. Consider a cluttered environment with a set of obstacles, $\mathcal{P}$, with the property of non-homogeneity in terms of obstacle size such that $\mathcal{A}(\mathcal{P}_i) \gg \mathcal{A}(\mathcal{P}_j)$ where $(i,j) \in \mathcal{I}^2$. Then, the result of the enlargement with the scaling coefficient employed in Algorithm 2 and 3 may not provide a global consistent corridor enlargement. □

Sketch of proof. Lets consider two obstacles, such that $\mathcal{A}(\mathcal{P}_1^0) \gg \mathcal{A}(\mathcal{P}_2^0)$ such that an adjacent facet, $\mathcal{F}_{1,2}^k = \text{adj}(\mathcal{X}_1^k, \mathcal{X}_2^k)$ is shared by $\mathcal{X}_1$ and $\mathcal{X}_2$ resulted from (2). The distances, d_1^k, d_2^k between the facet and these two obstacles can be constructed as $d_1^k = d(\mathcal{P}_1^0, \mathcal{F}_{1,2}^k)$ and $d_2^k = d(\mathcal{P}_2^0, \mathcal{F}_{1,2}^k)$. Clearly, the distance between these two obstacles is $\mathcal{D}_{1,2}^k = d(\mathcal{P}_1^k, \mathcal{P}_2^k)$ and that the aim of the enlargement is to achieve a ratio

$$(d_1^k + d_2^k)/\mathcal{D}_{1,2}^k \longrightarrow 1 \tag{20}$$

where $d_1^k \simeq d_2^k$. However, with the existence of a large size difference with the same enlargement coefficient, the relation between the volume will impact the distance and $d_1^{k+1} - d_1^k \gg d_2^{k+1} - d_2^k$. This relation causes $d_1^k < d_1^{k+1}$ whereas $d_2^k > d_2^{k+1}$ which provokes shrinkage in the $\mathcal{X}_2^{k+1}$ and that the enlargement with a coefficient as proposed in Algorithm 2 and 3 do not provide a proper corridor enlargement achieving (20). □

In other words, for obstacles of different sizes, the enlargement coefficient may not reflect the impact on the absolute corridor enlargement. In view of the insight offered by Proposition 2, the next methods aim to propose alternatives to achieve global enlargement by considering directly the performance indices in (16) and (17).

Algorithm 3 Chebyshev-centered enlargement with polytope scaling.

Input: $\mathcal{X}, \mathcal{P} = \bigcup_{i=1}^N \mathcal{P}_i, \epsilon, M > 0$, and $\bar{\lambda}_M$.
Output: $\{X_i\}_{i\in\mathcal{I}}$.
1: $\mathcal{P}^0 = \mathcal{P}$.
2: **while** $\lambda_M^k > \bar{\lambda}_M$ **do**
3: Find $\{\mathcal{X}_i\}_{i\in\mathcal{I}}$ by solving (2) with respect to $\mathcal{P}^k$ for $\mathcal{L}^k$ and project the facets of $\mathcal{L}^k$ into $\mathcal{X}$. as in (4)
4: Compute $c_{q,i}$ for each $\mathcal{P}_i^k$, regarding the problem (18).
5: For each $(\mathcal{P}_i^k, \mathcal{X}_i^k)$, solve the optimization problem in (19) for $\lambda_{M,i}^k$, then find $\lambda_M^k = \min_{i\in\mathcal{I}} \lambda_{M,i}^k$.
6: Compute set $\mathcal{P}^{k+1}$ by scaling each $\mathcal{P}_i^k$ using (14).
7: **end while**

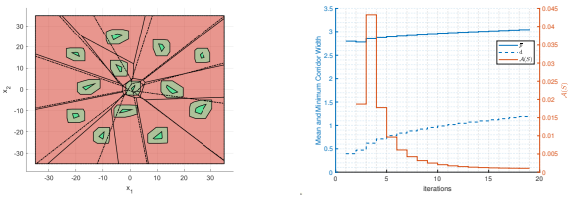
4.2 Minkowski-sum based enlargement approach

The methods presented so far perform the enlargement regardless of the distance between obstacles and facets. As an alternative to such a relative procedure, an absolute enlargement can be obtained by means of the Minkowski addition of a properly adjusted set to each of the obstacles in the environment. The main advantage of the Minkowski-sum-based enlargement is the capability of adapting the enlargement, considering each critical distance between obstacles and facets while preserving the feasibility of the lifting.

The method is summarized in Algorithm 4. To obtain a common set that provides enlargement, at first, all adjacent facets of the generated state partition are identified, and each distance, d_i and d_j, between the adjacent facet and the obstacle located as a subset of the partition is calculated. The resulting hyperplane is treated as a separating hyperplane between the concerning obstacle and the facet. The same process is applied to the obstacles located in the adjacent partition. When all hyperplanes are found for all adjacent facets, the intersection of the hyperplanes results in a polytopic common set that is used in enlargement, ensuring that $\mathcal{P}_i^{k+1} \subseteq \mathcal{X}_i^k$ for $\forall i \in \mathcal{I}$.

Algorithm 4 Enlargement with Minkowski-Sum

Input: $\mathcal{X}, \mathcal{P}^0 = \bigcup_{i=1}^N \mathcal{P}_i^0, \epsilon, M > 0$, and $\bar{a}$.
Output: $\{X_i\}_{i\in\mathcal{I}}$.
1: **while** $\mathcal{A}(S^k) > \bar{a}$ **do**
2: Find $\{\mathcal{X}_i\}_{i\in\mathcal{I}}$ by solving (2) with respect to $\mathcal{P}^k$ for $\mathcal{L}^k$ and project the facets of $\mathcal{L}^k$ into $\mathcal{X}$ as in (4) and find all edges as $\mathcal{E} = \mathcal{F}(\{\mathcal{X}_i\}_{i\in\mathcal{I}}^k)$.
3: **for** each $\mathcal{E}_l \in \mathcal{E}$ **do**
4: Find adjacent partition pairs such that $\mathcal{E}_l = \text{adj}(\mathcal{X}_i^k, \mathcal{X}_j^k)$.
5: Compute the basis $w = \min(\boldsymbol{d}_i, \boldsymbol{d}_j)$ where $\boldsymbol{d}_i = d(\mathcal{E}_l, \mathcal{P}_i^k)$.
6: Compute separating hyperplanes $\{x \in \mathbb{R}^n : H_m x \leq w\}, m \in \{i,j\}$ which defines the boundary of the partition for each index, i and j.
7: $S^k = \bigcap_m \{x \in \mathbb{R}^n : H_m x \leq w\}, m \in \{i,j\}$
8: **end for**
9: Calculate enlargement, $\mathcal{P}_i^{k+1} = \mathcal{P}_i^k \oplus S^k$, for each $\mathcal{P}_i^k \in \mathcal{P}^k$.
10: **end while**



(a) Illustration. (b) Results.

Fig. 6. Enlargement using the Minkowski-sum with a ball.

For the illustration, the Algorithm 4 is performed for the same cluttered environment as before with the same convex lifting optimization parameters, and the result is shown in Fig. 5a and Fig. 5b. The constant, $\bar{a}$, is selected as 0.001, and the procedure lasts 16 iterations to reach the threshold value. In Fig. 5b, the monotonic decrease in the volume of set S^k can be observed. Also, the minimum corridor width converges approximately to the limit value of 1.5. In contrast, the Minkowski-sum-based method provides better performance regarding the performance index in (17). As a drawback of the Minkowski-sum-based enlargement, the computed common set can become complex due to the change of shape at each iteration.

4.3 Minkowski-sum based enlargement using a predefined shape (ball)

The drawback mentioned in the previous subsection can be addressed readily by pre-imposing the shape or the generators of the operand in the Minkowski-sum. A polyhedral set generated by a ball with the appropriate norm (1 or ∞) avoids the aforementioned complexities. The method is summarized in Algorithm 5, and it proceeds similarly to Algorithm 4. As a distinction, instead of computing distance from facets, the minimum distance is calculated as in (17). Then, the 1-norm ball, $\mathbb{B}^1_{p,r}$, is defined by the minimum distance instead of a common set in Algorithm 4. The norm of the ball is chosen to be the 1-norm for its simplicity and to maintain feasibility.

The final enlargement and the result of the procedure are illustrated in Fig 6a and Fig 6b, respectively. The optimization parameters are the same as for the previous methods. The simplicity of the final obstacles can be observed in the related figure.

Algorithm 5 Enlargement with Minkowski-sum by using ball

Input: $\mathcal{X}, \mathcal{P}^0 = \bigcup_{i=1}^{N} \mathcal{P}_i^0, \epsilon, M > 0$, and $\bar{a}$.
Output: $\{X_i\}_{i\in\mathcal{I}}$.
1: **while** $\mathcal{A}(\mathbb{B}^1_{0,d}) > \bar{a}$ **do**
2: Find $\{\mathcal{X}_i\}_{i\in\mathcal{I}}$ by solving (2) with respect to $\mathcal{P}^k$ for $\mathcal{L}^k$ and project the facets of $\mathcal{L}^k$ into $\mathcal{X}$ as in (4) and find all edges as $\mathcal{E} = \mathcal{F}(\{\mathcal{X}_i\}^k_{i\in\mathcal{I}})$.
3: Compute the minimum distance between obstacles and the facets, $d = \min_{P_j\in\mathcal{P}} d(P_j, \mathcal{E}_i)$, for $i = 1, ..., |\mathcal{E}|$
4: Calculate enlargement, $\mathcal{P}_i^{k+1} = \mathcal{P}_i^k \oplus \mathbb{B}^1_{0,d}$, for each $\mathcal{P}_i^k \in \mathcal{P}^k$.
5: **end while**

5. CONCLUSION

The paper reviewed the previous developments in the path-planning framework using convex lifting. The necessity of the enlargement of the corridors was recalled for the MPC-based navigation. The purpose of this enlargement is to create wider paths for an agent to maneuver and perform obstacle avoidance in a cluttered area. The previous enlargement attempts were recalled, and the drawbacks of the respective algorithm were pointed out. A series of novel enlargement approaches are proposed to overcome the identified drawbacks. Most importantly, the paper proposes the use of a novel performance index based on the segment of the path. Compared with the existing methods, an improved efficiency is obtained with respect to the convergence rate and minimum corridor width. On the other hand, it is shown that fine-tuned enlargement sets can bring computational burden, and the use of predefined shapes can represent adequate alternatives. Future work will focus on the development of the same theoretical foundation for dynamic cluttered environments.

REFERENCES

Boyd, S.P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Claussmann, L., Revilloud, M., Gruyer, D., and Glaser, S. (2019). A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(5), 1826–1848.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107. doi:10.1109/tssc.1968.300136. URL `https://doi.org/10.1109/tssc.1968.300136`.

Hennet, J.C. (1989). Une extension du lemme de farkas et son application au probleme de régulation linéaire sous contraintes. *CR Acad. Sci. Paris*, 308(I), 415–419.

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.

Hsu, D., Latombe, J.C., and Kurniawati, H. (2007). On the probabilistic foundations of probabilistic roadmap planning. In *Robotics Research: Results of the 12th International Symposium ISRR*, 83–97. Springer.

Ioan, D. (2021). *Safe Navigation Strategies within Cluttered Environment*. Ph.D. thesis, Université Paris-Saclay.

Ioan, D., Prodan, I., Olaru, S., Stoican, F., and Niculescu, S.I. (2020). Navigation in cluttered environments with feasibility guarantees. *IFAC-PapersOnLine*, 53(2), 5487–5492.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*. Taipei, Taiwan.

Lozano-Pérez, T. and Wesley, M.A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10), 560–570.

Mirabilio, M., Olaru, S., Dórea, C.E., Iovine, A., and Di Benedetto, M.D. (2022). Path generation based on convex lifting: optimization of the corridors. *IFAC-PapersOnLine*, 55(16), 260–265.

Nguyen, N.A., Gulan, M., Olaru, S., and Rodriguez-Ayerbe, P. (2017). Convex lifting: Theory and control applications. *IEEE Transactions on Automatic Control*, 63(5), 1243–1258.

Paden, B., Čáp, M., Yong, S.Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1), 33–55.

Sugihara, K. (1993). Approximation of generalized voronoi diagrams by ordinary voronoi diagrams. *CVGIP: Graphical Models and Image Processing*, 55(6), 522–531.