# Mixed-Integer Programming Techniques in Distributed MPC Problems

Ionela Prodan, Florin Stoican, Sorin Olaru, Cristina Stoica, Silviu-Iulian Niculescu

# Mixed-Integer Programming Techniques in Distributed MPC Problems

Ionela Prodan, Florin Stoican, Sorin Olaru, Cristina Stoica, Silviu-Iulian Niculescu

**Abstract** This chapter proposes a distributed approach for the resolution of a multi-agent problem under collision and obstacle avoidance conditions. Using hyperplane arrangements and mixed integer programming, we provide an efficient description of the feasible region verifying the avoidance constraints. We exploit geometric properties of hyperplane arrangements and adapt this description to the distributed scheme in order to provide an efficient Model Predictive Control (MPC) solution. Furthermore, we prove constraint validation for a hierarchical ordering of the agents.

## 1 Introduction

*Distributed control* usually means a decomposition of a large scale system into a set of several smaller subsystems ("neighborhoods"). The rationale of this approach is to provide subsystems which have fewer decision variables and are affected by fewer constraints (thus making the optimization problems easier to solve). This design requirement means that we aim for subsystems which are loosely inter-coupled, i.e., a given subsystem is affected by only a few other subsystems [11].

A classical area in which distributed control can be applied is the control of *multi-agent systems* [1]. Here, the subsystems affect each other mainly through obstacle and collision avoidance requirements. The primary challenge in applying these requirements is that they model a non-convex feasible region, i.e., the agent state trajectory has to avoid a convex region representing an obstacle (static constraints) or another agent (dynamic constraints - leading to a parametrization of the set of constraints with respect to the current state).

Ionela Prodan, Sorin Olaru, Cristina Stoica
SUPELEC Systems Sciences (E3S) - Automatic Control Department, Gif Sur Yvette, France,
e-mail: ionela.prodan,sorin.olaru,cristina.stoica@supelec.fr

Florin Stoican
Norwegian University of Science and Technology (NTNU) - Department of Engineering Cybernetics, Trondheim, Norway, e-mail: florin.stoican@itk.ntnu.no

Ionela Prodan, Silviu-Iulian Niculescu
SUPELEC Systems Sciences (E3S) - Signals and Systems Laboratory, Gif Sur Yvette, France,
e-mail: ionela.prodan,silviu.niculescu@lss.supelec.fr

An extensively used method for formulating this type of problems is represented by *Mixed-Integer Programming* (MIP) [2] which provides the advantage of *explicitly* including non-convex constraints and discrete decisions in the optimization problem. These techniques have proven their usefulness in various applications. Among them, we cite [8] for task assignment with coordinated control of multiple agents, subject to dynamics and collision avoidance constraints, and [12] for fault detection and isolation.

A sensitive aspect of MIP techniques is the *computational complexity* which can increase exponentially with the number of binary variables used in the problem formulation. There are some works where the original decision problems are reformulated in a simplified MIP form [14], but the complexity still remains significant. Other works try to reduce the number of binary variables, e.g., through a logarithmic formulation, as recalled in [13]. A similar technique with geometric insights into the description of the feasible region is discussed in [9] and provides a notable improvement of the overall MIP formulation. The main improvement is the use of hyperplane arrangements which give a formal way of describing the usually *non-connected and non-convex* feasible region.

With all these improvements considered, the problem at hand is still difficult to solve, even more so when a Model Predictive Control (MPC) scheme is applied (both the dimension of the solution space and the difficulty of describing the feasible region become larger with an increase in the length of the prediction horizon). MPC and, naturally, Distributed Model Predictive Control (DMPC) can be used to tackle this problem and the present chapter aims to offer a MIP solution in the context of predictive control. Here we propose to present a two-pronged approach. On one side, we consider the geometric description of the feasible region and on the other side we introduce a distributed MPC approach, upon the principles in [11]. In both cases we exploit the topology of the problem: we consider neighborhoods which partition the agents into groups and give a MIP description of the feasible region in which they stand. With an adequate communication between the resulting groups of agents, we are able to respect performance and stability constraints while in the same time we greatly reduce the computational load.

The rest of the chapter is organized as follows. The preliminaries are given in Section 2. Improvements in the geometric representation are shown in Section 3 and a hierarchical distributed MPC is discussed in Section 4. The conclusions are drawn in Section 5.

The following notation will be used throughout the paper. The closure of a set $S$, $cl(S)$, is the intersection of all closed sets containing $S$. The collection of all possible combinations of $N$ binary variables will be noted $\{0,1\}^N := \{(b_1,\ldots,b_N) : b_i \in \{0,1\}, \forall i = 1:N\}$. For a scalar $x \in \mathbb{R}$, we denote by $\lceil x \rceil$ the upper integer part. A finite intersection of inequalities which describes a non-empty region is called a polyhedral set. A polytope is a bounded polyhedral set. Minkowski's addition of two sets $\mathcal{X}$ and $\mathcal{Y}$ is defined as $\mathcal{X} \oplus \mathcal{Y} = \{x+y : x \in \mathcal{X},\ y \in \mathcal{Y}\}$. We write $\mathbf{R} \succ 0$ to denote that $\mathbf{R}$ is a positive definite matrix and $\mathbf{Q} \succeq 0$ a positive semidefinite matrix.

## 2 Preliminaries and Prerequisites

In this section, we recall results from [9] and the appropriate framework which permit to describe a non-convex and non compact feasible region using the MIP formalism. In the second part, we use this codification to construct a typical centralized optimization problem for a multi-agent system [8]. Lastly, we shed light on the shortcomings afflicting this construction (most importantly the

numerical difficulties) and provide a sketch of the approaches we consider in the rest of the chapter for improving the solution.

## *2.1 Mixed integer representation of a non-convex feasible region*

Let us consider a collection of $N > 0$ hyperplanes

$$\mathcal{H}_i = \{x \in \mathbb{R}^n : h_i x = k_i\}, i = 1 : N, \tag{1}$$

with $(h_i, k_i) \in \mathbb{R}^{1\times n} \times \mathbb{R}$, each of them partitioning the space $\mathbb{R}^n$ into two disjoint regions (up to their common boundary – the hyperplane $\mathcal{H}_i$):

$$\mathcal{R}^+(\mathcal{H}_i) = \{x \in \mathbb{R}^n : h_i x \leq k_i\}, \quad \mathcal{R}^-(\mathcal{H}_i) = \{x \in \mathbb{R}^n : -h_i x \leq -k_i\}. \tag{2}$$

Assuming their intersection non-empty and bounded, we define the polytopic set $P \subset \mathbb{R}^n$ through its implicit half-space description[1]:

$$P = \bigcap_{i=1:N} \mathcal{R}^+(\mathcal{H}_i), \tag{3}$$

and its complement as:

$$\mathcal{C}_X(\mathbb{P}) = cl(X \setminus P) = \bigcup_{i=1:N} \mathcal{R}^+(\mathcal{H}_i). \tag{4}$$

In order to obtain a manageable formulation for (4), one has to use mixed integer techniques with the aim of defining a polyhedra in the extended space of *state* and *auxiliary binary variables* of the form:

$$-h_i x \leq -k_i + M\alpha_i, i = 1 : N \tag{5a}$$

$$\sum_{i=1:N} \alpha_i \leq N - 1, \tag{5b}$$

with a positive scalar $M$ chosen appropriately (that is, significantly larger than the rest of the variables in the right hand side of the inequalities)[2].

*Remark* 1. Inequality (5a) becomes active when its associated binary variable $\alpha_i$ is "0" and redundant when the binary variable is "1" (due to term $M$, the right hand side is much larger than the left hand side and variable "x" can have any value). E.g., region $\mathcal{R}^-(\mathcal{H}_i)$ can be obtained from (5a)–(5b) by projecting along:

$$\alpha_i := (1, \ldots, 1, \underbrace{0}_{i}, 1, \ldots, 1). \tag{6}$$

Condition (5b) forces at least one binary variable "0" and thus, makes at least one inequality active. ♦

[1] We have made the simplifying convention that all the half-spaces appearing in (3) are of form $\mathcal{R}^+(\cdot)$.

[2] Sometimes this construction is called in the literature the "big M" formulation.

**Illustrative example**

Consider the following illustrative example depicted in Figure 1, where the complement of the triangle $P = R^+(\mathcal{H}_1) \cap R^+(\mathcal{H}_2) \cap R^+(\mathcal{H}_3)$ represents the feasible region. In formulation (5a)–(5b)

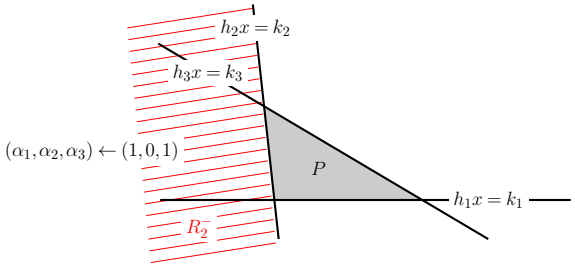


**Fig. 1** Exemplification of mixed integer codification.

there corresponds a unique binary variable to each inequality. Then, by choosing adequate combinations of binary variables, each region $\mathcal{R}^-(\mathcal{H}_i)$ can be recovered from the extended formulation (5a)–(5b). For example, by taking $(\alpha_1, \alpha_2, \alpha_3) \leftarrow (1, 0, 1)$ we can recover region $\mathcal{R}^-(\mathcal{H}_2)$, as depicted in Figure 1.

## *2.2 Mixed integer representation of a non-convex and non-connected feasible region*

The results shown in Section 2.1 have several limitations. Firstly, the feasible region is the complement of a polytope and thus has a restricted feasibility and secondly, the used MIP formulation requires a large number of binary variables (one for each inequality). Here we address both these shortcomings by considering the feasible region as the complement of a union of bounded polyhedral sets and by employing a logarithmic formulation for the MIP description.

Let us define a non-convex and possibly non compact feasible region as the complement of a union of bounded polyhedral sets $\mathbb{P} = \bigcup_l P_l$, with $P_l = \bigcap_{i \in \mathcal{K}_l} \mathcal{R}^+(\mathcal{H}_i)$ where $\mathcal{K}_l$ denotes the set of indices of the hyperplanes defining polytope $P_l$:

$$\mathcal{C}_X(\mathbb{P}) = cl(X \setminus \mathbb{P}). \tag{7}$$

We denote the collection of hyperplanes which define $\mathbb{P}$ as $\mathbb{H}$ and their number as $N \triangleq \sum_l |\mathcal{K}_l|$. The reduced notation $\mathcal{C}(\mathbb{P})$ is used whenever $X \subseteq \mathbb{R}^n$ is presumed known or is considered to be the entire space $\mathbb{R}^n$.

To describe (7) we introduce the next combinatorial notion.

**Definition 1** (Hyperplane arrangements – [15])**.** A collection of hyperplanes $\mathbb{H} = \{\mathcal{H}_i\}_{i=1:N}$ will partition the space into a union of disjoint cells $\mathcal{A}_l$ defined as follows:

$$\mathcal{A}(\mathbb{H}) = \bigcup_{l=1,\dots,\gamma(N)} \left( \underbrace{\bigcap_{i=1}^{N} \mathcal{R}^{\sigma_l(i)}(\mathcal{H}_i)}_{\mathcal{A}_l} \right) \tag{8}$$

where $\sigma_l \in \{-,+\}^N$ denotes feasible combinations of regions (2) that are obtained for the hyperplanes in $\mathbb{H}$. □

We note that there exists a subset $\{\mathcal{B}_j\}$ of *feasible* cells from (8) which describes region (7):

$$\mathcal{C}_X(\mathbb{P}) = \bigcup_{j=1,\dots,\gamma^b(N)} \mathcal{B}_j, \text{ with } \mathcal{B}_j \in \left\{ \mathcal{A}_i \in \mathcal{A}(\mathbb{H}) : \mathcal{A}_i \bigcap \mathcal{P} = \emptyset \right\} \tag{9}$$

where $\gamma^b(N) \leq \gamma(N)$ denotes the number of cells $\mathcal{B}_j$.

Recall that any of the cells of (9) is described by a unique sign tuple $(\mathcal{B}_j \leftrightarrow \sigma_j)$. As such, we obtain that the cells are disjoint and cover the entire feasible space (7). For our purposes we are satisfied with any collection of regions not necessarily disjoint which covers the feasible space.

We can formally represent the problem by requiring that (7) is described by a union of regions $\{C_k\}_{k=1:\gamma^c(N)}$ (where $\gamma^c(N)$ denotes the number of cells),

$$\mathcal{C}_X(\mathbb{P}) = \bigcup_{k=1,\dots,\gamma^c(N)} C_k, \tag{10}$$

which verifies the next conditions:

- the new polyhedra are formed as unions of the old polyhedra (i.e., for any $k$ there exists a set $\mathcal{I}_k$ which selects indices from $1:\gamma^b(N)$ such that $C_k = \bigcup_{j \in \mathcal{I}_k} \mathcal{B}_j$);
- the union is minimal, i.e., the number $\gamma^c(N)$ of regions is minimal.

Existing merging algorithms are usually computationally expensive but here we can simplify the problem by noting that the sign tuples $\sigma_l$ describe an adjacency graph since any two cells whose sign tuples differ at only one position are neighbors and that the union of any two adjacent cells is a polyhedron. In particular, in [9], it is shown that boolean algebra methods can be applied for the generation of the merged cells.

We can now state the mixed integer formulation of the non-convex region (10) which allows to express it as a polyhedra in the extended space $X \times \{0,1\}^q$ of *state and auxiliary binary variables* as follows:

$$\left.\begin{array}{c} \vdots \\ \sigma_k(i_1) h_{i_1} x \leq \sigma_k(i_1) k_{i_1} + M\alpha_k(\lambda) \\ \vdots \\ \sigma_k(i_s) h_{i_s} x \leq \sigma_k(i_s) k_{i_s} + M\alpha_k(\lambda) \\ \vdots \end{array}\right\} \quad C_k \tag{11}$$

with functions $\alpha_l(\cdot)$ with binary arguments which link (10) with (11). The sequence $1 \leq i_1 < i_2 \cdots < i_s \leq N$ denotes the hyperplanes appearing in the definition of cell $C_k$.

In order to provide an explicit representation of the binary part of (11) we recall a slightly modified form of Proposition 3.1 from [9].

**Proposition 1.** *For each cell $C_k$ we associate a unique combination of binary variables $\lambda^k \in \{0,1\}^q$ where $q = \lceil log_2 \gamma^c(N) \rceil$. Then, we can construct the functions $\alpha_k : \{0,1\}^q \to \{0\} \cup [1,\infty)$ such that they are affine in $\lambda$ and verify the relations:*

$$\alpha_k(\lambda) = \begin{cases} 0, \text{ for } \lambda = \lambda^k \\ \geq 1, \text{ for } \lambda \neq \lambda^k \end{cases} \tag{12}$$

*and are defined as follows:*

$$\alpha_k(\lambda) = \sum_{i=0}^{q} \left( \lambda_i^k + (1 - 2\lambda_i^k) \cdot \lambda_i \right). \tag{13}$$

*Index 'i' denotes the $i^{th}$ variable and $\lambda_i^k$ its value for the tuple $\lambda^k$, associated to cell $C_k$. A similar construction can be found in [13].* □

*Sketch of proof.* See the proof of Proposition 3.1 of [9].

*Remark* 2. The validation of (12) for functions (13) assures that projecting (11) along the direction $\lambda = \lambda^k$ results in the space $\mathbb{R}^n$ in the cell $C_k$ (since only $\alpha_k(\lambda^k)$ will be zero). Note also that the converse is false: if a binary variable is "1", the corresponding inequality degenerates such that it covers any point $x \in \mathbb{R}^n$ (this represents the limit case for $M \to \infty$). The last issue means that all the combinations of binary variables not associated with cells from (10) need to be made infeasible by additional inequalities. See Corollary 3.1 and Proposition 3.2 of [9] for constructive details. □

### Illustrative example

Consider the following illustrative example depicted in Figure 2, where the complement of the union of two triangles ($\mathbb{P} = P_1 \cup P_2$) represents the feasible region. We take $\mathbb{H} = \{\mathcal{H}_i\}_{i=1:4}$ the collection of $N = 4$ hyperplanes (given as in (1)) which define $P_1$ and $P_2$ as follows: $P_1 = R^+(\mathcal{H}_1) \cap R^+(\mathcal{H}_2) \cap R^+(\mathcal{H}_3)$, $P_2 = R^-(\mathcal{H}_1) \cap R^-(\mathcal{H}_2) \cap R^+(\mathcal{H}_4)$.

We observe that we have 11 cells obtained as in the arrangement (8). From them, a total of 9, $\mathcal{B}_1, \ldots, \mathcal{B}_9$, describe the non-convex region (7). To each of them corresponds a unique tuple of signs, e.g., $B_1 \leftrightarrow (-,-,-,-)$ means that $B_1$ is the result of intersecting half-spaces $\mathcal{R}^-(\mathcal{H}_1), R^-(\mathcal{H}_2), R^-(\mathcal{H}_3)$ and $R^-(\mathcal{H}_4)$.

Applying the merging methods of [9] we obtain 4 overlapping regions: $C_1 = B_1 \cup B_2 \cup B_3 \cup B_4$, $C_2 = B_4 \cup B_5 \cup B_6$, $C_3 = B_6 \cup B_7 \cup B_8 \cup B_1$ and $C_4 = B_8 \cup B_9 \cup B_1 \cup B_2$, depicted in Figure 2 through the dotted lines. Consequently, we note that $q = 2$ binary variables suffice in coding these regions. As for (11) and (13), we are now able to describe cells $\{C_i\}$ in the MIP formalism attaching to each of the regions a tuple in lexicographical order as seen in the left side of Figure 2.

Note that, in addition to reducing the number of regions from 9 to 4, we also have reduced the number of hyperplanes appearing in the region's half-space representation. In this particular case no tuple remains unallocated and thus, no additional constraints need to be added (see Remark 2).

$$-h_3 x \le -k_3 + M(\quad\ \lambda_1+\lambda_2)\Big\} \quad C_1$$

$$\left.\begin{array}{l} -h_1 x \le -k_1 \\ -h_4 x \le -k_4 \end{array}\right\} + M(1+\lambda_1-\lambda_2) \quad C_2$$

$$-h_4 x \le -k_4 + M(1-\lambda_1+\lambda_2)\Big\} \quad C_3$$

$$\left.\begin{array}{l} h_1 x \le\ \ k_1 \\ -h_2 x \le -k_2 \end{array}\right\} + M(2-\lambda_1-\lambda_2) \quad C_4$$

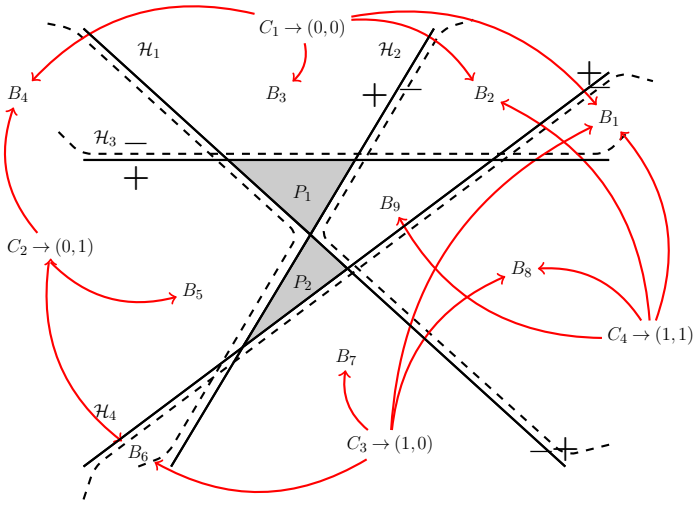


**Fig. 2** Exemplification of hyperplane arrangement and cell merging.

## *2.3 Numerical issues*

Although functional, the representation of a feasible region defined as in Section 2.1 does not scale favorably with increases in space dimension and number of hyperplanes. More precisely, the number of feasible cells of an arrangement of form (8), denoted as $\gamma(N)$, is bounded by Buck's formula (see for more details [9]):

$$\gamma(N) \le \sum_{i=0}^{d} \binom{N}{i} \tag{14}$$

with equality satisfied if the hyperplanes are in general position and $X = \mathbb{R}^n$ (in relation with the space dimension – $d$ and the number of hyperplanes – $N$).

The increase in the number of cells makes their enumeration more difficult and increases (even after applying merging algorithms) the number of necessary binary variables. Therein lies the second issue which plagues the centralized formulation: the mixed programming algorithms increase in worst case situations exponentially with respect to the number of binary auxiliary variables and the computation becomes fragile for high dimensions.

To alleviate these issues, we propose two enhancements:

- to reduce the complexity of the problem and the dimension of the space in which the problem is solved, we consider a distributed MPC approach. We partition the agent collection into neighborhoods and compute the solutions locally while using information provided by the other neighborhoods. The stability of the overall formation and a reasonable performance of the agents also needs to be verified;
- to reduce the complexity of the feasible region representation we decompose only the "visible" part of the feasible region (i.e., the region which is reachable along the prediction horizon of the agents from a given neighborhood).

In the forthcoming sections we will detail these improvements and show that the computational load is reduced significantly while the performance of the scheme remains within acceptable bounds.

## 3 Feasible region decomposition improvements

Consider a set of $N_a$ linear systems which model the behavior of individual agents.

Let us define:

$$\mathcal{I} \triangleq \{1, \ldots, N_a\}, \tag{15}$$

as the collection of all agents indices. In the following, a basic multi-agent problem is considered. Besides normal constraints upon input/output magnitude and rate of variance, we need to consider collision and obstacle avoidance constraints. Let us for the moment ignore the dynamics governing the agents trajectory and discuss only the non-convex constraints coming from the obstacle and collision avoidance. To this end, let $x_i \in \mathbb{R}^d$ be the state and $S_i$ the invariant with respect to time safety region (see [7]) associated with the $i^{th}$ agent. The collection of fixed obstacles is described through a union of polyhedra $\{\mathcal{O}\}_{o\in\mathcal{I}_o}$ with $\mathcal{I}_o \triangleq \{1, \ldots, N_o\}$ and where $N_o$ denotes the number of polyhedra $\mathcal{O}_o$.

This allows us to write the collision/obstacle avoidance conditions as follows:

$$(\{x_i\} \oplus \mathcal{S}_i) \cap (\{x_j\} \oplus \mathcal{S}_j) = \emptyset, \qquad i, j \in \mathcal{I},\ i \neq j, \tag{16a}$$

$$(\{x_i\} \oplus \mathcal{S}_i) \cap \mathcal{O}_o = \emptyset, \qquad i \in \mathcal{I},\ o \in \mathcal{I}_o. \tag{16b}$$

Concatenating the variables $x_i$ into the extended variable $\mathbf{x} \triangleq \begin{bmatrix} x_1^T \ldots x_{N_a}^T \end{bmatrix}^T$ and using the fact that $[(\{a\} \oplus A) \cap (\{b\} \oplus B) = \emptyset] \leftrightarrow [a - b \notin B \oplus \{-A\}]$we reach the equivalent formulation

$$(P_i - P_j)\mathbf{x} \notin (\{-S_i\} \oplus S_j), \qquad i, j \in \mathcal{I},\ i \neq j \tag{17a}$$

$$P_i\mathbf{x} \notin (\{-S_i\} \oplus \mathcal{O}_o), \qquad i = 1 : N_a,\ o \in \mathcal{I}_o, \tag{17b}$$

where matrix $P_i \triangleq [0 \ \ldots \ \underbrace{I}_{i} \ \ldots \ 0]$ "extends" the state $x_i$ into the extended state $\mathbf{x}$ (i.e., $x(k)_i = P_i\mathbf{x}(k)$). Clearly, conditions (17) describe a non-convex region of form (7) which ultimately results in a description of form (10). We abuse the notation and thereafter we use the same notation for describing region (17):

$$\mathbf{x} \notin \mathbb{P}, \quad \mathcal{C}(\mathbb{P}) = \bigcup_{k=1,\ldots,\gamma^c(N)} C_k. \tag{18}$$

The above construction corresponds to a centralized approach where all the agents are considered simultaneously. In contrast, the distributed approach partitions the set of agent indices into subsets ("neighborhoods") and solves a series of local optimization problems.

For further use, we introduce the following definition of a neighborhood of agents.

**Definition 2.** Let $\mathbb{N} \triangleq \bigcup_i \mathcal{N}_i$ be the collection of neighborhoods $\mathcal{N}_i \subseteq \mathcal{I}$ which are considered to be *disjoint* (for any $i \neq j$, $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$) and to cover $\mathcal{I}$ (for any $i \in \mathcal{I}$ there exists $j$ such that $i \in \mathcal{N}_j$), with $\mathcal{I}$ defined as in (15). ♦

For each optimization involving the agents of a given neighborhood, the feasible region can be obtained from conditions (17) by selecting only the constraints involving the agents of the current neighborhood and taking the other agents as additional obstacles. This approach has the drawback of necessitating a continuous recalculation of decomposition (10). An alternative solution which uses the previously calculated $\mathcal{C}(\mathbb{P})$ to compute the feasible region corresponding to $\mathcal{N}_i$ is proposed in the following statement.

**Proposition 2.** *Let the variable $\mathbf{x}_{\mathcal{N}_i} \triangleq \left[x_{i_1}^T\ x_{i_2}^T\ \ldots\right]^T$ denote the agents of the neighborhood $\mathcal{N}_i$ and $\mathbf{x}_{\mathcal{I}\setminus\mathcal{N}_i}$ denote the remaining agents. With the notation of (17) and considering that $\mathbf{x}_{\mathcal{I}\setminus\mathcal{N}_i} \in \mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$ we have that the feasible region characterizing $\mathbf{x}_{\mathcal{N}_i}$ is defined as:*

$$\mathcal{C}(\mathbb{P})|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}} = \bigcup_{k=1,\ldots,\gamma^c(N)} C_k|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}}, \tag{19}$$

*with*

$$C_k|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}} \triangleq \left\{\mathbf{x}_{N_i} : \begin{bmatrix}\mathbf{x}_{\mathcal{N}_i}\\ \mathbf{x}_{\mathcal{I}\setminus\mathcal{N}_i}\end{bmatrix} \in C_k, \forall x_{\mathcal{I}\setminus\mathcal{N}_i} \in \mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}\right\}. \tag{20}$$

□

*Proof.* The proof is constructive. By intersecting every cell $C_k$ with $\mathbb{R}^{n\cdot|\mathcal{N}_i|} \times \mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$ and then projecting along the $\mathbf{x}_{\mathcal{N}_i}$ subspace we obtain the restrictions $C_k|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}}$ of the from (20), allowing us to conclude the proof. □

## 4 Distributed MPC optimization problem

A number of commonly found situations in the control related to multi-agent systems imply a cost function that has to be minimized, while in the same time, the agent avoids collision with obstacles and other agents. To solve this problem, there exists various methods in the literature. Arguably, they can be gathered in methods which penalize through the cost function as the violation of the constraints (e.g. Potential Field Method [5] and Navigation Functions [10], and methods which impose hard constraints which cannot be broken. The latter group usually employs receding horizon techniques as they naturally take into account constraints.

We start by describing in Section 4.1 the centralized optimization problem and make use of it in Section 4.2 to construct and analyze a particular distributed approach.

### *4.1 Centralized multi-agent problem*

Let us recall the $N_a$ agents considered in Section 3 and provide for them the LTI dynamics:

$$x(k+1)_i = A_i x(k)_i + B_i u(k)_i, \quad i \in \mathcal{I}, \tag{21}$$

where $x(k)_i \in \mathbb{R}^n$ and $u(k)_i \in \mathbb{R}^m$ represent the $i^{th}$ agent state and input, respectively, at time[3] $k$. The matrices $A_i \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times m}$ describe the dynamics and the pair $(A_i, B_i)$ is controllable.

Furthermore, let us consider the centralized optimization problem formulation. We take a cost function $V(\mathbf{x},\mathbf{u}) : \mathbb{R}^{N_a\cdot n} \times \mathbb{R}^{N_a\cdot m} \to \mathbb{R}$ which aims at maintaining a formation, following a reference path or simply to gather the agents towards the origin and the constraints defined as in (18). The centralized optimization problem using a finite receding horizon technique is formulated as follows:

$$\mathbf{u}^* = \arg \min_{\mathbf{u}(k|k),\ldots,\mathbf{u}(k+N_p-1)|k)} \sum_{l=0}^{N_p-1} V(\mathbf{x}(k+l|k), \mathbf{u}(k+l|k)) \tag{22a}$$

$$\text{s.t.:} \quad \mathbf{x}(k+l+1|k) = \mathbf{A}\mathbf{x}(k+l|k) + \mathbf{B}\mathbf{u}(k+l|k), \quad l = 0 : N_p - 1, \tag{22b}$$

$$\mathbf{x}(k+l|k) \in \mathcal{C}(\mathbb{P}), \quad l = 1 : N_p. \tag{22c}$$

The optimization problem (22) requires the minimization of the cost function over a finite prediction horizon $N_p$. From the optimal sequence of inputs $\mathbf{u}(k)^*, \ldots, \mathbf{u}(k+N_p-1|k)^*$ the first control input, $\mathbf{u}(k)^*$, is selected and applied to the centralized system, thus closing the loop.

*Remark* 3. The problem can be further simplified by noting that the agents' dynamics are usually subject to operational constraints (e.g., magnitude or rate of variation constraints) which limit their actual range. This means that we need to consider in the description of $\mathcal{C}(\mathbb{P})$ only these cells (20) which are intersecting the *reachable set* of the agents. The simplification of the scheme can be significant, e.g., if no obstacle is in the "line of sight" of the agents and the agents themselves are far away from each other, then the resulting feasible domain will be convex. □

## *4.2 Distributed multi-agent problem*

As detailed in the introduction, the distributed approach has obvious computational benefits. Consequently, we reformulate problem (22) into a distributed form. For the agents of neighborhood $\mathcal{N}_i$, the local optimization problem becomes:

$$\mathbf{u}^*_{\mathcal{N}_i} = \arg \min_{\mathbf{u}_{\mathcal{N}_i}(k|k),\ldots,\mathbf{u}_{\mathcal{N}_i}(k+N_p-1|k)} \sum_{l=0}^{N_p-1} V_{\mathcal{I}\setminus\mathcal{N}_i}(\mathbf{x}_{\mathcal{N}_i}(k+l|k), \mathbf{u}_{\mathcal{N}_i}(k+l|k)) \tag{23a}$$

$$\text{s.t.:} \quad \mathbf{x}_{\mathcal{N}_i}(k+l+1|k) = \mathbf{A}_{\mathcal{N}_i}\mathbf{x}_{\mathcal{N}_i}(k+l|k) + \mathbf{B}_{\mathcal{N}_i}\mathbf{u}_{\mathcal{N}_i}(k+l|k), \quad l = 0 : N_p - 1, \tag{23b}$$

$$\mathbf{x}_{\mathcal{N}_i}(k+l|k) \in \mathcal{C}(\mathbb{P})|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}}. \tag{23c}$$

The use of indexing "$\mathcal{N}_i$" in (23) is to be understood as in Proposition 2, e.g., $\mathbf{A}_{\mathcal{N}_i}$ denotes the concatenation (block-diagonal in this case) of state matrices $A_j$ where the indices $j$ are found in the neighborhood $\mathcal{N}_i$.

The local cost function $V_{\mathcal{I}\setminus\mathcal{N}_i}(\mathbf{x}_{\mathcal{N}_i}, \mathbf{u}_{\mathcal{N}_i}) : \mathbb{R}^{|\mathcal{N}_i|\cdot n} \times \mathbb{R}^{|\mathcal{N}_i|\cdot m} \to \mathbb{R}$ is defined as

[3] Whenever the time instant is clear we abuse the notation and denote the current state, $x(k)_i$, as $x_i$. The same simplified notation is applied to the input.

$$V_{\mathcal{I}\setminus\mathcal{N}_i}(\mathbf{x}_{\mathcal{N}_i}, \mathbf{u}_{\mathcal{N}_i}) = \min_{\substack{\mathbf{u}_{\mathcal{I}\setminus\mathcal{N}_i}\in\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i} \\ \mathbf{x}_{\mathcal{I}\setminus\mathcal{N}_i}\in\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}}} V(\mathbf{x}, \mathbf{u}), \tag{24}$$

where $\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i}$ denotes the values taken by the inputs $u_{\mathcal{I}\setminus\mathcal{N}_i}$ (similarly with the definition of set $\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$).

*Remark* 4. The use of operator "min" assumes a cooperative approach (the agents exterior to the current neighborhood will try to accommodate the inputs/states suggested by the optimization problem). If replacing with the "max" operator we assume instead an adversarial or indifferent approach where the worst combination of inputs/states of the exterior agents has to be taken into account. □

Both the cost function (23a) and the feasible domain (23b)–(23c) depend explicitly upon the values found in the sets $\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i}$ and $\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$ which characterize the behavior (input and state) of the agents exterior to the current neighborhood. Consequently, the content of these sets can accommodate a large span of distributed control strategies. If no information is forthcoming from the exterior, then these sets are defined using reachable analysis, thus resulting in a decentralized control. At the other extreme, when the exact state of the exterior agents is communicated we have a distributed cooperative approach.

Usually, distributed approaches necessitate several iterations in-between consecutive discretization steps. That is, at discretization step $k$ for a state $x(k)_i$ we may have $\bar{p}$ iterations with the intermediate values $x_i^p$ where $p = 0 : \bar{p}$ and $x_i^0 \leftarrow x(k)$ and $x(k+1)_i \leftarrow x_i^{\bar{p}}$. The computations stop after some predefined number of iterations or when a consensus is reached.

Here we propose a hierarchical implementation which avoids by construction consensus verification and requires only one iteration. To this end we consider a hierarchical ordering of the neighborhoods[4]. For neighborhood $\mathcal{N}_i$, the remaining indices, represented by $\mathcal{I}\setminus\mathcal{N}_i$, are partitioned into $\mathcal{N}_i^-$ and $\mathcal{N}_i^+$ which denote the neighborhoods with lower, respectively higher, priority. Then, the sets $\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i}$ and $\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$ are constructed as follows:

$$\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i} = \left(\bigcup_{j\in\mathcal{N}_i^-}\{u_j^0\}\right) \cup \left(\bigcup_{j\in\mathcal{N}_i^+}\{u_j^1\}\right), \tag{25a}$$

$$\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i} = \left(\bigcup_{j\in\mathcal{N}_i^-}\{x_j^0\}\right) \cup \left(\bigcup_{j\in\mathcal{N}_i^+}\{x_j^1\}\right), \tag{25b}$$

where superscript "0" and "1" denote the current iteration for a certain variable (e.g., $u_j^0$ means that the state is not yet updated and the initial value $u(k)_j$ will be used, whereas $u_j^1$ means that we use the updated value, the one which will define the next discretization step, $u(k+1)_j$).

Under the aforementioned constructive assumptions we state the following lemma dealing with constraints verification.

**Lemma 1.** *Let the agents be in a feasible agent formation at discretization step k:*

[4] Note that in the hierarchical implementation the neighborhoods are disjoint, see also the definition of neighborhoods $\mathcal{N}_i$ in Section 3.

$$\mathbf{x}(k) \in \mathcal{C}(\mathbb{P}). \tag{26}$$

*By applying optimization problems* (23) *with sets* (25) *for each of the neighborhoods* $\mathcal{N}_i$ *which partition* $\mathcal{I}$*, we preserve the formation feasibility at the next discretization step* $k+1$*:*

$$\mathbf{x}(k+1) \in \mathcal{C}(\mathbb{P}). \tag{27}$$

□

*Proof.* The proof is inductive. Consider neighborhood $\mathcal{N}_i$ and its attached optimization problem (23). Due to the construction of sets $\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i}$ and $\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$ as in (25), the optimization problem "sees" the agents of higher order in their updated positions and the ones of lower order in their initial positions. Then, the resulting control $u_{\mathcal{N}_i}(k+1)$ will lead to a state $x_{\mathcal{N}_i}(k+1)$ which respects the new states of the higher order agents (since they are explicitly included in the constraint set) and the un-updated states of the lower-order agents. On the other hand, the lower-order agents will not break the constraints involving agents with index in $\mathcal{N}_i$ because from their point of view, this neighborhood has a superior position in the hierarchy.

The presence of the constraints associated to the lower order agents guarantees the feasibility of $\mathbf{x}(k+1)$. At the $i^{th}$ optimization problem, the initial state of the lower order agents is respected, thus, if no movement is possible for them, they can at lest keep the same state. This final argument completes the global recursive feasibility proof of the control scheme. □

**Algorithm 1:** Distributed MPC scheme

**Input**: obstacles $\{\mathcal{O}_o\}$, safety regions $S_i$, neighborhoods $\mathbb{N}$, initial inputs and states $\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i}, \mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$, agent dynamics $(A_i, B_i)$ and global cost function $V(\cdot,\cdot)$

describe $\mathbb{P}$ based on collision and obstacle avoidance conditions (17) and extract the collection of hyperplanes $\mathbb{H}$;
obtain the cell arrangement as in (8) for $\mathbb{H}$;
obtain the feasible cells (9) and merge them in representation (10);
**for** $k = 1, 2, \ldots$ **do**
  **foreach** $\mathcal{N}_i \in \mathbb{N}$ **do**
    for neighborhood $\mathcal{N}_i$ calculate $\mathcal{U}_{\mathcal{I}\setminus\mathcal{N}_i}$ and $\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}$ as in (25);
    construct the feasible region $\mathcal{C}(\mathbb{P})|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}}$ as in Proposition 2;
    write $\mathcal{C}(\mathbb{P})|_{\mathcal{X}_{\mathcal{I}\setminus\mathcal{N}_i}}$ in MI formulation as in (11) with Proposition 1;
    solve optimization problem (23);
    $k = k+1$;
  **end**
**end**

*Remark* 5. As mentioned in the proof of Lemma 1 we consider for lower order agents the un-updated state and thus we guarantee the existence of a solution (at worse, the lower order agent will be able

to keep the same state[5]). This approach can be generalized by assuming not a single point, but rather all the points that can be reached by those agents. In other words, this means that agents situated lower in the hierarchy could be "pushed-around" within acceptable bounds. □

*Remark* 6. Note that the partitioning between neighborhoods needs not to be time invariant. If we consider that the neighborhoods have a geometric meaning (i.e., take the indices of agents which are physically close) it may be necessary to change their content at every (few) discretization steps. In this sense, we note the $k$-means clustering algorithms, which permit partitioning a collection of agents into a predefined number of groupings and partitions the state space into Voronoi cells [3].□

To clarify the exposition we provide in Algorithm 1 a sketch of the distributed control problem.

### *4.3 Illustrative example*

For illustrative purposes let us consider the following example. Consider 3 agents, each one of them its own neighborhood: $\mathcal{N}_i = \{i\}$, $i = 1:3$. We order these neighborhoods lexicographically, that is, $\mathcal{N}_1 < \mathcal{N}_2 < \mathcal{N}_3$ in the hierarchical point of view and apply the optimization procedure (23). For clarity of the exposition we keep a one-step MPC problem (i.e., $N_p = 1$), such that only one step-ahead has to be considered in the constraints.

In Figure 3 we consider the 3 agents and show their evolution. Note that the first and last frames represent the current ($k$) and next discretization step ($k+1$) respectively and that the 3 intermediate frames represent the single iteration executed in-between the discretization steps. Further, we detail the execution of this iteration. In the second frame, we solve optimization (23) for $\mathcal{N}_1$. Since this neighborhood is the highest in the ordering ($\mathcal{N}_1^+ = \emptyset$ and $\mathcal{N}_1^- = \{2,3\}$) all the other agents are kept in their initial position and agent 1 positions itself as depicted by the dashed red contour. At the next frame, the third, we solve the optimization problem for $\mathcal{N}_2$ to which correspond $\mathcal{N}_2^+ = \{1\}$ and $\mathcal{N}_2^- = \{3\}$. In this case, $\mathcal{N}_2^+$ is not empty and thus the agent's 1 updated state is used and agent 3 has its initial state. It can be seen that agent 2 finds a better state which respects both the updated and the initial constraints. The same procedure is repeated at the next frame for $\mathcal{N}_3$ to which correspond $\mathcal{N}_3^+ = \{1,2\}$ and $\mathcal{N}_3^- = \emptyset$. In the last frame it can be seen that each of the agents has changed its position and that the constraints were respected.

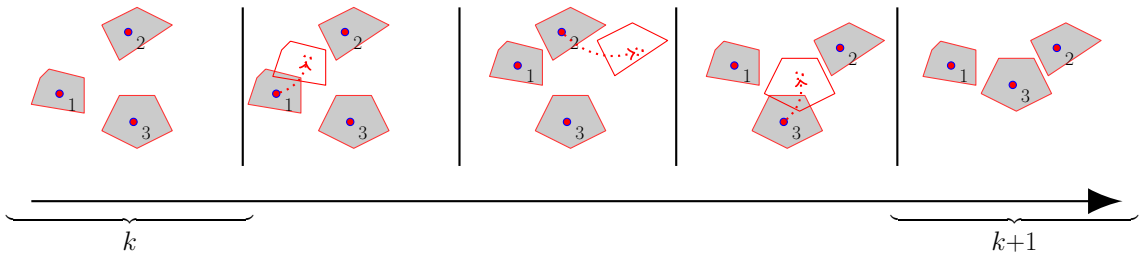


**Fig. 3** Exemplification of the hierarchical distributed approach.

[5] Not necessarily true when the dynamics describe systems which have a minimal velocity – unmanned aerial vehicles (UAVs) for example.

## 5 Conclusions

In this chapter we revisited a technique which transforms a non-convex and possibly non-connected region into a polyhedra in an augmented space (state and auxiliary binary variables) through the use of hyperplane arrangements. This enables the geometric description of the feasible regions and use them for describing a typical multi-agent collision and obstacle avoidance problem. Furthermore, the agents can be portioned into neighborhoods and a distributed Model Predictive problem can be solved through a hierarchical ordering of the neighborhoods in order to guarantee constraint validation and avoid consensus seeking. "Proof of concept" illustrations are provided. The interested reader is encouraged to seek further details on the mathematical concepts related to hyperplane arrangements in [6] and Mixed Integer Programming (MIP) based Model Predictive Control (MPC) in [4]. For the distributed MPC version, the implementation of these concepts leads to challenging open questions, as for example how to make use of reachable set calculation for MIP enhancement or how to efficiently decompose the feasible regions.

## Acknowledgements

The research of Ionela Prodan is financially supported by the EADS Corporate Foundation (091-AO09-1006).

## References

[1] D. Grundel, R. Murphey, and P. M. Pardalos. *Cooperative systems, Control and optimization*, volume 588. Springer Verlag, 2007.

[2] M. Jünger, M. Junger, T. M. Liebling, D. Naddef, G. Nemhauser, and W. R. Pulleyblank. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer Verlag, 2009.

[3] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[4] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Model predictive control for urban traffic networks via milp. In *Proceedings of the 29th American Control Conference*, pages 2272–2277, Baltimore, Maryland, USA, 30 June-2 July 2010.

[5] R. Olfati-Saber and R.M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proceedings of the 15th IFAC World Congress*, pages 346–352, Barcelona, Spain, 21-26 July 2002.

[6] P. M. Pardalos. Hyperplane arrangements in optimization. In *Encyclopedia of Optimization*, pages 1547–1548. Springer US, 2009.

[7] I. Prodan, S. Olaru, C. Stoica, and S.-I. Niculescu. Predictive control for tight group formation of multi-agent systems. In *Proceedings of the 18th IFAC World Congress*, pages 138–143, Milano, Italy, 28 August-2 September 2011.

[8] I. Prodan, S. Olaru, C. Stoica, and S.-I. Niculescu. On the tight formation for multi-agent dynamical systems. In *KES - Agents and Multi-agent Systems, Technologies and Applications, LNAI 7327*, pages 554–565. Springer, 2012.

[9] I. Prodan, F. Stoican, S. Olaru, and Niculescu S.-I. Enhancements on the Hyperplanes Arrangements in Mixed-Integer Techniques. 154(2):549–572, 2012. 10.1007/s10957-012-0022-9.

[10] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[11] R. Scattolini. Architectures for distributed and hierarchical model predictive control-a review. *Journal of Process Control*, 19(5):723–731, 2009.

[12] F. Stoican, S. Olaru, M. M. Seron, and J.A. De Doná. Reference governor design for tracking problems with fault detection guarantees. *Journal of Process Control*, 22(5):829–836, 2012.

[13] J.P. Vielma and G.L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1):49–72, 2011.

[14] M. P Vitus, V. Pradeep, G. Hoffmann, S. L Waslander, and C. J Tomlin. Tunnel-milp: Path planning with sequential convex polytopes. In *AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, USA, 18-21 August 2008.

[15] G. M. Ziegler. *Lectures on polytopes*, volume 152. Springer, 1995.