Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Navigation in cluttered environments with feasibility guarantees

Daniel Ioan [*] **Ionela Prodan** [**] **Sorin Olaru** [*] **Florin Stoican** [***]
Silviu-Iulian Niculescu [*,⋆]

[*] *University Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes (L2S), 3, rue Joliot Curie, F-91192, Gif-sur-Yvette, France. (e-mail: {daniel.ioan, sorin.olaru, silviu.niculescu}@l2s.centralesupelec.fr)*
[**] *Univ. Grenoble Alpes, Grenoble INP⋆⋆, LCIS, F-26000 Valence. (e-mail: ionela.prodan@lcis.grenoble-inp.fr)*
[***] *Department of Automatic Control and Systems Engineering, UPB, Romania (e-mail: florin.stoican@acse.pub.ro)*

Abstract: This paper addresses the navigation problem in a multi-obstacle environment and makes use of convex lifting in trajectory planning problems with anti-collision constraints. The design problem is commonly stated in the literature in terms of a constrained optimization problem over a non-convex domain. In our framework, the convex lifting approach plays an instrumental role in the partitioning of the feasible space in accordance with the distribution of obstacles and in the subsequent generation of corridors in cluttered environments. We consider an adaptation of the generic MPC (Model Predictive Control) trajectory tracking problem, aiming to guarantee the feasibility and the convergence. Simulation results and proof of concepts illustrations prove the effectiveness of the proposed approach.

Keywords: obstacle avoidance, constrained control, linear systems

1. INTRODUCTION

Navigation through multi-obstacle environments has received significant attention from both control and robotics communities (Janecek et al., 2017) due to its many applications including, among others, monitoring or surveillance (Puri, 2005), autonomous overtaking (Ballesteros-Tolosana et al., 2017) or precision agriculture (Jawad et al., 2017). The main challenge resides in the non-convexity of the feasible regions in the motion space.

To tackle the navigation problem, several approaches have been developed. To the best of authors knowledge, most of them divide the problem in two main sub-tasks: path/trajectory generation and trajectory tracking. Moreover, these (sub-)tasks are usually viewed as independent or decoupled problems (Latombe, 2012). For instance, the classical sample-based [1] approaches, Karaman and Frazzoli (2011), LaValle (1998), are prone to focus on the first task, neglecting the second one and, thus, simplifying the problem. Next, the optimization-based strategies, e.g. mixed-integer formulations Richards and How (2002); Prodan et al. (2015), convexification techniques Szmuk et al. (2017), potential field methods Chen et al. (2016) or set-theoretic approaches Franzè and Lucia (2015) merge the planning and tracking tasks at the expense of a higher computational complexity, especially in the case of cluttered/congested multi-obstacle environments. In addition, there are a few works, e.g., Weiss et al. (2017), that have a more intrinsic approach, considering planning and tracking as distinct but interacting tasks Berntorp et al. (2017) and employing appropriate methods from the previously mentioned categories. For instance, in this last category, we may include Berntorp et al. (2018), where the authors proposed an extension of the RRT(rapidly-exploring random tree), using feedback control and positively invariant sets to guarantee collision-free tracking.

The main drawback of the methods involving sample-based techniques is their *probabilistic completeness* (Barraquand et al., 1997). More specifically, the probability that the algorithm returns a feasible solution tends to 1 if the number of sample points is sufficiently large (approaching ∞), as it was empirically shown in Hsu et al. (2007). Moreover, these probabilistic completeness proofs do not provide any guarantee on the time in which the algorithm finds the optimal path (if there exists one). Therefore, the primary objective of our paper is the *global feasibility* and we aim to develop a method characterized by *completeness, eliminating the risks of an heuristic/unpredictable behavior.*

More precisely, we revisit the ideas proposed by Ioan et al. (2019) and we proceed further in providing global feasibility guarantees and enhancing the effective control strategy. Basically, in a first stage we neglect the differential constraints and the physical limitations that can appear in the motion planning in order to generate a geometric path. This path ensures the avoidance of obstacles and has the

⋆ The author is also with Inria Saclay, "DISCO" Team, France.
⋆⋆Institute of Engineering Univ. Grenoble Alpes
[1] Sample-based approaches are generally related to the construction of a graph structure.

potential to explicitly describe a feasible corridor similar to the ones in Liu et al. (2017) or in Faulwasser and Findeisen (2016). At a second stage, using the geometric path and the corridor as starting points, we find some appropriate trajectory respecting the agent's dynamics and constraints using a MPC strategy.

The main contributions of this paper are twofold:

i) providing a detailed analysis of the recursive feasibility of a corridor-constrained MPC;
ii) providing a generic navigation strategy for cluttered environments with feasibility guarantees.

The remaining of the paper is organized as follows: Section 2 presents some basic set-theoretic tools, introduces the formulation of the problem and provides a mechanism able to generate feasible geometric paths through the multi-obstacle environment. Section 3 presents a prototype control strategy with recursive feasibility guarantees and the generic navigation strategy, pointing to the extensions to the problem of multiple agents. Finally, some concluding remarks end the paper

Notation: The Minkowski sum of two sets is denoted as $A \oplus B = \{x : x = a + b,\ a \in A, b \in B\}$. Given a compact set $S \in \mathbb{R}^n$, $\mathcal{C}_X(S)$ denotes the complement of S over $X \in \mathbb{R}^d$, $\mathrm{Com}(S)$ the space of compact subsets of S, $\mathrm{int}(S)$ its interior and $\mathcal{V}(S)$ the set of its extreme points. As well, $\mathrm{proj}_{[d_1:d_2]} S$ stands for projection of S on the dimensions d_1 to d_2. For a polyhedron $P \in \mathbb{R}^d$, $\mathcal{V}(P)$ is the (finite) set of its vertices, and $\mathcal{F}_i^k(P)$ is the i-th face of the dimension $k < d$. For $x \in \mathbb{R}^d$ we denote $\|x\|_Q^2 = x^\top Q x$. $\mathbb{B}_{p,r} = \{x \in \mathbb{R}^d : \|x - p\| \leq r\}$ is a ball of radius $r \geq 0$ centered in $p \in \mathbb{R}^d$ w.r.t. a given norm.

2. PREREQUISITIES AND PROBLEM FORMULATION

Consider a finite dimensional output space $\mathbb{R}^d$ and a finite number of interdicted convex regions $P_j \subset \mathrm{Com}(\mathbb{R}^d)$, $j \in \mathcal{I} = \{1, \ldots, N_o\}$ as obstacles:

$$\mathbb{P} = \bigcup_{j=1}^{N_o} P_j;\ P_i \cap P_j = \emptyset, \forall i \neq j, \tag{1}$$

their union lies in a bounded [2] *cluttered environment* $\mathbb{X}$:

$$\mathbb{P} \subset \mathrm{int}(\mathbb{X}) \subset \mathbb{R}^d \tag{2}$$

while the obstacle-free (implicitly, non-convex) domain is

$$\mathcal{C}_{\mathbb{X}}(\mathbb{P}) \triangleq \mathbb{X} \setminus \mathbb{P}. \tag{3}$$

Definition 1. (Corridors - Ioan et al. (2019)). Given the obstacles $\mathbb{P}$, a corridor between two points $x_0,\ x_f \in \mathrm{int}(\mathcal{C}_{\mathbb{X}}(\mathbb{P}))$ is enabled by the existence of two continuous functions:

$$\gamma : [0,1] \to \mathcal{C}_{\mathbb{X}}(\mathbb{P}),\ \rho : [0,1] \to \mathbb{R}_{>0} \tag{4}$$

satisfying $\gamma(0) = x_0$, $\gamma(1) = x_f$ and $\gamma(\theta) \oplus \mathbb{B}_{0,\rho(\theta)} \subset \mathcal{C}_{\mathbb{X}}(\mathbb{P}), \forall \theta \in [0,1]$. The corridor is defined as:

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

Definition 2. A family of sets $\{X_i\}_{i \in \mathcal{I}}$ verifying:

i) $\mathbb{X} = \bigcup_{i=1}^{N_o} X_i$,
ii) $\mathrm{int}(X_i) \bigcap \mathrm{int}(X_j) = \emptyset, \forall i \neq j \in \mathcal{I}$,
iii) $P_i \subset \mathrm{int}(X_i), \forall i \in \{1 \ldots, N_o\}$

is called a partition of $\mathbb{X}$ induced by the obstacles $\mathbb{P}$. Furthermore, if the sets $\mathbb{X}$ and $X_i, \forall i$ are polyhedral, then $\mathbb{X} = \bigcup_{i=1}^{N_o} X_i$ is called a polyhedral partition. ♦

The problem of navigation in cluttered environments $\mathbb{X}$ can be divided in three sub-problems:

P1) Starting from the set of forbidden regions $\mathbb{P}$, determine a partition of the *cluttered environment* around them.
P2) Considering any two points $x_0, x_f \in \mathrm{int}(\mathcal{C}_{\mathbb{X}}(\mathbb{P}))$, construct a corridor linking them or provide a certificate of infeasibility.
P3) Given a non-empty corridor, select/generate a continuous trajectory $\tau : [t_0, t_f] \to \Pi$ guaranteeing collision avoidance, i.e. $\tau(\theta) \cap \mathbb{P} = \emptyset, \forall \theta \in [0,1]$.

Previous work Ioan et al. (2019) addressed the first two problems P1) and P2). By developing a systematic solution, we aim hereinafter to treat in detail problem P3) and, additionally, to provide some appropriate feasibility guarantees. Prior to main developments, we recall the constructive method for the resolution of P1) and P2).

Definition 3. (Nguyen et al. (2018)). Given a collection of obstacles $\mathbb{P} = \bigcup_{j=1}^{N_o} P_j$ with $P_i \cap P_j = \emptyset, \forall i \neq j$, as defined in (1), and a partitioning of the cluttered environment $\mathbb{X} \supset \mathbb{P}$, the function $z : \mathbb{X} \to \mathbb{R}$ is called a *piecewise affine lifting* if:

$$z(x) = a_i^\top x + b_i,\ x \in X_i, \tag{6}$$

with X_i satisfying $\mathrm{int}(X_i) \supset P_i$, $\forall i$, $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$.

Taking P_i as polyhedral sets (i.e, having a finite number of extreme points), the lifting can be constructed as the result of the following convex optimization problem:

$$\min_{a_i, b_i} \sum_{i=1}^{N_o} \| \begin{bmatrix} a_i & b_i \end{bmatrix}^\top \|_2^2 \tag{7}$$

$$\text{s.t. } a_i^\top v + b_i \geq a_j^\top v + b_j + \epsilon, \forall v \in \mathcal{V}(P_i), \forall i \neq j, \tag{8a}$$

$$a_i^\top v + b_i \leq M, \forall v \in \mathcal{V}(P_i),\ \forall i. \tag{8b}$$

Based on the solutions of (7), we define the following "d+1"-dimensional polyhedron:

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

Projecting the facets of $\mathcal{P}$ on $\mathbb{X}$ provides a polyhedral partition $\{X_i\}_{i=1:N_o}$.

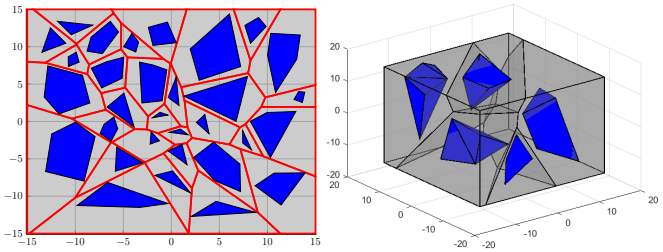
Corollary 4. The polyhedral partition $\{X_i\}_{i=1:N_o}$ has the following properties:

i) $P_i \subset \mathrm{int}(X_i)$, $\forall i$,
ii) $X_i \cap P_j = \emptyset$, $\forall j \neq i$.

Fig. 1 shows some partitioning of complicated 2D and 3D cluttered environments. Since in the motion planning context generally we do not go higher than 3D, we restricted the illustrations to these dimensions, but the construction from Corollary 4 holds also for arbitrary dimensions.

Once the partition $\{X_i\}_{i=1:N_o}$ of the cluttered environment $\mathbb{X}$ is available, the goal is to construct a graph in

[2] This assumption is always valid for some bounded $\mathbb{X}$, due to the boundedness assumptions on P_j, $j \in \mathcal{I}$.



(a) $N_o = 31$, $d = 2$ (b) $N_o = 5$, $d = 3$

Fig. 1. Space-partitioning $\{X_i\}_{i=1:N_o}$ from (7).

order to generate feasible paths through $\mathbb{X}$. Therefore, we have to select the nodes, the edges and the associated weights from the constructive parameters of the compact sets X_i (vertices and faces). For illustration, we consider the case $d = 2$. The resulting graph is $\Gamma_1(\mathcal{N}_1, \mathcal{E}_1, f_1)$, where the nodes are the vertices of the polyhedral regions X_i: $\mathcal{N}_1 = \bigcup_{i=1}^{N_o} \mathcal{V}(X_i)$; the edges are the facets of the partition regions X_i (i.e., their support hyperplanes): $\mathcal{E}_1 = \mathcal{F}^1(X_i)$; and the function f_1 gives the Euclidean distance between the incident nodes of the edge.

Finding the shortest path through the graph between the start and final points $x_i, x_f \in \mathcal{X} \setminus \mathbb{P}$ means adding them to the graph and finding the closest edge such that the connection to it does not intersect any obstacles:

$$(j^\perp, x^\perp)(x_i) = \arg \min_{j \in \mathcal{I}_{f_i}} \min_{x \in \text{facet}_j(X_i)} \|x - x_i\| \tag{9a}$$

$$\text{s.t.} \quad \alpha x + (1-\alpha) x_i \notin P_i, \quad x_i \in X_i, \tag{9b}$$

where $\mathcal{I}_{f_i}$ is the number of facets of X_i. These projections of the points are also new nodes of the graph. Thus, the graph $\tilde{\Gamma}(x_i, x_f)$ preserves the properties of Γ. Finally, a graph search algorithm (e.g., Dijkstra's Algorithm Karaman and Frazzoli (2011)) is employed and the shortest path between x_i and x_f is derived.

Remark 5. For further use, we denote the shortest path through the graph between x_i and x_f as $\text{Path}(x_0, x_f) = (\bar{x}_0 = x_i, \bar{x}_1, \ldots, \bar{x}_n, \bar{x}_{n+1} = x_f)$. This represents an ordered set of points where no segment defined by a pair of consecutive points cuts any of the obstacles. Altough it is not a path in the sense stated in problem P3), it represents a sufficient condition for the existence of a corridor (5).

Having the geometric path given by $\text{Path}(x_0, x_f)$, we can determine the values corresponding to $\gamma(\theta)$. Thus, the remaining part is to select the width of the corridor, namely $\rho(\theta)$. To ensure that $\Pi \cap \mathbb{P} = \emptyset$, $\rho(\theta)$ has to verify the inequality:

$$\rho(\theta) \leq \min_{P_i \in \mathbb{P}} d_H(P_i, \gamma(\theta)), \ \forall \theta \in [0, 1]; \tag{10}$$

Remark 6. The corridor Π can be written as $\Pi = \bigcup_{i=1}^{N_c} \Pi_i$ with $\Pi_i = \{x \in \mathbb{R}^d : \exists \tilde{\theta} \in [0,1] \quad \text{s.t. } x \in \gamma_i(\tilde{\theta}) \oplus \mathbb{B}_{0,\rho_i(\tilde{\theta})}\}$ where $\gamma_i(0) = \bar{x}_{i-1}$ and $\gamma_i(1) = \bar{x}_i$. ♦

For illustrating the construction of the corridor, we revisit the obstacle collection shown in Fig. 1a. Hence, we provide an approximation of the corridor width ρ (red area in Fig. 2 is the corridor defined in (5)). To compute the corridor width we sampled the continuous parameter θ and used the corresponding values in (10).

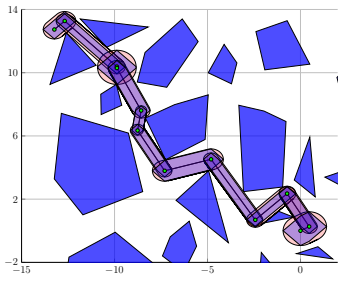


Fig. 2. $\text{Path}(x_i, x_f)$ and $\Pi = \bigcup_{i=1}^{N_c} \Pi_i$

Remark 7. Since we deal with a polytopic representation of the environment, for further use we consider the polytopic underapproximation of the corridors (blue area in Fig. 2). The construction of the corridor is based only on the topological characteristics, but the method proposed here is not unique. For example, instead of using Hausdorff distance, method in Liu et al. (2017) relies on finding the maximal ellipsoid including a segment from a given path and not intersecting the obstacles.

3. CORRIDOR-CONSTRAINED MPC

This section introduces the formulation of a MPC controller and its proof of recursive feasibility. The strategy proposed in the sequel exploits the existence of feasible corridors resulted from the partitioning of the environment. First we recall some concepts and definitions and next we present the formulation associated to a single compact corridor (a segment from (5)). Then the extension to the entire corridor (5) is tackled.

3.1 Prototype MPC with recursive feasibility guarantees

In what follows, our aim is to obtain a collision-free trajectory inside a corridor Π_i for an agent described by the following LTI dynamics:

$$x_{k+1} = Ax_k + Bu_k, \ \forall k \tag{11}$$

with $x_k \in \mathbb{R}^d$ the state vector, $u_k \in \mathbb{R}^{d_u}$ the input vector and the matrices A, B of appropriate dimension. Also, the physical constraints lead to the compact sets $\mathcal{X}$ and $\mathcal{U}$ from $\mathbb{R}^d$ and $\mathbb{R}^{d_u}$, respectively.

The constraints corresponding to the corridor $\Pi_i \subset \mathbb{R}^{d_1}$ are usually only on a subset of the state variables $\mathcal{X} \subset \mathbb{R}^d$ (without loss of generality considered next the corridor to be the first d_1 subcomponents). We use the corridors in the original state to be described by the set:

$$\tilde{\Pi}_i = \{x \in \mathcal{X} \mid \begin{bmatrix} I_{d_1} & 0_{d-d_1+1} \end{bmatrix} x \in \Pi_i\}. \tag{12}$$

For the MPC problem we consider a standard formulation with a quadratic cost:

$$\mathcal{J}(N_p, \bar{x}_i) = \Bigg(\overbrace{\|x_{k+N_p|k} - \bar{x}_i\|^2_{\mathbf{P}}}^{\mathbf{V}_f(x_k, u_k)} + \underbrace{\sum_{l=1}^{N_p-1} \|x_{k+l|k} - \bar{x}_i\|^2_{\mathbf{Q}} + \sum_{l=1}^{N_p-1} \|\Delta u_{k+l|k}\|^2_{\mathbf{R}}}_{\mathbf{V}(x_k, u_k)} \Bigg) \tag{13}$$

where N_p is the prediction horizon, the weight matrices $\mathbf{P}$ (terminal cost penalty), $\mathbf{Q}$ (output error penalty) and $\mathbf{R}$ (control action penalty) are positive semi-definite and of appropriate dimensions. For further implications, the value of $\mathbf{P}$ from the terminal cost $\mathbf{V}_f(x_k, u_k)$ is selected such that the function $V(x) = x^\top Px$ is a Lyapunov function for a pre-stabilizing state-feedback law $u = Kx$ with $K \in \mathbb{R}^{d\times d_u}$ in a vicinity of $\bar{x}_i$ (Mayne et al., 2000). As well, the values of $\mathbf{Q}$ and $\mathbf{R}$ from the cost per stage $\mathbf{V}(x_k, u_k)$ are chosen in order to enhance the tracking performances.

Therefore, the MPC problem to be solved at each time step throughout a corridor $\tilde{\Pi}_i$ can be formulated as:

$$\mathcal{P}(\tilde{\Pi}_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X}) : \min_{\mathbf{u}} \ \mathcal{J}(N_p, \bar{x}_i) \tag{14a}$$

$$\text{s.t. } x_{k+l+1|k} = Ax_{k+l|k} + Bu_{k+l|k}, \tag{14b}$$

$$u_{k+l|k} \in \mathcal{U}, \ \forall \ell = 1 : N_p - 1 \tag{14c}$$

$$x_{k+l|k} \in \tilde{\Pi}_i \tag{14d}$$

$$x_{k+N_p|k} \in \mathcal{X}_f, \tag{14e}$$

Having the formulation (14), the question to be addressed is *how to select the parameters* $\tilde{\Pi}_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X}$ *in order to ensure the recursive feasibility*. The recursive feasibility is one of the fundamental challenges in MPC literature. Basically, it represents the property that for all initial feasible states and for all optimal sequences of control inputs the MPC optimization problem remains feasible Mayne et al. (2000). Löfberg (2012) offers a broad overview on this topics, especially from the computational perspective.

Essentially, the selection of the parameters in (14) can be viewed as an additional analysis step, which builds on the, so-called, backward reachable set (BRS).

Definition 8. (N-step BRS). The N-step (BRS) is the set of all states that can reach a final position $\bar{x}_i$ in N steps-associated to the system described by LTI dynamics (11):

$$R_N^i = A^{-N}\left(\bar{x}_i \oplus \bigoplus_{j=0}^{N-1} -A^j BU\right). \tag{15}$$

Since we have "hard" constraints on state (the ones given by the corridor), we have to compute the N-step BRSs taking into account these constraints: $\tilde{R}_N^i$. In fact, the N-step BRSs remain the same as long as they check the corridor constraints. From that point, the computation can be done iteratively as in Algorithm 1.

Algorithm 1 Computing N-step BRS for (11) taking into account the state constraints $\tilde{\Pi}_i$

1: Find first $R_{N_{uc}}^i$ such that $R_{N_{uc}}^i \not\subset \tilde{\Pi}_i$
2: For $N < N_{uc}$: $\tilde{R}_N^i = R_N^i$
3: $\tilde{R}_{N_{uc}}^i = R_{N_{uc}}^i \cap \tilde{\Pi}_i$
4: For $N > N_{uc}$ the computation is iterative and relies on: $\tilde{R}_N^i = A^{-1}\left(\tilde{R}_{N-1}^i \oplus (-BU) \cap \tilde{\Pi}_i\right)$.

The N-step BRSs have an instrumental role in providing the parameters of the MPC problem (14). For instance, the theoretical minimal value of N_p is given by the condition $x_0 \in R_{N_p}^i$.

Proposition 9. $\mathcal{P}(\Pi_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X})$ is feasible for all feasible initial states if

$$N_p \geq \arg\min_N \ N \ \text{ s.t. } \ x_0 \in \tilde{R}_N^i. \tag{16}$$

Proof. It is straightforward that $x_0 \in \tilde{R}_{N_p}^i$. That is, there exists at least one sequence of N_p inputs so that the predicted final state is exactly the reference $\bar{x}_i$. As well, the existence of a terminal cost as in (13) ensures the convergence. Moreover, since $\bar{x}_i \in \mathcal{X}_f$, it is obvious that (14e) holds. Therefore, the feasibility of $\mathcal{P}(\Pi_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X})$ is guaranteed. ■

As a consequence of Proposition 9, the recursive feasibility of a MPC controller based on $\mathcal{P}(\cdot)$ follows straighforwardly.

3.2 Generic collision-free trajectory generation within cluttered environments

Since the recursive feasibility of an MPC strategy based on (14) is guaranteed for certain parameters, we aim to exploit these degrees of freedom for the resolution of the problem P3). To this end, an intuitive solution is to iteratively compute the parameters ensuring feasibility for each corridor segment. Specifically, for each initial position ($\gamma_i(0)$) within the corridor segment, we compute the N-step BRS centered in x_f^i. The computational effort corresponding to this may be substantial relative to a consequent real-time implementation. To overcome this drawback, we can split the strategy since we are aware of the corridor characteristics.

(1) (OFF-LINE) For each $\tilde{\Pi}_i$ and x_f^i
- compute the BRSs (as inAlgorithm 1) and $N_{p_{min}}^i$:

$$N_{p_{min}}^i = \arg\min_N \ N \ \text{ s.t. } \ \tilde{\Pi}_i \subseteq \tilde{R}_N^i, \tag{17}$$

- $\mathcal{X}_f^i = (\tilde{\Pi}_i \cap \tilde{\Pi}_{i+1})$ excepting the last segment, for which we have $\mathcal{X}_f^i = \tilde{R}_1^i$

(2) (ON-LINE) Apply Algorithm 2

As its name suggests, the idea behind the "Relay MPC" strategy is to ensure the transitions from a segment of corridor ($\tilde{\Pi}_i$) to the next ($\tilde{\Pi}_{i+1}$). For that reason, we choose the terminal sets $\mathcal{X}_{f_i}$ as the intersections between two consecutive segments.

Algorithm 2 "Relay MPC"

Input: $\Pi = \bigcup_{i=1}^{N_c} \Pi_i$, x_0, $N_{p_{min}}^i$, $\mathcal{X}_f^i$
1: **for each** $\tilde{\Pi}_i \subset \tilde{\Pi}$ **do**
2: $N_p = N_{p_{min}}^i$;
3: **repeat**
4: Apply MPC strategy solving $\mathcal{P}(\tilde{\Pi}_i, N_p, \mathcal{X}_f^i, \bar{x}_i, \mathcal{X})$
5: **until** $x_{k+1|k} \in \mathcal{X}_f^i$
6: $x_0 = x_{k+1|k}$
7: Update the parameters of $\mathcal{P}$
8: **end for**

Proposition 10. If $\text{Path}(x_0, x_f)$ exists and the control law based on $\mathcal{P}(\cdot)$ (14) is recursively feasible then the convergence $x_0 \to x_f$ is guaranteed.

Proof. The existence of Path(x_0, x_f) leads to the construction of the corridor $\tilde{\Pi} = \bigcup_i \tilde{\Pi}_i$. The recursive feasibility and the selection of terminal sets as in (1) ensure the transition from $\tilde{\Pi}_i$ to $\tilde{\Pi}_{i+1}$. Moreover, for the last segment we have $\mathcal{X}_f^i = \tilde{R}_1^i$ and the initial state in $\tilde{R}_{N_p}$ which directly leads to x_f being reachable, and, by consequence, it proves the convergence of the scheme. ■

For illustration purposes, consider an agent described by the dynamics (11) in $\mathbb{R}^4$ with:

$$A = \begin{bmatrix} O_2 & I_2 \\ O_2 & -\frac{\mu}{M} I_2 \end{bmatrix}, \quad B = \begin{bmatrix} O_2 \\ M^{-1} I_2 \end{bmatrix}, \tag{18}$$

where $\mu = 3$ and $M = 60$. The agent's state is composed of position and velocity components $x = [p_x \ p_y \ v_x \ v_y]^\top$, whereas the input is given by the accelerations $u = [a_x \ a_y]^\top$. Both state and input are constrained to :

$$\mathcal{X} = \{x \in \mathbb{R}^4 : |x| \leq [15 \ 15 \ 0.35 \ 0.35]^\top\}$$
$$\mathcal{U} = \{u \in \mathbb{R}^2 : |u| \leq [10 \ 10]^\top\}$$

Using dynamics (18) we revisit the obstacle collection depicted in Fig. 1a to which we apply Algorithm 2. As well, in Fig. 4 the values of the acceleration and velocity are plotted along the simulation horizon.

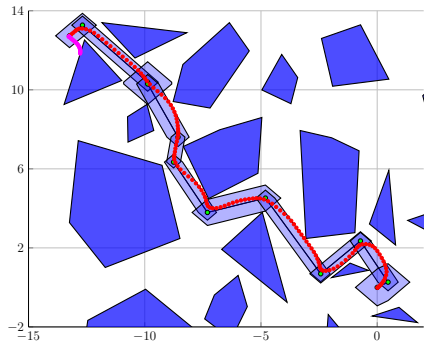


Fig. 3. Algorithm 2 over the example in Fig. 1a

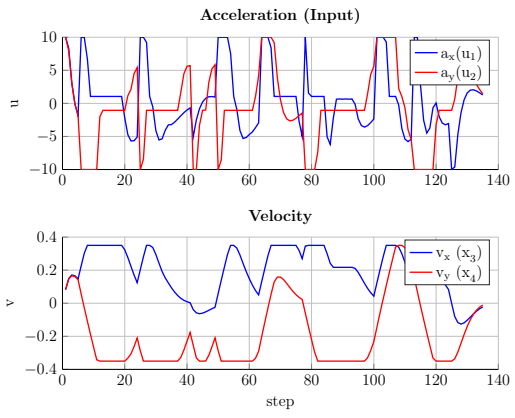


Fig. 4. Input and velocity during Algorithm 2

As a side remark, for the same initial and final position an MIP based MPC strategy can be employed, considering the entire set of obstacles and imposing a large prediction horizon ($N_p = 40$). Despite the large prediction horizon, the resulting trajectory does not converge to the final position, the agent remaining on the boundary of one of the obstacles (the pink trajectory in Fig. 3).

Regarding the offline part, as stated above, the computational effort is substantial, e.g., for the trajectory in Fig. 3 is around 3 minutes, but this can be improved by replacing the polyhedral representation of the sets with a zonotopic one (Althoff, 2015). Intuitively, there is a connection between the length of a corridor segment ($\Delta\gamma_i = \|\gamma_i(1) - \gamma_i(0)\|$) and the length of the minimal prediction horizon($N_{p_{min}}$). Therefore, we depict in Table 1 the values corresponding to (17).

i	1	2	3	5	8	10	11
$\|\Delta\gamma_i\|$	0.77	4.02	0.09	1.27	4.50	2.39	0.53
$N_{p_{min}}$	11	24	13	14	29	19	13

Table 1.

Remark 11. In practice, whenever the length of a segment of the corridor is greater than a user-defined value, that segment can be split. In this way, one can manage the trajectory tracking with similar length of the prediction horizons in concordance with the available computational constraints.

To emphasize the importance of the corridor and how it reflects in a real-world application, we can assume that the dynamics (18) are the nominal ones and any disturbance impacts, in fact, the dimension of the corridor. That is, the function $\rho(\cdot)$, as in (5), becomes $\tilde{\rho}(\cdot) = \rho(\cdot) + \Delta\rho$. In Table 2 we delineate: N_{goal} - the number of steps to attain a neighborhood of the final point, t_{goal} - the total time to compute the trajectory and ℓ_t - the trajectory length. As it can be seen in Table 2, the computing time t_{goal} has similar values (slight modifications), while N_{goal} and ℓ_t increase with the values of $\Delta\rho$. This behavior is counter-intuitive, but it can be explained by the fact that the decreasing in terms of steps (or distance) gained along of the corridor are wasted with the maneuvers associated to changing the segments of the corridor. However, by moving all complex operations from on-line to off-line, we note that the computing time is kept reasonable, allowing a comparison with the classical sampled-based methods.

$\Delta\rho$	t_{goal}(sec)	N_{goal}	ℓ_t(m)
0	1,200	135	23.920
3e-3	1,219	138	24,068
7e-3	1,227	139	24,095
1e-2	1,231	139	24,089
5e-2	1.254	141	24,066
7e-2	1.277	142	24,059

Table 2.

3.3 Insights on the extensions to multi-agent system

Up to now we consider only one agent within the static multi-obstacle environment. There exist however applications which can be arguable better performed by a team of agents. Thus, a second strong requirement is the avoidance of collisions among the agents. As stated in Section 1, the method herein has two phases: planning and tracking. Clearly, we can eliminate the risk of collision from the planning phase by selecting paths with disjoint sets of nodes (and, implicitly, of edges). Nevertheless, this may lead to trajectories too far (in terms of performance or consumed energy) from the optimal ones. Hence, in the planning phase we can only impose that the paths of the two agents share a minimal number of common nodes.

Since the graph is constructed relying on the partitioning from Corollary 4, it is straightforward that the degree (or valency) [3] of the nodes is 3, excepting the nodes from the boundaries of $\mathbb{X}$. Thus, the minimal number of common nodes between the paths of the agents is two and, implicitly, the paths share a common edge. Consequently, in the tracking phase we have to treat the behavior at the proximity of these/this common nodes/edge. Concretely, we adjust Algorithm 2, specifically an adaptation of the parameters involved in Step 4. Since the corridor constraints affect only a subset of the state variables, we may affect the behavior of the agent in the proximity of conflict zones by changing the bounds on the remaining subset of variables. Thus, we modify the parameter $\mathcal{X}$ of the MPC strategy from Algorithm 2. A prospective solution is to tune the bounds $\mathcal{X}$ according to a reachability analysis. That is, by considering the reachable sets mapped on the velocity subspace, we can find the set of velocities that would lead to collision with a moving entity. Enforcing velocities apart from this set ensures that the agents are able to attain a slowdown before the potential conflict zones.

4. CONCLUSIONS

The paper presents a constructive solution for the generation of collision-free trajectories between two points in an environment containing multiple obstacles in a d-dimensional space. We use the geometry of the obstacles and the convex lifting procedure to describe a graph around the obstacles. This graph represents a key element in order to generate collision-free trajectories employing MPC controllers with recursive feasibility guarantees. As a perspective of this work we will extend the results for time-varying multi-obstacle environments, considering, as well, re-configuration and adjustments of the graph.

ACKNOWLEDGEMENTS

The research of Daniel Ioan is financially supported by the Ministry of the Armed Forces - Defence Procurement Agency (DGA) - no.2017352. The authors acknowledge Jacques Blanc-Talon, expert DGA, for the fruitful discussions.

REFERENCES

Althoff, M. (2015). Cora 2015 manual. *TU Munich*, 85748.

Ballesteros-Tolosana, I., Olaru, S., Rodriguez-Ayerbe, P., Pita-Gil, G., and Deborne, R. (2017). Collision-free trajectory planning for overtaking on highways. In *2017 Conference on Decision and Control (CDC)*, 2551–2556.

Barraquand, J., Kavraki, L., Latombe, J.C., Motwani, R., Li, T.Y., and Raghavan, P. (1997). A random sampling scheme for path planning. *The International Journal of Robotics Research*, 16(6), 759–774.

Berntorp, K., Danielson, C., Weiss, A., and Di Cairano, S. (2018). Positive invariant sets for safe integrated vehicle motion planning and control. In *2018 IEEE Conference on Decision and Control (CDC)*, 6957–6962.

Berntorp, K., Weiss, A., Danielson, C., Kolmanovsky, I.V., and Di Cairano, S. (2017). Automated driving: safe motion planning using positively invariant sets. In *20th International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. IEEE.

Chen, Y.b., Luo, G.c., Mei, Y.s., Yu, J.q., and Su, X.l. (2016). UAV path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science*, 47(6), 1407–1420.

Faulwasser, T. and Findeisen, R. (2016). Nonlinear model predictive control for constrained output path following. *IEEE Transactions on Automatic Control*, 61(4).

Franzè, G. and Lucia, W. (2015). The obstacle avoidance motion planning problem for autonomous vehicles: A low-demanding receding horizon control scheme. *Systems & Control Letters*, 77, 1–10.

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

Ioan, D., Olaru, 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.

Janecek, F., Klauco, M., Kaluz, M., and Kvasnica, M. (2017). OPTIPLAN: A Matlab Toolbox for Model Predictive Control with Obstacle Avoidance. *IFAC-PapersOnLine*, 50(1), 531–536.

Jawad, H., Nordin, R., Gharghan, S., Jawad, A., and Ismail, M. (2017). Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors*, 17(8).

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894.

Latombe, J.C. (2012). *Robot motion planning*, volume 124. Springer Science & Business Media.

LaValle, S.M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report.

Liu, S., Watterson, M., Mohta, K., Sun, K., Bhattacharya, S., Taylor, C., and Kumar, V. (2017). Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robotics and Automation Letters*, 2(3), 1688–1695.

Löfberg, J. (2012). Oops! i cannot do it again: Testing for recursive feasibility in mpc. *Automatica*, 48(3), 550–555.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Nguyen, N.A., Gulan, M., Olaru, S., and Rodriguez-Ayerbe, P. (2018). Convex lifting: Theory and control applications. *IEEE Transactions on Automatic Control*, 63(5), 1243–1258.

Prodan, I., Stoican, F., Olaru, S., and Niculescu, S.I. (2015). Mixed-integer representations in control design: Mathematical foundations and applications. Springer.

Puri, A. (2005). A survey of unmanned aerial vehicles uav for traffic surveillance. *Department of computer science and engineering, Univ. of South Florida*, 1–29.

Richards, A. and How, J.P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the 2002 American Control Conference.*, volume 3, 1936–1941. IEEE.

Szmuk, M., Pascucci, C.A., Dueri, D., and Acikmese, B. (2017). Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance. In *2017 International Conference on Intelligent Robots and Systems (IROS)*.

Weiss, A., Danielson, C., Berntorp, K., Kolmanovsky, I., and Cairano, S.D. (2017). Motion planning with invariant set trees. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 1625–1630.

[3] The number of edges that are incident to that node.