



HAL
open science

Multi-agent collision avoidance based on buffered Voronoi cells and B-splines

Cong Khanh Dinh, Ionela Prodan, Florin Stoican

► **To cite this version:**

Cong Khanh Dinh, Ionela Prodan, Florin Stoican. Multi-agent collision avoidance based on buffered Voronoi cells and B-splines. APCA International Conference on Automatic Control and Soft Computing, Jul 2024, Porto, Portugal. hal-04684077

HAL Id: hal-04684077

<https://hal.science/hal-04684077v1>

Submitted on 2 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.

Multi-agent collision avoidance based on buffered Voronoi cells and B-splines ^{*}

Cong Khanh Dinh¹, Ionela Prodan¹, and Florin Stoican²

¹ Univ. Grenoble Alpes, Grenoble INP[†], LCIS, F-26000, Valence, France,

[†] Institute of Engineering and Management

{`cong-khanh.dinh`, `ionela.prodan`}@`lcis.grenoble-inp.fr`

² Faculty of Automation Control and Computer Science, University Politehnica of Bucharest, Romania,
`florin.stoican@upb.ro`

Abstract. This paper tackles the challenging issue of collision avoidance among multiple dynamical agents, a problem known for its non-convex nature and computational complexity. Our proposed solution involves decomposing the space into time-varying Voronoi cells with buffer zones, exploiting the versatility of B-splines to reconfigure trajectories when collisions are predicted. This approach is integrated into a Model Predictive Control (MPC) framework, implemented in a distributed manner where agents communicate predicted control point sequences defining the B-spline curves. Through simulation testing across various scenarios, our approach demonstrates significant promise, paving the way for experimental validation and real-world applications.

Keywords: Multi-agent dynamical systems, Collision avoidance, B-splines parametrization, Trajectories prediction, Distributed Model Predictive Control (MPC).

1 Introduction

Motivation and exiting work: Collision avoidance is crucial in scenarios involving multiple agents, such as autonomous vehicles, drones, or robotic systems navigating shared environments [4]. By incorporating collision avoidance into the control systems of each agent, it becomes possible to anticipate potential collisions and proactively prevent them. These preventive measures may involve adjusting velocity or reconfiguring trajectories to avoid imminent collisions [12]. However, collision avoidance poses significant challenges due to its non-convex nature. Unlike convex optimization problems with clear and efficient solutions,

^{*} The work of the first two authors is supported by La Région Auvergne-Rhône-Alpes, Pack Ambition Recherche 2021 - PlanMAV, RECLAMALCIR and Ambition Internationale 2023, Horizon-TA C7H-REG24A10, France. The third author is supported by a grant from the National Program for Research of the National Association of Technical Universities - GNAC ARUT 2023; Project ID: 207, UNSTPB, Romania.

non-convex problems lack globally optimal solutions and can be computationally demanding to solve [9]. Furthermore, the presence of multiple agents adds complexity, requiring coordinated movements to ensure collision-free paths while still achieving individual objectives [10].

Soft constraints offer a promising approach to address the non-convexity of optimization problems in collision avoidance scenarios, thereby reducing the computational burden to a more tractable level [5]. Instead of strictly adhering to avoidance maneuvers, agents may deviate from their planned trajectories within predefined bounds, accepting a certain level of collision risk if necessary. By incorporating penalty terms or cost functions associated with collision risk, the optimization problem becomes convex or semi-convex, enabling more efficient solution methods [1].

Contributions: Expanding upon the concept of control point space mapping, our approach utilizes the inherent properties of B-splines, as discussed in previous work [2], in conjunction with distributed Model Predictive Control (MPC) techniques to effectively address collision and constraint handling in multi-agent systems. B-splines offer a versatile mathematical framework for representing and manipulating trajectories [6]. By exploiting the control point space mapping, we can further benefit from the flexibility and adaptability of B-splines to dynamically adjust trajectory configurations in response to potential collision threats. In parallel, the distributed MPC algorithm enables agents to collaboratively plan and execute their trajectories while considering both individual objectives and shared constraints. The goal is to facilitate real-time coordination among agents, allowing them to efficiently navigate through shared spaces while avoiding collisions and satisfying the imposed constraints. Briefly, in this paper, we propose the following contributions:

- i) Utilizing the convexity and locality properties of B-splines, we introduce a method for enforcing collision avoidance through a time-varying Voronoi partitioning approach.
- ii) We integrate these collision avoidance constraints into a distributed MPC framework. Within this framework, agents communicate the predicted sequence of control points, which subsequently dictate their trajectories.

These theoretical contributions are validated across a range of simulated scenarios, laying the groundwork for future experimental investigations.

Notation: Vectors are represented by bold letters. Capital letters in bold represent matrices. \mathbf{I}_n and $\mathbf{0}_n$ are identity and zero matrix with dimension $n \times n$. $\|x\|_{\mathbf{Q}} \triangleq \sqrt{x^T \mathbf{Q} x}$. Otherwise, if not specified in the subscript, $\|\bullet\|$ represents the Euclidean norm. With notation $(\hat{\bullet})[k|k_t]$ denoting the predicted value of (\bullet) at time step $k + k_t$ from information known at time k_t , an agent's prediction model used inside the MPC problem is $\hat{\mathbf{x}}_i[k + 1|k_t] = \mathbf{A}\hat{\mathbf{x}}_i[k|k_t] + \mathbf{B}\hat{\mathbf{u}}_i[k|k_t]$, where $\hat{\mathbf{x}}_i[0|k_t] \leftarrow \mathbf{x}_i[k_t]$ denotes the vector of measured states.

2 Preliminaries

Agents dynamics: Although the approach described in the subsequent sections is applicable to arbitrary dynamics, for the sake of illustration, we will use the dynamics of a quadcopter. This model entails significant nonlinearities, which are mitigated through a flatness-based linearization method. Following discretization, we obtain a decoupled stacking of three double integrators:

$$\mathbf{x}_i[k+1] = \mathbf{A}\mathbf{x}_i[k] + \mathbf{B}\mathbf{u}_i[k], \quad (1)$$

with the matrices $\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & h\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} h^2\mathbf{I}_3/2 \\ h\mathbf{I}_3 \end{bmatrix}$ and the discretization step, h . The i -th agent has the state, $\mathbf{x}_i[k]$, defined by the position and velocity, $\mathbf{x}_i[k] = (\mathbf{p}_i^\top[k], \mathbf{v}_i^\top[k])^\top \in \mathbb{R}^6$, and the input, $\mathbf{u}_i[k] = \mathbf{a}_i[k] \in \mathbb{R}^3$, defined by the acceleration. If the agents are heterogeneous (i.e., they have different nonlinear dynamics), we can derive their linear models using either feedback linearization or Taylor approximation [3].

B-splines: Certainly, B-splines possess several advantageous properties that make them appealing for trajectory generation and optimization: i) continuity and locality ensure smooth curve generation and rapid reconfiguration [11]; ii) the convexity and endpoint interpolation play significant roles in constraint validation.

A family of B-spline basis functions $\{B_{i,p}(t)\}$ with $i \in \{0, \dots, N\}$ characterized by cardinality N , order p and knot-vector $\tau = \{\tau_1, \dots, \tau_m\}$ weighted by control points $\{P_i\}$ gives the curve

$$\mathbf{z}(t) = \sum_{i=0}^N P_i B_{i,p}(t) = \mathbf{P}\mathbf{B}_p(t), \quad \forall t \in [\tau_1, \tau_m]. \quad (2)$$

In equation (2), the column vector $\mathbf{B}_p(t)$ and control matrix \mathbf{P} come from stacking the B-spline basis functions $B_i(t)$ vertically and the control points P_i as columns, respectively.

Given the dynamics described by (1), where $\mathbf{z}(t) \in \mathbb{R}^{3 \times 1}$ represents the reference position to be tracked by the position component of the state, denoted as $Cx_i[k] = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} x_i[k]$.

Without delving into the specifics (refer to [8] for comprehensive details), it is noteworthy that the derivatives of (2) can be represented as B-spline curves of reduced order (specifically, order $p-r$ for the r -th order derivative of $\mathbf{z}^{(r)}(t)$), weighted by new control points linearly dependent on the original ones. For instance, we can express:

$$\mathbf{z}^{(2)}(t) = \mathbf{P}^{(2)}\mathbf{B}_{p-2}(t), \quad (3)$$

where $\mathbf{P}^{(2)} := \mathbf{P}\mathbf{M}_2$ and \mathbf{M}_2 is a matrix suitably computed (see [7]).

Cell partitioning: Partitioning of the environment by Voronoi is the foundation of several control techniques and path finding algorithms in robotics. In order to create a map to generate a trajectory for non-collision, it is often utilized to establish a zone that is near to an agent and separating from other agents. One fundamental baseline constraint that guarantees maintaining a minimum safety gap r_{min} between any two agents (i, j) is $\|\mathbf{p}_i - \mathbf{p}_j\|_2 \geq r_{min}$. In the formation of Voronoi diagram for a multi-agent system consisting of N_a agents, the Voronoi cell of the i -th agent is commonly defined using the distance metric:

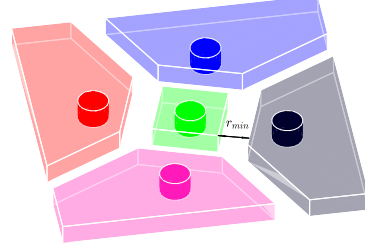


Fig. 1: Buffered Voronoi Cell partitioning in 3D with r_{min} the safety gap.

$$\mathcal{V}_i = \{\mathbf{p} \in \mathbb{R}^3 \mid \|\mathbf{p} - \mathbf{p}_i\| \leq \|\mathbf{p} - \mathbf{p}_j\|, \forall j \neq i\}. \quad (4)$$

To integrate the safety gap r_{min} , the notion of Buffered Voronoi cell is thus recalled as in [13][5]

Definition 1. Let N_a be number of agents. An i -th agent at position $\mathbf{p}_i \in \mathbb{R}^3$ can be associated with a Buffered Voronoi Cell (BVC) $\bar{\mathcal{V}}_i$ with a safety distance r_{min} defined as follow

$$\bar{\mathcal{V}}_i = \left\{ \mathbf{p} \in \mathbb{R}^3 \mid \left(\mathbf{p} - \frac{\mathbf{p}_i + \mathbf{p}_j}{2} \right)^\top (\mathbf{p}_j - \mathbf{p}_i) + \frac{r_{min}}{2} \|\mathbf{p}_j - \mathbf{p}_i\| \leq 0, \forall j \neq i \right\}. \quad (5)$$

Additionally, a sizing matrix Θ can be employed to generalize the distance so that the safety region of an agent can be viewed as an ellipsoid : $\xi_i = \{\mathbf{x} + \mathbf{p}_i \mid \mathbf{x}^\top \Theta^{-2} \mathbf{x} \leq r_{min}^2\}$

$$\bar{\mathcal{V}}_i = \left\{ \mathbf{p} \in \mathbb{R}^3 \mid \frac{(\mathbf{p}_i - \mathbf{p}_j)^\top \Theta^{-2} (\mathbf{p} - \mathbf{p}_i)}{\|\mathbf{d}_{ij}\|} \geq \frac{r_{min} - \|\mathbf{d}_{ij}\|}{2}, \forall j \neq i \right\}, \quad (6)$$

where $\mathbf{d}_{ij} = \Theta^{-1}(\mathbf{p}_j - \mathbf{p}_i)$

This BVC is thus equivalent to a combination of linear constraints as mentioned in our prior work [2]:

$$\bar{\mathcal{V}}_i = \{\mathbf{p} \in \mathbb{R}^3 \mid A_{ij} \mathbf{p} \leq b_{ij}, \forall j \neq i\}, \quad (7)$$

where $A_{ij} = -(\mathbf{p}_i - \mathbf{p}_j)^\top$ and $b_{ij} = A_{ij} \mathbf{p}_i - \frac{1}{2}(r_{min} - \|\mathbf{p}_i - \mathbf{p}_j\|)\|\mathbf{p}_i - \mathbf{p}_j\|$.

3 Spline-based distributed MPC

Optimization problem: This section refers back to the distributed control framework based on B-spline from our previous work [2], in which all agents resolve the trajectory planning problem in parallel following the receding horizon strategy. The collision constraint is then integrated with the reduced formulation for B-spline. Let consider the agent i -th where the optimization over N_p steps on the horizon is built up by summing certain cost functions which are subject to the constraints of the system.

Tracking error cost: We want to minimize the deviation from the desired trajectory or target over the prediction horizon. This is defined using a quadratic cost following:

$$\mathcal{L}_{tracking}^i = \sum_{s=0}^{N_p-1} \|\hat{\mathbf{p}}_i[s|k] - \mathbf{p}_i^f\|_{\mathbf{Q}[s]} + \|\hat{\mathbf{p}}_i[N_p|k] - \mathbf{p}_i^f\|_{\mathbf{Q}_p}, \quad (8)$$

where \mathbf{p}_i^f denotes the reference tracking position for the agent, $\hat{\mathbf{p}}_i[s|k] \in \mathbb{R}^3$ is the predicted position at s time steps given the information at k instant. $\mathbf{Q}[s]$ and $\mathbf{Q}_p \geq 0$ are matrices representing the weight at each step of the trajectory. The final term in the cost functions helps adding stability at the terminal position in the tracking problem. With parameterization of position using the B-spline as mentioned in [2]

$$\hat{\mathbf{p}}_i[s|k] = \mathbf{P}_i \mathbf{B}_p(t_s), \quad (9)$$

where $t_s = sh$ and h is the sampling time of the MPC problem. The cost function (8) can be reformulated using the B-spline as following:

$$\mathcal{L}_{tracking}^i = \mathcal{L}_{tracking}^i(\mathbf{P}_i) = \frac{1}{2} \mathbf{P}_i^\top \mathbf{H}_Q \mathbf{P}_i - \mathbf{f}_Q^\top \mathbf{P}_i, \quad (10)$$

which is in quadratic form with control points \mathbf{P}_i serving as decision variables.

Energy cost: To minimize the control energy spent along the trajectory, the cost defined by summation of squared input sequences is presented as

$$\mathcal{L}_{energy}^i = \sum_{s=0}^{N_p-1} \|\hat{\mathbf{u}}_i[s|k]\|_{\mathbf{R}}, \quad (11)$$

where $\mathbf{R} > 0$ denotes the weighted matrix for control action which is acceleration. Similarly, the conversion of (11) above using the derivative formulas for with control points of trajectory

$$\hat{\mathbf{u}}_i[s|k] = \mathbf{P}_i^{(2)} \mathbf{B}_{p-2}(t_s), \quad (12)$$

which leads to a quadratic function for the energy:

$$\mathcal{L}_{energy}^i = \mathcal{L}_{energy}^i(\mathbf{P}_i) = \frac{1}{2} \mathbf{P}_i^\top \mathbf{H}_R \mathbf{P}_i - \mathbf{f}_R^\top \mathbf{P}_i, \quad (13)$$

System dynamics constraints: For multi-agent navigation in a congested environment, the agents are restricted to certain operating conditions which are usually related to the dynamical limits of the agents. These constraints can also benefit from the B-spline parameterization which leads to linear inequality constraints.

$$\mathbf{A}_{dyn}^i \mathbf{P}_i \leq \mathbf{b}_{dyn}^i, \quad (14)$$

where \mathbf{A}_{dyn}^i and \mathbf{b}_{dyn}^i come from a compact form of the following constraints: $u_{min} \leq \hat{\mathbf{u}}_i[s|k] \leq u_{max}$, $v_{min} \leq \hat{\mathbf{v}}_i[s|k] \leq v_{max}$, $p_{min} \leq \hat{\mathbf{p}}_i[s|k] \leq p_{max}$.

Collision avoidance constraints: Recall the locality property of B-spline curves. For any sub-interval $[\tau_s, \tau_{s+1})$, the curve is bounded to lie inside the convex hull generated by the sequence $\{P_{i,s-d+1}, \dots, P_{i,s}\}$ of $d = p + 1$ consecutive control points, i.e.,

$$\hat{\mathbf{p}}_i(t) \in \mathcal{R}_{i,s}, \quad \forall t \in [\tau_s, \tau_{s+1}). \quad (15)$$

For convenience, we have denoted $\mathcal{R}_{i,s}$ as the convex region spanned by the sequence $\{P_{i,s-d+1}, \dots, P_{i,s}\}$.

A sufficient collision test from the viewpoint of the i -th agent is

$$\exists s \text{ s.t. } (\mathcal{R}_{i,s} \oplus \xi_i) \cap (\mathcal{R}_{j,s} \oplus \xi_j) \neq \emptyset, \quad \forall j \neq i. \quad (16)$$

where $\xi_i = \{\mathbf{x} + \mathbf{p}_i \mid \mathbf{x}^\top \Theta^{-2} \mathbf{x} \leq r_{min}^2\}$ is the ellipsoid representing the safety region of the agent with scaling matrix Θ . Making use of (15), we have that

$$\mathcal{R}_{i,s} \subseteq \bigcap_{t \in [\tau_s, \tau_{s+1})} \bar{\mathcal{V}}_i(t) \quad (17)$$

provides a sufficient avoidance condition. It is not easy to impose directly these conditions for real-time implementation due to very high conservativeness of this convex hull. Hence, it is necessary in our work to exploit the positivity properties of B-splines and give extra feasibility by introducing slack variables as in [2], which can be denoted as $\epsilon_i = [\epsilon_{i,1}, \epsilon_{i,2}, \dots, \epsilon_{i,N_a}]^\top \in \mathbb{R}^{N_a}$. Hence, the following collision avoidance constraints consider the control polygon over sub-interval as showed in Fig. 2:

$$\|\Theta^{-1}(P_{i,l-m} - \hat{\mathbf{p}}_j[\ell + 1|k - 1])\| \geq r_{min} + \epsilon_{i,j}, \quad \forall j \neq i, m = 0, \dots, d \quad (18)$$

where ℓ is the closest anticipating step, at which agent detects potential collision with condition: $\|\hat{\mathbf{p}}_i[\ell + 1|k] - \hat{\mathbf{p}}_j[\ell + 2|k - 1]\| < r_{min}$.

The relaxation term $\epsilon_{i,j}$ determines violation level to the distance between the two agents (i, j) .

$$\epsilon_{i,j} = \begin{cases} < 0 & \text{if } j \in \mathcal{N}_i^\ell \\ = 0 & \text{otherwise} \end{cases}$$

To mitigate the violation, we introduce an augmented cost correspondingly

$$\mathcal{L}_r(\epsilon_i) = \sum_{j=1}^{N_a} \eta \|\epsilon_{i,j}\|^2. \quad (19)$$

Equivalently, (18) can be reformulated using BVC as referring in (7)

$$A_{ij}^l P_{i,l-m} + \frac{\|\mathbf{d}_{ij}^\ell\|}{2} \epsilon_{i,j} \leq b_{ij}^l, \forall j \neq i \quad (20)$$

where $A_{ij}^l = -(\Theta^{-2}(\hat{\mathbf{p}}_i[\ell|k] - \hat{\mathbf{p}}_j[\ell+1|k-1]))^\top$,
 $b_{ij}^l = A_{ij}^l \hat{\mathbf{p}}_i[\ell|k] - \frac{(r_{min} - \|\mathbf{d}_{ij}^\ell\|)\|\mathbf{d}_{ij}^\ell\|}{2}$ and $\mathbf{d}_{ij}^\ell = \Theta^{-1}(\hat{\mathbf{p}}_j[\ell+1|k-1] - \hat{\mathbf{p}}_i[\ell|k])$.

Furthermore, (20) is compactly rewritten as:

$$A_i^l P_{i,l-m} + \mathbf{D}_i^l \epsilon_i \leq b_i^l, \quad \forall j \neq i, m = 0, \dots, d \quad (21)$$

where $\mathbf{D}_i^l = \frac{1}{2} \text{diag}([\|\mathbf{d}_{i1}^\ell\|, \|\mathbf{d}_{i2}^\ell\|, \dots, \|\mathbf{d}_{iN_a}^\ell\|])$.

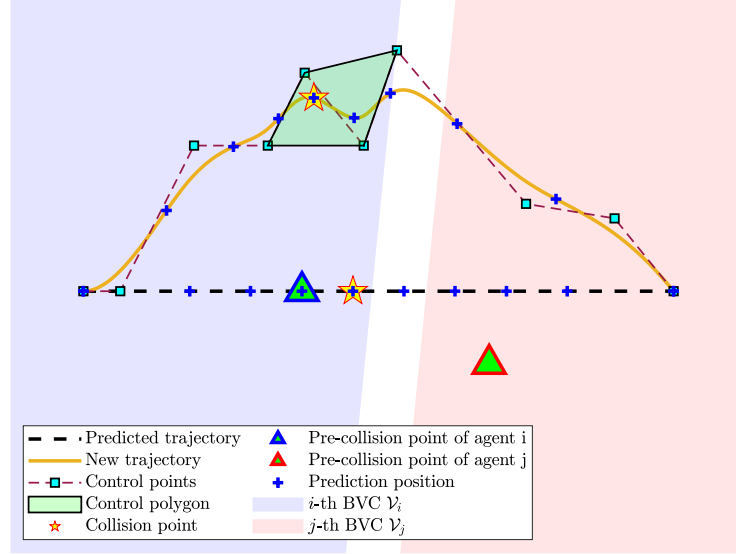


Fig. 2: Generating trajectory for an agent with convexity of B-spline

Gathering all the aforementioned constraints and cost functions, we obtain a standard quadratic programming problem with control points as decision variables as follow:

$$\text{Minimize:} \quad \mathcal{L}^i(\mathbf{P}_i, \epsilon_i) = \mathcal{L}_{tracking}^i + \mathcal{L}_{energy}^i + \mathcal{L}_r(\epsilon_i) \quad (22)$$

$$\text{Subject to:} \quad (14), (21) \quad (23)$$

4 Simulation scenarios and performance analysis

This section presents some useful simulation scenarios for testing the distributed planning algorithm proposed above. For the implementation of the optimal control problems (OCP) in (22), we use Matlab language with the open-source

nonlinear optimization tool CasADi to resolve the quadratic programming. Let us consider a system of aerial robots performing target tracking mission in a bounded environment with dimension $p_{min} = [-1.8, -1.8, 0.0]^T$ and $p_{max} = [1.8, 1.8, 2.0]^T$. The robots adhere to a safety gap $r_{min} = 0.3m$ with sizing matrix is designated to identity $\Theta = \mathbf{I}_3$ for simplicity. We assess the proposed distributed algorithm's performance in two scenarios: a four-agent scenario and an eight-agent scenario. This enables us to evaluate the algorithm's performance at different environmental densities. The values for weighted matrices are chosen by tuning the cost function as defined in (8) and (11): $\mathbf{Q}_s = \text{diag}([100, 100, 100])$, $\forall s \in \{1, 2, \dots, N_p\}$, $\mathbf{R} = \text{diag}([50, 50, 200])$ and $\eta = 100$. The parameters of the B-spline formulation is detailed in Table 1.

Table 1: Parameters for B-spline trajectory

Sampling time $h(s)$	Degree of B-spline p	No. control points $N + 1$	Prediction horizon N_p
0.2	3	6	15

Case 1: Square corner swapping for 4 robots (normal density)

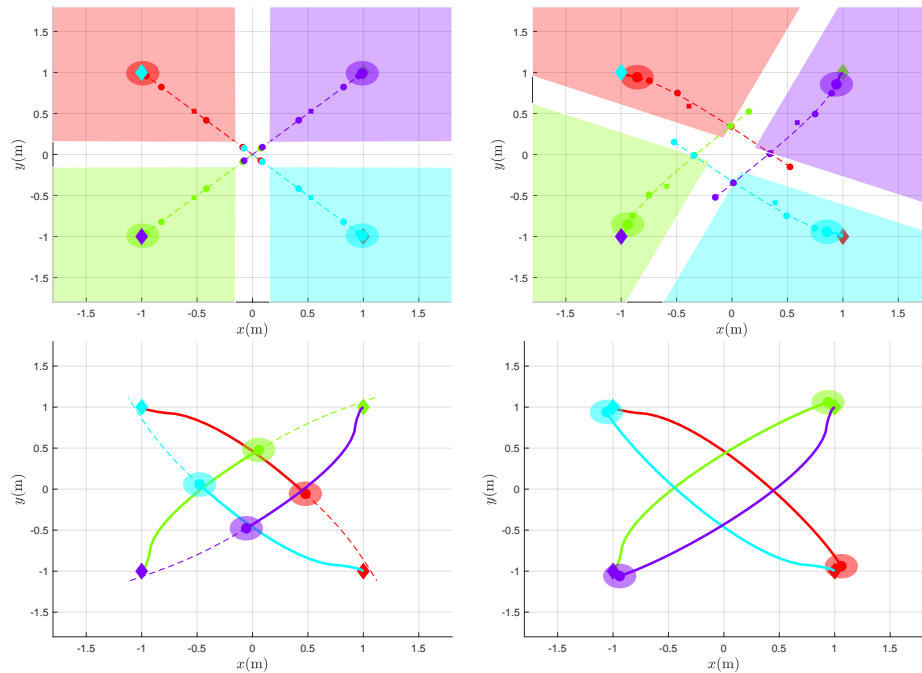


Fig. 3: Four robots exchanging position online. The diamond symbols represent the targets being tracked by agents of the corresponding color.

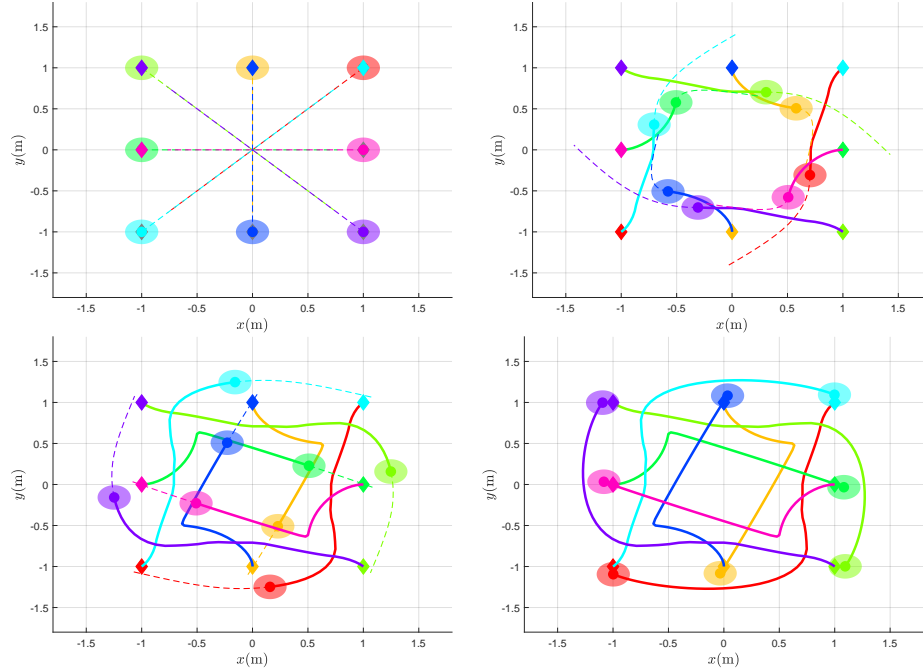
Case 2: 8 robots exchanging positions (high density)

Fig. 4: Eight drones tracking the target position online.

In the above simulations, the circular regions depict safety zones with of agents with radius r_{min} . The results in two cases are summarized in Table 2. As anticipated, the algorithm enables safe navigation for a multi-robot system within a confined space, achieving a termination time of less than 20 seconds. However, as the number of agents increases, the minimum inter-agent distance decreases, potentially reducing the feasibility of the problem. Therefore, the introduction of soft constraints proves highly beneficial in enhancing the problem's feasibility.

Table 2: Simulation results for two scenarios

	4 agents	8 agents
Termination time(s)	7.0	10.8
Minimum inter-agent distance(m)	0.45923	0.28724
Average QP solving time(ms)	13.875	18.354

5 Conclusion

This paper explores the concept of parameterizing distributed Model Predictive Control for multi-agent systems using B-spline curves to generate collision-free trajectories in complex environments. The properties of B-splines, along with relaxation techniques and Voronoi partition enable real-time execution, as demonstrated through several simulations provided. Future work will involve deploying these concepts in real systems for validation.

References

1. Adajania, V.K., Zhou, S., Singh, A.K., Schoellig, A.P.: Amswarm: An alternating minimization approach for safe motion planning of quadrotor swarms in cluttered environments. *arXiv preprint arXiv:2303.04856* (2023)
2. Dinh, C.K., Prodan, I., Stoican, F.: Online b-spline based trajectory planning for swarm of agents using distributed model predictive control. In: *2024 International Conference on Unmanned Aircraft Systems, ICUAS'24* (2024)
3. Do, H.T., Prodan, I.: Indoor experimental validation of mpc-based trajectory tracking for a quadcopter via a flat mapping approach. In: *2023 European Control Conference (ECC)*, pp. 1–6. IEEE (2023)
4. Hönig, W., Preiss, J.A., Kumar, T.S., Sukhatme, G.S., Ayanian, N.: Trajectory planning for quadrotor swarms. *IEEE Transactions on Robotics* **34**(4), 856–869 (2018)
5. Luis, C.E., Vukosavljev, M., Schoellig, A.P.: Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters* **5**(2), 604–611 (2020)
6. Marguet, V., Dinh, C.K., Prodan, I., Stoican, F.: Constrained pso-splines trajectory generation for an indoor nanodrone. In: *2024 International Conference on Unmanned Aircraft Systems, ICUAS'24* (2024)
7. Piegl, L., Tiller, W.: *The NURBS book*. Springer Science & Business Media (1996)
8. Prodan, I., Stoican, F., Louembet, C.: Necessary and sufficient lmi conditions for constraints satisfaction within a b-spline framework. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 8061–8066. IEEE (2019)
9. Prodan, I., Stoican, F., Olaru, S., Niculescu, S.I.: *Mixed-integer representations in control design: Mathematical foundations and applications*. Springer (2016)
10. Reis, M.F., Andrade, G.A., Aguiar, A.P.: Safe autonomous multi-vehicle navigation using path following control and spline-based barrier functions. In: *Iberian Robotics conference*, pp. 297–309. Springer (2023)
11. Rodrigues, R.T., Aguiar, A.P., Pascoal, A.: A b-spline mapping framework for long-term autonomous operations. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3204–3209. IEEE (2018)
12. Wang, L., Ames, A.D., Egerstedt, M.: Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics* **33**(3), 661–674 (2017)
13. Zhou, D., Wang, Z., Bandyopadhyay, S., Schwager, M.: Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells. *IEEE Robotics and Automation Letters* **2**(2), 1047–1054 (2017)