Commande prédictive sous contraintes de sécurité pour des systèmes dynamiques Multi-Agents

Minh Tri Nguyen

université PARIS-SACLAY

CentraleSupélec

NNT : 2016SACLC071

Thèse de doctorat
De
L'Université Paris-Saclay
Préparée à
CentraleSupélec

Ecole Doctorale N 580
Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Automatique

Par

Minh Tri NGUYEN

Commande prédictive sous contraintes de sécurité pour des systèmes dynamiques Multi-Agents

Thèse présentée et soutenue à Gif-sur-Yvette, le 10 Octobre 2016:

Composition du Jury:

M. George BITSORIS	Université de Patras	Rapporteur
M. Christophe LOUEMBET	Laboratoire d'analyse et d'architecture des systèmes	Examinateur
M. Hugues MOUNIER	Laboratoire des Signaux et Systèmes	Président du jury
M. Sorin OLARU	CentraleSupélec/Laboratoire des Signaux et Systèmes	Co-encadrant
Mme Ionela PRODAN	Institut Polytechnique de Grenoble, LCIS	Examinateur
Mme Cristina STOICA MANIU	CentraleSupélec/Laboratoire des Signaux et Systèmes	Directeur de thèse
M. Didier THEILLIOL	Université de Lorraine - CRAN	Rapporteur

Remerciement

Le travail de thèse présenté dans ce mémoire a été mené à l'Ecole CentraleSupélec/Laboratoire des Signaux et Systèmes.

Au bout de trois ans de thèse, avant tout, je tiens à exprimer toute ma reconnaissance et ma profonde gratitude à mes encadrants Cristina Stoica Maniu et Sorin Olaru. Leur soutien, la confiance qu'ils m'ont accordée, leurs conseils pour la rédaction et surtout leur suivi régulier ont largement contribué à l'aboutissement de ce travail.

Je tiens également à adresser de sincères remerciements à tous les membres du jury pour avoir accepté d'examiner ma présentation, malgré leur emploi du temps chargé et la distance géographique. En premier lieu, je voudrais remercier le Professeur Hugues Mounier pour avoir présidé ce jury. J'adresse ma profonde gratitude au Professeur George Bitsoris et au Professeur Didier Theilliol qui ont accepté d'être mes rapporteurs et qui ont pris aussi de leur temps précieux pour discuter avec moi. Leurs conseils pédagogiques m'ont permis de bien clarifier et de reformuler certaines problématiques, et même de perfectionner la prsentation de mes résultats. Je remercie également Ionela Prodan et Christophe Louembet qui ont accepté de faire partie du jury.

Je voudrais remercier tout le personnel du département Automatique de CentraleSupélec qui m'ont permis d'effectuer ce travail dans les meilleures conditions, ainsi que tous les collaborateurs avec lesquels j'ai été amené à travailler au cours des projets de recherche internationaux.

Je remercie tous mes amis vietnamiens et en particulier Bình, Hai, Dũng, Quang, Thanh, Son, Duy Minh, Anh Tuân, Phuong Uyên, Hoàng, Lý, qui sont tous actuellement doctorants et étudiants de l'Université Paris-Saclay. Je vous adresse ma plus profonde gratitude pour votre soutien au cours de ces trois années de thèse. Je vous offre ce travail en témoignage de ma reconnaissance et pour vous dire que je vous porte dans mon coeur.

J'adresse également des remerciements particuliers à Dai Viêt, Phú Thái, Trung Truc, Dai Duong et Hăng qui sont venus pour ma présentation et qui m'ont aidé à bien préparer le pot après la soutenance.

Enfin, je remercie tous mes amis et collègues qui sont venus vers moi, qui ont franchi les barrières de langue et de culture et qui ont toujours été présents dans les bons moments comme dans les plus difficiles.

Je remercie vivement Seif, Mohammed-Tahar qui ont assisté à ma soutenance et qui sont restés à mes côtés jusqu'au moment du verdict. Je tiens à remercier aussi mon ancien co-bureau Sofiane qui m'a encouragé non seulement le jour de la soutenance et tout au long des années où nous étions dans le même bureau. Je remercie aussi mon collègue Djawad qui m'a toujours donné les meilleurs conseils pour ma vie professionnelle future.

Je remercie également les collègues de mon deuxième bureau, Nolwenn, Guillaume et Ion pour avoir été si gentils avec moi, pour m'avoir beaucoup aidé à retoucher mon CV et en outre pour leur soutien dans ma recherche d'emploi.

Tout au long de mes trois années de thèse, j'ai pu apprécier vos compétences techniques et sociales qui font de vous des doctorants et des collègues très agréables au quotidien. Merci pour cela.

Enfin, je tiens à adresser mes sincères remerciements à tous mes amis et collègues qui ont été présents dans les bons moments comme dans les plus difficiles.

Dédicaces

Je dédie ce travail modeste qui est le fruit de mes efforts au bout de trois années

A mes parents et mon frère, qui m'ont accompagné, soutenu et encouragé tout au long de la réalisation de ma thèse. Pour moi vous êtes toujours ma source de motivation permanente pour franchir toutes les difficultés. Ce mémoire, plus précisément toute ma thèse, je vous dédie comme un cadeau d'un membre de notre famille pour toutes vos espérances.

A Tô Uyên, qui ne cesse jamais à me soutenir, à m'encourager tout au long de ces années. Je te dois mes promesses, plus de sept ans d'attente et de mes sincères excuses, juste à cause de mes décisions égoïstes.

A tous mes amis et collègues qui ont été présents dans les bons moments comme dans les plus difficiles.

Contents

Acronyms

AI	Artificial intelligence
AUV	Autonomous underwater vehicle
BMI	Bilinear matrix inequality
CC	Chebyshev configuration
CE	Control engineering
CMPC	Centralized Model Predictive Control
CSC	Chebyshev static configuration
CVC	Centroidal Voronoi Configuration
DMPC	Decentralized Model Predictive Control
dMPC	Distributed Model Predictive Control
FDI	Fault Detection and Isolation
FTC	Fault-tolerant control
GSC	General static configuration
HJB	Hamilton-Jacobi-Bellman
KKT	Karush-Kuhn-Tucker
LMI	Linear matrix inequality
LP	Linear programming
LTI	Linear time invariant
MAS	Multi-Agent system
MIP	Mixed-integer programming
MPC	Model Predictive Control
PI	Positive invariant
QP	Quadratic programming
RPI	Robust positively invariant
SC	Static configuration
TVV	Time-varying Voronoi partition
UAV	Unmanned aerial vehicle

Notation

The notations used in the manuscript are conventional and widely used in the control system/engineering literature. A short description is provided in the following.

In the sequel, $\mathbb{N}$ and $\mathbb{R}$ represent respectively the set of natural numbers and the set of real numbers. Let $\mathbb{R}_+$ denoting the set of the positive real numbers. We define also $\mathbb{R}_{[a,b)} = \{x \in \mathbb{R} | a \leq x < b\}$. Let $\mathbb{N}_{[n,n+m]} \triangleq \{n, n+1, ..., n+m\} \subset \mathbb{N}$ for $n, m \geq 0$ denoting the set of the natural numbers from n to $n+m$ in increasing order.

With respect to the vector operator, considering a column vector $x = [x_1 \; x_2 \; \dots \; x_n]^\top \in \mathbb{R}^n$ with $x_i \in \mathbb{R}$, the following norms are employed

- $|x| = [|x_1| \; |x_2| \; \dots \; |x_n|]^\top$: element-wise absolute value;
- $||x|| = \sqrt{x^\top x}$: Euclidean norm;
- $||x||_Q^2 = x^\top Q x$: Euclidean norm weighted by matrix Q.

Here, $I_n \in \mathbb{R}^{n \times n}$ and $0_n \in \mathbb{R}^{n \times n}$ denote respectively the unitary matrix and the zero matrix of $n \times n$ dimension. The symbol $0_{m,n}$ represents a zero matrix in $\mathbb{R}^{m \times n}$. The notation $1_n \in \mathbb{R}^n$ is used for the column vector whose elements are 1.

Considering the matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$. The notation $A \otimes B \in \mathbb{R}^{mp \times nq}$ is used to denote their Kronecker product

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

Moreover, we use $\sigma(A)$ to denote the sum of all elements of the matrix A, i.e. $\sigma(A) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}$. In addition, $A \succ 0$ (or $A \succeq 0$) means that A is a strictly (semi-) positive definite matrix. Similarly, $A \prec 0$ (or $A \preceq 0$) means that A is a strictly (semi-) negative definite matrix.

To denote a block matrix A formed by a given set of element matrices $A_{ij} \in \mathbb{R}^{m_i \times n_j}$, $i \in \mathbb{N}_{[1,M]}$, $j \in \mathbb{N}_{[1,N]}$, i.e.

$$A = \begin{bmatrix} A_{11} & \dots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MN} \end{bmatrix}$$

we use the notation $A = [A_{ij}]$, $\forall i \in \mathbb{N}_{[1,M]}$, $\forall j \in \mathbb{N}_{[1,N]}$. Matrix $A \in \mathbb{R}^{m \times n}$ has the appropriate dimension according to the dimensions of the element matrices, i.e. $m = \sum\limits_{i=1}^{M} m_i$ and $n = \sum\limits_{i=1}^{N} n_i$.

Similarly, a block diagonal matrix A formed by a given set of N element matrices $A_i \in \mathbb{R}^{m_i \times n_i}$, with $i \in \mathbb{N}_{[1,N]}$ is denoted by

$$A = \text{diag}\{A_1, A_2, \dots, A_N\} = \begin{bmatrix} A_1 & 0_{m_1,n_2} & \dots & 0_{m_1,n_N} \\ 0_{m_2,n_1} & A_2 & \dots & 0_{m_2,n_N} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{m_N,n_1} & 0_{m_N,n_2} & \dots & A_N \end{bmatrix}$$

Considering a set $\mathcal{P} \in \mathbb{R}^n$, $C(\mathcal{P}) = \mathbb{R}^n \setminus \mathcal{P}$ denotes its complement set. The cardinality of $\mathcal{P}$ is denoted by $|\mathcal{P}|$. By $\text{int}(\mathcal{P})$ it will be denoted the interior of the set $\mathcal{P}$ and by $\partial\mathcal{P}$ its frontier set. $\mathcal{C}(G, \xi)$ denotes the polyhedral cone $\mathcal{C}(G, \xi) = \{x \in \mathbb{R}^n | Gx \leq \xi,\ G \in \mathbb{R}^{p \times n}, \xi \in \mathbb{R}^p\}$. If $\xi = 0_{p,1}$, then it is a proper cone denoted by $\mathcal{C}(G)$. In addition, the notation $\mathbb{B}(x, r)$ is used to denote a ball centered at $x \in \mathbb{R}^n$, with the radius $r \in \mathbb{R}_+$.

We introduce next a collection of Multi-Agent system basic notions used throughout the manuscript.

The set notation $\mathcal{N} = \mathbb{N}_{[1,N]} \triangleq \{1, 2, ..., N\}$, with $N \in \mathbb{N}$, will denote the set of agents index in the MAS. The set $\mathcal{N}_E \in \mathbb{N}_{[1,N]}$ denotes the subset containing the indices of the eliminated agents due to the fault occurrence. Hence the complement set of the faulty agents' indices is $\mathcal{N}_R = \mathbb{N}_{[1,N]} \backslash \mathcal{N}_E$.

Considering an agent indexed by $i \in \mathcal{N}$, the following vector notions are introduced:

- $\check{x}_i(k)$ denotes the *one-step predicted state* of the i^{th} agent;
- $\bar{x}_i$ denotes the *target position* of the i^{th} agent in the case that the common reference of the entire MAS reduces to the origin;
- $\breve{x}_i(k)$ denotes the *trajectory reference* of the i^{th} agent, once one configuration for the entire system is determined.

Two following notions of finite-time receding horizons will be used for the Model Predictive Control framework's presentation:

- *Prediction horizon* denoted by N_p - will be employed as a finite-time window in the future;

- *Fault monitoring horizon* denoted by N_m - a finite-time window in the recent past, used to characterize the time to detect a fault.

The set $\mathcal{N}_i \subset \mathcal{N}$ contains the indices of the agents neighbor of the i^{th} agent. Its *safety region* represented by means of bounded polyhedral set is denoted by $\mathcal{S}_i$.

Denote by $\mathbb{V}$ the Voronoi partition of a considered Euclidean working space into n cells $\mathbb{V}_i$, with $i \in \mathbb{N}_{[1,n]}$ and $\mathbb{V} = \bigcup\limits_{i=1}^{n} \mathbb{V}_i$.

Résumé

Introduction aux systèmes dynamiques Multi-Agents

La notion de système Multi-Agent (MAS – "Multi-Agent System") était longtemps associée à l'Intelligence Artificielle (IA – "Artificial Intelligence"), qui représente des grands systèmes informatiques composés par plusieurs agents autonomes. Depuis la naissance de ce domaine en 1956, plusieurs définitions du système Multi-Agent ont été proposées. Du point de vue informatique, un MAS est reconnu typiquement par le nombre de ses agents, l'environnement (ou l'espace de travail dans lequel les agents interagissent) et finalement le niveau d'autonomie des agents. Il faut noter que la définition de l'agent n'est pas unique, mais en principe un agent doit porter les quatre propriétés caractéristiques suivantes :

- Autonomie : l'agent capable de prendre une décision le concernant ;
- Coopérativité : cette propriété implique la communication entre les agents ;
- Perceptivité : l'agent s'adapte aux changements et à l'infuence de l'environnement de travail ;
- Proactivité : l'agent se focalise sur le but de la mission commune (la formation des agents).

A côté des points communs avec l'AI, un système Multi-Agent est largement considéré dans l'Automatique, en raison de l'explosion du nombre des applications qui peuvent être décomposées en plusieurs sous-systèmes. En principe, la définition de MAS-Automatique (MAS-IC) est similaire à celle de MAS-AI, à l'exception que le comportement d'un agent du point de vue de l'IC peut être caractérisé par une équation dynamique. Pour cette raison, dans les applications de IC, un système Multi-Agent est souvent appelé système dynamique Multi-Agent. Plus spécifiquement, le concept de régulation d'un système dynamique Multi-Agent porte sur la supervision des interactions (coopératives ou adversaires) et il implique donc de trouver la meilleure stratégie de commande pour que tous les agents acquirent un but commun.

Depuis cinq décennies, un système Multi-Agent est utilisé comme un nouveau paradigme dans l'Automatique. Plusieurs applications sont concernées, telles que smart- ou micro-grid, surveillance de consommation énergétique des bâtiments, réseaux de distribution aquatique, réseaux de transport, déploiement d'un groupe de capteurs mobiles, des mini-robots ou des drones. Toutes ces applications ont les points caractéristiques ci-dessous :

- La nature de dynamique de l'agent (linéaire ou non-linéaire, variant ou invariant, etc.) ;
- Le type de graphe de communication (faible ou fort connecté, direct ou indirect, découplé ou non-découplé, etc.) ;
- L'environnement de travail (dans l'air, sous-marin, présence des obstacles, etc.) ;
- Les contraintes liées aux limitations des actionneurs, à la consommation énergétique et à l'anti-collision.

Choix du graphe de communication et méthodes pour la synthèse de la loi de commande

Notons que le graphe d'intercommunication entre les agents est l'un des points caractéristiques les plus importants d'un MAS. C'est l'élément principal pour décider la façon d'interaction des agents et de calculer la commande.

Les trois approches suivantes souvent considérées pour résoudre les problèmes de type MAS :

- centralisée ;
- distribuée ;
- décentralisée.

Chacune d'entre eux a ses propres avantages et inconvénients au sujet de l'objectif de commande, les limites des ressources de calcul et les contraintes issues de l'environnement de travail (perturbations, incertitudes, non-homogénéité de l'environnement, etc.).

L'approche centralisée est généralement privilégiée dans la littérature à être employée pour résoudre un problème de de type MAS. Suivant cette approche, tous les agents du MAS doivent être connectés à un centre de supervision. Ce centre est responsable de surveiller et de calculer les actions de commande puis les envoyer à tous les agents. En revanche, ceci implique que le graphe de communication doit être complètement connecté. Le calcul de commande est donc facilement perturbé par les défauts dans les informations échangées. D'autre part, le calcul centralisé est coteux et ainsi n'est pas appropriée en cas de large dimension du MAS ou lorsque les contraintes dans le cadre du MAS sont nombreuses.

L'approche distribuée et l'approche décentralisée sont donc mentionnées pour surmonter les inconvénients ci-dessus. Dans l'approche distribuée, chaque agent n'est connecté qu'avec ses plus proches voisins. Le graphe de communication devient donc partiellement connecté. Plus précisément, le graphe est décomposé en plusieurs sous-groupes dont chacun se compose d'un agent et ses proches voisins. On peut donc économiser l'énergie consommée pour la communication. Les tâches de calculs sont réduites aussi puisque elles ne sont plus centralisées, mais réparties au niveau de chaque sous-groupe. Par conséquent, la robustesse de commande est améliorée par rapport au cas centralisé.

Comme l'approche distribuée, un graphe décentralisé est aussi partiellement connecté, mais plus efficace en terme de minimisation des tâches de communication. La différence principale entre ces deux approches est la façon de calculer l'action de commande. Dans l'approche distribuée, les calculs sont partagés entre les agents dans un même sous-groupe, alors que dans le cas décentralisé, les calculs sont réalisés uniquement par chaque agent en exploitant ses propres informations et celles échangées avec ses voisins. L'un des avantages les plus remarquables de l'approche décentralisée consiste à permettre de rajouter ou enlever facilement un agent sans perturber significativement le système global. Ceci s'adapte forcément à la philosophie de la commande tolérante aux défauts. Pour toutes ces raisons, nous avons privilégié d'utiliser l'approche décentralisée pour développer les structures de commande dans cette thèse.

Après avoir décidé la structure de communication suivant ces trois approches ci-dessus, nous allons choisir la méthode appropriée pour synthétiser les actions de commande. Nombreuses méthodes ont été menées et développées dans la littérature, telles que la théorie des graphes, la théorie des jeux, la théorie de la viabilité, la méthode de construction à base de fonctions de potentiel, etc.

Parmi toutes les méthodes, nous privilégions l'utilisation des méthodes à base d'optimisation. Ce choix couvre les techniques de commande optimale et commande prédictive qui seront développées tout au long de ce manuscrit. Le plus grand avantage est que ces méthodes permettent de formuler la détermination de l'action de commande sous forme d'un problème d'optimisation. Ainsi, la détermination de la loi de commande peut être résolue en utilisant les outils tels que les solveurs LP ("Linear Programming"), QP ("Quadratic Programming") ou MIP ("Mixed-Integer Programming"). Ainsi les différentes contraintes peuvent être prises en compte dans les calculs. Un autre avantage est la possibilité d'utiliser des outils ensemblistes pour résoudre un problème d'optimisation.

Pour toutes ces raisons, dans cette thèse, les méthodes ensemblistes avec les techniques d'optimisation sont choisies comme principaux outils pour synthétiser les structures de commande tolérante aux défauts.

Méthodes ensemblistes appliquées dans le contexte de la commande tolérante aux défauts pour systèmes dynamiques Multi-Agent

Commande tolérante aux défauts (FTC – "Fault Tolerant Control") est une branche de recherche largement exploitée dans la littérature. Récemment, les méthodes ensemblistes commencent à être employées dans le développement de FTC. Cependant, très peu de résultats concernant l'application des méthodes ensemblistes pour la commande tolérante aux défauts des systèmes Multi-Agent ont été trouvés. Pour cette raison, l'objectif du présent manuscrit est de développer les nouvelles structures de commande tolérante aux défauts pour les systèmes dynamiques Multi-Agent. Cette commande est développée en suivant les principes d'optimisation et les méthodes ensemblistes. Ainsi sa synthèse peut être formulée comme un problème d'optimisation sous contraintes. Dans le but de préserver la sécurité de la mission et aussi la sureté de fonctionnement, une région de sécurité (sous la forme d'un polyèdre convexe et invariant) est associée à chaque agent. La collision est alors évitée en assurant que ces régions ne se superposent pas. Ceci est mis sous forme des contraintes non convexes rajoutées au calcul de la commande. Résoudre un tel problème est équivalent à réguler un système global en évitant un obstacle qui contient le point d'équilibre standard à l'intérieur. L'accès à ce point est donc infaisable. Par conséquent, un nouveau point d'équilibre est choisi tel qu'il se situe sur l'une des frontières définissant la forme de l'obstacle. L'anti-collision est donc garantie en assurant que le système tend vers ce point en évitant la collision avec l'obstacle. D'une façon équivalente, une formation est aussi un point d'équilibre pour le système Multi-Agent. Ce point généralisé peut en plus se décomposer en plusieurs points d'équilibre dont chacun est associé à un agent. Ceci permet donc de déterminer la commande d'une façon décentralisée pour chaque agent. Le problème à vérifier est la faisabilité de cette commande locale. De plus, les zones de sécurité sont utilisées pour construire les ensembles caractérisant les modes de fonctionnement de chaque agent et donc nous permettent de détecter si un agent est défectueux.

Les méthodes ensemblistes appliquées dans le contexte de la théorie de commande portent sur la théorie des ensembles et, en particulier, l'algèbre Brunn-Minkowski. Ces résultats sont ensuite employés pour les systèmes dynamiques caractérisés par les équations différentielles.

Les premières tentatives d'application des méthodes ensemblistes dans le domaine de sureté de fonctionnement portent sur la construction des ensembles caractérisant les modes de fonctionnement du système surveillé. Les défauts considérés dans ces travaux se limitent au niveau des défaillances dans les composants (tels que les actionneurs ou les capteurs). Dans le contexte de supervision d'un système Multi-Agent, la collision, la dégradation des informations échangées, le changement du nombre des agents et même les perturbations issues de l'environnement de travail sont aussi considérés comme les défauts. Parmi ceux-ci, l'évitement des collisions jusqu'à présent est toujours considéré comme l'un des défis typiques dans la régulation du système Multi-Agent. En pratique, la collision peut être évitée à condition que les régions de sécurité caractérisant chaque agent ne se heurtent pas. Ces zones peuvent être représentées en termes des ensembles.

Plusieurs familles d'ensemble peuvent être menées, telles que les ensembles en forme d'étoile, les zonotopes, les ellipsoïdes, etc. Chacune d'entre eux a ses propres avantages et inconvénients qui sont bien exploités dans la littérature. Nous choisissons des polyèdres comme la famille d'ensemble principale utilisée dans ce manuscrit, en raison de leur convexité géométrique et flexibilité en termes de représentation.

L'application des outils ensemblistes demande de faire appel à des notions d'invariance qui associent la dynamique du système avec un ensemble caractéristique statique dans l'espace d'état (Blanchini and Miani (2007), Aubin (2009)). Les conditions algébriques d'invariance et d'invariance contrôlée dans le cas des polyèdres sont bien détaillées dans Bitsoris (1988a).

Le choix des polyèdres non seulement donne des avantages au niveau du calcul grâce à la versatilité de sa double représentation (demi-espace/sommets), mais ce choix autorise également de formuler la détermination de l'action de commande sous la forme de la solution d'un problème d'optimisation. Nous présentons dans le manuscrit comment construire la zone de sécurité autour d'un agent en termes de polyèdre invariant borné, en fonction des bornes connues des perturbations à l'entrée.

Les hypothèses ci-dessous sont d'ailleurs utilisées dans le but de simplifier la présentation de nos contributions sans faire perdre la généralité :

- Le système Multi-Agent ne se compose que des agents mobiles et l'équation dynamique de chaque agent est supposée linéaire, invariante, commandable et observable ;
- L'intercommunication entre les agents est maintenue de faon permanente et il ne s'agit pas de dégradation ou de perte d'information échangée ;
- La perturbation à l'entrée de chaque agent a une distribution uniforme dans un ensemble borné prédéterminé ;
- Le déclenchement du défaut est brusque, sans transition et sa propagation n'est pas considérée.

Nous allons voir comment appliquer les méthodes ensemblistes pour déterminer des commandes décentralisées sous contraintes d'anticollision dans la partie suivante.

Une solution constructive pour satisfaire les contraintes de type anti-collision par approches ensemblistes

Du point de vue ensembliste, les contraintes d'évitement des collisions sont satisfaites si et seulement si toutes les conditions ci-dessous sont garanties :

- Pas d'intersection entre les régions de sécurité des agents ;
- Pas d'intersection entre les régions de sécurité des agents et les obstacles ;
- Pas d'intersection entre les zones de fonctionnement des agents ;
- Pas d'intersection entre les zones de fonctionnement des agents et les obstacles.

Il faut noter que la régulation d'un système sous contraintes d'évitement des collisions est équivalente à conduire sa sortie vers un point d'équilibre en dehors de son domaine d'attraction. Le principe de l'invariance positive (élaboré auparavant par Bitsoris (1988a)) a été utilisé pour calculer une commande linéaire dans le but de ne pas rentrer en collision avec des obstacles. La synthèse de commande a été détaillée dans Bitsoris and Gravalou (1999) en suivant les principes d'optimisation linéaire.

Dans ce contexte, notre objectif est d'utiliser les méthodes ensemblistes pour construire une nouvelle structure de commande décentralisée dans le but d'éviter les collisions. Une structure de type leader-suiveur est choisie dans la suite.

Afin de simplifier la présentation des résultats, la méthodologie sera présentée dans l'espace $\mathbb{R}^2$. La généralisation aux dimensions plus importantes est possible et sera étudiée dans les travaux futures. Ce travail peut être considéré comme une extension naturelle des résultats théoriques présentés dans Bitsoris and Olaru (2013), mais appliqués à un système Multi-Agent et développés de manière décentralisée pour des raisons numériques.

Plus spécifiquement, l'espace de travail est partitionné en une collection de zones non superposées. Chacune d'entre eux est associée à un seul agent et nommée sa zone de fonctionnement. Le leader joue le rôle de déterminer la zone de fonctionnement pour chacun de ses suiveurs. Ensuite, le suiveur doit à son tour déterminer un point d'équilibre et une loi de commande locale en suivant la méthode de Bitsoris and Olaru (2013), afin de fonctionner seulement à l'intérieur de la zone autorisée.

Si la commande est faisable, dans la plupart des cas, l'invariance n'est garantie que pour un sous-ensemble à l'intérieur de la zone de fonctionnement. Cette zone est caractérisée en termes de le plus grande ensemble des points accessibles associé à la commande trouvée. Pour les points initialement à l'extérieur de cet ensemble, nous présentons deux stratégies de commande pour conduire la sortie de l'agent vers le sous-ensemble invariant afin d'activer la commande locale.

1. Construction d'ellipsoïde paramétrée: L'idée est de garantir la condition d'invariance à l'intérieur d'un ellipsoïde qui est enfermé dans la zone de fonctionnement et à la fois contient la sortie courante du suiveur. Par la suite, en conduisant la sortie du suiveur vers le centre de cet ellipsoïde, nous pouvons maintenir la sortie du suiveur dans la zone de fonctionnement et donc préserver sa condition d'invariance.

2. Commande par techniques d'interpolation : Nous construisons un ensemble invariant contrôlable ("controlled invariant set") à l'intérieur de la zone de fonctionnement, qui entoure le plus grand ensemble des points accessibles associé à la commande locale. L'action de commande du suiveur est interpolée à partir de cette commande locale et une action de commande associée au nouvel ensemble invariant contrôlable. La commande interpolée est calculée de telle sorte qu'il est aussi proche que possible de la commande locale pour enfermer la sortie du suiveur strictement à l'intérieur de sa zone de fonctionnement.

Si la commande locale n'est pas faisable, on doit choisir un autre point d'équilibre qui appartient cette fois strictement à l'intérieur de la zone de fonctionnement. Ainsi, on peut appliquer la méthode de construction des ellipsoïdes (voir Chapitre 3 pour plus de détails) pour garantir l'invariance de la zone de fonctionnement. La question qui se pose est comment on décompose l'espace de travail pour obtenir des zones de fonctionnement. Ceci va être détaillé dans la prochaine suivante.

Commande décentralisée par des approches ensemblistes pour le déploiement dynamique du système Multi-Agent

Comme présenté ci-dessus, l'évitement des collisions est garanti en assurant que chaque agent ne sort pas de sa zone de fonctionnement et que les zones de fonctionnements ne se superposent pas. La construction de telles zones nécessite de partitionner l'espace de travail en temps réel, en fonction des positions courantes des agents.

La méthode de Voronoi peut être utilisée pour obtenir une partition de l'espace. Chaque cellule de Voronoi est donc associée à un agent et nommée sa zone de fonctionnement. Comme la partition est construite en fonction de la position courante des agents, elle varie dans le temps. Cette problématique est nommée *partition dynamique de Voronoi* ou *déploiement de base de Voronoi*, qui est connu dans de nombreuses applications de type déploiement telles que la surveillance, le suivi environnemental/géologique, les opérations de sauvetage, etc.

L'objectif est de stabiliser le déploiement à base de Voronoi autour d'une configuration permettant de maximiser la couverture de l'espace considéré. Celle-ci est considérée comme une partition/configuration optimale du système Multi-Agent sur tout l'espace de travail. La plus connue des applications telles que les réseaux de capteurs mobiles, les systèmes multi-robots ou les multi-véhicules est la *configuration centroïde de Voronoi* (CCV), où la position de chaque agent se confond avec le centre de masse de sa cellule de Voronoi. A noter qu'une configuration centroïde de Voronoi peut être obtenue en conduisant chaque agent vers le centre de masse de sa cellule de Voronoi correspondant. Cependant, le plus grand inconvénient consiste en la complexité du calcul des intégrations pour obtenir la valeur précise du centre de masse.

Dans le chapitre 4, nous voulons remplacer le centre de masse par un autre centre d'équilibre interne qui permet d'interpréter les notions et outils ensemblistes. Le but final est donc de conduire le système Multi-Agent vers une configuration stable où la sortie de chaque agent se confond avec son nouveau centre d'équilibre interne. Pour acquérir cet objectif, nous allons proposer une commande décentralisée à base d'optimisation pour ramener la sortie de chaque agent vers son centre correspondant. Par conséquent, l'évitement de collision est assuré de manière décentralisée via son inclusion dans sa cellule de Voronoi.

Le premier choix mentionné est le *centre de Chebyshev*, qui est le centre de la plus grande boule inclue à l'intérieur de la cellule de Voronoi. Sa détermination peut être mise sous forme d'un problème d'optimisation dont les variables sont le centre et le rayon de Chebyshev. En revanche, pour certains géométries spécifiques, l'unicité du centre de Chebyshev n'est pas garantie. Ainsi, même si tous les agents sont confondus avec leur centre de Chebyhev, la configuration obtenue n'est pas statique.

Pour éviter ce problème, nous proposons d'utiliser un autre centre construit par déflation récursive après l'étape de déterminer le rayon de Chebyshev. Ce centre est nommé *centre généralisé* et il est considéré comme nouveau point d'équilibre pour l'agent. Pour un polyèdre borné et convexe, ce centre est certainement unique d'après le principe de déflation. La configuration correspondant à ce choix de centre est unique aussi.

En principe, n'importe quelle commande qui peut rendre la cellule de l'agent contrôlée λ-contractive ("controlled λ-contractive") peut garantir la convergence du système Multi-Agent vers une configuration statique et unique. La synthèse de commande est locale en fonction de la géométrie caractéristique de la cellule de Voronoi associée et suivant les techniques de commande prédictive, à condition que la cellule soit contrôlée λ-contractive vers la fin de l'horizon de prédiction.

Nous avons étudié également un autre centre d'équilibre interne en se basant sur l'interpolation des sommets de la cellule de Voronoi (voir Annexe A). Chaque sommet est supposé être un point d'équilibre accessible par rapport à la dynamique de l'agent. Des lois de commande décentralisée pour conduire l'agent vers ce type de centre, en utilisant les principes de commande optimale et ensuite de commande prédictive sous contraintes, ont été proposées dans ce manuscrit.

D'ailleurs, toujours dans le contexte des applications de déploiement, dans l'article de Moarref and Rodrigues (2014), les auteurs ont proposé une commande optimale décentralisée pour stabiliser le déploiement d'un système Multi-Agent autour d'une CCV. Le modèle dynamique des agents est un système de premier ordre et la commande est développée en temps continu. Nous avons revisité ces résultats (voir Annexe B) en développant une nouvelle commande optimale décentralisée pour le modèle dynamique à temps discret. La stabilité de notre solution suit le principe de l'équation dynamique Hamilton-Jacobi-Bellman (HJB).

La troisième contribution dans cette thèse qui consiste en détection et isolation des agents défectueux est présenté dans la partie suivante.

Reconfiguration de la formation par des approches ensemblistes et des lois de commande prédictive

La détection et l'isolement de défaut (FDI – "Fault Detection and Isolation") sont définis comme une couche supplémentaire rajouté dans la supervision du système dans le but d'enlever les anomalies et de maintenir le fonctionnement. Plusieurs études ont été menées dans la littérature sur ce sujet et divers résultats ont été obtenus. La plupart des résultats obtenus sont fondés sur le même principe de génération des signaux résiduels. Dans le contexte de FDI, un résidu est caractérisé par l'écart de suivi entre la sortie mesurée et celle estimée issue du modèle de système. Un bloc de FDI est toujours succédé par une stratégie de reconfiguration partielle/complète du système, dans le but de couvrir les impacts des défauts détectés.

Bien que des nombreuses études de FDI pour un seul système sont menées dans la littérature, jusqu'au présent très peu de résultats de la littérature concernent le développement de techniques FDI pour les systèmes Multi-Agent. Nous rappelons que la définition de sécurité pour les systèmes Multi-Agents est plus généralisée que celle d'un seul système (d'un seul agent). Plus spécifiquement, le fonctionnement du MAS n'est pas affecté uniquement par les défauts au niveau des composants de l'agent, mais la collision peut aussi être considérée comme un type de défaut important pour le système global.

Nous avons déjà montré comment utiliser les méthodes ensemblistes pour construire les régions de sécurité pour les agents et ensuite éviter la collision pendant le fonctionnement. D'ailleurs, elles peuvent être employées aussi pour construire les ensembles caractérisant les différents modes de fonctionnement. Ainsi, un défaut est détectable si son ensemble caractéristique et l'ensemble caractérisant le fonctionnement non défectueux ne se superposent pas (s'il y a une séparation de ces deux ensembles).

Trois modes de fonctionnement sont mentionnées dans cette partie :

- Sain : il n'y a pas d'anomalie ou de défaut dans le comportement ;
- Suspecté : l'état de l'agent sort de son ensemble sain, mais est encore inclus dans la formation courante ;
- Défectueux : l'état de l'agent sort de la formation courante.

Dans cette thèse, précisément dans Chapitre 5, l'objectif est d'employer les outils ensemblistes pour concevoir une nouvelle structure de FDI centralisée pour un groupe d'agents homogènes. Cette structure FDI doit être capable de détecter et d'isoler en temps réel les agents défectueux, puis permettre de reconfigurer le système courant, y compris la formation et la commande. Deux scénarios de défaut sont étudiés :

- Le bloc FDI est capable de détecter si un agent a des anomalies dans son comportement par rapport aux autres agents dans la formation. Si les anomalies sont significatives, l'agent est certifié en défaut et ensuite éliminé de la formation courante. Le bloc va faire la mise à jour du nombre des agents restant dans la formation, puis reconfigurer et aussi calculer la commande centralisée à l'instant suivant ;
- Une zone de sécurité est construite pour contenir la formation. Elle permet donc de détecter s'il y a un agent de l'extérieur qui veut rentrer dans la formation. Après une certaine durée, cet agent sera intégré dans le système global, puis à l'instant suivant la formation et la commande centralisée seront mises à jour.

Nous supposons que la communication entre les agents est complètement connectée, autrement dit chaque agent peut savoir les positions précises de tous les autres agents et il est capable de leur envoyer ses propres informations. De plus, nous faisons l'hypothèse qu'il n'y a donc pas de dégradation/retard de l'information échangée.

Le mécanisme de reconfiguration utilisé consiste à mettre à jour des informations (y compris les dynamiques, les régions de sécurité) des agents sains dans le système surveillé, de sorte que les agents ne se heurtent pas durant la reconfiguration.

Conclusions

Le présent manuscrit propose des nouvelles structures de commande tolérante aux défauts, en particulier liés à l'évitement de collision et à la surveillance des défauts. Les approches développées se basent sur les méthodes ensemblistes et les techniques d'optimisation. Le but est de préserver la sécurité de mission et la sureté de fonctionnement du système Multi-Agent, malgré les différents défauts. La régulation d'un groupe d'agents autonomes n'est pas nouvelle, mais l'évitement de collision reste encore un défi très connu jusqu'au présent. L'originalité de cette thèse consiste à concevoir les ensembles caractérisant les différentes modes de fonctionnement de chaque agent, dans le but de détecter si un agent n'est plus coopérative avec les autres agents, et d'ailleurs en synthèse de la commande prédictive décentralisée sous contraintes d'un point de vue ensembliste. Nous rappelons que bien que les méthodes ensemblistes ont été déjà employées dans plusieurs applications de commande, très peu de résultats sur la commande tolérante aux défauts pour les systèmes Multi-Agent se trouvent dans la littérature.

Sans perte de généralité, nous avons supposé que le comportement de chaque agent est représenté par une équation dynamique linéaire et invariante dans le temps. L'entrée de chaque agent subit une perturbation dont sa distribution est supposée uniforme dans un ensemble borné déjà déterminé. Ainsi, une région de sécurité autour la position nominale de l'agent peut être construit en suivant la méthode de Kofman et al. (2007). Elle prend donc la forme d'un polyèdre convexe et borné. Pour toutes ces raisons, la position nominale avec la région de sécurité deviennent les propriétés caractéristiques d'un agent tout au long du manuscrit.

En supposant connu le modèle exact caractérisant les modes de chaque agent, nous pouvons construire les ensembles de seuil caractérisant le comportement sain/suspecté/défectueux de chaque agent. Après cette étape, un signal résiduel est utilisé pour évaluer le comportement de l'agent. L'ensemble de seuil pour la formation est l'enveloppe convexe des régions de sécurités de tous les agents. Dans le manuscrit, nous nous concentrons sur les cas critiques où le déclenchement de défaut est brusque et les phases de transition ne sont donc pas considérées. Deux scénarios de défauts sont traités :

- Détection et élimination d'un agent défectueux de la formation courante ;
- Détection et intégration des nouveaux agents externes dans la formation.

Le mécanisme de reconfiguration consiste à mettre à jour le nombre des agents pris en compte dans le calcul de la commande centralisée et d'une nouvelle formation.

tant considérée comme la meilleure solution pour surmonter l'inconvénient du côté numérique de la structure centralisée, l'approche décentralisée est bien reconnue par sa capacité de plug-and-play. Dans le contexte de régulation du système Multi-Agent, nous avons proposé de localiser chaque agent dans une zone bornée, qui est construite en se basant sur les informations échangées entre cet agent et ses voisins les plus proches. Une telle zone est nommée la zone de fonctionnement de cet agent. L'objectif de commande est donc de garantir l'invariance contrôlée de cette zone via la commande locale associée à cet agent. Cependant, dans la plupart des cas, une telle commande locale ne peut préserver l'invariance que dans un sous-ensemble à l'intérieur de la zone de fonctionnement. Pour cette raison, nous avons proposé deux stratégies de commande décentralisée afin de ramener l'agent vers le sous-ensemble invariant associé à sa commande locale. L'une consiste en construire de faon itérative des ellipsoïdes contractifs à l'intérieur de la zone de fonctionnement, et l'autre se base sur les techniques dinterpolation.

Une autre application plus complexe étudiée dans le manuscrit porte sur le déploiement d'un groupe des agents sur une région bornée. A chaque instant, la région sur laquelle les agents sont déployés est partitionnée d'après l'algorithme de Voronoi. Chaque agent est ensuite associé à une cellule de Voronoi, qui est aussi nommé la zone de fonctionnement de cet agent. La commande décentralisée est calculée en choisissant tout d'abord un centre d'équilibre à l'intérieur de la cellule de Voronoi, ensuite une commande locale est déterminée pour conduire la sortie de l'agent vers le centre choisi. Le centre de masse est le choix conventionnel dans la littérature. L'algorithme de Lloyd dont la nature est équivalente au principe de commande décentralisée, est largement utilisé pour stabiliser le déploiement autour d'une configuration où la sortie de chaque agent est confondue avec son centre de masse. En raison de la complexité du calcul de centre de masse, nous l'avons remplacé par le centre de Chebyshev. Une autre solution moins conservatrice consiste à choisir un autre point d'équilibre nommé centre généralisé pour surmonter l'inconvénient d à la non-unicité de détermination du centre de Chebyshev dans certains cas spécifiques. De plus, nous avons étudiés autre types de centres tels que le centre interpolé à partir des sommets de chaque cellule de Voronoi et aussi le centre de masse. L'originalité commune est que nous avons revisité la commande décentralisée correspondant à chaque choix de centre d'équilibre, en se basant sur les principes d'optimisation et les méthodes ensemblistes.

Perspectives

Dans le cadre d'application des méthodes ensemblistes sur des systèmes Multi-Agent, certaines côtés du calcul numérique restent à considérer dans les travaux futurs. Afin d'éviter des problèmes numériques et sans perte de généralité, des hypothèses simplificatrices ont été prises en compte dans ce manuscrit. A côté de ces problèmes, vu que le concept de la sureté de fonctionnement pour un système Multi-Agent est plus large et plus généralisé que celui d'un système seul (un agent), plusieurs points peuvent être considérés comme des perspectives à exploiter dans le futur. Nous n'allons présenter les perspectives les plus significatifs correspondant respectivement à chaque contribution principale du manuscrit.

Nous insistons sur le fait que les problématiques menées ont été beaucoup simplifiés en considérant des hypothèses simplificatrices. Une dynamique nominale à été utilisée pour chaque agent. La présence des incertitudes dans le modèle de chaque agent sera étudiée dans la suite.

Dans ce manuscrit, nous avons supposé des défauts qui intervient aux moments de temps distincts. La simultanéité des différents types de défauts doit être considérée dans des travaux futures.

D'autre part, des perturbations issues de l'espace de travail peuvent générer du retard, dégrader ou même déformer gravement les informations échangées entre les agents. Dans ce cas, il serait intéressant d'étudier la robustesse des solutions proposées dans le manuscrit à côté de la performance de détection de défaut. L'idée est de diminuer/compenser/annuler les impacts de défauts afin de robustifier les commandes proposées.

La couche de détection et identification des défauts (FDI) proposée peut être développée d'une façon décentralisée dans les études à venir. En outre, la connexion entre la FDI et la commande appliquée au système Multi-Agent doit être étudiée plus profondément. Il faudra chercher la solution de commande pour éviter les risques de collision après l'étape de reconfiguration.

Une autre direction consiste à développer les résultats obtenus pour un système Multi-Agent hétérogène. Ce cas d'application semble plus compliqué, mais à la fois intéressant à étudier en raison de l'ajout des contraintes concernant la dynamique de chaque agent dans la synthèse de la commande.

Notre synthèse de commande décentralisée tolérante aux défauts dépend strictement de la géométrie de la zone de fonctionnement associée à l'agent. Cette zone est considérée comme l'union des demi-espaces qui décident la faisabilité de détermination de la commande décentralisée. Pour cette raison, il reste à trouver une partition de l'espace de travail telle qu'elle reste invariante par la commande par rapport à l'ensemble donnée des commandes décentralisées des agents. Cette partition doit assurer que chaque agent avec sa région de sécurité est inclus strictement à l'intérieur de sa zone de fonctionnement correspondante.

Dans le contexte du déploiement d'un système Multi-Agent sur une région bornée, l'objectif de commande est de maximiser la qualité de couverture d'une façon décentralisée. Nous avons choisi le diagramme de Voronoi comme outil principal pour partitionner l'espace de travail. En revanche, la géométrie propre de chaque agent n'est pas encore prise en compte dans la partition. Dans le futur proche, nous pouvons étudier comment obtenir une partition de l'espace de travail plus généralisée en fonction des positions des agents, leurs régions de sécurité et aussi de leur propres commandes prédéterminées.

En outre, les résultats obtenus sont fondés sur l'hypothèse que tout l'espace de travail du système Multi-Agent est de type "controlled invariant" par rapport à la dynamique des agents. Une autre perspective consiste à étudier le cas où l'invariance contrôlée n'est plus préservée pour tout l'espace, mais seulement pour une partition de l'espace considéré.

L'un des points les plus difficiles est la prédiction de la partition de Voronoi. Ce problème reste à être étudié dans le temps qui suit, par manque d'une formulation explicite caractérisant l'évolution dans le temps d'une telle partition. Cependant, trouver une telle formulation s'avère compliqué.

Une autre direction porte sur la commande à base d'optimisation en tenant compte d'autres types de contraintes telles que l'efficacité énergétique ou la durée effective de l'opération. Toutes ces contraintes demandent des façons spécifiques pour obtenir la commande convenable.

Nous rappelons que les contributions dans le cadre de ce manuscrit sont développées dans un cadre théorique. Bien qu'ils sont bien formulés et présentés, pour les perspectives, nous voudrions les appliquer sur les systèmes Multi-Agent réels tels que les formations de drones, des mini-robots ou des véhicules. Le but principal est de valider l'efficacité et d'évaluer la performance de nos solutions sur des systèmes Multi-Agents réels.

Chapter 1

Introduction

1.1 Interdisciplinary overview of Multi-Agent systems

1.1.1 From Artificial Intelligence to Control Engineering

According to the early works mentioning this syntagma, the notion of *Multi-Agent system* (MAS) was mainly used by the computer science community. Widely considered as a sub-branch of Artificial Intelligence (AI), this new concept aims to provide efficient tools to supervise complex systems involving multiple agents[1]. Since 1956, there were numerous different ways to define MAS and basically from the computer science point of view, a MAS is defined as a large-scale system characterized by the number of agents[2], the environment where the system operates and the autonomy level of the agents. Moreover, the definition of an agent is not unique. According to Russell et al. (1995) and Stone and Veloso (2000), there is no unified definition of "agent" in AI scope. We cite here the agent's definition of Wooldridge et al. (1995)

"a software (or hardware) entity that is situated in some environment and is able to autonomously react to changes in that environment".

According to this definition, an agent has to possess the four following basic properties:

- Autonomy: it limits the human interventions;
- Cooperativeness: it relates the inter-communication between the agents;
- Perceptiveness: it reacts to the impact of the environment[3];
- Pro-activeness: exhibit goal-directed behavior.

[1] By following Crevier (1993) and McCorduck (2004), the birth of the Artificial Intelligence is officially recognized after The Dartmouth Conference of 1956 when John McCarthy persuaded the attendees to accept "Artificial Intelligence" as the name of the *"Logic Theorist"* introduced by Allen Newell and Herbert A. Simon.

[2] Wooldridge (2009) shortly describes MAS as a computer system composed of at least two intelligent agents.

[3] Thus refers to the working environment and meanwhile the impact from the other agents.

Beside the natural cohesion with the AI development, the MAS concept is also employed widely in *control engineering* (CE), due to the flourish of various control applications that can be decomposed into multiple sub-systems each of which qualifying with respect to the previous properties. The definition of MAS-CE is basically similar to MAS-AI, meaning that MAS is a group of multiple intelligent (decision making) agents interacting within an environment subject to constraints. From the CE point of view, the main difference is that each agent is characterized by a *dynamical equation* beside the four properties above. Therefore a Multi-Agent system in the control engineering framework is usually called *dynamical Multi-Agent system.*

For this reason, throughout this manuscript, we use the acronym MAS to denote uniquely a dynamical Multi-Agent system in a control engineering perspective and related finite-dimensional case-studies.

The control concept of MAS-CE is translated in terms of supervising the agents interaction and further making the best control strategy to achieve a common goal. In the last five decades, MAS was employed as a new paradigm in CE applications, ranging from military operations (environmental/meteorological monitoring, geological exploration etc.) to domestic utilities (mini mobile robots, manufacturing line production etc.). Plenty of notable examples can be listed such as smart or micro grid (Dimeas and Hatziargyriou (2005), Pipattanasomporn et al. (2009)), building energy management (Wang et al. (2010), Zhao et al. (2013)), water distribution networks (Van Overloop et al. (2010), Ocampo-Martinez et al. (2013)), traffic and transportation networks management (Tomlin et al. (1998), Negenborn et al. (2008), de Oliveira and Camponogara (2010)), mobile sensing networks (Cortes et al. (2002), Cortes et al. (2005)), robots/vehicles deployment (Schwager et al. (2009), Moarref and Rodrigues (2014)), multi-vehicles formation control and tracking (Stipanović et al. (2004), Prodan (2012)). Furthermore, each application has its own characteristics e.g.

- the nature of agent's dynamics (linear or nonlinear, time-variant or time-invariant, etc.);
- the type of communication graph (weakly or strongly connected, directed or undirected, decoupled or interconnected, etc.);
- the working environment (air, underwater, presence of obstacles, hazardous environment, etc.);
- the constraints (limitation of actuators, energy consumption, collision avoidance etc.);

leading to various opened research directions.

Moreover, it is worth to mention that one of the most important characteristics of MAS relates to the topology of the inter-communication graph. This point is inherited from the AI development. The reason is that it is the prior element to supervise the agents behavior and determine the computation of the control decision. Some major topologies will be presented next.

1.1.2 Networks topology and control structures

The supervision of a group of relatively independent agents can be classified into three main approaches: centralized, distributed and decentralized approaches. Each of them has its own advantages and also the drawbacks ranging from the objective of operation, the limitations of resources authorized for the computation and the constraints/difficulties issued from the working environment (large disturbance, drift, propagation in a non-homogeneous environment...). The *centralized approach* is usually considered as the easiest way to solve the problem of MAS regulation, by means of considering all agents as an unique extended system. However, the main inconvenience is that it burns a considerable computation effort and it is not appropriate when the dimension of MAS increases along with the number of constraints. The most effective conventional solution to overcome this drawback is to use the *distributed approach* or the *decentralized approach*. The main difference between these two last approaches is that distributed framework simplifies the central computation by distributing it to each agent's computation block and thus handling at the central level only their inter-dependence by communication and agreement on the results. This often translates into a block-decomposition of the central computation into multiple sub-computation problems. In the decentralized approach, the graph of communication is minimized and each agent uses its own local information feedback to compute its control input (Siljak (2011)). Each type of topology decides the organization of control computation.

Therefore, we present in the following these three topologies by means of

- Architecture of graph topology;
- Structure of feedback control computation.

1.1.2.1 Centralized graph

In graph theory, a centralized graph is defined as a topology in which each node connects to a central supervising node (see the red node in Fig. 1.1). The graph is totally connected and any connexion is undirected (see Mesbahi and Egerstedt (2010)), allowing the central node to have the full knowledge of all system's nodes. From the control point of view, this node represents a centralized controller which is responsible for the system supervision and the control computation. Figure 1.2 illustrates a typical closed-loop controller of centralized MAS which is usually recalled in the MAS control literature, e.g. Feng (1981), Milutinovic and Lima (2006), Shamma (2007), Meskin and Khorasani (2011), etc. The information of all agents is sent to the central controller C which decides the behavior of each agent in the system. From outside, a centralized MAS can be equivalent to an extended global system offering centralized effective supervision. Several notable centralized control techniques related to typical MAS problems can be found in the literature. For instance, the authors of Kempker et al. (2011) propose a formation flying algorithm for Autonomous Unmanned Vehicles (AUVs) by using a centralized optimal control combined with a set of observers to estimate the full velocity. The same authors in Kempker et al. (2012) analyze the controllability and observability from a centralized point of view for a group of coordinated linear interconnected agents. Olfati-Saber (2006) and Olfati-Saber et al. (2007) give a centralized solution to obtain the consensus of flocking based on a graph-theoretic approach. Nowadays, centralized Model Predictive Control (CMPC) is employed widely in many large-scale applications due to its advantage to take into account constraints into the computation of the optimal decision over a finite prediction horizon. More precisely, Prodan (2012) developed CMPC algorithms for MAS subject to collision avoidance constraints, allowing to safely drive all agents in a predetermined configuration. However, notice that the central computation requires the full knowledge of all agents, implying that the connexion is permanently maintained. All these reasons make the centralized approach very sensitive to *communication faults*[4]. Notable works related to fault diagnosis in order to recover the MAS faults are presented in Meskin and Khorasani (2011), Nguyen et al. (2014b), Nguyen et al. (2014a).

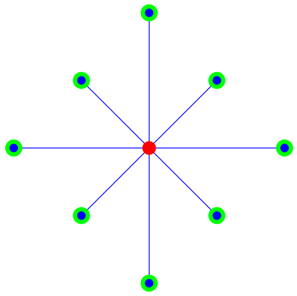

Figure 1.1: Centralized topology.

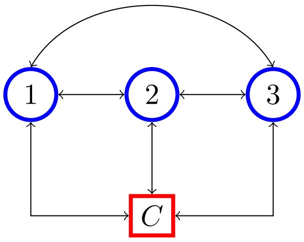

Figure 1.2: Centralized control structure.

1.1.2.2 Distributed graph

A distributed graph is defined as a topology in which each node is connected uniquely with its closest neighbors. Therefore, the graph becomes partially connected in comparison with the particular centralized case as shown in Fig. 1.3 and thus the communication task is reduced. A central controller is not necessary and consequently the computation is distributed over all nodes in the network. Concretely, the control is calculated within a sub-group of agents whose communication graph is a sub-graph of the MAS centralized topology. As a consequence, the control computation load is reduced significantly with respect to the centralized approach. An example of a distributed control architecture is shown in Fig. 1.4. Each agent has its own controller which collects the agent's state but also shares the computation with other controllers. An impressive work on the state-of-art of cooperative control of distributed MAS is given by Shamma (2007), Scattolini (2009) and Cao et al. (2013). Concerning the application

[4] Notice that any loss or degradation of information exchanged can impact seriously the communication. Furthermore, any change in the topology leads to modify the common behavior of MAS, requiring thus to fully reconfigure the entire system.

of distributed control, the paper Dimarogonas et al. (2012) proposes distributed event-triggered control for MAS. Johansen and Storaa (2002) introduces an energy-based control of a distributed solar collector field, relying on a distributed parameter nonlinear plant model and including feed-forward from the solar irradiation and inlet temperature. Recently distributed Model Predictive Control (dMPC) receives considerable attention. Beside the constraints handling as in the CMPC case, dMPC aims to achieve the constrained control objective using local MPC controllers. Maestre and Negenborn (2014) give a detailed overview of numerous methods concerning dMPC. A study on dMPC for consensus problem is presented by Keviczky and Johansson (2008), where agents negotiate to compute an optimal consensus point using an incremental sub-gradient method. Other recent notable works of Grancharova and Johansen (2011), Grancharova and Johansen (2014) focus on explicit dMPC for nonlinear interconnected systems compared with centralized non linear Model Predictive Control (NMPC). This work is extended to discrete-time polytopic system in Grancharova and Olaru (2014).

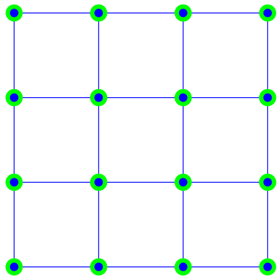

FIGURE 1.3: Distributed topology.

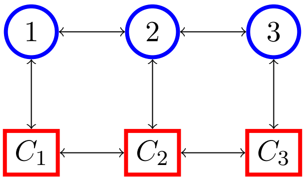

FIGURE 1.4: Distributed control structure.

1.1.2.3 Decentralized graph

Similar to a distributed structure, a decentralized graph is also partially connected, but more efficient with respect to the communication. This is obtained via hierarchical role-determining The significant difference to distinguish decentralized and distributed structures is the control computation. In the distributed case, the computation is shared between the close agents, e.g. two or three agents share together the task, whereas the computation of the decentralized control of each agent is based uniquely on its own state information, even if this agent exchanges also the information with other neighbors. An overview of decentralized control development is detailed in the survey paper Siljak (2011). Several notable decentralized applications concern multi-robots motion planning or flocking problem. For example, in the works of Khatib et al. (1996), the authors discuss and extend some methodologies to mobile manipulation systems and propose a decentralized control structure for cooperative tasks. The paper Tanner et al. (2005) introduces a graph-theoretic decentralized control to achieve the consensus of nonholonomic agents. The most well-known decentralized organization is the leader-follower structure (or hierarchical structure as seen in Fig. 1.5), where a MAS is decomposed into multiple hierarchical sub-groups. Each group is characterized by a leader followed by a set of followers (or sub-group having a lower level of hierarchy). The local/individual control of each agent is computed by itself based uniquely on its own state. Most of leader-follower applications can be found in robotic motion control, path planing (Dimarogonas et al. (2009), Consolini et al. (2008), Basilico et al. (2009)), platoon vehicles routing (Stanković et al. (2000), Farokhi and Johansson (2015)). The leader-follower approach reduces the communication and the computation tasks to minimum, and in addition it offers fault-tolerant advantages via the plug-and-play mechanism. For instance, a faulty agent can be unplugged immediately without reconfiguration of the remaining agents in MAS (Riverso et al. (2013), Nguyen et al. (2015b)). The robustness to such phenomena is one of the advantages of a decentralized control structure. In the last two decades, decentralized Model Predictive Control (DMPC) flourished significantly. Prodan (2012) provided a DMPC framework by considering a cost function composed of two potential components: repulsive (to avoid the collision) and attractive (to achieve a formation meanwhile preserving the tracking effectiveness), according the previous works concerning potential function construction in Tanner et al. (2007), Barnes et al. (2009) and Wu et al. (2010). Polyhedral or sum functions (Blanchini (1995), Camacho and Bordons (2013)) are considered to construct these two components with respect to collision avoidance constraints and the tracking requirement. The authors of Keviczky et al. (2008) introduce DMPC for the coordination in formation control. The authors use a conventional cost function subject to conflict-free constraints. Other notable results in the DMPC framework for a class of nonlinear dynamics are given by Magni and Scattolini (2006). The authors in this work emphasized that the stability proof for DMPC framework could rely on the inclusion of a contractive constraint in the formulation.

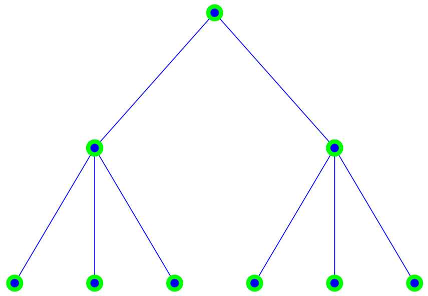

FIGURE 1.5: Decentralized topology.

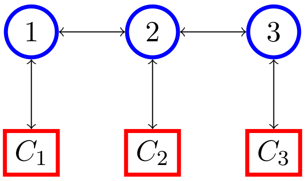

FIGURE 1.6: Decentralized control structure.

We presented above three major network topologies to construct the graph of communication for a Multi-Agent system. The choice of network topology describes how the agents interact and more important, it decides the structure and also preselect the methods of control computation. Next we will give a short description of the design techniques.

1.1.3 Conventional approaches for MAS control

Beside choosing one from three topologies mentioned for the MAS control, we focus now on the theoretical tools used to synthesize the control action. We will summarize below the main ideas of the mostly employed methods.

Graph-theoretic methods: The agents interaction can be expressed in terms of communication topology, therefore the graph theory comes as a natural approach when dealing with Multi-Agent systems. In this context, algebraic graph theory is the main theoretic tool used to solve the control problem by taking into account the agent's dynamics. An overview of graph-theoretic methods used in MAS control is detailed in Mesbahi and Egerstedt (2010) and various results are developed by Murray (2003), Lafferriere et al. (2004), Olfati-Saber and Murray (2004), Tanner et al. (2007), etc. Each agent is denoted by a node in the graph and an interaction is represented by a link between two nodes.

Game theory: Being traditionally employed as a framework tool for describing the behavior in societal systems, game theory recently becomes a powerful approach for controlling MAS. Each agent is considered as a *player* (intelligent rational decision-makers) and has to follow the game rules scenarios (*role-playing game*) which mimic human social phenomena, e.g. conflict or cooperation (see Bauso et al. (2008), Maestre et al. (2011)). Numerous results can be found, e.g. Karfopoulos and Hatziargyriou (2013)) applied game theory for controlled charging of a large population of electric vehicles, or Semsar-Kazerooni and Khorasani (2009) used cooperative game theory to achieve the MAS consensus.

Potential field approaches: Positive scalar potential function is constructed such that its minimum is obtained when the agent reaches its goal. Outside the ideal configuration, this function is defined such that it decreases towards the goal configuration, so that the agent can reach the goal by following the negative gradient of the potential. Normally, in the context of motion planning, a potential function is built on the distance between the agent's position and the desired goal. The construction can also take into account the specific shape of obstacles to concept the repulsive field beyond the attractive field determined around the goal (see Khatib (1986), Prodan (2012)). In Koditschek (1992) a historical review of the potential field approach can be found. However, this method can lead to configurations which are characterized by a local minima. Additionally, generating a navigation function is a computationally involved design problem and thus not suitable for many control applications. Possibly due to these particularities FTC design based on potential field approach remains an unexplored topic.

Viability theory: Being introduced by Aubin (1991), viability theory develops mathematical methods for investigating the adaptation to viability constraints of evolutions governed by complex (continuous, discrete-time or even hybrid dynamics) systems under uncertainties or state/input constraints, with plenty of applications in many domains e.g. biological evolution, financial market supervision, cognitive/perceptive studies etc. It opens new directions to develop the control solution for various MAS applications relating social phenomena such as automated highways or air traffic management (application on cars crash prevention in Gao et al. (2004)), ecosystem approach to fisheries (Cury et al. (2005)), management of renewable resources (Aubin and Saint-Pierre (2007)).

Optimization-based approaches: This class covers a wide range of control synthesis methods from optimal control to receding horizon optimization. The advantage is that these approaches formulate the control objective as an optimization problem, providing efficient tools (using LP, QP or MIP solvers) to find the optimal solution with respect to the considered criterion. Among the successful optimization-based control methods, MPC is known to be the best choice for constraints handling due to the receding finite horizon principle with respect to the conclusion of Mayne et al. (2000). Moreover, it allows using the notion of controlled invariance together with set computations to treat the constraints and guarantee the stability. The MPC design principles are well understood and documented in classical textbooks Maciejowski (2002). The papers Mayne et al. (2000), Rawlings and Mayne (2009) and Camacho and Bordons (2013) detail the constrained MPC particularities. Over the last two decades, many successful results related to MPC development are obtained, e.g. Bemporad et al. (2002) build the explicit solution of constrained MPC by means of a Piecewise Affine description (the solution is obtained by exploiting the KKT condition of the constrained MPC framework). In the same context, Riverso et al. (2013) contribute to plug-and-play decentralized constrained MPC, Kothare et al. (1996) developed robust constrained MPC using LMI approach. Camacho and Bordons (2013) give an overview of Min-Max MPC in presence of additive disturbance on the system's state. The book Olaru et al. (2015) presents an overview of model-based optimization and control with a collection of recent applications.

For all of the above arguments, optimization-based approaches with set-theoretic tools have been chosen as key methods for the MAS control design in the present work and will be further detailed throughout the present manuscript.

1.2 Safety of functioning: from FDI to FTC

Besides the performance quality, mission safety becomes an intensive research field for MAS applications, composed of Fault detection and isolation (FDI) - Fault tolerant control (FTC) study and development. FDI is considered as a supplementary layer to detect and isolate the faults, succeeded by a reconfiguration strategy to fully/partially reconfigure the system once the faults are located. We emphasize that the concept of MAS safety is more general than the definition of safety for single system. Practically, the functioning of MAS is not uniquely impacted by the damage occurrence on each agent (Meskin and Khorasani (2009a)) but the collision, the large environmental disturbance (strong drift, or disturbances violating the expected bound, etc.) (Meskin et al. (2010), Meskin and Khorasani (2009b)) or the change in the communication network (loss or degradation of information exchanged between the agents) can also be considered as faults for MAS. Hence FTC in the context of MAS is understood as a paradigm to design the control so that serious damages due to collision or environmental disturbances are reduced to minimum.

In the following, we present firstly the FDI design for MAS, followed by an introduction of FTC applied for formation control subject to collision avoidance constraints and ended by some details of safe deployment regulation. All these approaches are based on set-theoretic methods.

1.2.1 FDI design for MAS

Although FDI study for single dynamical system is widely known and developed in the literature (see Zhou and Frank (1998), Theilliol et al. (2002), Blanke and Schröder (2006), Noura et al. (2009) and Puig (2010)), just few results relate FDI concept for MAS. Recently designing supplementary fault diagnosis layers becomes a highly required priority for MAS. Many studies in the literature have been conducted on this topic and various results were obtained. Precisely the authors of Meskin and Khorasani (2009a), Meskin and Khorasani (2010), Meskin et al. (2010) have developed a set of FDI filters to detect the actuator faults in presence of large environmental disturbances. Then the faulty functioning of MAS is recovered by applying the Markov chain theory. Other works (Stanković et al. (2010), Antonelli et al. (2013) and Kempker et al. (2011)) have used model-based Fault Detection to generate residual signals for MAS. Recently, set theory[5] is widely employed in many research fields of automation. The reason is that set theory is proved to be a powerful tool to monitor and enhance effectively the system safety. Some notable results are listed e.g. application of set-theoretic methods to detect and recover the damage on actuators (Franze et al. (2012)) or the degradation in the sensing channels (see Olaru et al. (2010)). Stoican and Olaru (2013) employed set-theoretic methods to firstly detect and isolate the faulty sensor in a multi-sensors system. The reconfiguration step consists in eliminating the faulty sensor and switching to other healthy mode such that the closed-loop stability is not degraded. Recent results have been reported on the application of set-theoretic methods for MAS safety guarantee. Some new results in Nguyen et al. (2015a) and Rosich et al. (2014) relate to construct the dynamical threshold sets to detect the faulty agents inside the MAS and the agents from outside which need to integrate the MAS system. These works provide also algorithms to reconfigure the entire MAS after eliminating the faulty agents or integrating the new/recovered agents.

1.2.2 Fault-tolerant formation control

A formation is equivalent to the coordination via consensus (Olfati-Saber et al. (2007)) and thus, the ultimate control objective is to move all agents states toward a desired predefined formation. For a network of dynamical agents, ”consensus” means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. In the context of the cooperative behavior objective, the requirement to control a group of mobile agents and to track a predefined path while keeping a desired formation/configuration (Kempker et al. (2011), Egerstedt and Hu (2001)) are integrated on problem formulation. The theoretical framework for posing and solving consensus problems for networked dynamic systems was introduced by Murray (2003), Olfati-Saber and Murray (2004) building on the earlier work of Fax and Murray (2004). In these works, graph theory is widely employed to analyze the stability of MAS consensus or in other words, the stabilization of MAS at a predefined network configuration. An overview of graph theoretic methods used in MAS networks is formulated in Mesbahi and Egerstedt (2010). Furthermore,

[5] The full application of set-theoretic methods in control is detailed in Blanchini and Miani (2007).

the formation can be static but also time-varying, depending on the control context. In presence of arbitrary switching in the network topology, the results of Tanner et al. (2007) point out that regardless of the graph switching, convergence to a common velocity vector and stabilization of inter-agent distances are still guaranteed in a decentralized manner as long as the network remains connected at all times. Decentralized formation control applied for MAS stabilization is also developed by Lafferriere et al. (2005), as long as the graph topology has a rooted directed spanning tree. These results are extended in presence of multiple obstacles (Tanner (2004), Saber and Murray (2003)) and nonholonomic dynamics (Tanner et al. (2005)). As consequence, according to the graph-theoretic point of view, the network topology (fully/partially) is taken into account in the control computation. Recently, in the set-theoretic context, a formation can be understood as an implicit set-based form where the inter-distance between the agents is constant (Nersesov et al. (2010), Fontes et al. (2009), Kempker et al. (2011)) or a union of non-overlapping polyhedral sets such that each set represents uniquely one agent (Prodan (2012), Nguyen et al. (2015b), Bemporad and Rocchi (2011)). Some examples are presented in Figs. 1.7-1.10. Concretely, Figs. 1.7 and 1.8 illustrate respectively a centralized tight formation for homogeneous and heterogeneous MAS[6]. Figure 1.10 shows that a MAS reaches a tight formation from its given initial configuration (see Fig. 1.9) in a decentralized manner[7]. However, while converging toward the expected formation, the agents can collide between them or with some obstacles, leading to degrade seriously the mission safety. Therefore, formation control subject to anti-collision constraints becomes highly interesting today.

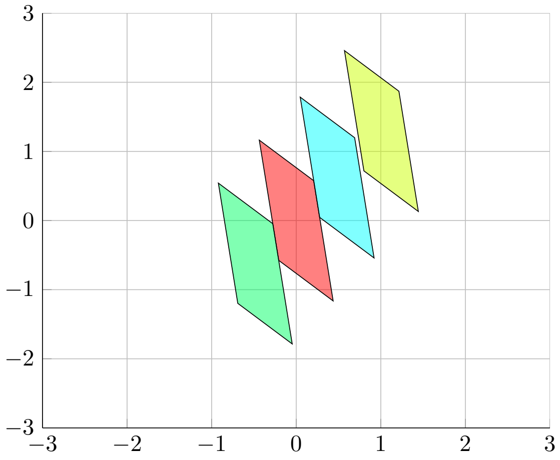

FIGURE 1.7: Homogeneous minimal configuration achieved by centralized approach.

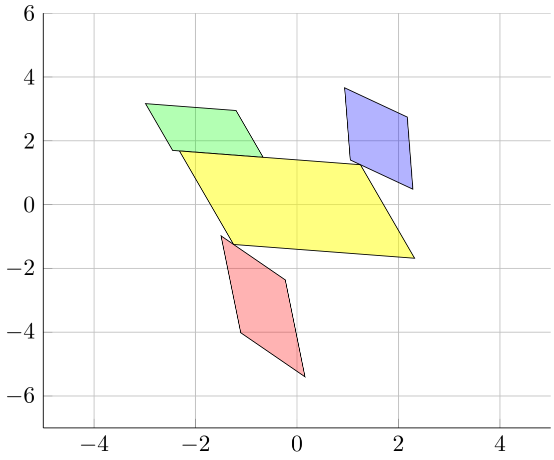

FIGURE 1.8: Heterogeneous minimal configuration achieved by centralized approach.

[6] Each polyhedron in the figures illustrating the formation represents an agent. The position of the agent is placed at the center of polyhedron and the shape of the polyhedron characterizes the geometry/safety zone around the agent. The construction of these polyhedrons with respect to the agents dynamics will be given in Chapter 2. A homogeneous MAS has a common polyhedral shape for all the agents. In case of heterogeneous MAS, each agent has its own polyhedral shape.

[7] Each agent is authorized to operate uniquely in a zone with respect to the position of its neighbors. In Fig. 1.9 and 1.10, these zones are represented by the large polyhedral region enclosing the agent's polyhedral contour. In the context of a leader-follower structure, these zones are built on the hyperplanes selected with respect to the leader's position, in order to separate the followers. The details of such a construction will be presented in Chapter 3.

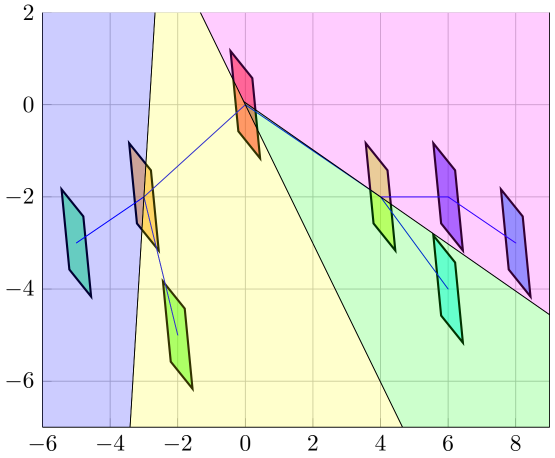

FIGURE 1.9: Partition of hierarchical leader-follower organization.

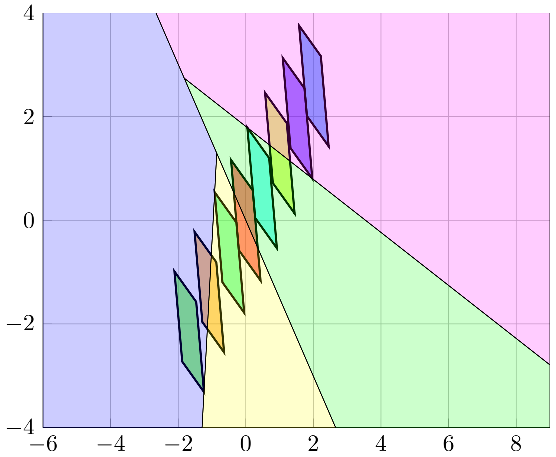

FIGURE 1.10: Homogeneous minimal configuration achieved by decentralized approach.

1.2.3 Deployment control problem

In a wide range of MAS applications, such as environmental/meteorological monitoring, surveillance, search and rescue operations planning or vehicle routing problems, the common principle is to let a group of cooperative mobile agents[8] (e.g. vehicle, robots, Unmanned Aerial Vehicle (UAV), Autonomous Underwater Vehicle (AUV) etc.) self-deploy within a target region (see Murray (2007), Tanner et al. (2007), Adib Yaghmaie et al. (2015)). The main objective of such cooperative task is to maximize the coverage quality subject to constraints. The most challenging problem aside the conventional formation control and tracking reference is that the final static configuration is not known a priori. Many works employed dynamic Voronoi partition as a conventional tool to drive the MAS close to a stationary configuration over a given bounded region. The concept of the "Dirichlet-Voronoi Diagram" introduced by Dirichlet (1850) and generalized after by Voronoï (1908), has numerous practical and theoretical applications in many research fields, including computer vision, computational geometry, computer-aided design, vehicles/robots routing and mobile sensor networks (Boissonnat (1984), Cortes et al. (2002), Schwager et al. (2009)). In principle, it describes a special partition of a topological space, which is equipped with a metric distance function, by means of an union of non-overlapping regions called *Dirichlet domains* or *Voronoi cells*. Each region is associated uniquely with an element from a given set of points, known as the set of Voronoi generators. The construction of the Voronoi diagram for different types of "distance" functions can be found in Aurenhammer (1991) and Okabe et al. (2009). According to the survey, the metric can differ from Euclidean norm such as Manhattan distances or other metric distance weighted by power constraints or minimum time to go (relating Zermelo navigation problem[9] (Sugihara (1992))).

[8] In general, all of these results are developed mainly for first-order linear continuous-time mobile system, i.e. $\dot{x}_i = u_i$, where x_i and u_i denote respectively the position and the speed of the i^{th} agent.

[9] Zermelo's problem (see Zermelo (1931)) can be illustrated via a simple example: Considering a boat initially resided in a river, find the control law to drive it towards a target by minimizing the traveling time subject to the disturbances issued by the current drift.

We emphasize that the partition evolves with the evolution of the agents and thus becomes time-varying. Hence, many works in the literature focus on stabilizing the Voronoi based deployment close to a configuration such that the coverage is maximal. The most well-known configuration is the *Centroidal Voronoi Configuration* (CVC) which is obtained when the position of each agent coincides with the centroid (or center of mass) of its associated Voronoi cell. The Lloyd's algorithm (see Lloyd (1982), like the closely related *k-means clustering* algorithm of MacQueen et al. (1967)) is widely used to obtain the CVC by moving the agents toward their centers of mass. This is equivalent to the decentralized control design principle. The principle of the Lloyd's algorithm is to repeat the following steps until a CVC is approximately obtained

- Step 1: Compute the Voronoi diagram of the generator points $x_1(k), \dots, x_N(k)$;
- Step 2: Calculate the centroid of each Voronoi cell;
- Step 3: Move each generator toward its associated centroid.

In Kwok and Martinez (2010), the authors propose a generalized power-weighted Voronoi partition and modify the Lloyd's algorithm to solve the power constrained deployment problem. The results in Song et al. (2014) reduce the step size in the Lloyd's algorithm and optimize the communication exchange. In the presence of large disturbance issued from the working environment such as strong wind/current drift, Sugihara (1992) and further Bakolas and Tsiotras (2013) propose a novel constructive method of dynamic Voronoi partition, called *Zermelo Voronoi diagram*. This allows an agent that resides within a domain of the partition at a given time to reach the generator associated with this domain faster than any other agent outside this domain at the same instant of time. The construction is based on the optimal control technique with respect to minimizing the time to go of the Zermelo navigation problem. Notable recent works in the field of mobile sensors networks are introduced by Cortes et al. (2002), Cortes et al. (2005) and other references therein. The authors present a novel decentralized optimal control which is distributed over the Delaunay graph of a dynamic Voronoi partition, to approach a centroidal Voronoi configuration by assuming that the working region is bounded and additionally the density distribution over this region is time-invariant. The conservation of mass law is used in combination with the LaSalle invariance principle (see Khalil and Grizzle (1996)) to prove the stability of the convergence into a CVC. Inspired by the last works, Moarref and Rodrigues (2014) extend the optimal decentralized control to deal with energy-efficient constraints. Other interesting works in the same direction is presented by Schwager et al. (2009) where different control strategies are given for multi-robots self-deployment, including respectively geometric, probabilistic, and potential field approaches. To the best of the author's knowledge, the safety of MAS withing the deployment procedure was not addressed in the literature.

1.3 Thesis orientation

As presented in the literature overview of dynamical MAS in control engineering, the notions and studies of FTC represent a well established branch in the control literature. Relatively few results exploit set-theoretics tools to solve the MAS control problem, but do not prevent/cover the faults occurrence to maintain the MAS functioning.

For this reason, the present manuscript fixes as objective to employ optimization based control synthesis combined with tools inherited from set theory in FDI-FTC design for dynamical MAS. In this way, it is possible to formulate FTC and FDI construction for MAS as constrained optimization problems. A time-invariant safety set around each agent is built by using the invariance notion linked to agent's dynamics. This region is considered a priori as the agent's safety set to avoid collision and additionally it can be used as geometrical residual to detect and isolate the faulty functioning to protect globally the formation. Moreover, set theory gives us the advantage to translate formation determination in terms of a constrained optimization problem. As a consequence, the optimal-formation solution is the fixed point in the appropriate extended space built with respect to the given constraints, thus ensuring the uniqueness from the geometrical point of view. In the presence of non-convex constraints (obstacle or collision avoidance), the fixed point becomes restricted by adversary constraints making thus the original equilibrium infeasible. The formation or extended fixed point of MAS can be decomposed into local fixed point determination for each agent, creating the appropriate framework for decentralized control design subject to convex constraints. Once the local structure of the control is defined within the decentralized design, the remaining problem is to examine the feasibility of the local control problem.

The following hypotheses are used in order to simplify the theoretical developments:

- the MAS is composed by mobile agents and the discrete-time agent's dynamics is assumed to be Linear Time Invariant (LTI), controllable and fully observable;
- the inter-communication is perfectly maintained and there is no loss or degradation of the exchanged information;
- the (additive) uncertainties/disturbances considered in the dynamical systems model are supposed to belong to a bounded set;
- the fault occurrence[10] is abrupt and additionally its propagation is not considered.

The second part of the thesis is dedicated to the stabilization of the Voronoi based deployment by using a decentralized control approach. The stability will be discussed in the sense of the Lyapunov function analysis. For this problem, it is important to assume that all the Voronoi

[10] Faults can manifest at sensor, actuator level or even due to non-cooperative behavior denying the formation requirement.

cells, and also the region within the agents deployment problem, are bounded and for computational reasons are considered to have polyhedral description. In this perspective, we focus in the current work on how to stabilize the self-deployment, while the anti-collision constraints are not considered in this topic but separately as a standalone problem. The metric distance function chosen is the Euclidean distance.

1.4 Thesis contribution

The contribution builds upon three main results set-based formation control (Prodan (2012)), set-theoretic approach for FTC of multi-sensor scheme based on bounding/invariant sets as a tool for FDI implementation (Stoican (2011)) and optimal control approach for decentralized coverage problem (Cortes et al. (2002)). With respect to the above results, we advocate a set-theoretic FTC philosophy based on the basic principle of decentralized control thus contributing towards the establishment of a safe plug-and-play MAS control and monitoring.

The linear feedback synthesis involving collision avoidance is following the principles exposed in Bitsoris and Olaru (2013) in terms of regulating the system state toward an equilibrium point lying on the boundary of a prohibited region, and complemented Prodan (2012) with respect to the centralized anti-collision MPC formation control. In this framework, our results in Nguyen et al. (2015b) relate to the anti-collision decentralized control exploring the well-known leader-follower structure. The main objective is to partition the working space into a collection of functioning zones. Each of these zones is associated with only one agent which will be kept operating in the interior of this zone, offering anti-collision guarantees. Further contribution consists in analyzing the restrictions of the method developed in Bitsoris and Olaru (2013), i.e. its feasibility limits and infeasible situations, and further propose two control strategies. The first one is based on the iterative construction of a set of contracted ellipsoids inside of the agent's functioning zone (Boyd and Vandenberghe (2004)), and the second approach employs the interpolation technique (Nguyen et al. (2013)).

Most of the works in MAS Voronoi based self-deployment focus on steering each agent to its *center of mass/centroid* (considered as the target point) thus the entire MAS approaches a Centroidal Voronoi Configuration according to the Lloyd's algorithm. Therefore CVC is an optimal configuration associated with the center of mass. Other target point can be selected leading to an alternative optimal configuration. For this reason, in Nguyen et al. (2016c) we first describe a basic approach based on the *Chebyshev center* as the target point. Its advantage is that the Chebyshev center can be expressed in geometric terms with respect to its associated Voronoi cell. However, when the Chebyshev center is not unique, the control strategy cannot lead to a stable configuration. In order to keep driving the agents into a stable static configuration, we propose a novel concept based on the computation of a so-called *general center* which leads to a unique center by deflation in the degenerated cases. Moreover, other choice of center is studied such as inner fixed point defined as convex combination of the vertices of the corresponding Voronoi cell (Nguyen and Maniu (2016)). This type of center is named *vertex interpolated center*. It is worth to mention that the agent dynamics in all of the previous works concerning centroidal Voronoi configuration is in continuous-time form. In the paper Nguyen et al. (2016a), we revisit the decentralized control to drive a MAS towards a CVC in the context of discrete-time dynamics.

In the recent paper Nguyen et al. (2015a) which follows the ideas presented in the previous developments Nguyen et al. (2014a), Nguyen et al. (2014b), set-theoretic tools are employed to design centralized FDI layer for homogeneous MAS. The a priori objective is to ensure the formation safety by supervising the inner agents functioning and additionally, detecting if some agents from exterior try to integrate the current formation. Moreover, we investigate the use of residual for the fault monitoring with respect to set-based FDI mechanism introduced in (Stoican and Olaru (2013)). In the context of MAS, residual is defined as the difference of behavior between the agent and its monitoring model. The residual's inclusion in healthy/faulty precomputed invariant set monitors the status of the corresponding agent. The reconfiguration step consists in eliminating the faulty agents and adjusting the formation control computation according to the remaining healthy agents.

We provide here the complete list of submitted/accepted publications.

Book chapter:

- **Minh Tri Nguyen**, Cristina Stoica Maniu, Sorin Olaru, and Alexandra Grancharova. Formation reconfiguration using Model Predictive Control techniques for Multi-Agent dynamical systems. In *Developments in model-based optimization and control*, pages 183-205. Springer, 2015.

Accepted conference papers:

- **Minh Tri Nguyen**, Cristina Stoica Maniu, Sorin Olaru, and Alexandra Grancharova. About formation reconfiguration for Multi-Agent dynamical systems. In *Automatics and Informatics'2014*, Sofia, Bulgaria, pages 141-144, 2014.
- **Minh Tri Nguyen**, Cristina Stoica Maniu, Sorin Olaru, and Alexandra Grancharova. Fault tolerant predictive control for Multi-Agent dynamical systems: formation reconfiguration using set-theoretic approach. In *Control, Decision and Information Technologies (IEEE CODIT)*, Metz, France, pages 417-422, 2014.
- **Minh Tri Nguyen**, Cristina Stoica Maniu and Sorin Olaru. Control invariant partition for heterogeneous Multi-Agent dynamical systems. In *System Theory, Control and Computing (IEEE ICSTCC)*, Cheile Gradistei, Romania, pages 354-359, 2015.
- **Minh Tri Nguyen**, Cristina Stoica Maniu, and Sorin Olaru. Decentralized constructive collision avoidance for Multi-Agent dynamical systems. In *European Control Conference (ECC)*, Aalborg, Denmark, pages 1526-1531, 2016.

- **Minh Tri Nguyen** and Cristina Stoica Maniu. Voronoi based decentralized coverage problem: from optimal control to Model Predictive Control. In *Mediterranean Conference on Control and Automation (IEEE MED)*, Athens, Greece, 2016.

- **Minh Tri Nguyen**, Luis Rodrigues, Cristina Stoica Maniu, and Sorin Olaru. Discretized optimal control approach for dynamic Multi-Agent decentralized coverage. In *IEEE Multi-Conference on Systems and Control*, Buenos Aires, Argentina, 2016.

Submitted paper:

- **Minh Tri Nguyen**, Cristina Stoica Maniu and Sorin Olaru. Optimization-based control for Multi-Agent deployment based on dynamic Voronoi partition. In *IFAC World Congress*, Toulouse, France, 2017.

1.5 Organization of the manuscript

The outline excluding Chapter 1 is organized as follows, with numerous illustrative examples presented throughout the manuscript.

Chapter 2 provides the mathematical background used throughout the manuscript, starting from the conventional description of the MAS dynamics and followed by a short introduction of set theory applied in control engineering. The set-based prerequisites are shortly recalled by means of basic set operations and invariance notions linked to discrete-time LTI equation dynamics. Subsequently, a set-theoretic approach for collision avoidance guarantee is given, starting from robust invariant agent's safety region construction. The basic method founded by Bitsoris and Olaru (2013) to compute the control action subject to collision avoidance requirement is reminded briefly. For the Voronoi based deployment problem, a basic definition of Voronoi tessellation for a bounded polyhedral space associated with the Euclidean norm is recalled and the computation of some inner center point of bounded convex polytope is introduced next, with some illustrative examples.

Chapter 3 presents the use of set-theoretic tools in decentralized formation control synthesis. A leader-follower structure is considered, leading to the followers functioning zone assignment decided by the leader. After, the local linear feedback control is computed such that the follower operates strictly inside its authorized zone, offering anti-collision guarantees. The main ingredients of local control are detailed, such as the local fixed point suitable for the leader position and the linear gain determination based on the method proposed by Bitsoris and Olaru (2013). The feasibility limits and infeasible situations are discussed and further we provide two control strategies exploring ellipsoidal construction and interpolation techniques to deal with both these cases.

Chapter 4 focuses on revisiting the decentralized stabilization of the Voronoi based deployment using an inner target driver in the context of discrete-time LTI dynamics. The main novelty is to consider the Chebyshev center as the inner target for each agent, leading to an optimization-based decentralized control design. A different choice of center related to the principle of space deflation is introduced in case where the Chebyshev center computation is impossible.

Chapter 5 aims to present the use of set-theoretics tools to design a centralized FDI layer for dynamical MAS. Two case-study will be presented, followed by a numerical simulation. The first one relates to faulty agents inside the formation. The second one focus on the case where some external agents try to integrate the current formation. The set-based FDI allows detecting and isolating these faulty agents to protect the current formation. After the faults certification, the formation containing the remaining healthy agents will be reconfigured using for instance centralized control techniques.

Chapter 6 completes the manuscript by means of concluding remarks and formulates several interesting perspectives.

Appendix A presents a novel decentralized framework for the Multi-Agent dynamical coverage problem subject to anti-collision constraints. The main contributions are related to the optimization-based decentralized control design by revisiting the optimal control approach and Model Predictive Control with respect to the vertex interpolated center chosen as the agent's inner target. The control objective is to obtain an optimal coverage by moving each agent towards its corresponding vertex interpolated center. This center is computed based on the vertices of the Voronoi cell characterizing the functioning zone. The proof of stability for both approaches is given with some illustrative examples.

Appendix B presents a novel decentralized framework to obtain a centroidal Voronoi configuration. The main contribution consists in extending the optimal decentralized control in Moarref and Rodrigues (2014) for discrete-time systems. The well-known Hamilton-Jacobi-Bellman equation is employed together with the mass conservation law in order to prove the stability proof and also the equivalence between the proposed discretized optimal control with the continuous-time optimal solution of Moarref and Rodrigues (2014).

Due to the laborious calculation and in order to simplify the presentation of this manuscript, we prefer to add the two Appendices, rather than formulating these results as chapters of the PhD thesis.

The manuscript organigram which describes the connexion between the chapters above is depicted in Fig. 1.11.

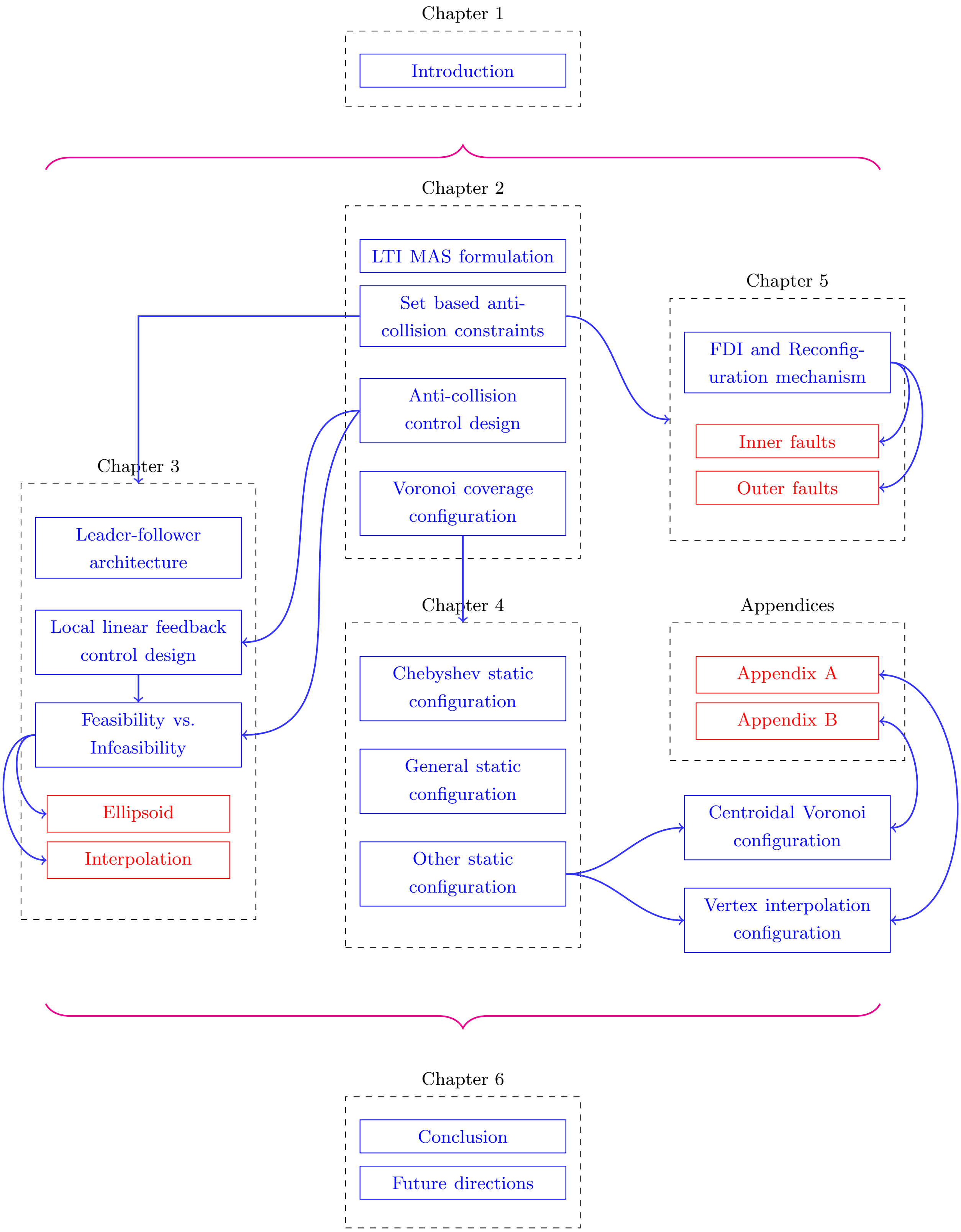
Chapter 1
Introduction
Chapter 2
LTI MAS formulation
Set based anti-collision constraints
Anti-collision control design
Voronoi coverage configuration
Chapter 5
FDI and Reconfiguration mechanism
Inner faults
Outer faults
Chapter 3
Leader-follower architecture
Local linear feedback control design
Feasibility vs. Infeasibility
Ellipsoid
Interpolation
Chapter 4
Chebyshev static configuration
General static configuration
Other static configuration
Appendices
Appendix A
Appendix B
Centroidal Voronoi configuration
Vertex interpolation configuration
Chapter 6
Conclusion
Future directions

FIGURE 1.11: Organigram of the manuscript.

Chapter 2

Set-theoretic notions for Multi-Agent system analysis and design

Since the early developments, set-theoretic framework in control engineering relies on the mathematical set theory and particularly on the Brunn-Minkowski algebra (see Schneider (2013) for more details) and exploit its results for dynamical systems described by ordinary differential equations or difference equations. The present chapter builds on applying set-theoretic methods in the context of safety guarantee for Multi-Agent systems composed of independent mobile agents evolving in a common environment. The class of agent dynamics considered of this thesis is described in Section 2.1.

The concept of safety relates to the collision avoidance during the agents operation. It is worth mentioning that collision avoidance is often one of the challenges in the context of the MAS operation, since the nature of anti-collision constraints is non-convex. The principle is to ensure that the safety regions which are considered around each agent do not intersect whenever they are translated along the trajectory of the agents. The use of set will be advocated to construct the agent safety region. Polyhedral sets (Motzkin et al. (1953)), star-shaped sets (Rubinov and Yagubov (1986)), zonotopic sets (Fukuda (2004)) and ellipsoidal sets (Kurzhanski and Varaiya (2000)) are among the most popular set families widely employed in many research and application fields. The strengths and weaknesses of each set family were already mentioned and commented in the literature.

Briefly, ellipsoidal sets are known due to their simple numerical representations which avoid the vertices/half-spaces enumeration. However this family of sets is often considered as conservative in shape representation and further not suitable for set operations. Star-shaped sets allow representing non-convex shape but the main complexity relates its non-convexity in the computation. Zonotopic sets represent a particular class of polytopes built on convex combination of a set of generator vectors with respect to a selected center. Similar to ellipsoidal sets, zonotopic sets exhibit symmetry but more effective in geometric representation with a similar computational resources.

We choose polyhedral set family as the principal set concept in the present manuscript for their convexity properties and flexibility with respect to the shape approximation. The complexity of the polyhedral sets need however a particular attention as their dual representation in terms of constraints or generators can be relatively high.

As an additional characteristic, the application of set theory in MAS control needs to remind the invariance notions allowing us to link the dynamical systems to static geometrical sets in the state-space (Blanchini and Miani (2007), Aubin (2009)). The results relating the algebraic invariance conditions and controlled invariance notions are mature for the case of polyhedral sets (Bitsoris (1988a)). As long as the dynamics of the agents are linear, ellipsoidal set will be recommended as valid option and employed on several instances of the next developments. The overview of our choices of sets family with the prerequisites of invariance notions will be provided in Section 2.2.

The choice of polyhedral set family is supported not only by the versatility of its dual (half-spaces/vertices) representation leading to computation advantages and straightforward implementation (construction of safety region, obstacle), but it is known to be a suitable formulation for the optimization-based control framework (construction of feasible sets). In Section 2.3, we will explain how to construct the agents safety regions in terms of invariant polyhedron according to the boundedness of additive uncertainties.

Solving an optimization problem subject to non-convex constraints leads to using Mixed-Integer Programming (MIP) (Jünger et al. (2009)). However, MIP is not always the best solution due to its computational complexity which can increase exponentially with the number of binary variables used in the problem formulation. Earl and D Andrea (2001) tried to reformulate the original decision problem in a simplified tractable MIP form but the complexity still remains significant. Recent results of Prodan et al. (2012) and further developed in Prodan et al. (2015) use hyperplane arrangements to reduce the number of non-convex feasible regions, thus avoiding the explosion of the number of binary variables.

In the present chapter the objective is to show that non-convexity can be avoided by solving the same optimization based control problem with respect to each feasible convex subset forming the non-convex region. This leads naturally to a decentralized approach. Note that it offers a suboptimal solution with respect to the MIP solution but provides significant advantages by exploiting the subsets convexity. Indeed, the decentralized solution consists in partitioning the working space into an union of non-overlapping convex zones and the control will be computed locally to keep the agent operating strictly within its corresponding zone. In most of deployment applications such as mobile sensing networks (Cortes et al. (2002)) or multi-robots deployment (Schwager et al. (2009)), the Voronoi tessellation is used to fulfill the partition objective, leading to the so-called Voronoi-based deployment. The ultimate objective of Voronoi-based deployment is to drive the agents towards an optimal coverage over a bounded working region. The concept of *optimal coverage* (see Cortes et al. (2002), Cortes et al. (2005)) for such applications received several interpretations, but the common feature is the stable static configuration over the working region. Mathematical definitions and the notions linked to optimal Voronoi-based coverage will be given in Section 2.3.

A conclusion will be provided at the end of the present section to recall briefly the main ideas.

2.1 Multi-Agent system description

Consider a Multi-Agent system $\boldsymbol{\Sigma}$ composed of N agents. Each agent has its own discrete-time linear time-invariant (LTI) dynamics

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) \\ y_i(k) = C_i x_i(k) \end{cases} \tag{2.1}$$

with the state vector $x_i \in \mathbb{R}^{n_i}$, the control input $u_i \in \mathbb{R}^{m_i}$, the constrained output $y_i \in \mathbb{R}^{p_i}$, $\forall i \in \mathcal{N} = \mathbb{N}_{[1,N]}$. The output space is a subspace of the state-space, such that $p_i \leq n_i, \forall i \in \mathcal{N}$. The matrices A_i, B_i and C_i have the appropriate dimensions.

Assumption 1. *The matrices (A_i, B_i) are controllable and (C_i, A_i) are observable, with $\forall i \in \mathcal{N}$.*

Each individual agent $i \in \mathcal{N}$ has access to its full state measurement as well as the output of the other agents. It is supposed that the output space is shared among the agents and that $p \leq n_i, \forall i \in \mathcal{N}$.

The dynamics of the entire system $\boldsymbol{\Sigma}$ can be further aggregated into a global model

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \end{cases} \tag{2.2}$$

where $\mathbf{x} = \left[x_1^\top \; x_2^\top \ldots x_N^\top\right]^\top \in \mathbb{R}^{\sum\limits_{i\in\mathcal{N}} n_i}$, $\mathbf{u} = \left[u_1^\top \; u_2^\top \ldots u_N^\top\right]^\top \in \mathbb{R}^{\sum\limits_{i\in\mathcal{N}} m_i}$ and $\mathbf{y} = \left[y_1^\top \; y_2^\top \ldots y_N^\top\right]^\top \in \mathbb{R}^{\sum\limits_{i\in\mathcal{N}} p_i}$ denote respectively the collective state, the control input vector and the constrained output vector of $\boldsymbol{\Sigma}$. The matrices $\mathbf{A} = \operatorname{diag}\{A_1, A_2, \ldots, A_N\}$, $\mathbf{B} = \operatorname{diag}\{B_1, B_2, \ldots, B_N\}$ and $\mathbf{C} = \operatorname{diag}\{C_1, C_2, \ldots, C_N\}$ collect all the matrices corresponding to each agent by juxtaposition.

The non-decoupling representation (2.2) makes the agents look apparently independent. In fact, the safety guarantee implies that at least each agent needs to know the position of its nearest neighbors in order to avoid the collision. This requires its corresponding control computation taking into account the position of other remaining agents, i.e.

$$u_i = u_i\big(x_1, x_2, \ldots, x_N\big), \text{ with } i \in \mathcal{N}$$

making the entire MAS interconnected by means of closed-loop behavior despite the non-decoupling open-loop representation (2.2).

In this way, the control input $\mathbf{u}$ of the dynamics (2.2) can be considered as a function of its agents states, i.e.

$$\mathbf{u} = \mathbf{u}(\mathbf{x})$$

Notice that the explicit form of this input $\mathbf{u}$ in terms of the state vector $\mathbf{x}$ depends on the network topology or precisely on the way to explore the feedback states of the agents via the control input u_i. The three structures of agent's control computation corresponding to three main network topologies detailed in Section 1.1.2 are given in Table 2.1. We use the notation $\mathcal{N}^i$ to denote the indices set of the neighbors of the i^{th} agent.

Table 2.1: Structures of agent's feedback control.

Agent's control input u_i	**Control structure**	**Illustrative figure**
• $u_i = u_i\big(x_1, x_2, \ldots, x_N\big)$, with $i \in \mathcal{N}$	Centralized	Fig. 1.2
• $u_i = u_i\big(x_j\big)$, with $i \in \mathcal{N}$ and $\forall j \in \mathcal{N}^i$	Distributed	Fig. 1.4
• $u_i = u_i\big(x_i\big)$, with $i \in \mathcal{N}$	Decentralized	Fig. 1.6

In the following, we will provide the set-theoretic notions and methods linked to dynamical system and further explain their applications in collision avoidance guarantee for Multi-Agent system.

2.2 Set-theoretic prerequisites

First let us concentrate on the basic notions related to the polyhedral and ellipsoidal set representations. Some fundamental set algebra will be also recalled in terms of mathematical description in Section 2.2.1 and the basic application of set in control theory will be mentioned further in Section 2.2.2.

2.2.1 Basic notions and set operations

We start by giving below two fundamental definitions related to the construction of polyhedra based on vertices enumeration and unidirectional rays.

Definition 2.1. The *convex hull* of a given finite set of vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_p\}$, with $v_i \in \mathbb{R}^n$ is defined as

$$\text{conv}\{\mathcal{V}\} = \left\{ x \in \mathbb{R}^n \middle| x = \sum_{i=1}^{p} \lambda_i v_i, \lambda_i \in \mathbb{R}_+, \sum_{i=1}^{p} \lambda_i = 1 \right\}$$

Definition 2.2. The *cone* of a given finite set of unidirectional rays $\mathcal{T} = \{t_1, t_2, \dots, t_q\}$, with $t_i \in \mathbb{R}^n$ is defined as

$$\text{cone}\{\mathcal{T}\} = \left\{ x \in \mathbb{R}^n \middle| x = \sum_{i=1}^{q} \gamma_i t_i, \gamma_i \in \mathbb{R}_+ \right\}$$

We provide next the basic definition of a polyhedron with its implicit mathematical representations.

Definition 2.3. A *polyhedron* $\mathcal{P} \in \mathbb{R}^n$ is defined as the combination of linear inequalities[1] (see Schrijver (1998)), i.e.

$$\mathcal{P} = \{x \in \mathbb{R}^n | Hx \leq \theta\} \tag{2.3}$$

with $H \in \mathbb{R}^{m \times n}$ and $\theta \in \mathbb{R}^m$.

From a geometrical point of view, the polyhedron $\mathcal{P}$ in (2.3) can be considered equivalent to the intersection of a finite number of closed half-spaces. Thus, such a form is named $\mathcal{H}$-representation. Moreover, if $\mathcal{P}$ is a bounded polyhedron, it can be defined as the convex hull of its vertices according to Definition 2.1 ($\mathcal{V}$-representation).

Two examples illustrating the convex hull and cone presentation are given in Figs. 2.1 and 2.3.

Example 2.1. Consider the polyhedron $\mathcal{P}$ in Fig. 2.1 built on the vertices. Its $\mathcal{H}$-representation is

$$\mathcal{P} = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 0.1374 & -0.5494 \\ 0.5345 & 0.2673 \\ 0 & 0.7071 \\ -0.5883 & 0.1961 \end{bmatrix} x \leq \begin{bmatrix} 0.8242 \\ 0.8018 \\ 0.7071 \\ 0.7845 \end{bmatrix} \right\}$$

and its $\mathcal{V}$-representation is

$$\mathcal{P} = \text{conv}\left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right\}$$

or equivalently

$$\mathcal{P} = \left\{ x \in \mathbb{R}^2 \middle| x = \lambda_1 \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2 \\ -1 \end{bmatrix} + \lambda_4 \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \lambda_i \in \mathbb{R}_+, \sum_{i \in \{1,2,3,4\}} \lambda_i = 1 \right\}$$

Example 2.2. Consider the cone $\mathcal{C}$ in Fig. 2.3 which is built on its two unidirectional rays $t_1 = \begin{bmatrix} 3 \\ 0.5 \end{bmatrix}$, $t_2 = \begin{bmatrix} 0.5 \\ 3 \end{bmatrix}$ i.e.

$$\mathcal{C} = \left\{ x \in \mathbb{R}^2 \middle| x = \gamma_1 \begin{bmatrix} 3 \\ 0.5 \end{bmatrix} + \gamma_2 \begin{bmatrix} 0.5 \\ 3 \end{bmatrix}, \gamma_1, \gamma_2 \in \mathbb{R}_+ \right\}$$

[1] In general, a polyhedron $\mathcal{P}$ is built by combining both linear inequalities and also equalities $\mathcal{P} = \{x \in \mathbb{R}^n | H_{in} x \leq \theta_{in}, H_{eq} x = \theta_{eq}\}$.

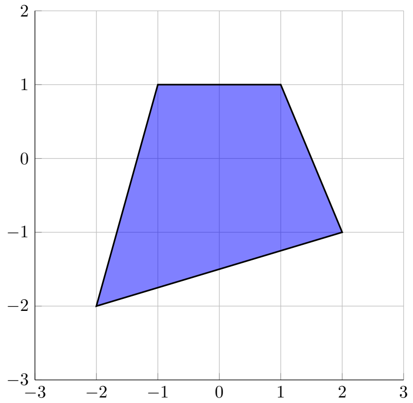

FIGURE 2.1: Convex hull of a set of given vertices v_i.

Its $\mathcal{H}$-representation is

$$\mathcal{C} = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 1 & -6 \\ -6 & 1 \end{bmatrix} x \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

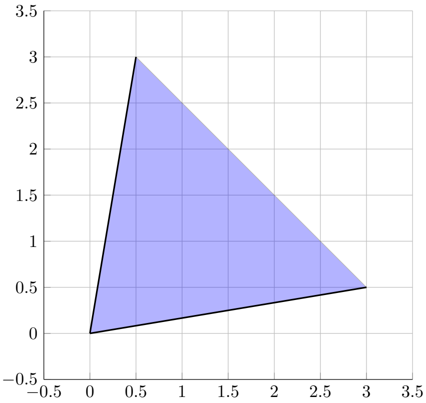

FIGURE 2.2: Cone.

Remark 2.1. A polyhedron can be represented as a linear combination of vertices and unidirectional rays according to two Definitions 2.1 and 2.2. Such representation is called the *dual Minkowski representation*[2] (see Motzkin et al. (1953)) by regrouping the concept of convex hull and cone in an unified representation, i.e.

$$\mathcal{P} = \left\{ x \in \mathbb{R}^n \middle| x = \sum_{i=1}^{p} \lambda_i v_i + \sum_{j=1}^{q} \gamma_j t_j, \lambda_i \in \mathbb{R}_+, \sum_{i=1}^{m} \lambda_i = 1, \gamma_j \in \mathbb{R}_+ \right\}$$

Remark 2.2. The dual Minkowski representation of a polyhedral cone having the $\mathcal{H}$-representation

$$\mathcal{C}(H, \theta) = \left\{ x \in \mathbb{R}^n | Hx \leq \theta, H \in \mathbb{R}^{m \times n}, \theta \in \mathbb{R}^m \right\}$$

[2] In general, with respect to the paper Schrijver (1998), a polyhedron is equivalent to a linear combination of vertices, unidirectional rays and also bidirectional rays. $\mathcal{P} = \left\{ x \in \mathbb{R}^n \middle| x = \sum_{i=1}^{p} \lambda_i v_i + \sum_{j=1}^{q} \gamma_j t_j + \sum_{k=1}^{r} \mu_k r_k, \lambda_i \in \mathbb{R}_+, \sum_{i=1}^{m} \lambda_i = 1, \gamma_j \in \mathbb{R}_+, \forall \mu_k \in \mathbb{R} \right\}$, with $v_i, t_i, r_i \in \mathbb{R}^n$. However, the interpretation of bidirectional rays is not considered in this present manuscript.

is

$$\mathcal{C}(H,\theta) = \left\{ x \in \mathbb{R}^n \middle| x = \lambda + \sum_{i=1}^{q} \gamma_i t_i, t_i \in \mathbb{R}^n, \lambda \in \mathbb{R}_+, \gamma_i \in \mathbb{R}_+ \right\}$$

Moreover, a cone becomes *proper cone* if $\theta = 0_{m,1}$, which implies $\lambda = 0$.

Example 2.3. Consider the unbounded polyhedron $\mathcal{P}$ in Fig. 2.3. Its $\mathcal{H}$-representation is

$$\mathcal{P} = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 1 & -6 \\ -6 & 1 \end{bmatrix} x \leq \begin{bmatrix} 15 \\ -20 \end{bmatrix} \right\}$$

and its dual Minkowski representation is

$$\mathcal{P} = \left\{ x \in \mathbb{R}^2 \middle| x = \begin{bmatrix} 3 \\ -2 \end{bmatrix} + \gamma_1 \begin{bmatrix} 3 \\ 0.5 \end{bmatrix} + \gamma_2 \begin{bmatrix} 0.5 \\ 3 \end{bmatrix}, \gamma_1, \gamma_2 \in \mathbb{R}_+ \right\}$$

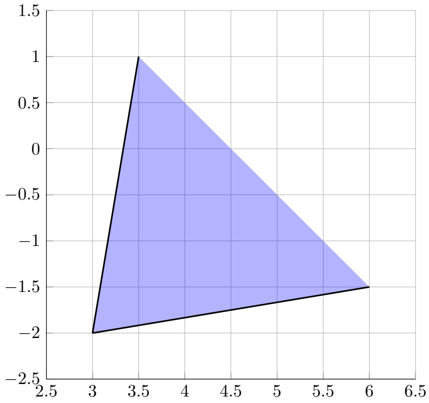

FIGURE 2.3: Unbounded polyhedron.

Beside polyhedral sets family, we employ also ellipsoidal set whose definition is given below.

Definition 2.4. An ellipsoid $\mathcal{E}(c,P,d) \in \mathbb{R}^n$ with the center $c \in \mathbb{R}^n$, the shape matrix $P \in \mathbb{R}^{n\times n}$ and the range $d \in \mathbb{R}$ is defined as $\mathcal{E}(c,P,d) = \{x \in \mathbb{R}^n | (x-c)^\top P(x-c) \leq d\}$, with $P = P^\top \succ 0$ and $d > 0$.

Example 2.4. In Fig. 2.4, we present three ellipsoids centered at the origin with different shape matrices P and ranges d.

- $\mathcal{E}(c_1,P_1,d_1)$: $c_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $d_1 = 0.4$
- $\mathcal{E}(c_2,P_2,d_2)$: $c_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $P_2 = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$, $d_2 = 2$
- $\mathcal{E}(c_3,P_3,d_3)$: $c_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $P_3 = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$, $d_3 = 2$

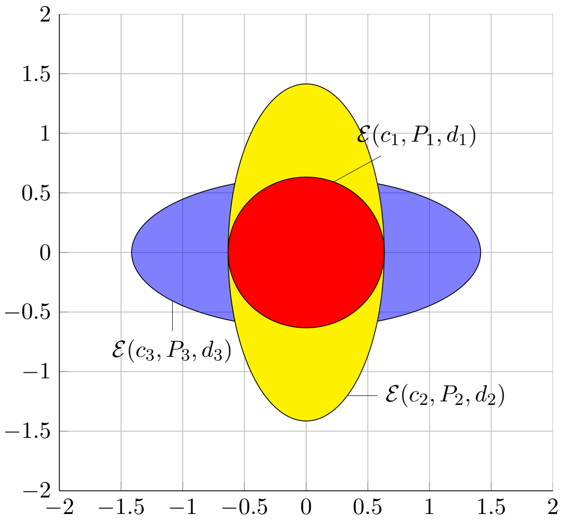

FIGURE 2.4: Ellipsoids centered at the origin.

Definition 2.5. Given two sets $\mathcal{A}$ and $\mathcal{B}$, their *Minkowski sum* is defined as

$$\mathcal{A} \oplus \mathcal{B} = \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}$$

Definition 2.6. Given two sets $\mathcal{A}$ and $\mathcal{B}$, their *Pontryagin difference* is defined as

$$\mathcal{A} \ominus \mathcal{B} = \{a \in \mathcal{A} | a + b \in \mathcal{A}, \forall b \in \mathcal{B}\}$$

Example 2.5. Consider two following polyhedron $\mathcal{A}$ and $\mathcal{B}$

$$\mathcal{A} = \left\{ x \in \mathbb{R}^2 \left| \begin{bmatrix} 0.1374 & -0.5494 \\ 0.5345 & 0.2673 \\ 0 & 0.7071 \\ -0.5883 & 0.1961 \end{bmatrix} x \le \begin{bmatrix} 0.8242 \\ 0.8018 \\ 0.7071 \\ 0.7845 \end{bmatrix} \right. \right\}$$

$$\mathcal{B} = \left\{ x \in \mathbb{R}^2 \left| \begin{bmatrix} 0 & -0.9806 \\ 0.8123 & 0.5415 \\ -0.8123 & 0.5415 \end{bmatrix} x \le \begin{bmatrix} 0.1961 \\ 0.2166 \\ 0.2166 \end{bmatrix} \right. \right\}$$

Their Minkowski sum set $\mathcal{A} \oplus \mathcal{B}$ and Pontryagin difference set $\mathcal{A} \ominus \mathcal{B}$ are given in Figs. 2.5

Definition 2.7. A λ*-scaled set* of a set $\mathcal{A} \in \mathbb{R}^n$ is defined as

$$\lambda\mathcal{A} = \{x \in \mathbb{R}^n | x = \lambda a, a \in \mathcal{A}\}$$

with $\lambda \in \mathbb{R}$ denoting the scale factor.

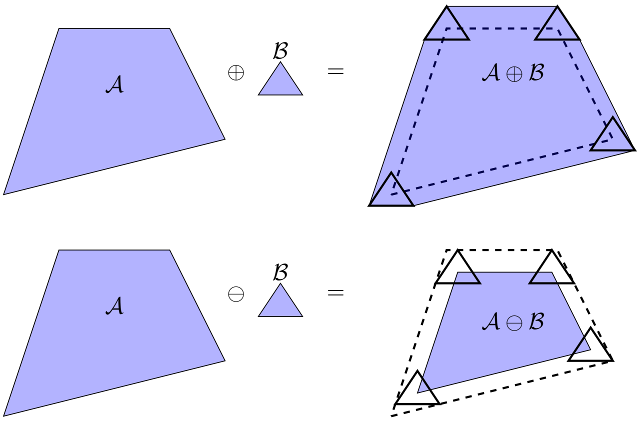

FIGURE 2.5: Minkowski sum $\mathcal{A} \oplus \mathcal{B}$ and Pontryagin difference $\mathcal{A} \ominus \mathcal{B}$ representations of two given sets $\mathcal{A}$ and $\mathcal{B}$.

Remark 2.3. Consider a polyhedron $\mathcal{A}$ having its $\mathcal{H}$-representation $\mathcal{A} = \{x \in \mathbb{R}^n | Hx \leq \theta\}$, the $\mathcal{H}$-representation of its λ-scaled set $\lambda\mathcal{A}$ is

$$\lambda\mathcal{A} = \{x \in \mathbb{R}^n | Hx \leq \lambda\theta\} \tag{2.4}$$

Remark 2.4. Consider a set $\mathcal{A}$. If the origin belongs to the strict interior of the set $\mathcal{A}$ then it belongs also to the strict interior of its λ-scaled set $\lambda\mathcal{A}$. In this case, if $\lambda \in \mathbb{R}_{(0,1)}$, then the set $\lambda\mathcal{A}$ is called λ-*contraction*.

Example 2.6. Consider the polyhedron $\mathcal{A}$ of Example 2.5. Its λ-scaled sets corresponding respectively to $\lambda = 0.5$ and $\lambda = 1.5$ are shown in Fig. 2.6.

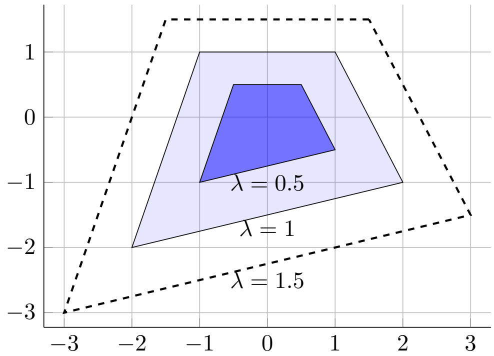

FIGURE 2.6: λ-scaled set of $\mathcal{A}$.

Remark 2.5. The multi-parametric toolbox MPT3 (see Herceg et al. (2013)) has been used to compute the solutions and plot the results of the illustrative examples related to the set operations.

We will present next the basic set approaches in control theory.

2.2.2 Sets in control theory

In this section we introduce the fundamental concepts related to dynamics and sets, starting from some basic definitions of invariance linked to the autonomous LTI dynamic equation. Note that a domain, or a set of points in the state-space is invariant with respect to a given dynamics if the future evolutions of its points are enclosed by this set.

Definition 2.8. (Blanchini and Miani (2007)) Consider an autonomous discrete-time LTI system $x(k+1) = Ax(k)$, where A is a Schur matrix[3]. A set $\mathcal{S}$ is called *positive invariant* (PI) for this system, if $x(k+1) \in \mathcal{S}$ for all $x \in \mathcal{S}$, which is equivalent to $A\mathcal{S} \subseteq \mathcal{S}$.

Definition 2.9. (Blanchini and Miani (2007)) Consider an autonomous LTI system $x(k+1) = Ax(k) + w(k)$, where A is a Schur matrix. A set $\mathcal{S}$ is called *robustly positive invariant* (RPI) for this system, if $x(k+1) \in \mathcal{S}$ for all $x(k) \in \mathcal{S}$ and $w(k) \in \mathcal{W}$, which is equivalent to $A\mathcal{S} \oplus \mathcal{W} \subseteq \mathcal{S}$.

The earlier work of Bitsoris (1988c) extends the concept of invariant set with respect to non-autonomous discrete-time LTI dynamics. Precisely, the invariance is obtained via the existence and interpretation of an exogenous input action making the considered set controlled invariant. It is recalled in the definition below.

Definition 2.10. (Bitsoris (1988c)) Given the discrete-time LTI system dynamics $x(k+1) = Ax(k) + Bu(k)$, with (A, B) controllable. A set $\mathcal{S} \in \mathbb{R}^n$ is *controlled invariant* with respect to this system, if for any $x \in \mathcal{S}$ there exists a control law u such that $Ax(k) + Bu(k) \in \mathcal{S}$. If the control has the linear feedback form $u(k) = Kx(k)$, then $\mathcal{S}$ is *linearly controlled invariant.*

In the present thesis, for any polyhedron in $\mathcal{H}$-representation, we will employ the algebraic invariance conditions proposed in Bitsoris (1988c) as a main criterion to check the invariance of a given polyhedral set. The following theorem summarizes these conditions.

Theorem 2.1. *(Bitsoris (1988b)) A set $\mathcal{S} = \{x \in \mathbb{R}^n | Gx \leq \theta\}$, with $G \in \mathbb{R}^{m \otimes n}$ and $\theta \in \mathbb{R}^m$, is positively invariant with respect to the discrete-time LTI system dynamics $x(k+1) = Ax(k)$ if and only if there exists a matrix $H \in \mathbb{R}^{n \times n}$ with non-negative elements, such that $GA = HG$ and $H\theta \leq \theta$.*

Beside the advantage of invariance certification, this theorem is useful for the control design due to its formulation as an optimization problem.

Concerning the construction of invariant sets, there are several methods in the literature but the complexity is their common drawback. A recent notable method is the work of Kofman et al. (2007) related to the construction of a RPI set for discrete-time LTI autonomous systems with bounded additive disturbances. This method reduces significantly the computation load by exploiting the matrix structural properties of the system's dynamics and allows us to off-line predict the shape of the RPI set. We recall the main results in the following lemma.

[3] The notion in Bhatia (2013) is used here, where a Schur matrix is a square matrix with real entries and with eigenvalues of absolute value less than one.

Lemma 2.2. *(Kofman et al. (2007)) Consider the system $x(k+1) = Ax(k) + w(k)$, with the matrix A assumed to be a Schur matrix and a non-negative vector $w(k)$ such that $|w(k)| \leq \bar{w}$, $\forall w(k) \in \mathcal{W} \subset \mathbb{R}^n$. Let $A = VJV^{-1}$ be the Jordan decomposition of A, with V, $J \in \mathbb{R}^{n\times n}$. Then the set*

$$\mathcal{S} = \{x \in \mathbb{R}^n : |V^{-1}x| \leq (I - |J|)^{-1}|V^{-1}\bar{w}|\} \tag{2.5}$$

is robustly invariant with respect to the system dynamics.

Example 2.7. Let us consider the system $x(k+1) = Ax(k)+w(k)$, with $A = \begin{bmatrix} 0.1589 & -0.0451 \\ 0.3112 & 0.5411 \end{bmatrix}$. The disturbance is bounded, i.e. $|w(k)| \leq \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$. Using Lemma 2.2, we obtain the RPI set $\mathcal{S}$

$$\mathcal{S} = \left\{ x \in \mathbb{R}^2 \left| \begin{bmatrix} -1.0374 & -0.1370 \\ 1.0374 & 1.1370 \\ 1.0374 & 0.1370 \\ -1.0374 & -1.1370 \end{bmatrix} x \leq \begin{bmatrix} 0.2936 \\ 0.8698 \\ 0.2936 \\ 0.8698 \end{bmatrix} \right. \right\}$$

corresponding to the Jordan decomposition $A = VJV^{-1}$, with $J = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.5 \end{bmatrix}$ and $V = \begin{bmatrix} -1.0961 & -0.1321 \\ 1 & 1 \end{bmatrix}$.

Briefly, we presented above the basic notions of set and its application in control theory. The next section will detail how to use Lemma 2.2 to construct the agent's safety region and how to apply set-theoretic notions to represent the collision avoidance constraints in the context of Multi-Agent system.

2.3 Set-based collision avoidance guarantee

The collision avoidance can be understood as keeping a minimal distance between any pair of agents. This leads to associate each agent with a safety region and hence the collision is avoided by ensuring that these regions are non-overlapping. The construction of a safety region based on Lemma 2.2 and with respect to the agent's dynamics stabilized at the origin will be provided in Section 2.3.1. Subsequently, we will explain in Section 2.3.2 how to formulate the anti-collision constraints for Multi-Agent system using the notion of safety regions.

2.3.1 Robust tube-based safety region of an agent

We consider discretized linear time-invariant dynamical systems. Bounded additive disturbances are considered, i.e.

$$\tilde{x}_i(k+1) = A_i\tilde{x}_i(k) + B_i\tilde{u}_i(k) + w_i(k),\ i \in \mathcal{N} \tag{2.6}$$

where $w_i(k) \in \mathcal{W}$ is the disturbance vector, $\tilde{x}_i \in \mathbb{R}^n$ and $\tilde{u}_i \in \mathbb{R}^m$ are the i^{th} agent's state and input vector, respectively. In the present work it is assumed that the set $\mathcal{W} \subset \mathbb{R}^n$ is bounded and contains the origin in its interior. If the robust control input vector $\tilde{u}_i$ in (2.6) accounts for a nominal control action u_i and a linear disturbance rejection term

$$\tilde{u}_i(k) = u_i(k) + K_i\left(\tilde{x}_i(k) - x_i(k)\right) \tag{2.7}$$

then, by denoting $e_i = \tilde{x}_i - x_i$ as the tracking error of the i^{th} agent, the following expression is obtained

$$e_i(k+1) = (A_i + B_iK_i)e_i(k) + w_i(k) \tag{2.8}$$

Applying Lemma 2.2 for the tracking error equation, a robustly positive invariant set $\mathcal{S}_i$ can be constructed [4]. This ensures that the tracking error $e_i(k) \in \mathcal{S}_i$ at each time instant if the initial condition satisfies $e_i(0) \in \mathcal{S}_i$. The set $\mathcal{S}_i$ is considered as a basic geometrical domain describing the safety region around the i^{th} agent. Furthermore, although the real state $\tilde{x}_i$ is unknown due to disturbances w_i, its trajectory is always bounded by the parameterized tube

$$\mathcal{S}(x_i(k)) = \{x_i(k)\} \oplus \mathcal{S}_i \tag{2.9}$$

Therefore, the nominal dynamics (2.1) together with its robust tube-based safety region are used to characterize the behavior of an agent.

Example 2.8. Let us consider the i^{th} agent of a MAS, which has the real dynamics (2.6) and the nominal dynamics (2.1), with $A_i = \begin{bmatrix} -0.2 & 0.5 \\ 0.2 & 0.71 \end{bmatrix}$ and $B_i = \begin{bmatrix} 0.71 \\ 0.22 \end{bmatrix}$. The disturbance is bounded, i.e. $|w(k)| \leq \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$. The feedback gain $K_i = \begin{bmatrix} -0.5055 & 0.7677 \end{bmatrix}$ of the control action (2.7) is obtained by using pole placement technique, with 0.2 and 0.5 chosen as stable poles. Lemma 2.2 can be employed to construct a RPI set with respect to the closed-loop dynamics (2.8) of the tracking error $e_i = \tilde{x}_i - x_i$ between the real state and the nominal state. As illustrated in Fig. 2.7, the real state $\tilde{x}_i$ (red line) is bounded at each time instant by a tube $\mathcal{S}(x_i)$ composed of its nominal state x_i (blue line) and its safety region $\mathcal{S}_i$ (polyhedron bounded by black line).

2.3.2 Collision avoidance based on safety regions

The collision avoidance constraint between two agents can be described via the non-overlapping of their safety regions, or in other words, their safety regions do not intersect, i.e.

$$\mathcal{S}(x_i) \cap \mathcal{S}(x_j) = \emptyset, \forall i \neq j$$

[4] This construction was introduced by Mayne et al. (2005) and applied for MAS in Prodan et al. (2011).

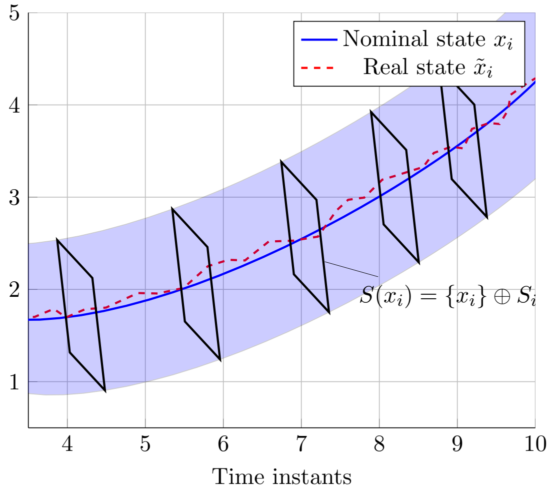

FIGURE 2.7: Robust tube-based safety region of an agent.

Using (2.9), this means that

$$(\{x_i\} \oplus \mathcal{S}_i) \cap (\{x_j\} \oplus \mathcal{S}_j) = \emptyset$$

or equivalently

$$x_j - x_i \notin (-\mathcal{S}_j) \oplus \mathcal{S}_i, \forall i \neq j$$

Example 2.9. Consider a MAS composed of $N = 3$ homogeneous agents having the same dynamics with the agent in Example 2.8. Applying the tube-based construction to enclose each agent nominal state x_i within a safety tube $\mathcal{S}(x_i)$, the anti-collision constraints while tracking a reference signal x_{ref} (red dashed line in Fig. 2.8) are then guaranteed by means of avoiding the intersection between the tubes $\mathcal{S}(x_i)$.

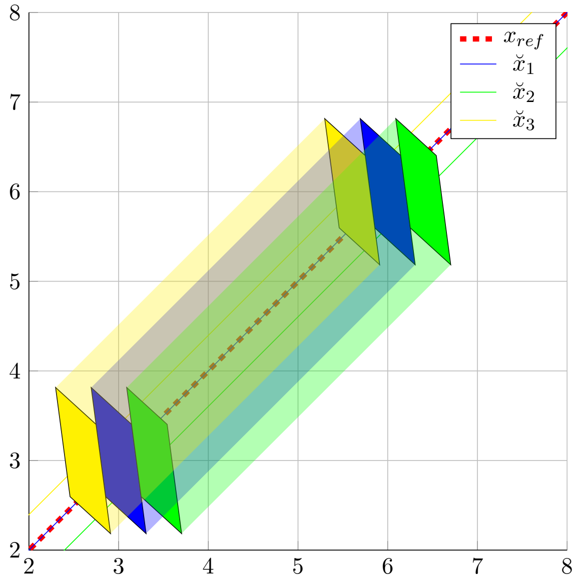

FIGURE 2.8: Formation of 3 homogeneous agents.

The constraint $x_j - x_i \notin (-\mathcal{S}_j) \oplus \mathcal{S}_i, \forall i \neq j$ can be translated into

$$x_j - x_i \in C((-\mathcal{S}_j) \oplus \mathcal{S}_i)$$

with the complement set $C((-\mathcal{S}_j) \oplus \mathcal{S}_i)$ being the union of the half-spaces[5] sharing uniquely the boundary with the prohibited set $(-\mathcal{S}_j) \oplus \mathcal{S}_i$, i.e.

$$C((-\mathcal{S}_j) \oplus \mathcal{S}_i) = \bigcup_{k \in \mathcal{N}_{\partial((-\mathcal{S}_j) \oplus \mathcal{S}_i)}} \mathcal{C}^k$$

We use $\mathcal{N}_{\partial((-\mathcal{S}_j) \oplus \mathcal{S}_i)}$ to denote the indices set of the boundaries of $(-\mathcal{S}_j) \oplus \mathcal{S}_i$.

Example 2.10. Consider two agents with their corresponding safety regions as shown in Fig. 2.9. The $\mathcal{V}$-representations of their safety region in $\mathbb{R}^2$ are respectively

$$\mathcal{S}_1 = \text{conv}\left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}$$

and

$$\mathcal{S}_2 = \text{conv}\left\{ \begin{bmatrix} -1.4 \\ -0.7 \end{bmatrix}, \begin{bmatrix} 1.4 \\ -0.7 \end{bmatrix}, \begin{bmatrix} 0 \\ 1.4 \end{bmatrix} \right\}$$

The collision avoidance constraint for these two agents are written as $x_2 - x_1 \notin (-\mathcal{S}_2) \oplus \mathcal{S}_1$

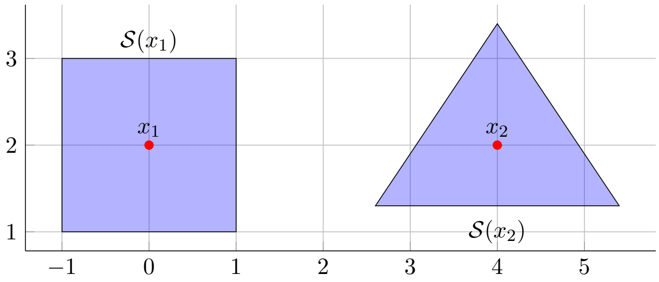

FIGURE 2.9: Two agents with their corresponding safety region.

where the prohibited set $(-\mathcal{S}_2) \oplus \mathcal{S}_1$ (colored in red in Fig. 2.10) has its $\mathcal{H}$-representation

$$(-\mathcal{S}_2) \oplus \mathcal{S}_1 = \left\{ x \in \mathbb{R}^2 \left| \begin{bmatrix} 0 & -0.3846 \\ 0.3491 & -0.2327 \\ 0.3846 & -0 \\ 0 & 0.5070 \\ -0.3846 & 0 \\ -0.3491 & -0.2327 \end{bmatrix} x \leq \begin{bmatrix} 0.9231 \\ 0.9077 \\ 0.9231 \\ 0.8619 \\ 0.9231 \\ 0.9077 \end{bmatrix} \right. \right\}$$

[5] This way of decomposition is not unique, i.e. the complement set $C((-\mathcal{S}_j) \oplus \mathcal{S}_i)$ can be described as the union of cones.

The anti-collision constraint $x_2 - x_1 \notin (-\mathcal{S}_2) \oplus \mathcal{S}_1$ can be translated into $x_2 - x_1 \in C((-\mathcal{S}_2) \oplus \mathcal{S}_1)$. This non-convex complement set $C((-\mathcal{S}_2) \oplus \mathcal{S}_1)$ is covered by the blue colored half-spaces in Fig. 2.10.

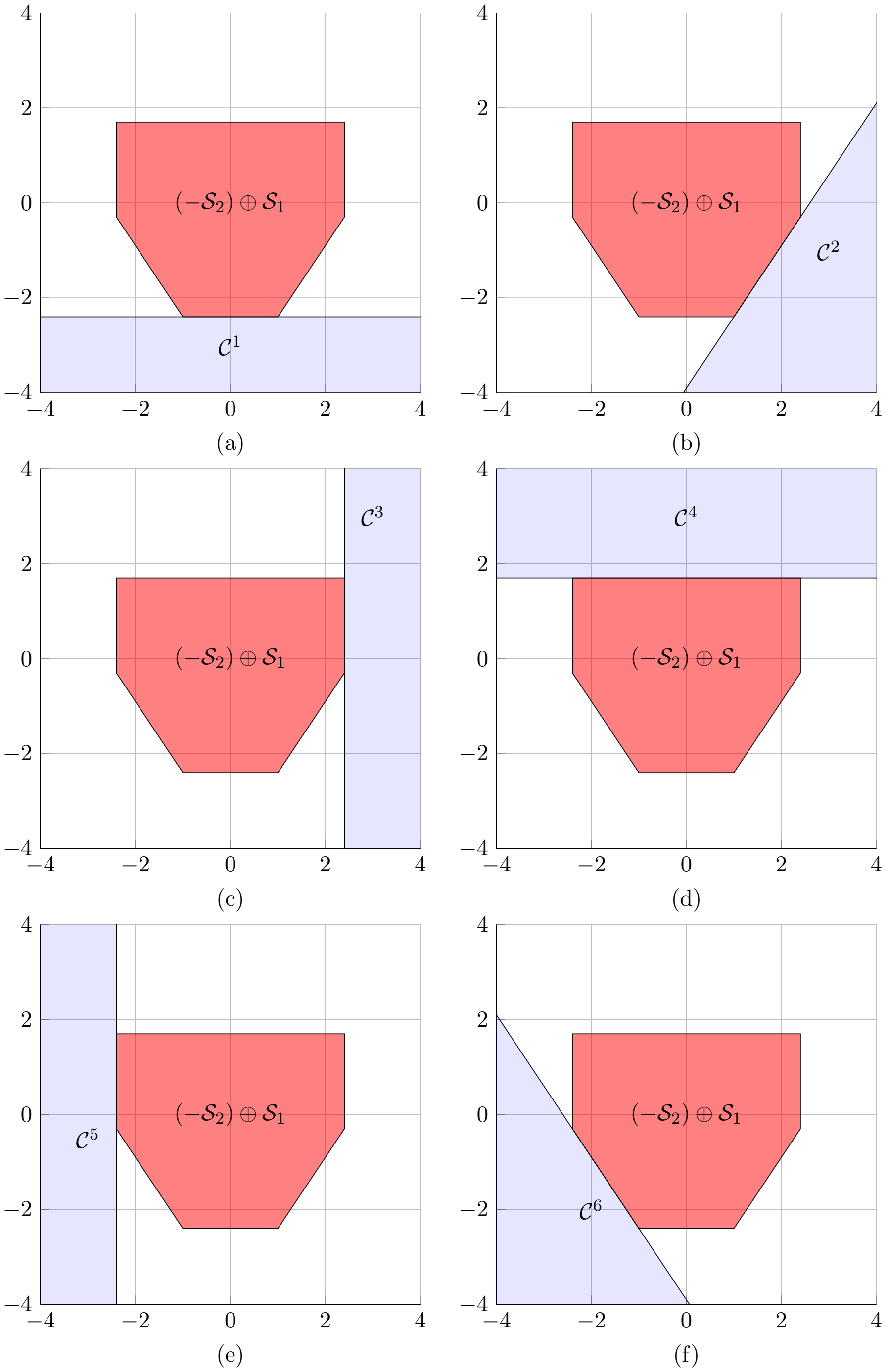

FIGURE 2.10: Prohibited anti-collision set $(-\mathcal{S}_2) \oplus \mathcal{S}_1$ (red) and its complement set $C(-\mathcal{S}_2 \oplus \mathcal{S}_1)$ (blue).

Each half-spaces has its own $\mathcal{H}$-representation derived from $x_2 - x_1 \notin (-\mathcal{S}_2) \oplus \mathcal{S}_1$:

- Fig. 2.10 (a) $\mathcal{C}^1 = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 0 & -0.3846 \end{bmatrix} x \geq 0.9231 \right\}$

- Fig. 2.10 (b) $\mathcal{C}^2 = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 0.3491 & -0.2327 \end{bmatrix} x \geq 0.9077 \right\}$
- Fig. 2.10 (c) $\mathcal{C}^3 = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 0.3846 & -0 \end{bmatrix} x \geq 0.9231 \right\}$
- Fig. 2.10 (d) $\mathcal{C}^4 = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 0 & 0.5070 \end{bmatrix} x \geq 0.8619 \right\}$
- Fig. 2.10 (e) $\mathcal{C}^5 = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} -0.3846 & 0 \end{bmatrix} x \geq 0.9231 \right\}$
- Fig. 2.10 (f) $\mathcal{C}^6 = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} -0.3491 & -0.2327 \end{bmatrix} x \geq 0.9077 \right\}$

Briefly, in the present section, we presented the main set-theoretic ingredients necessary to design formation control subject to anti-collision constraints. In particular, we provided the construction of the safety region around each agents state based on Lemma 2.2 combined with a linear feedback control (2.7) taking into account the difference between the nominal state and the real state of the agent. The anti-collision constraints are formulated by means of the exclusion between the safety regions and thus emphasize their non-convex nature. We will provide our proposed control design subject to these constraints in Chapter 3.

In the following, we will mention the prerequisites related to safe deployment of Multi-Agent system, which is one of the main topics discussed in this thesis beside the formation control subject to anti-collision constraints. The kind of deployment studied is Voronoi-based deployment.

2.4 Voronoi-based deployment

This section provides the basic mathematical elements for the Voronoi-based deployment formulation, starting from the definition of the classical Voronoi tessellation (see Voronoï (1908)) and the generalization proposed by Dirichlet (1850) in Section 2.4.1. We will emphasize also in some cases, the superposition of agents outputs do not allow using classical Voronoi formulation to construct the partition. These cases are known as singularities of Voronoi partition. An algorithm is provided in Section 2.4.1 to cover these singularities. Moreover, the remaining subsections will formulate the Voronoi-based optimal configuration (in Section 2.4.2) and further present several types of optimal configuration associated to the inner points chosen as the tessellation generators. We will introduce the basic definitions of the chosen inner points such as the center of mass in Section 2.4.2.1 (see Cortes et al. (2002)) and the Chebyshev center in Section 2.4.2.2 (see Boyd and Vandenberghe (2004)) with their corresponding computation framework and optimal configuration. Notice that the center of mass with its corresponding centroidal Voronoi partition is a well-known problem recalled usually in the literature (see Cortes et al. (2002), Cortes et al. (2005), Schwager et al. (2009), Moarref and Rodrigues (2014) and Nguyen et al. (2016a)). Beside this, the Chebyshev center can also be employed to find the optimal Voronoi-based coverage (see Nguyen et al. (2016c)). Its main advantage comes from a simplified computation based on set-theoretic operations.

2.4.1 Conventional Voronoi partition

We introduce next the *neighborhood* corresponding to an agent output y_i when this agent is part of the tuple $(y_1, y_2, \dots, y_N) \in \mathcal{W}^N$. A partition of the working space $\mathbb{V}(y_1, \dots, y_N)$ needs to be computed, by decomposing $\mathcal{W}$ into a union of non-overlapping sets

$$\mathcal{W} = \mathbb{V}(y_1, \dots, y_N) = \bigcup_{i=1}^{N} \mathbb{V}_i, \quad \text{with } \mathbb{V}_i \cap \mathbb{V}_j = \emptyset, \ \forall i, j \in \mathcal{N} \tag{2.10}$$

A natural mathematical definition of such a decomposition is provided by the Voronoi partition, which characterizes the *neighborhood* $\mathbb{V}_i(y_i)$ as

$$\mathbb{V}_i = \{y \in \mathcal{W} | \|y_i - y\| \leq \|y_j - y\|, \forall j \neq i\} \tag{2.11}$$

From this definition, it follows that $\|y_i - y\|^2 \leq \|y_j - y\|^2$ yielding $(y_i - y)^\top (y_i - y) \leq (y_j - y)^\top (y_j - y)$. This leads to $2(y_j - y_i)^\top y \leq \|y_j\|^2 - \|y_i\|^2$ and thus allows to obtain the $\mathcal{H}$-representation of $\mathbb{V}_i$, i.e.

$$\mathbb{V}_i = \{y \in \mathcal{W} | 2(y_j - y_i)^\top y \leq \|y_j\|^2 - \|y_i\|^2, \forall j \neq i\} \tag{2.12}$$

It is worth mentioning that each set $\mathbb{V}_i$ is a polytope as a consequence of the boundedness of $\mathcal{W}$ and the structure of the constraints in (2.11).

The cardinality remains constant in (2.10), as well as the structure of the constraints in (2.12), and thus each point is associated to a set called *neighborhood* within the partition $y_i \leftrightarrow \mathbb{V}_i$.

Example 2.11. Let us consider a group of 4 agents deployed over a bounded region

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} -10 \\ -10 \end{bmatrix}, \begin{bmatrix} 15 \\ -10 \end{bmatrix}, \begin{bmatrix} 15 \\ 10 \end{bmatrix}, \begin{bmatrix} -10 \\ 10 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

The positions of the agents are respectively $x_1 = \begin{bmatrix} -2 \\ 4 \end{bmatrix}$, $x_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$, $x_3 = \begin{bmatrix} -7 \\ -6 \end{bmatrix}$ and $x_4 = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$. The Voronoi partition corresponding to these agents positions are shown in Fig. 2.11.

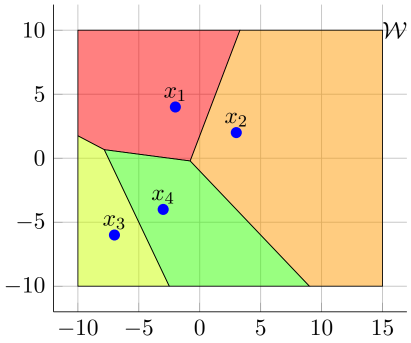

FIGURE 2.11: Voronoi partition for a group of 4 given points in $\mathbb{R}^2$.

Remark 2.6. The use of the Voronoi partition based on (2.12) for the Multi-Agent system $\mathbf{\Sigma}$ builds on the definition of the Voronoi cell which handles the difference $\|y_j - y_i\|$. Note that the superposition of agents outputs in $\mathbf{\Sigma}$ leads to a singularity on the half-space definition in (2.12) and calls for a generalization of the Voronoi partition.

The principle of the partition of the output space remains however valid, the only problem being related to the singularity in (2.12). In the present framework, such singularities are handled within the following algorithm.

Algorithm 1: Compute Voronoi partition corresponding to the current agents outputs

```
Input  : Agents outputs $y_1, y_2, \ldots, y_N$
Output: Voronoi partition $\mathbb{V} = \mathbb{V}(y_1, y_2, \ldots, y_N)$
1 - Find the distinct positions between the outputs $y_1, y_2, \ldots, y_N$ and store the multiplicity of
    each point (the number of agents sharing the same position);
2 - Build the Voronoi partition for the distinct positions;
3 - Replace each cell corresponding to points with multiplicity higher than 1, by a conic
    sub-partitioning via a number of hyperplanes (equal to the number of agents) passing through
    their common position.
```

Remark 2.7. Notice that the step 3 of Algorithm 1 is trivial in dimension $p \geq 2$ where the choice of the sub-partitioning in r cells can follow a partition of $\mathbb{R}^2$ in cones with angles of $\dfrac{2\pi}{r}$.

Remark 2.8. The case $p = 1$ represents singularity with respect to the above space partition. The reason is that the bounded working region in this case has the form of a segment. Therefore, the sub-partitioning can be fulfilled by dividing the working region into a set of segments such that the number of segments equals the multiplicity. It is worth to mention that the generator point may not be included inside the generated segments.

In the sequel, we provide some examples to illustrate respectively the cases of singularities in $\mathbb{R}$, $\mathbb{R}^2$ and $\mathbb{R}^3$.

Example 2.12. In Fig. 2.12, the red segments represent the working region $\mathcal{W} = \{x \in \mathbb{R} | -40 \leq x \leq 60\}$. The blue point $y_i = 52$ is the agent position. If the singularity is higher than 1, which means the superposition of more than one agent in this blue point. The partition is realized to obtain a set of equal segments corresponding to the level of multiplicity. Note that the agents may not be included inside the generated segments, which is coherent with Remark 2.8. We show in Fig. 2.12 respectively the partition of $\mathcal{W}$ into 2, 3 and 4 equal segments corresponding to the number of agents superposed.

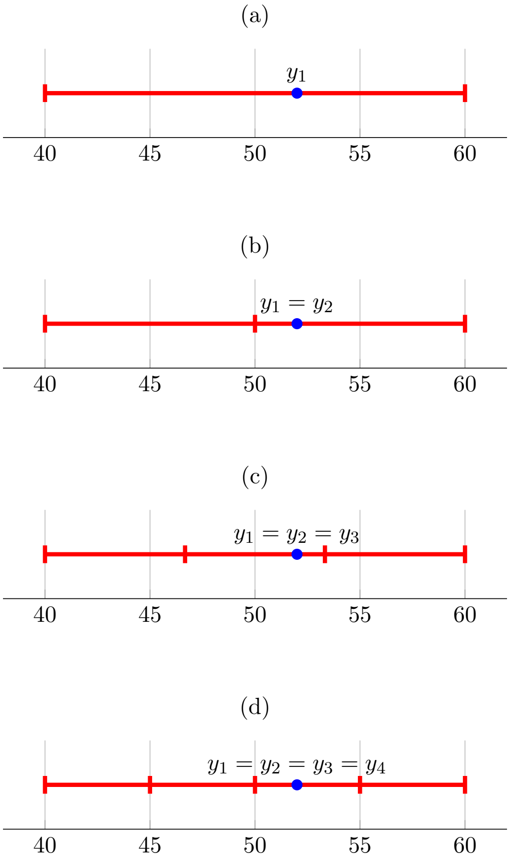

FIGURE 2.12: Singularity cases and Voronoi partition in $\mathbb{R}$.

Cases	Multiplicity	Number of equal segments
Fig. 2.12 (a)	1	1
Fig. 2.12 (b)	2	2
Fig. 2.12 (c)	3	3
Fig. 2.12 (d)	4	4

Example 2.13. Four examples illustrating the singularity cases of Voronoi partition in $\mathbb{R}^2$ are shown in Fig. 2.13. In case (a), obviously there is no superposition in case of one unique agent thus the singularity is avoided. In the three remaining cases, the singularity is caused by the superposition of the agents positions at $y_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and the partition is obtained by following Remark 2.7.

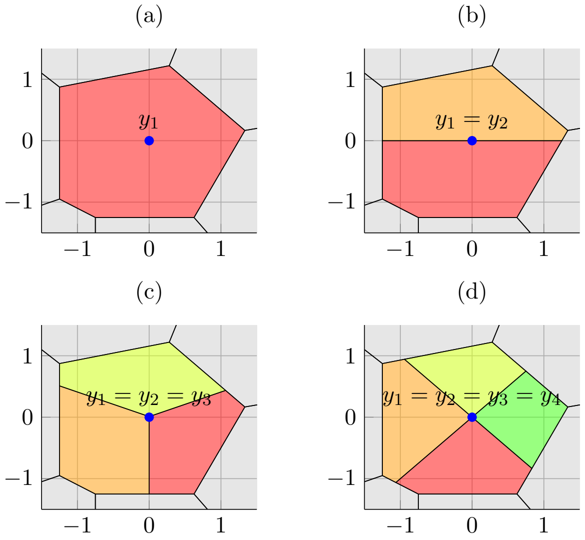

FIGURE 2.13: Singularity cases and Voronoi partition in $\mathbb{R}^2$.

Cases	Multiplicity	Number of cells	Angle
Fig. 2.13 (a)	1	1	$\frac{2\pi}{1}$
Fig. 2.13 (b)	2	2	$\frac{2\pi}{2}$
Fig. 2.13 (c)	3	3	$\frac{2\pi}{3}$
Fig. 2.13 (d)	4	4	$\frac{2\pi}{4}$

Example 2.14. Fig. 2.14 shows the Voronoi partition in $\mathbb{R}^3$ of a bounded region $\mathcal{W}$ i.e.

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ 80 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 80 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 40 \end{bmatrix}, \begin{bmatrix} 100 \\ 0 \\ 40 \end{bmatrix}, \begin{bmatrix} 100 \\ 80 \\ 40 \end{bmatrix}, \begin{bmatrix} 0 \\ 80 \\ 40 \end{bmatrix} \right\} \subset \mathbb{R}^3$$

In Fig. 2.14 (a) we show the case when there is no singularity in $\mathbb{R}^3$ because the multiplicity is one. In Fig. 2.14 (b), 4 agents are present at the position $y = \begin{bmatrix} 50 \\ 40 \\ 25 \end{bmatrix}$. According to Remark 2.7, the sub-partitioning is done in the sub-dimensional $\mathbb{R}^2$ thus we get 4 non-overlapping cells as similar to Example 2.13. These cells are further translated reversely into $\mathbb{R}^3$.

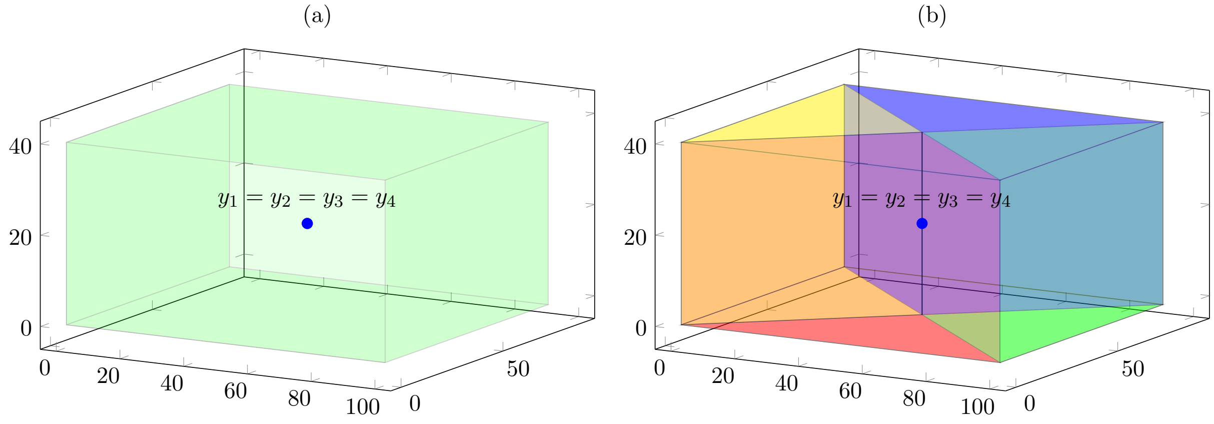

FIGURE 2.14: Singularity cases and Voronoi partition in $\mathbb{R}^3$.

Cases	Multiplicity	Number of cells in $\mathbb{R}^2$	Angle
Fig. 2.14 (a)	1	1	$\frac{2\pi}{1}$
Fig. 2.14 (b)	4	4	$\frac{2\pi}{4}$

In conclusion, we recalled in this section the basic formulation of Voronoi partition. Its limitations in case of superposition of the agents outputs were provided with our algorithm to cover these limitations. We will formulate next the application of Voronoi partition in the context of Multi-Agent systems deployment.

2.4.2 Voronoi based optimal configuration

Let us consider the Multi-Agent system $\boldsymbol{\Sigma}$ in (2.2). Using its available output measurement (which satisfies the assumption that $y_i(k) \in \mathcal{W}$) at the time instant k, the geometric formulation (2.10) leads to a *time-varying Voronoi partition* (TVV)

$$\mathbb{V}(y_1(k), \ldots, y_N(k)) = \bigcup_{i=1}^{N} \mathbb{V}_i(k) \tag{2.13}$$

Given this static notion, the goal from the control design point of view will be to stabilize a time-varying Voronoi partition or more specifically, maximize the coverage over the deployed region by driving the agents towards an optimal configuration not predetermined. The effective developments will be detailed in Chapter 4 but we provide here the definition of the Voronoi-based optimal configuration for the Multi-Agent system (2.2)

Definition 2.11. A Voronoi-based configuration of dynamical agents is optimal if the output $y_i(k)$ coincides with the selected inner target (equilibrium) point $\bar{y}_i(k)$ inside its corresponding Voronoi cell.

Remark 2.9. The formulation Voronoi-based can be extended to ”Generalized Voronoi partitions” which allow the target points to be outside the cell itself. However, this theoretical extension has not been considered in the present work due to the lack of interest for such a pathological distribution of the optimal configuration.

According to Definition 2.11, the type of optimal configuration is not unique but strictly depends on the choice of the inner target point. In the next subsections, we will present the computation of some selected inner target point studied in the thesis such as the center of mass in Section 2.4.2.1 and and the Chebyshev center in Section 2.4.2.2 and also provide their corresponding Voronoi based optimal configurations. We give also Examples 2.16 and 2.18 to illustrate these optimal configurations and further detail their computation in Chapter 4.

2.4.2.1 Centroidal Voronoi configuration based on the center of mass

The *mass* $M_{\mathbb{V}}$ and the *center of mass* $C_{M_{\mathbb{V}}}$ (so called the *centroid*) of a given bounded convex polyhedron $\mathbb{V} \subset \mathcal{W}$ are respectively defined as detailed in Moarref and Rodrigues (2014)

$$M_{\mathbb{V}} = \int_{\mathbb{V}} \phi(q) dq \tag{2.14}$$

$$C_{M_{\mathbb{V}}} = \frac{\displaystyle\int_{\mathbb{V}} q\phi(q) dq}{\displaystyle\int_{\mathbb{V}} \phi(q) dq} \tag{2.15}$$

with the density function $\phi : \mathcal{W} \to \mathbb{R}_+$ denoting the priority of coverage at a point $q \in \mathcal{W}$.

It is worth to mention that the equations (2.14) and (2.15) require a lot of computational resources because their complexity increases with the dimension of the state-space and the kind of density function ϕ. Moreover, to the best of our knowledge, there is no explicit formulation to compute the mass and the center of mass with a general density function[6] ϕ. We show a bounded polyhedron with its mass and center of mass in Example 2.15.

Example 2.15. Consider a bounded polyhedron shown in Fig. 2.15

$$\mathbb{V} = \text{conv}\left\{ \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \begin{bmatrix} 4 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

[6] The explicit formulations to compute the mass and the center of mass of a closed polyhedron in $\mathbb{R}^2$ with a uniform density function, i.e. $\phi(q) = 1, \forall q$ are introduced by Cortes et al. (2002).

The density function is uniform, i.e. $\phi(q) = 1, \forall q$.

Using the equations (2.14) and (2.15), the mass and the center of mass (yellow point) are respectively $M_{\mathbb{V}} = 13.5277$ and $C_{M_{\mathbb{V}}} = \begin{bmatrix} 0.3338 \\ -0.6664 \end{bmatrix}$.

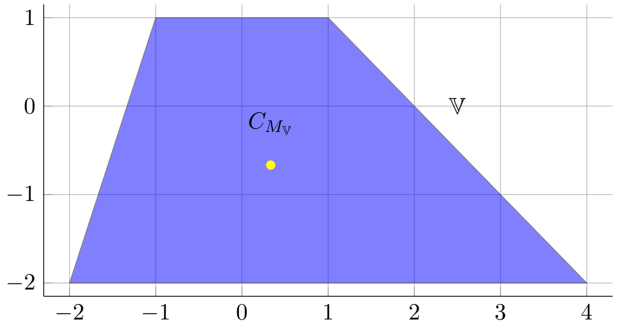

FIGURE 2.15: Mass $M_{\mathbb{V}}$ and center of mass $C_{M_{\mathbb{V}}}$ of polyhedron $\mathbb{V}$ with the uniform density function.

According to two formulations (2.14)-(2.15) used to compute the mass and the center of mass of a given polyhedron, we can apply these results for a Multi-Agent system deployment after each time of partition. Therefore, the Voronoi-based optimal configuration corresponding to the center of mass chosen as inner target point is defined in Definition 2.12. In Example 2.16, we show the CVC over a region $\mathcal{W}$ where a MAS composed of $N = 7$ agents is deployed.

Definition 2.12. A *Centroidal Voronoi Configuration* of MAS (2.2) is defined as a configuration where each agent observed output y_i coincides with the center of mass of its corresponding Voronoi cell, i.e. $y_i = C_{M_{\mathbb{V}_i}}, \forall i \in \mathcal{N}$.

Example 2.16. Consider a MAS composed of $N = 7$ agents. These agents are deployed within a bounded region

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 6 \end{bmatrix}, \begin{bmatrix} 0 \\ 6 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

A Centroidal Voronoi configuration for this MAS is shown in Fig. 2.16 where the position of each agent (denoted by blue circle) coincides with its center of mass (denoted by red dot).

The positions of the agents are respectively $y_1 = C_{M_{\mathbb{V}_1}} = \begin{bmatrix} 3.224 \\ 3.135 \end{bmatrix}$, $y_2 = C_{M_{\mathbb{V}_2}} = \begin{bmatrix} 4.738 \\ 4.913 \end{bmatrix}$, $y_3 = C_{M_{\mathbb{V}_3}} = \begin{bmatrix} 5.013 \\ 2.254 \end{bmatrix}$, $y_4 = C_{M_{\mathbb{V}_4}} = \begin{bmatrix} 3.553 \\ 0.644 \end{bmatrix}$, $y_5 = C_{M_{\mathbb{V}_5}} = \begin{bmatrix} 1.303 \\ 1.11 \end{bmatrix}$, $y_6 = C_{M_{\mathbb{V}_6}} = \begin{bmatrix} 0.921 \\ 3.354 \end{bmatrix}$ and $y_7 = C_{M_{\mathbb{V}_7}} = \begin{bmatrix} 1.707 \\ 5.167 \end{bmatrix}$.

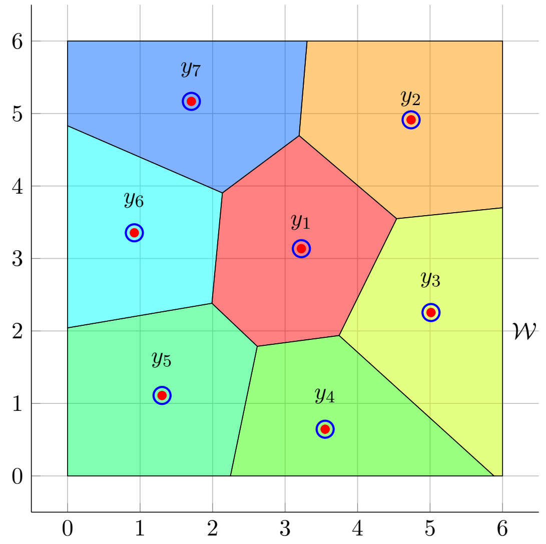

FIGURE 2.16: Centroidal Voronoi configuration over $\mathcal{W}$ of $N = 7$ agents.

2.4.2.2 Chebyshev static configuration based on the Chebyshev center

Consider a bounded convex polyhedron $\mathbb{V} = \left\{x \in \mathbb{R}^n | a_i^\top x \leq b_i,\ i \in \mathbb{N}_{[1,m]}\right\}$. The Chebyshev center $\bar{x}$ of $\mathbb{V}$ is defined as the center of the largest ball $\mathbb{B} = \{x \in \mathbb{R}^n | \, \|x - \bar{x}\| \leq r\}$ included in $\mathbb{V}$, with r denoting its radius. The values of $(\bar{x}, r)$ are obtained by solving this optimization problem:

$$\max r$$
$$\text{s.t:} \begin{cases} a_i^T \bar{x} + \|a_i\| r \leq b_i,\ \forall i \in \mathbb{N}_{[1,m]} \\ r \geq 0 \end{cases} \tag{2.16}$$

By exploiting the Karush-Kuhn-Tucker condition (see Boyd and Vandenberghe (2004)) of the problem (2.16), we get the equations defining the center $\bar{x}$ and the radius r, i.e.:

$$a_{j*}^T \bar{x} + \|a_{j*}\| r = b_{j*},\ j^* \in \{1, \ldots, m\} \tag{2.17}$$

with j^* denoting the index of the activated constraints. Hence if $\begin{bmatrix} a_{j*} & ||a_{j*}|| \end{bmatrix}$ is invertible, one can have a unique relationship which links directly the Chebyshev center $\bar{x}$ and radius r to the facets describing $\mathbb{V}$, i.e.

$$\begin{bmatrix} \bar{x} \\ r \end{bmatrix} = \begin{bmatrix} a_{j*} & \|a_{j*}\| \end{bmatrix}^{-1} b_{j*} \tag{2.18}$$

Notice that compared with the equation (2.15) used to compute the center of mass, the computation of Chebyshev center in (2.16) is simpler. This comes from its explicit formulation (2.18) where the Chebyshev center and radius are the solution of a combination of linear equations. In Example 2.15, we show a bounded polyhedron with its Chebyshev center.

Example 2.17. Let us consider the polyhedron $\mathbb{V}$ in Example 2.15. Its $\mathcal{H}$-representation is

$$\mathbb{V} = \left\{ x \in \mathbb{R}^2 \left| \begin{bmatrix} 0 & -0.4472 \\ 0.4082 & 0.4082 \\ 0 & 0.7071 \\ -0.5883 & 0.1961 \end{bmatrix} x \leq \begin{bmatrix} 0.8944 \\ 0.8165 \\ 0.7071 \\ 0.7845 \end{bmatrix} \right. \right\}$$

Solving the problem (2.16) gives us the Chebyshev center $\bar{x} = \begin{bmatrix} 0.0811 \\ -0.5 \end{bmatrix}$ (red dot in Fig. 2.17) and the radius $r = 1.5$, corresponding to the set of activated constraints

$$\begin{bmatrix} 0 & -0.4472 \\ 0 & 0.7071 \\ -0.5883 & 0.1961 \end{bmatrix} x \leq \begin{bmatrix} 0.8944 \\ 0.7071 \\ 0.7845 \end{bmatrix}$$

and the following equations to compute $\bar{x}$ and r with respect to the equation 2.18.

$$\begin{bmatrix} 0 & -0.4472 & \| 0 & -0.4472 \| \\ 0.4082 & 0.4082 & \| 0.4082 & 0.4082 \| \\ 0 & 0.7071 & \| 0 & 0.7071 \| \\ -0.5883 & 0.1961 & \| -0.5883 & 0.1961 \| \end{bmatrix} \begin{bmatrix} \bar{x} \\ r \end{bmatrix} = \begin{bmatrix} 0.8944 \\ 0.8165 \\ 0.7071 \\ 0.7845 \end{bmatrix}$$

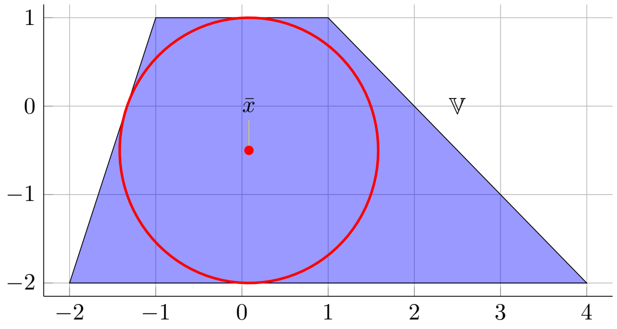

FIGURE 2.17: Chebyshev center of a polyhedron $\mathbb{V}$.

According to the equation (2.18) used to compute the Chebyshev center and radius of a given polyhedron, we apply this equation for a Multi-Agent system deployment after each time of partition. Therefore, the Voronoi-based optimal configuration corresponding to the Chebyshev center chosen as inner target point is defined in Definition 2.13. In Example 2.18, we show the CC over a region $\mathcal{W}$ where a MAS composed of $N = 7$ agents is deployed.

Definition 2.13. A *Chebyshev configuration* (CC) of MAS (2.2) is defined as a configuration where each agent observed output y_i coincides with the Chebyshev center of its corresponding Voronoi cell, i.e. $y_i = \bar{y}_i, \forall i \in \mathcal{N}$.

Example 2.18. Consider the MAS and the region $\mathcal{W}$ given in Example 2.16. A Chebyshev configuration for this MAS is shown in Fig. 2.18 where the position of each agent (denoted by blue circle) coincides with its center of mass (denoted by red dot).

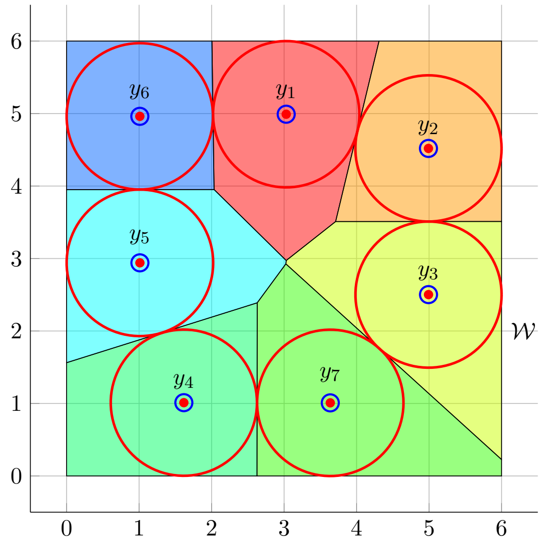

FIGURE 2.18: Chebyshev configuration over $\mathcal{W}$ of $N = 7$ agents.

The positions of the agents are respectively $y_1 = \bar{y}_1 = \begin{bmatrix} 3.029 \\ 5 \end{bmatrix}$, $y_2 = \bar{y}_2 = \begin{bmatrix} 4.991 \\ 4.519 \end{bmatrix}$, $y_3 = \bar{y}_3 = \begin{bmatrix} 4.992 \\ 2.502 \end{bmatrix}$, $y_4 = \bar{y}_4 = \begin{bmatrix} 1.617 \\ 1.011 \end{bmatrix}$, $y_5 = \bar{y}_5 = \begin{bmatrix} 1.011 \\ 2.94 \end{bmatrix}$, $y_6 = \bar{y}_6 = \begin{bmatrix} 1.01 \\ 4.962 \end{bmatrix}$ and $y_7 = \bar{y}_7 = \begin{bmatrix} 3.637 \\ 1.009 \end{bmatrix}$.

Briefly, in this section, we firstly recalled the basic definition of Voronoi partition and further extend this to Multi-Agent system deployments. The ultimate goal for such Multi-Agent system application is to maximize the coverage over a deployed region while converging towards an optimal configuration not predetermined. Two kinds of optimal configurations were described and compared in terms of computational aspects, corresponding respectively to the center of mass and the Chebyshev center.

2.5 Some concluding remarks

This present Chapter provided the set-theoretical prerequisites necessary for the manuscript. We started by introducing in Section 2.1 the linear time-invariant dynamics characterizing the agent's behavior in a Multi-Agent system.

After, we recalled in Section 2.1 some of the fundamental notions in set theory related to polyhedral and ellipsoidal sets families. Their adaptation in view of the application on dynamical Multi-Agent systems analysis and design was detailed subsequently. The choice of polyhedral set is supported by the versatility of its dual (half-spaces/vertices) representation leading to computation advantages and straightforward implementation (construction of safety region, obstacle) and also by its formulation for the optimization-based control framework (construction of feasible sets). Ellipsoidal set is employed further in Chapter 3 in order to imply the controlled invariance without regarding the feasibility of the control solution. The invariance notion is used to construct the robust tube-based safety region of an agent based on the knowledge of agent's dynamics and the boundedness of additive uncertainties. This allows defining a formation in terms of geometrical interpretation, and further formulating collision avoidance constraints as the non-overlapping of safety regions. These properties are the main ingredients to design the formation control subject to anti-collision constraints in Chapter 3 and further the fault detection and isolation layer in Chapter 5.

The remaining sections in this chapter focused on introducing the backgrounds of Voronoi partition and Voronoi-based deployment. We provided also our constructive solution to cover the singularities in the classical Voronoi partition. Furthermore, in the context of Voronoi-based deployment, the ultimate goal is to maximize the coverage over the deployed region of a Multi-Agent system by driving the agents towards an optimal configuration associated with a selective target point inside each Voronoi cell. The conventional point is the center of mass, which leads to an optimal configuration called Centroidal Voronoi configuration. The control solution to obtain such configuration will be detailed in Appendix B. Beside this, we chose also the Chebyshev center as the inner target and provide the formulation of the optimal configuration associated with this center. Chapter 4 will detail how to obtain an optimal configuration. Other kind of target point such as vertex interpolation center will be considered later in Appendix A in with their corresponding Voronoi-based optimal configuration.

Chapter 3

A constructive solution for decentralized collision avoidance

Collision, understood as non-respect of constraints, represents one of the most difficult challenges for the control of dynamical systems in general and by consequence for mobile agents formation based applications in particular. This problem attract considerable attention principally due to the major impact relating to serious damages during the MAS functioning (e.g. robotics, UAV or AUV maneuvering), and thus making collision avoidance the prior objective of MAS supervision. In the context of mission safety guarantee, many works focus on development of fault-tolerant formation control subject to collision avoidance constraints.

From the set-theoretic point of view, collision avoidance objective can be translated into non-overlapping conditions for the safety regions characterizing each agent/obstacle. Concretely, this objective can be described as an exclusion of the operating zones for any pairs of agents (Nguyen et al. (2015b)), or exclusion of the functioning region from the obstacles (Raković and Mayne (2005), Prodan (2012)). The paper Raković and Mayne (2005) presents a set-based exclusion of the obstacles and then determination of admissible regions, by using polyhedral algebra and linear programming framework.

A notable contribution in applying set-theoretic tools in anti-collision control is proposed by Bitsoris and Olaru (2013). In this work, the authors point out that regulating a system subject to anti-collision constraints is equivalent to steer the state towards an equilibrium point outside its convex polyhedral domain of attraction. The positively invariance principle which had been developed before by Bitsoris (1988a) was employed to compute the linear feedback control avoiding the collision with prohibited obstacles. The computation follows the linear programming based control design presented in Bitsoris and Gravalou (1999).

These theoretical notions have been pointed out by Prodan (2012) as viable design tools for MAS centralized formation control subject to anti-collision constraints. This work proposes to compute beforehand a set of linear piecewise affine control solutions subject to anti-collision constraints, which allows keeping the entire MAS functioning strictly within a corresponding safe polyhedral zones inside the non-convex region built on anti-collision constraints (see Section 2.3.2). After finding the feasible solutions, a centralized MPC framework can be derived according to the principle of inverse optimality (see Larin (2003)).

Borrelli et al. (2004) and Keviczky et al. (2008) developed decentralized collision-free formation control using MPC for a group of vehicles. Collision avoidance is ensured by considering an emergency maneuver which is implemented in case where the agents are going to disobey the constraints. Bounds on the agents speeds and accelerations are computed off-line such that the implementation of the emergency maneuver leads to collision-free trajectories. Other contributions of Nersesov et al. (2010) relate to applying set-theoretic methods to design decentralized control for the multi-vehicle coordinated motion, but the main ingredients focus on studying the stability proof in the sense of a vector Lyapunov function for a group of continuous-time dynamics (see Lakshmikantham et al. (1991), Nersesov and Haddad (2006)).

Bemporad and Rocchi (2011) proposed an hierarchical control scheme using decentralized MPC techniques to avoid the collision with the obstacle. The authors of this work approximate the non-convex feasible space where the agent can navigate without collision with the obstacles by a convex polyhedron. Bencatel et al. (2011) proposed sliding mode control based on keeping a constant distance between the agents, thus offering also collision avoidance guarantee while trading the tightness of the formation. Khatib (1986) constructed a family of potential and navigation functions to guide a group of multi-robots avoiding the obstacle.

In this context, the aim of the present chapter is to propose a new decentralized approach to guarantee the collision avoidance using the leader-follower structure. In order to simplify the presentation of the results, the methodology will be described in $\mathbb{R}^2$, while the principles remain general. This work can be considered as a natural extension of the theoretical results presented in Bitsoris and Olaru (2013) but applied for MAS and developed in a decentralized manner from the control computation point of view. We will analyze the restrictions of the method developed in Bitsoris and Olaru (2013), i.e. its feasibility limits and infeasible situations. Our main objective is to partition the working space into a collection of safety zones. Each of these zones is associated with only one agent and our novel control strategy will keep the agent operating in the interior of this zone, offering anti-collision guarantees. The results of this chapter are built on the published contributions in Nguyen et al. (2015b) and Nguyen et al. (2016b).

In the following, Section 3.1 recalls the main ingredients of the decentralized structure for dynamical MAS. The main contribution is described in Section 3.2. Here, we analyze the cases leading to feasibility limitations or infeasibility of the decentralized control action for Multi-Agent system and further propose new guaranteed solutions based on set-theoretic tools. Numerical simulation results are illustrated in Section 3.3, followed by some concluding remarks in Section 3.4.

3.1 Decentralized approach

The decentralized structure considered in this chapter is the leader-follower architecture Stanković et al. (2000). In principle, an agent must be in relative formation and observe a number of constraints with respect to their leader.

Assumption 2. *Each follower is connected uniquely to one leader which is the only one who possesses the complete information on its followers.*

Figure 3.1 below illustrates a simple example emphasizing the difference between the decentralized (the 1st agent is the leader) and the centralized approaches.

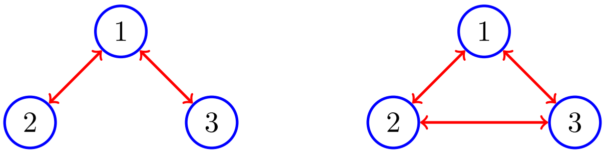

Figure 3.1: Decentralized (left) and centralized (right) communication graph.

In this section, we will present firstly the leader-follower structure in Section 3.1.1 and further the control strategy used to ensure the collision avoidance in Section 3.2. For brevity of the notation and presentation, the group of agents will be considered to evolve in $\mathbb{R}^2$ and the i^{th} agent will be considered as the leader.

3.1.1 Leader-follower architecture

Consider a group of N agents with the i^{th} agent being the leader. $\mathcal{F}^i \subseteq \mathbb{N}_{[1,N]} \setminus \{i\}$ is the set of indices of its followers identified by the existence of a connection to the node i.

The local closed-loop of the j^{th} follower is

$$x_j^+ = A_j x_j + B_j u_j \tag{3.1}$$

with its local control

$$u_j = \bar{u}_j^i + \mathcal{K}(x_j, \bar{x}_j^i) \tag{3.2}$$

The pair $(\bar{x}_j^i, \bar{u}_j^i) \in \mathbb{R}^n \times \mathbb{R}^m$ represents the tracking reference of the j^{th} follower, according to the current state and control action $(x_i, u_i) \in \mathbb{R}^n \times \mathbb{R}^m$ communicated by the leader. An offset free pair $(\bar{x}_j, \bar{u}_j) \in \mathbb{R}^n \times \mathbb{R}^m$ is added to $(\bar{x}_j^i, \bar{u}_j^i)$ to avoid the collision with the leader, i.e.

$$\begin{aligned} \bar{x}_j^i &= \bar{x}_j + x_i \\ \bar{u}_j^i &= \bar{u}_j + u_i \end{aligned}$$

The collision avoidance is guaranteed if each follower operates strictly in its *functioning zone* relative to the position of the leader. This zone is defined by the leader and identified by a separating hyperplane for any pair of followers. Its construction is presented in the following subsection.

3.1.2 Partitioning in functioning zones

The leader uses a set of hyperplanes to separate the functioning zone of its followers, more precisely it partitions the working space into a union of polyhedral non-overlapping *separation sets*. Each set is associated to a follower and defines its functioning zone. For the j^{th} follower, its separation set $\mathcal{C}_j^i$ is defined as a cone

$$\mathcal{C}_j^i = \{x \in \mathbb{R}^n | C^i(x - x_i) \leq 0, j \in \mathcal{F}^i\} \tag{3.3}$$

where $C^i(x - x_i) = 0$ denotes the set of hyperplanes used to isolate the j^{th} follower from its neighbors, and $\mathcal{F}^i$ denotes the indices set of the followers.

Remark 3.1. Instead of determining the functioning zone relative to the position of the leader, it can be translated without loss of generality to a positioning relative to the origin, i.e. $(x_i, u_i) = (0, 0)$.

Clearly, the hyperplanes pass through the leader current state, now reduced to the origin by a change of coordinates.

Remark 3.2. The choice of the separating hyperplanes $C^i x = 0$ is important because it decides the existence of a feasible control action on the boundary of the separation set.

Assumption 3. *In the remaining of this chapter it is assumed that the hyperplanes $C^i x = 0$ are given and the sets $\mathcal{C}_j^i$ are non-overlapping.*

In order to ensure the collision avoidance, the state x_j has to satisfy the two following constraints:

$$x_j \notin \text{int}\,\{(-\mathcal{S}_j) \oplus \mathcal{S}_i\} \tag{3.4a}$$

$$\{x_j\} \oplus \mathcal{S}_j \in \mathcal{C}_j^i \tag{3.4b}$$

with $\mathcal{S}_i$, $\mathcal{S}_j$ denoting respectively the safety regions of the leader and the j^{th} follower. $\mathcal{C}_j^i$ denotes the separation set of this follower. The constraint (3.4a) guarantees the collision avoidance between the j^{th} follower and the leader, with respect to Section 2.3.2. This constraint is derived from $x_j - x_i \notin (-\mathcal{S}_j) \oplus \mathcal{S}_i$, with x_i being the origin (see Remark 3.1). The constraint (3.4b) guarantees the collision avoidance with the other followers. The j^{th} follower with its safety region[1] is the unique agent belonging to the separation set $\mathcal{C}_j^i$. This implies the anti-collision with its neighbors, i.e. $\mathcal{S}(x_j) \in \mathcal{C}_j^i$. Using (2.9), we obtain $\{x_j\} \oplus \mathcal{S}_j \in \mathcal{C}_j^i$. These two constraints

[1] Note that, the effect of the possible disturbances is taken into account via the safety region $\mathcal{S}_j$. In fact, a hard constraint is imposed via (3.4b) for the center x_j of the tube in steady functioning.

determine the non-convex *functioning zone* $\mathcal{Z}_j^i \subset \mathcal{C}_j^i$ in which x_j has to evolve in order to ensure collision-free maneuvering of the formation.

$$\mathcal{Z}_j^i = \left\{x \in \mathbb{R}^n | \{x\} \oplus \mathcal{S}_j \in \mathcal{C}_j^i, x \notin \operatorname{int}\{(-\mathcal{S}_j) \oplus \mathcal{S}_i\}\right\} \tag{3.5}$$

The collision avoidance is guaranteed by means of stabilizing the follower's dynamics around a fixed point of its functioning zone. We will detail next how to determine this fixed point.

3.1.3 Local fixed point determination

Starting from the characterization of the functioning zone sent by the leader, each follower has to find the best suitable equilibrium point relative to the origin (considered as the leader's state with respect to Assumption 3.1). This represents one of the conceptual contributions of the present work with respect to the existing results which impose the local target via the centralized decision or directly by communication with the leader.

In the present framework, the local fixed point is characterized by solving locally the optimization problem:

$$\bar{x}_j^* = \arg\min_{\bar{u}_j} \|\bar{x}_j\| \tag{3.6a}$$

$$\text{s.t.} \quad \bar{x}_j \in \mathcal{Z}_j^i \tag{3.6b}$$

$$\bar{x}_j = A_j\bar{x}_j + B_j\bar{u}_j \tag{3.6c}$$

The constraint (3.6b) implies that the non-convex set $\mathcal{Z}_j^i$ called *operating zone* contains the local fixed point $\bar{x}_j$, as illustrated in Fig. 3.2 (with $i = 1$ and $\mathcal{F}^i = \{2, 3\}$). Note that the set $\mathcal{Z}_j^i$ does not contain the origin with respect to the condition (3.4a). The remaining constraint (3.6c) enforces the fixed point characterization.

Remark 3.3. In order to avoid the non-convexity of $\mathcal{Z}_j^i$, the constraint (3.6b) will be replaced for the local feedback design by $x_j \in \mathcal{H}_j^i$, with $\mathcal{H}_j^i \subset \mathcal{Z}_j^i$ a convex region with the following property:

$$\mathcal{H}_j^i = \left\{\mathcal{C}_j^i \ominus \mathcal{S}_j\right\} \cap \{h_z x \leq k_z\} \tag{3.7}$$

with $\{h_z x = k_z\}$ the constraints activated when verifying the inclusion $\bar{x}_j^* \in \{(-\mathcal{S}_j) \oplus \mathcal{S}_i\}$.

Example 3.1. Consider a MAS composed of 3 homogeneous agents. The 1^{st} agent is the leader and the two remaining agents are its followers. Moreover, the leader resides at the origin. We will illustrate the construction of the functioning zones for the followers in Fig. 3.2. The forbidden blue set represents the anti-collision constraints $(-\mathcal{S}_j) \oplus \mathcal{S}_i$ between the leader and its followers (see the constraint (3.4a)). The green line represent the hyperplane used to separate the two followers, which in fact defines the separation sets $\mathcal{C}_2^1$ (on its left hand side) and $\mathcal{C}_3^1$ (on its right hand side) (see the definition (3.3)). Therefore, using the formulation (3.5), we can built the non-convex zones $\mathcal{Z}_2^1$ and $\mathcal{Z}_3^1$ bounded by the black dash-line. The fixed points ($\bar{x}_2^*$,

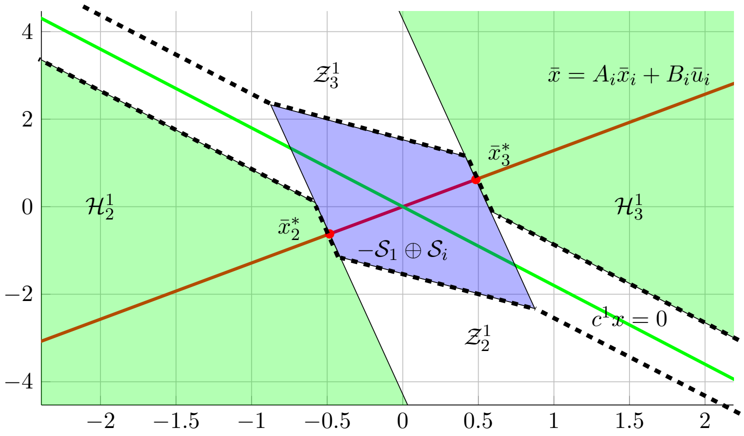

FIGURE 3.2: Determination of $\mathcal{H}_2^1$ and $\mathcal{H}_3^1$ for the 2^{nd} and 3^{rd} agent.

$\bar{x}_3^*$) found by solving (3.6) are plotted as the circle-red points[2]. Using (3.7), we can choose the functioning zones $\mathcal{H}_2^1$ and $\mathcal{H}_3^1$ based on the hyperplanes activated by the fixed points and the hyperplane used to separate the follower. These functioning zones $\mathcal{H}_2^1$ and $\mathcal{H}_3^1$ are covered by green in Fig. 3.2.

Once the feasibility of the static optimization problem is guaranteed, each follower has to design its individual control policies to keep its own trajectories within its functioning zone $\mathcal{H}_j^i$ and to asymptotically converge to the fixed point $\bar{x}_j$ solution of the optimization problem (3.6). This control law will be developed in the next subsection.

3.1.4 Individual feedback control

In the sequel, for brevity, we will neglect the follower indices in the notations. The nominal local dynamics of a follower $x^+ = Ax + Bu$, its functioning zone $\mathcal{H}$ and its fixed point $\bar{x}$ relative to the origin are considered known.

It will be considered that $\mathcal{H} \subset \mathbb{R}^2$ is represented as the combination of linear inequalities $\mathcal{H} = \{x \in \mathbb{R}^n | hx \leq k\}$ similar to the one illustrated in the Fig. 3.2 where not all hyperplanes forming $\mathcal{H}$ are activated by $\bar{x}$ (as part of them are inherited from the functioning zone constraints). We denote by $\{h_1x = k_1\}$ the set of hyperplanes activated by $\bar{x}$ and $\{h_2x = k_2\}$ the remaining hyperplanes[3].

The goal is to ensure that the follower's state x will converge asymptotically to the fixed point $\bar{x}$ lying on the boundary $h_1x = k_1$, while the trajectory remains in the functioning zone $\mathcal{H}$

[2] In the present case, they are placed on the same read line of fixed-points due to the homogeneity of the agents' dynamics but in the general case, they follow the structure imposed by the local dynamics.

[3] If the fixed point saturates both constraints, then all boundaries are activated by this point, and should be taken into account in the local control design. It remains to consider the feasibility of the control computation, according to the conclusion in Bitsoris and Olaru (2013)

according to the constraints (3.4). This requires the design of a feedback linear control[4]:

$$u = K(h_1, k_1)x \tag{3.8}$$

which makes the set $\mathcal{H}$ positive invariant and also stabilizes the closed-loop dynamics of the follower. According to Bitsoris and Olaru (2013), such a control law (3.8) exists if and only if there exist a non-negative matrix F and a symmetric positive definite matrix $P = P^\top \succ 0$ such that the following feasibility problem is verified

$$h_1\,(A + BK) = Fh_1 \tag{3.9a}$$

$$\begin{bmatrix} \delta P & (A+BK)P \\ P^\top (A+BK)^\top & P \end{bmatrix} \succeq 0 \tag{3.9b}$$

with the decision variables F, K and P. The constraint (3.9a) expresses the invariance condition (see Definition 2.10 and Theorem 2.1), while (3.9b) formulates the Lyapunov stability constraint, with $0 < \delta < 1$ the rate of convergence which is a parameter of this optimization problem.

By using a new variable $Y = KP$, the problem (3.9) becomes

$$h_1\,(AP + BY) = Fh_1P \tag{3.10a}$$

$$\begin{bmatrix} \delta P & AP+BY \\ (AP+BY)^\top & P \end{bmatrix} \succeq 0 \tag{3.10b}$$

which can be numerically solvable by choosing an appropriate value[5] for F.

However, solving this problem for $h_1x \leq k_1$ could face particular (infeasible) situations which will be discussed in the next section.

3.2 Main contribution

This section considers the two possibilities of solving the fesibility problem (3.9):

1) *Feasibility:* In this case, we can compute a control $u = K(h_1, k_1)x$ which ensures the controlled invariance property of the domain described by the half-space active at $\bar{x}$, i.e. $h_1x \leq k_1$. What remains to be considered is the satisfaction of the inactive constraints at $\bar{x}$, i.e. $h_2x \leq k_2$, or equivalently, the controlled invariance of this domain with respect to the control u.

2) *Infeasibility:* The set $\mathcal{H}$ is not controlled invariant even if the closed-loop dynamics is stabilized. In this case, the constraints (3.9) hold for a strict subset of $\mathcal{H}$ (i.e. the control law (3.8) makes a strict subset of $\mathcal{H}$ positive invariant).

[4] If $k_1 \neq 0$, this control is affine, i.e. $u = K(h_1, k_1)x + l(h_1, k_1)$, with $l(h_1, k_1)$ representing the affine term.

[5] According to Bitsoris and Olaru (2013), an eigenstructure assignment approach can be employed to solve the problem 3.10. By this way, the matrix F will be composed of the eigenvalues corresponding to the eigenvectors h_1.

Both of these two cases were not completely analyzed in the literature. The main contribution of the present section is to employ set-theoretic tools in order to analyze and to propose novel control strategies to deal with both the feasible and the infeasible cases. Two approaches are provided for the feasible case in the subsection 3.2.1. A relaxed construction is proposed for the infeasible case in the subsection 3.2.2.

3.2.1 Feasible fixed point $\bar{x}$

If the optimization problem (3.9) is feasible, then there exists a control $u = K(h_1, k_1)x$ ensuring the invariance of the subspace $h_1 x \leq k_1$. Otherwise, this means that the entire set $\mathcal{H}$ is not controlled invariant, then there exists an invariant subset $\Omega \subseteq \mathcal{H}$.

The set Ω can be characterized in terms of the *Maximal Output Admissible Set* (MOAS) (see Gilbert and Tan (1991) for more details) associated with the local control $u_\Omega = K(h_1, k_1)x$. Knowing that $\Omega \subseteq \mathcal{H}$ and $A + BK(h_1, k_1)$ is a Schur matrix, there exist a constructive method to obtain iteratively the set Ω as proposed in Gilbert and Tan (1991). We denote $\Omega(t)$ the MOAS obtained at each iteration t, with $\Omega(t) = \{x \in \mathbb{R}^n | h_\Omega x \leq k_\Omega\}$ and $\Omega(t) \subset \mathcal{H}$. The computation of $\Omega(t+1)$ is derived by recurrence, i.e.

$$\Omega(t+1) = \big\{x \in \mathbb{R}^n | x \in \Omega(t), \big(A + BK(h_1, k_1)\big)x \in \Omega(t)\big\} \tag{3.11}$$

or equivalently

$$\Omega(t+1) = \big\{x \in \mathbb{R}^n \,\big|h_\Omega x \leq k_\Omega,\ h_\Omega\big(A + BK(h_1, k_1)\big)x \leq k_\Omega\,\big\} \tag{3.12}$$

We recall the following theorems presented and proved by Gilbert and Tan (1991) concerning the determinedness of MOAS with respect to the discrete-time autonomous dynamics

$$\begin{cases} x(k+1) = Ax(k) \\ y(k) = Cx(k) \end{cases}$$

and the constraint

$$y(k) \in \mathcal{H}$$

Theorem 3.1. *(Gilbert and Tan (1991)) The maximal output admissible set Ω is finitely determined if and only if $\Omega(t+1) = \Omega(t) = \Omega$ with $t \geq t^*$.*

Theorem 3.2. *The following statements summarize the conditions of maximal output admissible set Ω existence.*
(i) *If (C, A) is observable*[6] *and the constrained output's set $\mathcal{H}$ is bounded, then the set Ω is bounded;*
(ii) *If the matrix A is Lyapunov stable and $0 \in int(\mathcal{H})$, then $0 \in int(\Omega)$.*

[6] In the present chapter, we assume that $C = I_n$, i.e. $y(k) = x(k)$.

It is worth to mention that the finite determinedness[7] allows a construction as the one illustrated in Example 3.2

Example 3.2. This example shows a case of finitely determined MOAS in Fig. 3.3, starting from $\Omega(0) = \mathcal{H}$ (bounded by the dash line) and contracting successively to $\Omega(1)$, $\Omega(2)$ and $\Omega(3) = \Omega$ (darkest blue colored). The polyhedron $\mathcal{H}$ is chosen as a cone defined as

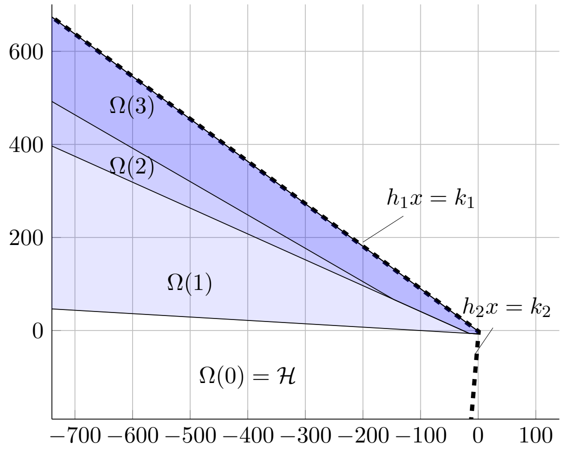

FIGURE 3.3: Construction of Ω.

$$\mathcal{H} = \left\{ x \in \mathbb{R}^n | \begin{bmatrix} 0.6740 & 0.7387 \\ 0.5120 & -0.0353 \end{bmatrix} x \leq \begin{bmatrix} -1.1302 \\ 0.5 \end{bmatrix} \right\}$$

and the closed-loop dynamics is $x(k+1) = \begin{bmatrix} -0.0003 & -0.2196 \\ 0.2619 & 0.4870 \end{bmatrix} x(k)$. The construction of the set $\Omega(k)$ follows the formulation (3.11)-(3.12) with respect to the follower's closed-loop dynamics.

Example 3.3. This example considers the case of undetermined MOAS illustrated in Fig. 3.4.The polyhedron $\mathcal{H}$ (bounded by dashed lines) is chosen as

$$\mathcal{H} = \text{conv} \left\{ \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \end{bmatrix} \right\}$$

The closed-loop matrix dynamics is $x(k+1) = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.2 \end{bmatrix} x(k)$. We employ the formulation 3.11-3.12 con construct iteratively the set $\Omega(k)$ with respect to the follower's closed-loop dynamics. However, the set $\Omega(10)$ is unbounded thus not enclosed inside $\mathcal{H}$. As consequence, Ω in this example is not finitely determined.

In the sequel, we consider uniquely the cases when the determinedness is guaranteed.

[7] The maximal output admissible set Ω is not necessary finitely determined (meaning infinitely determined or even undetermined). Some examples illustrating the singularity of the dynamical MOAS construction can be found in the paper of Gilbert and Tan (1991). Moreover, a reachability analysis can be employed to derive the condition to stop the invariant subset Ω construction in Theorem 3.1.

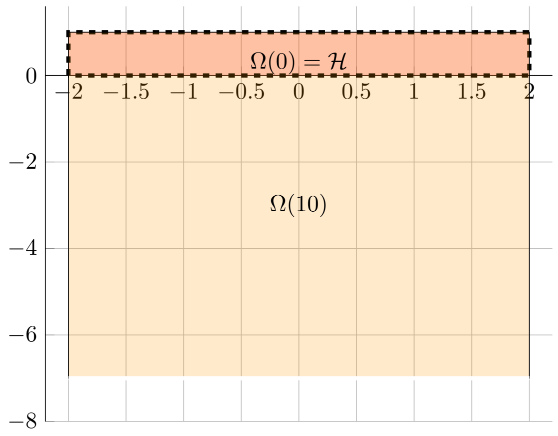

Figure 3.4: Undetermined case of Ω.

Remark 3.4. According to the conclusion of Gilbert and Tan (1991), the finite determinedness is not ensured in case of unbounded polyhedra, but we can still apply the constructive method (3.12) by replacing the unbounded polyhedron with a bounded polyhedron with arbitrary large bounds in order to cover the admissible configurations.

Remark 3.5. The control law $u_\Omega = K(h_1, k_1)x$ makes the entire half-space $h_1 x \leq k_1$ controlled invariant. But $\mathcal{H}$ is a subset of this half-space and it has $h_1 x \leq k_1$ as its boundary. It follows that one of the boundary of Ω is $h_1 x = k_1$.

Remark 3.6. If $\mathcal{H}$ is controlled invariant by u_Ω, then $\Omega = \mathcal{H}$.

We note that the local control is restricted in the subset Ω, and it will be activated only when the current state belongs to Ω. Whenever $x \notin \Omega$, the linear control law is not admissible. One has to devise a procedure to drive the agent state x into the strict interior of Ω while ensuring the invariance of $\mathcal{H}$.

In the following, we propose two approaches to treat this problem. The first one is based on the iterative construction of a set of contracted ellipsoids within $\mathcal{H}$. The second approach employs the interpolation technique for the case of unbounded functioning zones.

3.2.1.1 Approach 1 - Parameterized contractive ellipsoid

This approach uses the existence of an ellipsoid $\mathcal{E}(c, P, d)$ (see Definition 2.4) in the current functioning zone to drive the agent's state towards the Maximal output admissible set Ω. It takes the form of an optimization problem (3.13) with a given shape matrix P of the ellipsoidal set (which enjoys contractiveness property with respect to the closed-loop dynamics around the fixed point), while the center c and the range d are considered as parameters. Recall the explicit form of the functioning zone $\mathcal{H} = \left\{x \in \mathbb{R}^n | a_l x \leq b_l,\ \forall l \in \mathbb{N}_{[1,p]}\right\}$, with p denoting the number of constraints. With these elements, the optimization problem to find the ellipsoid center c is formulated as

$$
\begin{aligned}
c^*(x) = \arg \min_{c,d,u_c} & \; log(|c - \bar{x}|) && \text{(3.13a)}\\
\text{s.t.} \quad & c = Ac + Bu_c && \text{(3.13b)}\\
& \left(\sqrt{a_l P^{-1} a_l^\top}\right)\sqrt{d} + a_l c \le b_l && \text{(3.13c)}\\
& (x - c)^\top P(x - c) \le d && \text{(3.13d)}
\end{aligned}
$$

The constraint (3.13b) implies that the center c is a fixed point. The shape matrix P is pre-imposed as the solution of (3.9) for $h_1 x \le k_1$. The constraints (3.13c) and (3.13d) enforce respectively that the ellipsoid is inside $\mathcal{H}$ and contains the current state vector x. We develop further the properties of the solution of the problem (3.13) and the relationship with the above constraints. The following theorem is given to confirm that the existence of such ellipsoid depends on the feasibility of (3.13) and further ensure the controlled invariance inside this ellipsoid.

Theorem 3.3. *For a given x, if (3.13) is feasible, then there exists an ellipsoid $\mathcal{E}(c, P, d)$ included inside of $\mathcal{H}$ and $x^+ \in \mathcal{E}(c, P, d)$.*

Proof. The optimization problem corresponds to an inclusion of an ellipsoid in a polyhedral set (see for example Hindi (2004) for a discussion on the canonical form of the constraints in this respect). We recall that feasibility of the optimization problem (3.9) for the half-space $h_1 x \le k_1$ implies the existence of a normalized ellipsoid

$$
\mathcal{E}(\bar{x}, P, 1) = \left\{x \in \mathbb{R}^n | (x - \bar{x})^\top P(x - \bar{x}) \le 1\right\}
$$

such that at the instant k, the state vector x belongs to this ellipsoid. We want to check if by using $u = K(h_1, k_1)x$, the constraint $x^+ \in \mathcal{H}$ holds for every $x \in \mathcal{H}$. This can be translated to finding an ellipsoid included in the strict interior of $\mathcal{H}$ which satisfies:

(i) the current state vector x is included in the ellipsoid, as illustrated by the constraints (3.13d);

(ii) it preserves the shape matrix P and optimizes its scaling factor d;

(iii) its center c is situated on the fixed point hyperplane, see the constraint (3.13b);

(iv) its center c is as close as possible to the fixed point $\bar{x}$.

Let us start from the fact that the ellipsoid in (3.13d) has the parameterized form

$$
\mathcal{E}(c, P, d) = \left\{x \in \mathbb{R}^n | (x - c)^\top P(x - c) \le d\right\}
$$

Consider $y = \left(\frac{P}{d}\right)^{1/2}(x-c)$. The ellipsoid definition becomes

$$\mathcal{E}(c,P,d) = \left\{ \left(\frac{P}{d}\right)^{-1/2} y + c \in \mathbb{R}^n |\, \|y\|^2 \leq 1 \right\}$$

Considering $\mathcal{E}(c,P,d) \subset \mathcal{H}$, then for all $l \in \mathbb{N}_{[1,p]}$ and $\|y\|^2 \leq 1$. From $a_l x \leq b_l$, we derive

$$a_l \left(\left(\frac{P}{d}\right)^{-1/2} y + c \right) \leq b_l$$

Using this result combined with $\|y\|^2 \leq 1$, we obtain

$$\left\| \left(\frac{P}{d}\right)^{-1/2} a_l^\top \right\| + a_l c \leq b_l$$

Using

$$\left\| \left(\frac{P}{d}\right)^{-1/2} a_l^\top \right\| = \sqrt{\left(\left(\frac{P}{d}\right)^{-1/2} a_l^\top \right)^\top \left(\left(\frac{P}{d}\right)^{-1/2} a_l^\top \right)}$$

we get finally

$$\left(\sqrt{a_l P^{-1} a_l^\top}\right)\sqrt{d} + a_l c \leq b_l$$

Whenever the ellipsoid $\mathcal{E}(c,P,d)$ exists, by the inclusion of $\mathcal{E}(c,P,d)$ in the strict interior of the functioning zone $\mathcal{H}$, one can guarantee the stability of the closed-loop, and the recursive constraint satisfaction as long as $x^+ \in \mathcal{H}$. □

For a given x, its convergence to the fixed point $\bar{x}$ is guaranteed by minimizing the distance between c and $\bar{x}$. This optimization problem is repeated until the state trajectory reaches the MOAS Ω as long as the linear feedback law is known to be feasible in $\Omega \subset \mathcal{H}$.

Example 3.4. This example illustrates the application of parameterized contractive ellipsoid approach. The result is shown in Fig. 3.5.

The functioning zone $\mathcal{H}$ with respect to the leader state is defined in terms of two hyperplanes $h_1 x = k_1$ and $h_2 x = k_2$, i.e.

$$\mathcal{H} = \left\{ x \in \mathbb{R}^2 \,| Hx \leq \gamma \right\}$$

with $H = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} 0.6740 & 0.7387 \\ 0.5120 & -0.0353 \end{bmatrix}$ and $\gamma = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} -1.1302 \\ 0.5 \end{bmatrix}$.

The red line denotes the equilibrium line $\bar{x} = \begin{bmatrix} -0.2 & 0.5 \\ 0.2 & 0.71 \end{bmatrix} \bar{x} + \begin{bmatrix} 0.71 \\ 0.22 \end{bmatrix} \bar{u}$. The red point is the follower's equilibrium point $\bar{x} = \begin{bmatrix} -0.6962 \\ -0.8948 \end{bmatrix}$ of $\mathcal{H}_2^1$ and closest to the leader state, obtained by solving (3.6).

Concerning the local linear control determination, by solving the problem (3.9) for the boundary $h_1 x = k_1$, we obtain the gain $K(h_1, k_1) = \begin{bmatrix} 0.2812 & -1.0135 \end{bmatrix}$ for the local linear feedback control (3.20), associated with the matrix $P = \begin{bmatrix} 2.7132 & 0.0031 \\ 0.0031 & 2.7132 \end{bmatrix}$ employed as the shape matrix for the ellipsoids. The green ellipsoid $\mathcal{E}(\bar{x}, P, 1)$ denotes the ellipsoid associated with its center $\bar{x}$ (red dot) lying on the boundary $h_1 x = k_1$. The black dots are the centers of the ellipsoids obtained by solving iteratively (3.13) according to the feedback state x (blue dots). We can see that the state vector x approaches the fixed point $\bar{x}$ according to the convergence of the ellipsoid center $c(x)$ to this fixed point $\bar{x}$.

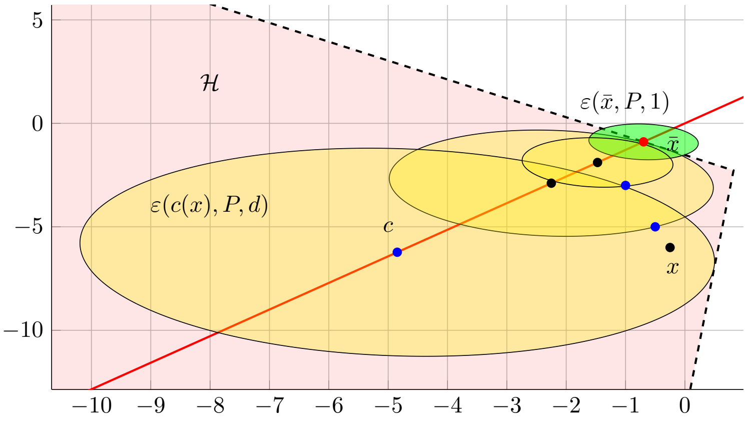

FIGURE 3.5: Feasibility test.

Remark 3.7. The square root terms makes the problem (3.13) nonlinear beside the form of the cost function. In order to avoid the complexity related to the nonlinear constraint (3.13c), we can replace $\sqrt{d}$ by a new variable $\theta \in \mathbb{R}$, i.e. $\left(\sqrt{a_l P^{-1} a_l^\top}\right) \theta + a_l c \leq b_l$. Then, we can subsequently use a strictly positive parameter ϵ to decrease the gap between the variables d and θ, i.e. $0 \leq \theta^2 - d \leq \epsilon$, and thus transform the optimization problem (3.13) into the following convex optimization problem

$$c^*(x) = \arg\min_{c,d,u_c} log(|c - \bar{x}|) \tag{3.14a}$$

$$\text{s.t.} \quad c = Ac + Bu_c \tag{3.14b}$$

$$\left(\sqrt{a_l P^{-1} a_l^\top}\right) \theta + a_l c \leq b_l \tag{3.14c}$$

$$(x - c)^\top P(x - c) \leq d \tag{3.14d}$$

$$0 \leq \theta^2 - d \leq \epsilon \tag{3.14e}$$

Remark 3.8. Solving the problem (3.13) does not guarantee to cover the maximal controllable subset of $\mathcal{H}$. In fact, for initial points on the vertices of $\mathcal{H}$ the feasibility of the ellipsoidal containment cannot be fulfilled as long as there are two different supporting hyperplanes active for the same point on the boundary.

In the light of Remark 3.8, we propose a second approach which deals with the infeasibility of the problem (3.13) for the current state vector$x$.

3.2.1.2 Approach 2 - Interpolation based control

In this subsection, the control action u will be obtained as a convex combination of

- the *local control* $u_\Omega = K(h_1, k_1)x$ in the MOAS Ω;
- a *vertex control law*[8] associated with a *M-step robustly controlled set* $\mathcal{P}_M$ (understood as an approximation of the maximal controllable set within $\mathcal{H}$).

The principles of such an interpolation based control for constrained dynamical systems were discussed in Nguyen et al. (2013) and applied in Nguyen et al. (2015b). The feasible region for the interpolation scheme will be the *maximal controllable set* inside the functioning zone $\mathcal{H}$, while the set Ω defines the feasibility region of a local control law u_Ω. This set can be obtained using classical reachability arguments (see Blanchini and Miani (2007) for details).

The present work consideres that either the set $\mathcal{P}_M$ or a convex (polyhedral) controlled invariant subset Φ, such that $\Omega \subseteq \Phi \subseteq \mathcal{P}_M$ is available together with a feasible control action on the boundaries (the so-called *vertex control* of this set, as proposed by Gutman and Cwikel (1986)). Therefore, the interpolated control action is described by a linear interpolation scheme:

$$u = \beta u_\Phi + (1-\beta)u_\Omega, \; 0 \leq \beta \leq 1 \tag{3.15}$$

where u_Φ and β have to be calculated in real time. The control component u_Φ will be activated i.e. $\beta = 1$ when $x \in \Phi \setminus \Omega$ and the scalar β has to be minimized in order to get the control input u as close as possible to the local control u_Ω.

We present next how to effectively obtain the vertex control action u_Φ and then the interpolation coefficient β.

The determination of u_Φ exploits the fact that its objective is to push x from the boundary of Φ towards its interior. This can be done by solving the following LP problem:

$$\begin{aligned} u_\Phi = \arg\min_{\alpha, u_\Phi} \alpha \\ \text{s.t.} \begin{cases} Ax + Bu_\Phi \in \; \alpha\Phi \\ 0 \leq \alpha \leq 1 \end{cases} \end{aligned} \tag{3.16}$$

with α denoting the minimal contraction factor.

For the determination of the scalar β, we exploit the fact that the control u obtained as an interpolation of u_Ω and u_Φ has to be as close as possible to u_Ω. This can be translated in

[8] A vertex control for a given polyhedral state-space set is defined as a control action used to drive a current state from the corresponding vertex toward the interior of the considered polyhedron.

terms of the optimization problem. Consider $u = \beta u_\Phi + (1-\beta)u_\Omega$, and x decomposed as $x = \beta x_\Phi + (1-\beta)x_\Omega$, where $x_\Phi \in \Phi$ and $x_\Omega \in \Omega$. Let $\Phi = \{x \in \mathbb{R}^n | F_\Phi x \leq k_\Phi\}$ and $\Omega = \{x \in \mathbb{R}^n | F_\Omega x \leq k_\Omega\}$, then the optimization problem allowing us to find the interpolation factor β is

$$\begin{aligned} &\min_{\beta, x_\Omega, x_\Phi} \ \beta \\ \text{s.t.} &\begin{cases} F_\Phi x_\Phi \leq k_\Phi \\ F_\Omega x_\Omega \leq k_\Omega \\ \beta x_\Phi + (1-\beta)x_\Omega = x \\ 0 \leq \beta \leq 1 \end{cases} \end{aligned} \tag{3.17}$$

Although (3.17) is nonlinear, we can translate it into a LP problem by using the following change of variable $r = \beta x_\Phi$. Hence the constraint $\beta x_\Phi + (1-\beta)x_\Omega = x$ of the problem 3.17 becomes $x - r = (1-\beta)x_\Omega$. In addition, the constraint $F_\Omega x_\Omega \leq k_\Omega$ is equivalent to $F_\Omega(1-\beta)x_\Omega \leq (1-\beta)k_\Omega$ and by replacing $(1-\beta)x_\Omega$ by $x - r$, we obtain $F_\Omega(x-r) \leq (1-\beta)k_\Omega$. Moreover, $F_\Phi x_\Phi \leq k_\Phi$ is translated to $F_\Phi \beta x_\Phi \leq \beta k_\Phi$ yielding $F_\Phi r \leq \beta k_\Phi$. Briefly, the problem (3.17) becomes:

$$\begin{aligned} &\min_{\beta, r} \beta \\ \text{s.t.} &\begin{cases} F_\Phi r \leq \beta k_\Phi \\ F_\Omega(x-r) \leq (1-\beta)k_\Omega \\ 0 \leq \beta \leq 1 \end{cases} \end{aligned} \tag{3.18}$$

Notice that the construction of the maximal controllable set is a relatively time consuming procedure. It has to be carefully considered whenever input saturations are imposed. In the case when such limitations are imposed over $\mathcal{H}$, there exists a simple candidate construction for the outer set Φ. Indeed, by considering Φ to be a scaled version of Ω with the restriction to $\mathcal{H}$. The scaling of the set Ω will be done with respect to the fixed point $\bar{x}$. The construction framework is detailed in the proposition below.

Proposition 3.4. *An outer candidate set for the interpolation Φ is defined as*

$$\Phi = (\{\bar{x}\} \oplus \mu\Omega_0) \cap \mathcal{H}$$

with μ found by solving the following linear programming (LP) problem:

$$\min_{\mu} \mu \quad s.t. \begin{cases} x \in \ \{\bar{x}\} \oplus \mu\Omega_0 \\ \mu \geq 0 \end{cases} \tag{3.19}$$

with $\Omega_0 = \{-\bar{x}\} \oplus \Omega$.

Remark 3.9. The set Φ, obtained via the optimal solution for the LP problem (3.19) ensures $\Omega \subseteq \Phi$ and $\Phi \subseteq \mathcal{H}$ via Proposition 3.4.

Example 3.5. This example illustrates the construction of the set Φ for the functioning zone $\mathcal{H}$ with the fixed point $\bar{x}$ in Example 3.4. The set $\mathcal{H}$ is bounded by $h_1 x = k_1$ and $h_2 x = k_2$

(dash line in Fig. 3.6). The fixed point (red) is on the boundary $h_1 x = k_1$. The orange set is the MOAS Ω obtained by using (3.11) with respect to the follower's dynamics given in Example 3.4. The follower's state vector is $x = \begin{bmatrix} -2 \\ -15 \end{bmatrix}$.

As shown in Fig. 3.6, this state x is inside $\mathcal{H}$ but outside the set Ω. By solving the problem (3.19) with respect to the follower's state vector x, we obtain the scalar $\mu = 2.2257$ to construct the set $\Phi \subset \mathcal{H}$ (covered by yellow) according to the formulation in Proposition 3.4.

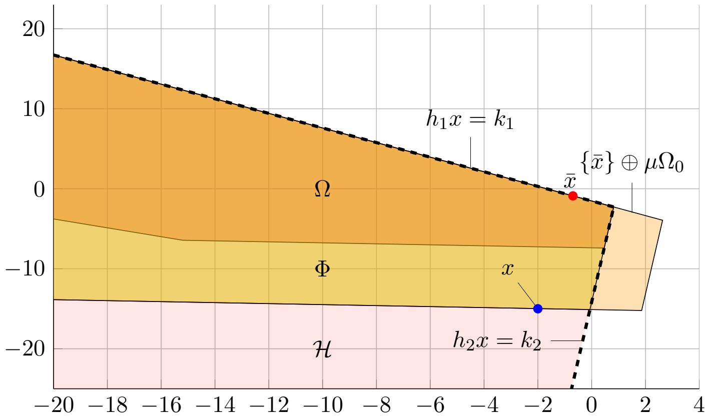

FIGURE 3.6: Construction of the interpolation set Φ.

Summing up, solving respectively (3.19), (3.16), (3.18) in real time provides the interpolation factor and implicitly the control u. We resume all of these procedures in the following algorithm which permit obtaining the control action in real time.

```
Algorithm 2: Real time calculation of $u$
Input  : The state vector $x$, the fixed point $\bar{x}$, the functioning zone $\mathcal{H}$,the Maximal output
         admissible set $\Omega$
Output: The interpolation based control $u$
- solve the problem (3.9) for $\mathcal{H}$ to get $K(h_1, k_1)$ for the local control $u_\Omega$;
- solve the problem (3.19), get the scalar $\mu$ and construct $\Phi$;
- solve the problem (3.16), get the scalar $\alpha$ and the control $u_\Phi$;
- solve the problem (3.17), get the scalar $\beta$;
- calculate the interpolated control (3.15);
```

3.2.2 Infeasible fixed point $\bar{x}$

The structure of the dynamical system may be such that a local linear control $u_\Omega = K(h_1, k_1)x$ can not guarantee the invariance of the half-space $h_1 x \leq k_1$ and implicitly of the set $\mathcal{H}$ as long as the fixed point is placed on the boundary (see Fig. 3.6 for the notations). Therefore, the local set Ω can not be constructed and the interpolation based control presented in the previous subsection can not be used.

In this case, one can always find a subset in the strict interior of $\mathcal{H}$ such that the controlled invariance is guaranteed. Now, the problem comes from the nonlinear nature of the constraints (3.9). Indeed, the detailed study available in Bitsoris and Olaru (2013) mentioned that only a subclass of design problem can be solved by eigenstructure assignment approaches. Here, we propose a simple and efficient method to solve this problem: by moving the fixed point $\bar{x}$ into the strict interior of $\mathcal{H}$ along the equilibrium points line. Once this relaxation is performed, any linear stabilizing feedback gain is admissible. Subsequently, the ellipsoidal construction approach presented above can be used to drive the current state x according to the ellipsoid whose center is parametrized to enable the feasibility. This construction ensures the recursive feasibility and approaches the fixed point arbitrarily close to the boundary of $\mathcal{H}$. Alternatively, an interpolation strategy can be employed after the construction of the maximal controlled invaraint subset of $\mathcal{H}$, upon the same premises used in Section 3.2.

3.3 Illustrative example

Consider a MAS $\mathbf{\Sigma}$ composed of $N = 3$ homogeneous agents. They have the same safety region. Their dynamics are described by

$$x_i(k+1) = \begin{bmatrix} -0.2 & 0.5 \\ 0.2 & 0.71 \end{bmatrix} x_i(k) + \begin{bmatrix} 0.71 \\ 0.22 \end{bmatrix} u_i(k)$$

Let us choose the 1^{st} agent as the leader. We will study the local feedback gain of the 2^{nd} agent (similar for the 3^{rd} agent). Its conic functioning zone with respect to the leader state is defined in terms of two hyperplanes $h_{2_1}x = k_{2_1}$ and $h_{2_2}x = k_{2_2}$, i.e.

$$\mathcal{H}_2^1 = \left\{ x \in \mathbb{R}^2 \,|H_2 x \leq \gamma_2 \right\}$$

with $H_2 = \begin{bmatrix} h_{2_1} \\ h_{2_2} \end{bmatrix} = \begin{bmatrix} 0.6740 & 0.7387 \\ 0.5120 & -0.0353 \end{bmatrix}$ and $\gamma_2 = \begin{bmatrix} k_{2_1} \\ k_{2_2} \end{bmatrix} = \begin{bmatrix} -1.1302 \\ 0.5 \end{bmatrix}$

In all scenarios below, we represent the functioning zone $\mathcal{H}_2^1$ by a conic set bounded by two dashed black line. The red line denotes the equilibrium line $\bar{x}_2 = A_2\bar{x}_2 + B_2\bar{u}_2$. The red point is the follower's equilibrium point $\bar{x}_2$ of $\mathcal{H}_2^1$ and closest to the leader state. By solving (3.6), we obtain

$$\bar{x}_2 = \begin{bmatrix} -0.6962 \\ -0.8948 \end{bmatrix} \text{ and } \bar{u}_2 = -0.5466$$

The time-domain evolution of the follower is denoted by the blue dot line.

Concerning the local linear control determination, by solving the optimization problem (3.9) for the boundary $h_{2_1}x = k_{2_1}$, we obtain the gain $K_2(h_{2_1}, k_{2_1}) = \begin{bmatrix} 0.2812 & -1.0135 \end{bmatrix}$ for the local linear affine feedback control (3.20) (which is coherent with the expression (3.8)), i.e.

$$u_2 = \bar{u}_2 + K_2(h_{2_1}, k_{2_1})\big(x_2 - \bar{x}_2\big) \tag{3.20}$$

We remind that this local control ensures uniquely the controlled invariance of the half-space $h_{2_1}x \leq k_{2_1}$ but not with the entire $\mathcal{H}_2^1$.

Let us consider in the first scenario the initial point $x_2(0) = \begin{bmatrix} -2 \\ -6 \end{bmatrix}$. Notice that this chosen point is already included in $\mathcal{H}_2^1$, thus the parameterized contractive ellipsoid approach can be applied. Concretely, the optimization problem (3.13) is successfully solved at each iteration, providing all of the ellipsoids centers c (black dots) to which x_2 converges. This result is shown in Fig. 3.7. The agent's state x_2 goes towards the fixed point $\bar{x}_2$ with respect to the slippery of the center c towards $\bar{x}_2$ over the equilibrium line, meanwhile the shape of contractive ellipsoid (yellow filled) converges to the green ellipsoid centered at $\bar{x}_2$.

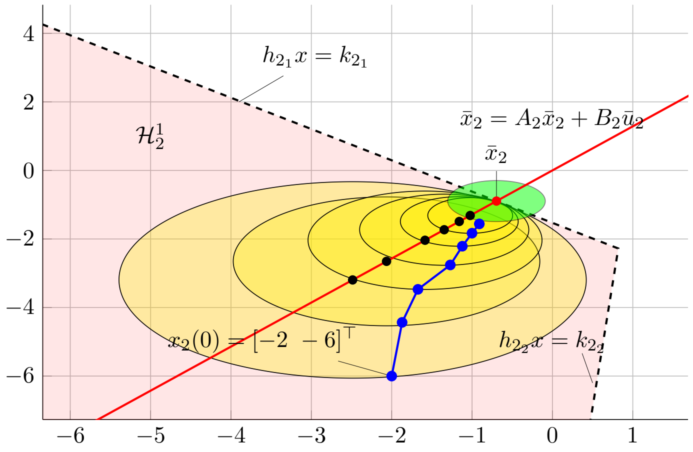

FIGURE 3.7: Evolution of the 2^{nd} agent from $x_2(0) = [-2 \ \ -6]^\top$ using the ellipsoid approach.

In the second scenario, we consider in Fig. 3.8 another initial point $x_2(0) = \begin{bmatrix} 0 \\ -14.1643 \end{bmatrix}$ which resides already on the boundary $h_{2_2}x = k_{2_2}$. We reuse the notations of the first scenario. As a consequence, the choice of $x_2(0)$ in the inactive boundary implies that the ellipsoid approach can not be applied. In fact, there do not exist any non-empty ellipsoid which contains $x_2(0)$ and which is included in the functioning zone $\mathcal{H}_2^1$. In this case, solving the optimization problem (3.13) leads to infeasibility. Moreover, we see that the control (3.20) can not ensure the controlled invariance of $\mathcal{H}_2^1$ even if x_2 still converges to the fixed point. This is illustrated via the jump out of $\mathcal{H}_2^1$ of the follower's state in Fig. 3.8.

In this case, the interpolation-based approach is further applied. Starting from the construction of the maximal output admissible set Ω_2^1 (using the algorithm of Gilbert and Tan (1991)) and the controlled invariant subset $\Phi_2^1 \subseteq \mathcal{H}_2^1$ enclosing Ω_2^1 (solving the optimization problem (3.19)),

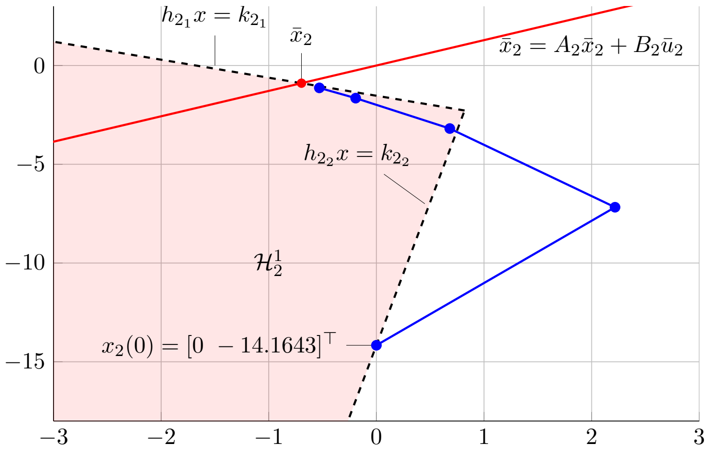

FIGURE 3.8: Evolution of the 2^{nd} agent from $x_2(0) = [0 \ \ -14.1643]^\top$.

the interpolated control action of the 2^{nd} agent is

$$u_2 = \beta_2 u_{\Phi_2^1} + (1-\beta_2)u_{\Omega_2^1},\ 0 \le \beta \le 1$$

with the local control being

$$u_{\Omega_2^1} = \bar{u}_2 + K_2(h_{2_1}, k_{2_1})\big(x_2 - \bar{x}_2\big)$$

Solving in real time respectively (3.19) to construct the set Φ_2^1, (3.16) to get the vertex control $u_{\Phi_2^1}$ and (3.18) to obtain the interpolation factor $\beta_2$for the control u_2. The behavior of the 2^{nd} agent's state x_2 within its functioning zone $\mathcal{H}_2^1$ is given in Fig. 3.9.

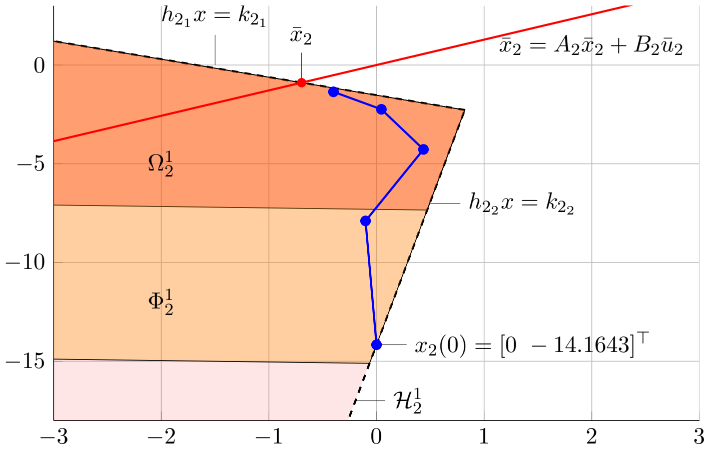

FIGURE 3.9: Evolution of the 2^{nd} agent from $x_2(0) = [0 \ \ -14.1643]^\top$ using the interpolation based approach.

At the beginning, because the initial point $x_2(0)$ is outside Ω_2^1, the interpolated control u_2 is computed such that the follower's state x_2 is moved towards Ω_2^1 meanwhile enclosing the state x_2 inside the set Φ_2^1. When $x_2 \in \Omega_2^1$, the local control $u_{\Omega_2^1}$ is activated as the dominant action to keep the state x_2 strictly inside the MOAS Ω_2^1.

We show the evolution of the interpolation factor β_2 in Fig. 3.10. The case $x_2 \notin \Omega_2^1$ implies $0 < \beta_2 < 1$. At the time instant $k = 3$, the value $\beta_2 = 0$ corresponds to the fact that the state vector x_2 is inside the MOAS Ω_2^1.

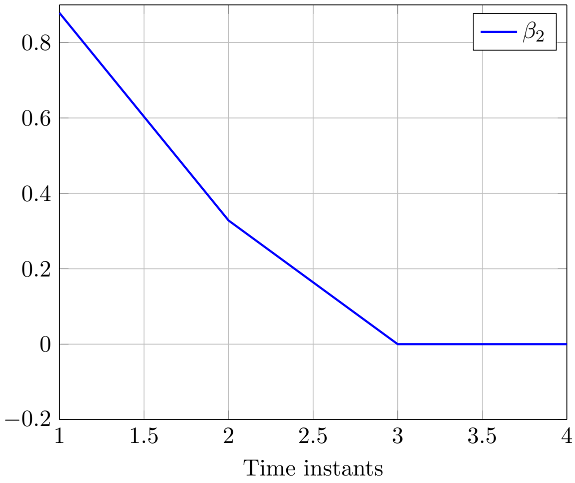

FIGURE 3.10: Evolution of the interpolation factor β_2.

The values of u_2 with the vertex control $u_{\Phi_2^1}$ and the local control $u_{\Omega_2^1}$ are plotted in Fig. 3.11. We see that before $k = 3$, u_2 (blue line) is upper-bounded by the control $u_{\Phi_2^1}$ (red dotted line) and lower-bounded by the local control $u_{\Omega_2^1}$ (green dotted line). After $k = 3$ when $\beta_2 = 0$, the evolution of the control u_2 coincides with the local control $u_{\Omega_2^1}$.

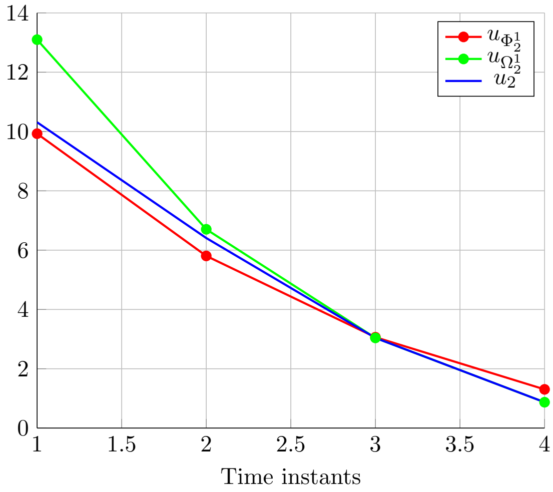

FIGURE 3.11: Interpolated control u_2, vertex control $u_{\Phi_2^1}$ and local control $u_{\Omega_2^1}$.

3.4 Conclusion

This chapter presents a decentralized approach to deal with the collision avoidance of Multi-Agent Systems. The main tool employed is the optimization-based control design with the goal of enforcing the controlled invariance of the safety functioning zone. The main advantage is that the local feasibility can be handled, which is considered novel compared to set-theoretic based decentralized control in the literature. We consider the cases when there exists a local control action to keep each agent in its strict functioning zone. The feasibility of this determination however is limited in a subset of the functioning zone, assimilated to a controlled invariant region via linear feedback. This region is characterized in terms of maximal output admissible set assimilated with the control action on the activated boundaries of the functioning zone. For the initial conditions which are not covered by this set, we propose two possible control strategies to drive the agent state towards the region to activate the local linear control action. The first one is the parameterized contractive ellipsoid approach and the second one is the interpolation based control.

In the first approach, the idea is to guarantee the controlled invariance inside an ellipsoid enclosed by the functioning zone and containing the follower's state. Subsequently, by driving the follower's state towards this ellipsoid's center, we can ensure keeping the follower's state inside the functioning zone.

In the second approach, we construct a controlled invariant set inside the functioning zone, which encloses the maximal output admissible set. The control action of the follower is interpolated from the local control of the maximal output admissible set and a control action associated with the new controlled invariant set. This control is computed such that it is as close as possible to the local control to imply the follower's state staying within the functioning zone.

It should be mentioned that the present framework, the separation hyperplanes between the functioning zones are imposed by the leader via real-time communication. The choice of these hyperplanes is important because they impact the feasibility of the control action and two ideas for further studies are related to the analysis of the maximal violation time in the case of infeasibility. This can be further used in a constraint tightening procedure. It seems possible to update of the separation hyperplanes by enforcing local feasibility. The level of communication (and the information exchanged with the leader) might need to be revisited in this case.

In this chapter, we have presented our constructive solutions for decentralized collision avoidance using set-theoretic methods. These solutions are oriented a priori for application concerning formation control. We emphasized that the collision avoidance is guaranteed by enclosing each agent strictly inside a so-called functioning zone such that these zones are non-overlapping. Next, these zones construction will be detailed in Chapter 4, which leads to a dynamical partition of the Multi-Agent system working space due to the evolution of the agents. Chapter 4 will present also our set-based decentralized control to stabilize the dynamical partition problem.

Chapter 4

Set-based decentralized control for dynamical Multi-Agent deployment

As presented in Chapter 3, the collision avoidance is guaranteed by enclosing each agent strictly inside a so-called functioning zone such that these zones are non-overlapping. The well-known Voronoi partition[1] can be employed in order to build these agents functioning zones and further allocates each Voronoi cell to one unique agent. It is worth to mention that Voronoi partition is obtained with respect to the position of the agents, making the partition time-varying due to the evolution of the agents. This problem is known as *Voronoi-based deployment* or *dynamic Voronoi partition*, which can be found in many Multi-Agent system deployment-based application such as surveillance, environmental/geological monitoring, rescue operations or vehicle routing problems (see Murray (2007), Tanner et al. (2007), Adib Yaghmaie et al. (2015)). The main objective of such cooperative application is to stabilize the Voronoi-based deployment close to a configuration such that the coverage is maximal. The most well-known configuration for the applications such as mobile sensing network (Cortes et al. (2002), Cortes et al. (2005)), multi-robots (Schwager et al. (2009)) or Unmanned aerial vehicles deployment (Bakolas and Tsiotras (2013), Moarref and Rodrigues (2014)) is the *centroidal Voronoi configuration* (CVC), where the position of each agent coincides with the center of mass of its Voronoi cell. Note that a CVC can be obtained by steering each agent individually towards the center of mass of its Voronoi cell. This is totally equivalent to the decentralized control design principle and known as *Lloyd's algorithm* (see Lloyd (1982)). There are many works in the literature related to improving the Lloyd's algorithm, or more specifically, developing decentralized control to drive a Multi-Agent system towards a centroidal Voronoi configuration. Notable recent works in the field of mobile sensors networks are introduced by Cortes et al. (2002), Cortes et al. (2005) and other references therein. The authors present a novel decentralized optimal control which is distributed over the Delaunay graph of a dynamic Voronoi partition, to approach a centroidal Voronoi configuration by assuming that the working region is bounded and additionally the density distribution over this region is time-invariant. The conservation of mass law is used in combination with

[1] The original formulation of Voronoi partition is introduced by Voronoï (1908).

the LaSalle's invariance principle (see Khalil and Grizzle (1996)) to prove the stability of the convergence into a CVC. Inspired by the last works, Moarref and Rodrigues (2014) extend the optimal decentralized control to deal with energy-efficient constraints. Other interesting works in the same direction is presented by Schwager et al. (2009) where different control strategies are given for multi-robots self-deployment, including respectively geometric, probabilistic, and potential field approaches. Kwok and Martinez (2010) propose a generalized power-weighted Voronoi partition and modify the Lloyd's algorithm to solve the power constrained deployment problem.

Instead of using the center of mass[2], the aim of this chapter is to consider other inner target points (e.g. Chebysev center) appropriate to set-theoretic formulation. A different approach based on the vertex interpolated center is presented in Appendix A. The ultimate goal now is to drive the entire Multi-Agent system towards a stabilized configuration where each agent's state coincides with its inner target point. In order to fulfill this goal, we will propose the optimization-based decentralized control to steer each agent towards this chosen target. In addition, the collision avoidance is guaranteed in a decentralized manner by means of the agent inclusion in the corresponding Voronoi cell.

We first formulate the problem in Section 4.1 with the necessary assumptions. Subsequently, we describe a basic approach for the design of solution based on the *Chebyshev center* as the target point in Section 4.2. Its advantage is that the Chebyshev center can be expressed in geometric terms with respect to its associated Voronoi cell. However, whenever the Chebyshev ball computation is not unique (see Section 4.3) the control strategy cannot lead to a stable configuration. In order to keep driving the agents into a stable static configuration, we propose a novel concept based on the computation of a so-called *general center* which leads to a unique center by deflation in the degenerate cases. These results have been detailed in Nguyen et al. (2016c). The overall design is resumed in Section 4.4 through an algorithm for the decentralized control synthesis. Some numerical simulations will be given in Section 4.5 at the end of this chapter to illustrate the performance of the proposed decentralized control.

4.1 Problem formulation

The existence of a stabilized configuration requires that each chosen target point has to be an admissible equilibrium point with respect to the agent's dynamics. Additionally, the whole working region where the agents are deployed needs to be controlled invariant to guarantee the feasibility of the decentralized control action. We consider in the following the assumptions related to the working region and the class of agent's dynamics. In the sequel, we use $\mathcal{Y}(k) = (y_1(k), \dots, y_N(k))$ to denote the tuple containing the aggregated indexed-ordered outputs of the agents. Each of these vectors $y_i(k), i \in \mathbb{N}_{[1,N]}$ are sharing the same $\mathbb{R}^p$ space or equivalently all agent's outputs share the same dimension, i.e. $p_i = p, \forall i \in \mathbb{N}_{[1,N]}$, and thus $\mathcal{Y}(k) \in \mathbb{R}^{Np}$.

[2] Notice that a novel result based on the center of mass is proposed in Appendix B.

Assumption 4. *All agents are deployed in a compact convex set $\mathcal{W} \subset \mathbb{R}^p$ which is denoted as the* working region *and corresponds to the output space shared by the individual agents. It is assumed that for any point $\bar{y}_i \in \mathcal{W}$ there exists a pair $(\bar{x}_i, \bar{u}_i) \in \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$ such that $(\bar{x}_i, \bar{u}_i, \bar{y}_i) \in \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \times \mathbb{R}^{p_i}$ characterize an equilibrium of* (2.1)*. In an equivalent formulation, the system of equations*

$$\begin{cases} \bar{x}_i = A_i \bar{x}_i + B_i \bar{u}_i \\ \bar{y}_i = C_i \bar{x}_i \end{cases} \tag{4.1}$$

is feasible for any $\bar{y}_i \in \mathcal{W}$.

Remark 4.1. Assumption 4 can be satisfied whenever the mapping $\bar{y}_i = C_i(I - A_i)^{-1} B_i \bar{u}_i$ is well-posed and surjective which implies $m_i \geq p_i$ and matrix A_i without eigenvalues at 1. Another particular model description satisfying Assumption 4 is the one corresponding to the integrator-like dynamics (eigenvalues at 1), e.g. for $A_i = I_{n_i}$ for which (4.1) is trivially satisfied.

The basic objective for a deployment problem is to design a control policy at the level of each dynamical agent allowing the coverage of the set $\mathcal{W}$ by the distribution of agents. The link between the classical stability notions and the coverage will be made via set-theoretic methods. The controllability at the level of each agent allows to exploit controlled contractive sets notions. As a particularity of multi-agent dynamics, these contractive sets will be defined in the output space by adapting the classical definitions in Blanchini and Miani (2007).

Definition 4.1. A convex set $\mathbb{V} \subset \mathbb{R}^p$ is controlled λ-contractive with respect to the dynamics of the i^{th} individual agent (2.1) if for any $\bar{y}_i \in \text{int}(\mathbb{V})$ and any admissible x_i satisfying $C_i x_i = y_i \in \mathbb{V}$, it exists u_i ensuring

$$C_i(A_i x_i + B_i u_i) \in \{\bar{y}_i \oplus \lambda(\mathbb{V} \oplus \{-\bar{y}_i\})\}.$$

The contractiveness is related to a scalar $0 \leq \lambda < 1$. If $\lambda = 1$ then the set is controlled-invariant.

The term $\bar{y}_i \oplus \lambda(\mathbb{V} \oplus \{-\bar{y}_i\})$ in Definition 4.1 denote the λ-scaled set of the given set $\mathbb{V}$ with respect to its inner center $\bar{y}_i$. This set is constructed by firstly computing $\mathbb{V} \oplus \{-\bar{y}_i\}$ to move the set $\mathbb{V}$ towards the origin such that it contains the origin inside its interior. According to Remark 2.4, the λ-scaled set $\lambda(\mathbb{V} \oplus \{-\bar{y}_i\})$ will contain also the origin thus we need to re-add the center $\bar{y}_i$ to move this new λ-scaled set back to its real position.

If the set $\mathbb{V}$ in the output space is λ-contractive according to the previous definition, it follows that starting from any point on the boundary of $\mathbb{V}$, there exist a trajectory in $\text{int}(\mathbb{V})$ which can be selected via control. The next definition introduces a generalization which will be shown to be useful in the framework of the multi-agent systems.

Definition 4.2. A convex set $\mathbb{V} \subset \mathbb{R}^p$ is N-step controlled λ-contractive with respect to the dynamics of an individual agent (2.1) if for any $\bar{y}_i \in \text{int}(\mathbb{V})$ and any admissible state $x_i(0)$ satisfying $C_i x_i(0) = y_i \in \mathbb{V}$, it there exists an input sequence $\{u_i(0), \ldots, u_i(N-1)\}$ ensuring

$$y_i(N) \in \{\bar{y}_i \oplus \lambda(\mathbb{V} \oplus \{-\bar{y}_i\})\}.$$

Theorem 4.1. *Let the system (2.1) satisfying Assumptions 1-4 and a bounded set of admissible states $\mathcal{X}$. Then, for any $0 < \lambda < 1$ there exists a finite integer $N(\lambda)$ such that any convex set $\mathbb{V} \subset \mathbb{R}^p$ to be $N-$step controlled λ-contractive.*

Proof. The $N-$step contractivity introduced in Definition 4.2 can be translated from the output-space towards the state-space by including the hypothesis of the theorem. Indeed, $y_i \in \mathbb{V}$ restricts the admissible states to:

$$\mathcal{V}_x = \{x \in \mathcal{X} : C_i x \in \mathbb{V}\}$$

The set $\mathcal{V}_x$ contains a fixed-point $\bar{x}_i$ for each $y_i \in \mathbb{V}$ within its interior, according to the Assumption 4. The controllability assumption together with the boundedness of $\mathcal{V}_x$ leads to the existence of a minimal number of steps N for which any point on the boundary can be driven towards the strict interior of $\mathcal{V}_x$. The proof is complete by observing that the number of steps N depends on the pre-imposed contractivity factor. □

Remark 4.2. In the following, we use $N(\lambda) = 1$ to simplify the description of our methodologies.

Assumption 5. *The agents evolve in a common convex and bounded working space $\mathcal{W}$ which is a proper subset of the output space $\mathbb{R}^p$. This set will be considered to be a polytope represented as the intersection of a set of half-spaces*

$$\mathcal{W} = \{y \in \mathbb{R}^p | Hy \leq \theta\} \tag{4.2}$$

The *global objective* is to control each agent independently so that the global position of the Multi-agent system in $\mathcal{W}$ converges towards a static configuration $\mathcal{Y}^e = \{y_1^e, y_2^e, \dots, y_N^e\}$. This configuration relates the agents' output y_i^e at the equilibrium with an associated *neighborhood* described by a nondegenerate set $\mathbb{V}_i^e \subset \mathcal{W}$. This collection of sets has to cover the working space

$$\mathcal{W} \subseteq \bigcup_{i=1}^{N} \mathbb{V}_i^e$$

Conversely, the output of each agent's dynamics y_i^e represents a relative *center* for its neighborhood $\mathbb{V}_i^e$.

The envisaged state feedback control law $u_i(k)$ will be

- *local (decentralized)*;
- *efficient.*

The *local* characteristic of the control action

$$u_i(k) = \mathcal{K}_i(x_i(k), \mathcal{Y}(k)) \tag{4.3}$$

is understood as a feedback law with respect to the i^{th} agent's state and the distribution of the neighbors at the time instant k. This implies that the design cannot rely on the knowledge of the global states of $\mathbf{\Sigma}$ in (2.1) nor on the communication between the agents with respect to their respective control decisions. The *efficiency* of the control policy will be evaluated in terms of a performance cost function. In the following we provide rigorous notions for *neighborhood* and *center* in order to cast the problem into a mathematical form.

We introduce next the *neighborhood* corresponding to an agent output y_i when this agent is part of the tuple $(y_1, y_2, \ldots, y_N) \in \mathcal{W}^N$. A partition of the working space $\mathbb{V}(y_1, \ldots, y_N)$ will to be computed, by decomposing $\mathcal{W}$ into a union of non-overlapping sets:

$$\begin{aligned} \mathcal{W} = \mathbb{V}(y_1, \ldots, y_N) = \mathbb{V}(\mathcal{Y}(k)) = \bigcup_{i=1}^{N} \mathbb{V}_i(\mathcal{Y}(k)), \\ \mathbb{V}_i(\mathcal{Y}(k)) \cap \mathbb{V}_j(\mathcal{Y}(k)) = \emptyset,\ \forall i, j \in \mathcal{N} \end{aligned} \tag{4.4}$$

The mathematical definition of such a decomposition is provided by the Voronoi partition (see Aurenhammer (1991)), which characterizes the *neighborhood* $\mathbb{V}_i(y_1, \ldots, y_N) = \mathbb{V}_i(\mathcal{Y}(k))$ as

$$\mathbb{V}_i(\mathcal{Y}(k)) = \{y \in \mathcal{W} | \|y_i - y\| \leq \|y_j - y\|, \forall j \neq i\} \tag{4.5}$$

From this definition, it follows $\|y_i - y\| \leq \|y_j - y\|$ and subsequently $2(y_j - y_i)^\top y \leq \|y_j\|^2 - \|y_i\|^2$ and thus

$$\mathbb{V}_i = \left\{ y \in \mathcal{W} | (y_j - y_i)^\top y \leq \frac{\|y_j\|^2 - \|y_i\|^2}{2}, \forall j \neq i \right\} \tag{4.6}$$

It is worth to be mentioned that each set $\mathbb{V}_i$ is a polytope as a consequence of the boundedness of $\mathcal{W}$ and the structure of the constraints in (4.5). Using the available output measurement of the Multi-Agent system $\mathbf{\Sigma}$ (which satisfies the assumption that $y_i(k) \in \mathcal{W}$) at the time instant k, the geometric formulation (4.4) leads to a time-varying partition $\mathbb{V}(\mathcal{Y}(k)) = \bigcup\limits_{i=1}^{N} \mathbb{V}_i(\mathcal{Y}(k))$. The cardinality remains constant in (4.4), as well as the structure of constraints in (4.6), and thus each point is uniquely associated to a set (called *neighborhood*) within the partition

$$y_i(k) \longleftrightarrow \mathbb{V}_i(\mathcal{Y}(k)) \tag{4.7}$$

Remark 4.3. Only the neighbor agents contribute to the definition of $\mathbb{V}_i(\mathcal{Y}(k))$ and the information needed for its contraction is local. This is obvious from equation (4.6) where part of the inequalities defininig the set $\mathbb{V}_i$ are redundant. Denoting by $\mathcal{N}_{\mathcal{Y}}^i(k)$ the subset of vectors in $\mathcal{Y}(k)$ involved in active (non-redundant) constraints for a given i in the expression (4.6), the association (4.7) becomes:

$$y_i(k) \longleftrightarrow \mathbb{V}_i(\mathcal{Y}(k)) = \mathbb{V}_i(\mathcal{N}_{\mathcal{Y}}^i(k)) \tag{4.8}$$

This remark has an implication on the structure of the control law (4.3) whenever this is designed based on the information coming from the Voronoi cell (defininig the neighborhood)

$$u_i(k) \quad = \quad \mathcal{K}_i(x_i(k), \mathcal{Y}(k))$$

or an equivalent formulation with an appropriate control function

$$u_i(k) = \mathcal{K}_i(x_i(k), \mathbb{V}_i(\mathcal{Y}(k)))$$

$$u_i(k) = \mathcal{K}_i(x_i(k), \mathbb{V}_i(\mathcal{N}^i_{\mathcal{Y}}(k)))$$
$$u_i(k) = \mathcal{K}_i(x_i(k), \mathcal{N}^i_{\mathcal{Y}}(k))$$

enforcing his local characteristics.

With the elements above, the control objective can be formulated readily: drive the global Multi-Agent system towards a static configuration by ensuring the convergence of each agent position towards the central point within its Voronoi cell.

The present work builds on the well-established notion of the *Chebyshev center* (see Boyd and Vandenberghe (2004)). There are structural and computation arguments for such a choice. This becomes apparent in the description of the Chebyshev center $\bar{y}_i(k)$ and the constraints defining $\mathbb{V}_i(\mathcal{N}^i_{\mathcal{Y}}(k))$, which are available via a convex optimization problem[3]:

$$\begin{aligned}(\bar{y}_i(k), \bar{r}_i(k)) = \arg \max_{\bar{y}_i(k), \bar{r}_i(k)} \; & \bar{r}_i(k) \\ \text{s.t.: } 2(y_j(k) - y_i(k))^\top \bar{y}_i(k) &+ 2\|y_j(k) - y_i(k)\|\bar{r}_i(k) \\ \leq \|y_j(k)\|^2 &- \|y_i(k)\|^2, \; \forall y_j(k) \in \mathcal{N}^i_{\mathcal{Y}}(k)\end{aligned} \tag{4.9}$$

This optimization involves constraints related to the neighboring agents $\mathcal{N}^i_{\mathcal{Y}}(k)$ and thus comply with the need of a *local* (decentralized) decision making for the control synthesis (as described in the next sections).

The explicit form of $\bar{y}_i(k)$ and $\bar{r}_i(k)$ can be obtained by exploiting the Karush-Kuhn-Tucker conditions of the parametric optimization problem (with $\mathcal{N}^i_{\mathcal{Y}}(k)$ as parameter vector):

$$\bar{y}_i(k) \quad = \quad f(\mathcal{Y}(k)) \tag{4.10}$$
$$\bar{r}_i(k) \quad = \quad g(\mathcal{Y}(k)) \tag{4.11}$$

The functions $f(.), g(.)$ are continuous with respect to a set of points $\mathcal{N}^i_{\mathcal{Y}}(k)$. Note however that the set of neighbours is time-varying in itself and can lead to a variation of the cardinalty of $\mathcal{N}^i_{\mathcal{Y}}(k)$. Thus in order to represent the time-varying nature of the Voronoi partition, a generic dependence on $\mathcal{Y}(k)$ has to be used in (4.10)-(4.11). Note also that the functions $f(.), g(.)$ may be discontinuous with respect to $\mathcal{Y}(k)$ by the same argument. Globally, the dynamics we are

[3] See subsection 2.4.2.2 for more details of determining the Chebyshev center and radius of a bounded polyhedron via the half-spaces defining this polyhedron.

interested in is resumed by the equation:

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i \mathcal{K}_i(x_i(k), \mathcal{N}^i_{\mathcal{Y}}(k)) \\ y_i(k) = C_i x_i(k) \\ \bar{y}_i(k) = f(\mathcal{Y}(k)) \end{cases} \tag{4.12}$$

The objective is to design of a decentralized *local* feedback-control law $u_i(k) = \mathcal{K}_i(x_i(k), \mathcal{N}^i_{\mathcal{Y}}(k))$ for each agent such that

$$\lim_{k \to \infty} y_i(k) - \bar{y}_i(k+1) = 0,$$

which implicitly corresponds to the convergence towards a static configuration which address the deployment problem.

The explicit form of this local control law will be analyzed in the next section, along with the stability proof related to the convergence of the Multi-Agent system towards a stabilized configuration.

4.2 Basic solution analysis

In the following, we propose a basic control solution for the Multi-Agent system deployment, based on the computation of the Chebyshev center and radius. The explicit form of this control solution will be given in subsection 4.2.1 with the notations denoting the distance from the agent's output to a frontier of its Voronoi cell. The convergence proof associated with this chosen decentralized control will be discussed in subsection 4.2.2.

4.2.1 Chebyshev radius tracking

For each agent's output y_i with $i \in \mathbb{N}_{[1,N]}$, we define a set $\mathcal{R}_i \subset \mathbb{R}_+$ which collects the distance from the agent's output y_i to the hyperplanes forming its Voronoi cell

$$\mathcal{R}_i = \{r \in \mathbb{R}_+ | r = \min ||y - y_i||, \ y \in \mathbb{V}_i \cap \mathbb{V}_j, \forall j \in \mathcal{N}_i\} \tag{4.13}$$

with $\mathbb{V}_i$ and $\mathbb{V}_j$ denoting respectively the Voronoi cell of the i^{th} agent and its neighbor. The set $\mathcal{N}_i$ contains the neighbors indices of the i^{th} agent. Obviously, $\mathcal{R}_i$ is also time-varying whenever $y_i(k)$ or $\mathbb{V}_i(k)$ is time-varying.

In the general case, the distance from the agent state y_i to a hyperplane is half of the distance between the positions of this agent and of the agent in its neighbor sharing this hyperplane. Hence we can define the finite set of distance $\mathcal{R}_i$ as

$$\mathcal{R}_i = \{r \in \mathbb{R}_+ | r = \frac{1}{2} \min ||y_j - y_i||, \forall j \in \mathcal{N}_i\} \tag{4.14}$$

Let us define

$$r_i^m(k) = \min \mathcal{R}_i(k) \tag{4.15a}$$
$$r_i^M(k) = \max \mathcal{R}_i(k) \tag{4.15b}$$

which characterize respectively the distance from the agent state x_i to the closest and the farthest hyperplanes. The following expressions depending explicitly on time are derived

$$r_i^m(k) \leq \bar{r}_i(k) \leq r_i^M(k) \tag{4.16}$$

with $\bar{r}_i$ the radius of the Chebyshev ball. We will illustrate the distance notations above in Example 4.1.

Example 4.1. Four agents are deployed within a bounded working space

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} -10 \\ -10 \end{bmatrix}, \begin{bmatrix} 15 \\ -10 \end{bmatrix}, \begin{bmatrix} 15 \\ 10 \end{bmatrix}, \begin{bmatrix} -10 \\ 10 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

as shown in Fig. 4.1. The blue points are the agents outputs y_i and the red points are their Chebyshev centers $\bar{y}_i$. The outputs of the agents are respectively $y_1 = \begin{bmatrix} -2 \\ 4 \end{bmatrix}$, $y_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$, $y_3 = \begin{bmatrix} -7 \\ -6 \end{bmatrix}$ and $y_4 = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$. The Voronoi partition corresponding to these agents outputs are shown in Fig. 4.1. The Chebyshev centers of the agents are respectively

$$\bar{y}_1 = \begin{bmatrix} -3.925 \\ 5.108 \end{bmatrix}, \; \bar{y}_2 = \begin{bmatrix} 7.737 \\ 1.535 \end{bmatrix}, \; \bar{y}_3 = \begin{bmatrix} -7.135 \\ -7.135 \end{bmatrix}, \; \bar{y}_4 = \begin{bmatrix} -0.156 \\ -6.207 \end{bmatrix}$$

Let consider the 2^{nd} agent. The dash lines denote the distances r_2 from this agent output y_2 to the facets of its Voronoi cell $\mathbb{V}_2$. The Chebyshev radius $\bar{r}_2$, the minimal distance to a frontier r_2^m and the maximal distance to a frontier r_2^M are also indicated.

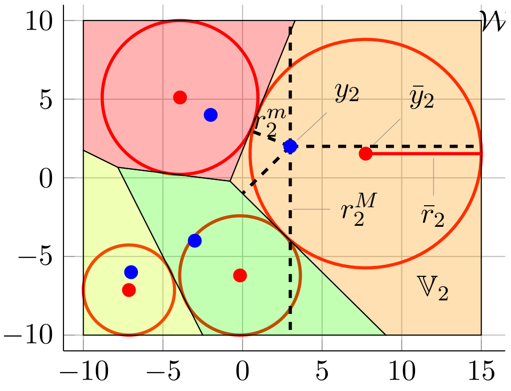

FIGURE 4.1: Deployment over a region $\mathcal{W}$ with the distance notation.

Furthermore, after the partition step, each agent is driven individually towards its Chebyshev center. This goal can be achieved by using the decentralized control

$$u_i(k) = \bar{u}_i(k) + \mathcal{K}_i(\bar{x}_i(k), x_i(k)) \tag{4.17}$$

and, in particular, we privilege the linear form

$$u_i = \bar{u}_i(k) + K_i(x_i - \bar{x}_i) \tag{4.18}$$

The definitions of a Static Configuration of MAS is further introduced in order to describe the limit behavior.

Definition 4.3. A Static Configuration (SC) of the MAS $\mathbf{\Sigma}$ is achieved whenever $u_i(k) = u_i(k_0)$ and $x_i(k) = x_i(k_0)$, $\forall k \geq k_0$.

Additionally, Chebyshev Configuration (CC) is already defined in Definition 2.13.

Remark 4.4. Not any SC is CC as it can be observed by choosing the null input $u_i = \bar{u}_i$ for a configuration where $y_i \neq \bar{y}_i$. Conversely, a CC is not necessary a SC for the homogeneous agents with $u_i \neq \bar{u}_i$, with $\forall i \in \mathcal{N}$.

Proposition 4.2. *The Multi-Agent system $\mathbf{\Sigma}$ in closed-loop with a set of decentralized control laws $u_i = \bar{u}_i(k) + K_i(x_i - \bar{x}_i)$ achieves a SC if and only if this is a CC.*

Proof. First we observe that for $x_i(k) = \bar{x}_i(k)$ the control action is $u_i(k) = \bar{u}_i(k)$ and thus it characterizes an equilibrium. Conversely, if $x_i(k) \neq \bar{x}_i(k)$ based on the controllability assumption, the control action $u_i(k) \neq \bar{u}_i$ and $x_i(k+1) \neq x_i(k)$, thus invalidating the assumption of static configuration. □

We analyze next the time-varying configuration and the convergence toward a so-called *Chebyshev Static Configuration* (CSC) which mixes the Static Configuration and Chebyshev Configuration notions.

Remark 4.5. Note that a CSC is not unique and depends on the initial agent state and implicitly on the control policies applied on the agents (feedback gain).

The convergence of MAS deserves a particular attention as long as the ultimate Static Configuration is not known a priori, and it will be analyzed in the next subsection.

4.2.2 Convergence proof

As mentioned above, a Chebyshev Static Configuration can be obtained by using the decentralized control (4.17) ensuring the convergence of each agent towards its Chebyshev center. In this subsection, we will complete this result with the stability proof.

Consider the following semi-positive function

$$V(\mathcal{Y}(k)) = \sum_{i\in\mathcal{N}} (\bar{r}_i(k) - r_i^m(k)) \tag{4.19}$$

From the first part of the inequality in (4.16), it follows that each term of the sum in (4.19) is positive and globally $V(\mathcal{X}(k)) \geq 0$. Another structural property inherited from the definition of the Chebyshev radius is resumed by Proposition 4.3.

Proposition 4.3. *A stationary Chebyshev configuration of* $\mathbf{\Sigma}$ *over the working space* $\mathcal{W}$ *is achieved if the function* $V(\mathcal{Y}(k)) = 0$.

Proof. The configuration at equilibrium is obtained when the output of each agent coincides with its equilibrium Chebyshev center, i.e. $y_i(k) = \bar{y}_i(k)$ and $u_i(k) = \bar{u}_i(k)$. This leads to $\bar{r}_i(k) = r_i^m(k)$, or equivalently the value function (4.19) becomes $V(\mathcal{Y}(k)) = 0$. □

Briefly, the convergence of the Multi-Agent system $\mathbf{\Sigma}$ towards a stationary configuration is analyzed on the basis of the value function (4.19). The approach is based on the monotonic decrease (or at least non-increase) of $V(\mathcal{X}(k))$ along the trajectories of $\mathbf{\Sigma}$. Consider

$$\begin{aligned} V(\mathcal{Y}(k+1)) - V(\mathcal{Y}(k)) &= \sum_{i\in\mathcal{N}} [(\bar{r}_i(k+1) - r_i^m(k+1)) - (\bar{r}_i(k) - r_i^m(k))] \\ &= \sum_{i\in\mathcal{N}} (\bar{r}_i(k+1) - \bar{r}_i(k)) + \sum_{i\in\mathcal{N}} (r_i^m(k) - r_i^m(k+1)) \end{aligned} \tag{4.20}$$

Remark 4.6. Denoting by $\theta_i(k) = r_i^m(k) - r_i^m(k+1)$, we remark that $\theta_i(k) = (\bar{r}_i(k) - r_i^m(k+1)) - (\bar{r}_i(k) - r_i^m(k))$. By using any control action ensuring the convergence of the i^{th} agent's minimal distance to a frontier r_i^m towards its Chebyshev radius $\bar{r}_i$, we can derive that $r_i^m(k+1) \geq r_i^m(k)$ yielding $(\bar{r}_i(k) - r_i^m(k+1)) \leq (\bar{r}_i(k) - r_i^m(k))$. Thus, $\theta_i(k) \leq 0$ is verified.

The monotonicity of $V(\mathcal{Y}(t))$ is not straightforward by analyzing equation (4.20) but it is possible to study the non increasing property over a longer horizon N_p. Extending the differences over N_p time steps forward yields

$$\begin{aligned} V(\mathcal{Y}(k+1)) - V(\mathcal{Y}(k)) &= \sum_{i\in\mathcal{N}} (\bar{r}_i(k+1) - \bar{r}_i(k)) + \sum_{i\in\mathcal{N}} \theta_i(k) \\ V(\mathcal{Y}(k+2)) - V(\mathcal{Y}(k+1)) &= \sum_{i\in\mathcal{N}} (\bar{r}_i(k+2) - \bar{r}_i(k+1)) + \sum_{i\in\mathcal{N}} \theta_i(k+1) \\ &\vdots \\ V(\mathcal{Y}(k+N_p)) - V(\mathcal{Y}(k+N_p-1)) &= \sum_{i\in\mathcal{N}} (\bar{r}_i(k+N_p) - \bar{r}_i(k+N_p-1)) + \sum_{i\in\mathcal{N}} \theta_i(k+N_p-1) \end{aligned} \tag{4.21}$$

Summing up these equations, we obtain

$$V(\mathcal{Y}(k+N_p)) - V(\mathcal{Y}(k)) \quad = \sum_{i\in\mathcal{N}} (\bar{r}_i(k+N_p) - \bar{r}_i(k)) + \sum_{l=0}^{N_p-1} \sum_{i\in\mathcal{N}} \theta_i(k+l) \tag{4.22}$$

Lemma 4.4. *The sum* $\sum_{i\in\mathcal{N}}(\bar{r}_i(k+N_p)-\bar{r}_i(k))$ *is upper bounded by* $N\cdot\gamma(\mathcal{W})$*, with* $\gamma(\mathcal{W})$ *denoting the radius of the largest ball enclosed in the working region* $\mathcal{W}$*.*

Proof. One has $0 \leq \bar{r}_i \leq \gamma(\mathcal{W})$. The upper bound $\gamma(\mathcal{W})$ is obvious from the boundedness assumptions on $\mathcal{W}$. The lower bound is derived by considering the worst case where all the agents positions are superposed which leads to $r_i^m = 0$. Summing up the N agents Chebyshev radius yields $0 \leq \sum_{i\in\mathcal{N}}\bar{r}_i \leq N\cdot\gamma(\mathcal{W})$ and finally $\sum_{i\in\mathcal{N}}(\bar{r}_i(k+N_p)-\bar{r}_i(k)) \leq \sum_{i\in\mathcal{N}}\bar{r}_i \leq N\cdot\gamma(\mathcal{W})$. □

Proposition 4.5. *The following equality holds* $\lim_{s\to\infty}\theta_i(k+s) = 0,\ \forall i \in \mathcal{N}$*.*

Proof. Lemma 4.4 ensures the boundedness of the first term in (4.22). Consider the remaining term $\sum_{l=0}^{N_p-1}\sum_{i\in\mathcal{N}}\theta_i(k+l)$ in 4.4 which depends uniquely on the distance between the agents. The limit $\lim_{s\to\infty} V(\mathcal{Y}(k+s))-V(\mathcal{Y}(k))$ is bounded if and only if the limit of the sum $\lim_{s\to\infty}\sum_{l=0}^{s}\left(\sum_{i\in\mathcal{N}}\theta_i(k+l)\right)$ exists and is bounded. Moreover, the existence of this bounded limit $\lim_{s\to\infty}\sum_{l=0}^{s}\left(\sum_{i\in\mathcal{N}}\theta_i(k+l)\right)$ implies the convergence of each terms towards zero, i.e. $\lim_{l\to\infty}\sum_{i\in\mathcal{N}}\theta_i(k+l) = 0$ or more precisely $\lim_{l\to\infty}\sum_{i\in\mathcal{N}}\theta_i(l) = 0$, according to the theory of series (see Brabenec (2004)), otherwise the convergence is not guaranteed. Furthermore, due to the expression $\theta_i(k) = r_i^m(k) - r_i^m(k+1)$ in Remark 4.6, the sum $\sum_{i\in\mathcal{N}} r_i^m$ becomes stationary. □

Corollary 4.6. *If the Chebyshev center associated to each Voronoi cell is unique, the Multi-Agent system converges to a Chebyshev Static Configuration.*

Proof. Proposition 4.5 proves the convergence of r_i^m towards $\bar{r}_i$ which is obtained when the output of each agent coincides with its equilibrium Chebyshev center, i.e. $y_i(k) = \bar{y}_i(k)$ and $u_i(k) = \bar{u}_i(k)$ if the Chebyshev center is uniquely associated to the Chebyshev radius $\bar{r}_i$. Thus in the limit case, $r_i^m = \bar{r}_i$ corresponds to a Chebyshev Static Configuration. □

Next, Example 4.2 intends to illustrate the analysis above on a simple 2D numerical case study involving only 2 agents for the simplicity of the graphical illustration.

Example 4.2. Consider two mobile agents deployed in a bounded region $\mathcal{W} \subset \mathbb{R}^2$ as shown in Fig. 4.2. Their output's evolution $y_i(k)$ is marked by the blue points. The red points denote their Chebyshev centers $\bar{x}_i(3)$ at instant $k = 3$. According to equation (4.14), we have $r_1(k) = r_2(k) = \frac{1}{2}||y_1(k) - y_2(k)||$. The initial positions satisfy $r_1^m(0) = r_2^m(0) = \frac{1}{2}||y_1(0) - y_2(0)||$. Each agent has its own local control action which steers the agent position towards its Chebyshev center. As illustrated in Fig. 4.2, at $k = 3$, the distance $\frac{1}{2}||y_1 - y_2||$ approaches $\frac{1}{2}||\bar{y}_1 - \bar{y}_2||$ due to the decrease of the distance between the position of the agent and its Chebyshev center.

Asymptotically, r_i^m becomes constant. Furthermore, the value function (4.19) applied for these two agents has $V(\mathcal{Y}(k)) - V(\mathcal{Y}(0))$ bounded for $k \geq 3$.

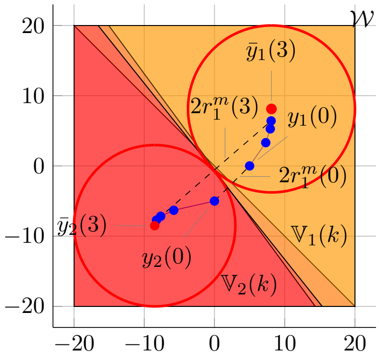

FIGURE 4.2: Convergence of two agents to their Chebyshev center.

In conclusion, this section proves that local control laws can be designed to decrease the distance between the minimal distance r_i^m and the Chebyshev radius and globally to drive the Multi-Agent system $\boldsymbol{\Sigma}$ toward a static configuration whenever the Chebyshev center is unique. The convergence proof enforces the decrease of the difference between these distances and can be generalized for the non-unique Chebyshev center case as detailed in the following section.

4.3 Solution analysis and generalization

We will show in subsection 4.3.1 some counter examples related to the non-uniqueness of the Chebyshev center computation, due to the geometry of the considered polyhedron. Other center chosen to overcome the Chebyshev computation problem will be presented in subsection 4.3.2.

4.3.1 Non-uniqueness of the Chebyshev center computation

For a given polyhedron, the feasibility of the optimization problem (4.9) is guaranteed by the non-emptiness and the boundedness properties of this polyhedron. The solution may not be unique.

Example 4.3. Consider the blue polyhedron $\mathbb{V}$

$$\mathbb{V} = \text{conv}\left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ 80 \end{bmatrix}, \begin{bmatrix} 0 \\ 80 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

as shown in Fig. 4.3. Obviously, every circle having the center on the yellow segment bounded by two points $\begin{bmatrix} 40 \\ 40 \end{bmatrix}$ and $\begin{bmatrix} 60 \\ 40 \end{bmatrix}$ is the largest circle enclosed inside the polyhedron $\mathbb{V}$. Thus, every point lying on the yellow segment is a feasible Chebyshev center of the polyhedron $\mathbb{V}$. The Chebyshev center is then non unique in case of such geometry of $\mathbb{V}$.

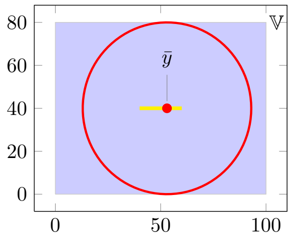

FIGURE 4.3: Non unique Chebyshev center.

Starting from such a configuration, it is possible to construct the following counterexample where the convergence towards a unique Chebyshev Static Configuration of a given MAS is not accomplished due to the non-uniqueness of the Chebyshev computation.

Example 4.4. Consider three agents deployed within a bounded region $\mathcal{W}$

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} -110 \\ -20 \end{bmatrix}, \begin{bmatrix} 110 \\ -20 \end{bmatrix}, \begin{bmatrix} 110 \\ 20 \end{bmatrix}, \begin{bmatrix} -110 \\ 20 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

as in Fig. 4.4. The real positions and the Chebyshev centers are represented respectively by the red and blue points.

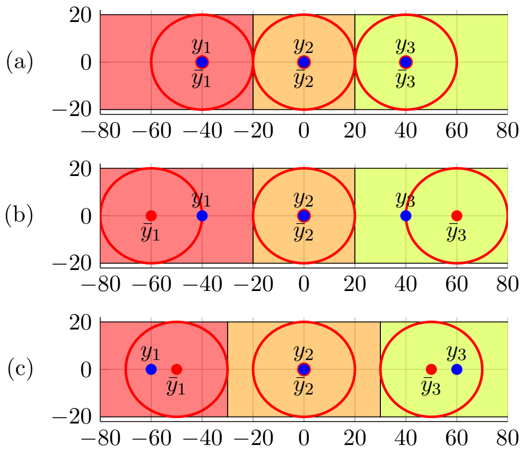

FIGURE 4.4: Oscillation-like behavior of 3 agents.

In Fig. 4.4 (a), each agent is positioned at its Chebyshev center and thus the global system is in a CSC, i.e.

$$y_1 = \bar{y}_1 = \begin{bmatrix} -40 \\ 0 \end{bmatrix}, \ y_2 = \bar{y}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ y_3 = \bar{y}_3 = \begin{bmatrix} 40 \\ 0 \end{bmatrix}$$

However, due to the non-uniqueness of the Chebyshev centers of the 1^{st} and 3^{rd} agents, a corrective control action will be initiated whenever the second configuration (Fig. 4.4 (b)) is issued by the Chebyshev center computation in (4.9). In Fig. 4.4 (b), the Chebyshev centers are

$$\bar{y}_1 = \begin{bmatrix} -60 \\ 0 \end{bmatrix}, \ \bar{y}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \bar{y}_3 = \begin{bmatrix} 60 \\ 0 \end{bmatrix}$$

and the agents positions are

$$y_1 = \begin{bmatrix} -40 \\ 0 \end{bmatrix}, \; y_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; y_3 = \begin{bmatrix} 40 \\ 0 \end{bmatrix}$$

Subsequently, the agents start moving in the opposite direction and then further evolve to the configuration 3 (Fig. 4.4 (c)) where the non-uniqueness of the Chebyshev center is exhibited for all the Voronoi cells in the partition. In Fig. 4.4 (c), the Chebyshev centers are

$$\bar{y}_1 = \begin{bmatrix} -50 \\ 0 \end{bmatrix}, \; \bar{y}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; \bar{y}_3 = \begin{bmatrix} 50 \\ 0 \end{bmatrix}$$

and the agents positions are

$$y_1 = \begin{bmatrix} -60 \\ 0 \end{bmatrix}, \; y_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; y_3 = \begin{bmatrix} 60 \\ 0 \end{bmatrix}$$

Hence the CSC is no longer maintained and all agents enter into an oscillation-like behavior according to the choice of the respective time-varying centers.

This phenomenon is caused by the non-uniqueness of the Chebyshev Configuration. This problem will be encountered for any choice of inner target points unless the non-uniqueness is enforced. The next subsection presents how to avoid this problem and to keep driving the agents into a stationary configuration.

4.3.2 General solution

The generalization of the basic result presented in the previous section is based on a *deflation* process. The control law will exploit the contractive control with respect to the Pontryagin difference (see Definition 2.6) between the cell $\mathbb{V}_i$ and the ball of radius r_i^m. Thus the Chebyshev center is replaced by a so-called *general center* inside the Voronoi cell, which is defined below.

Definition 4.4. Given a bounded convex polyhedron $\mathbb{V} \subset \mathbb{R}^n$, the general center $\bar{y}$ of $\mathbb{V}$ is defined such as

$$\bar{y} \oplus \bar{\mathbb{B}}^1_{r_1} \oplus \ldots \oplus \bar{\mathbb{B}}^p_{r_p} \subset \mathbb{V} \tag{4.23}$$

where the ball $\bar{\mathbb{B}}^i_{r_i}, i \in \mathbb{N}_{[1,p]}$ centered at the origin is defined as:

- $\bar{\mathbb{B}}^p_{r_p} \subset \mathbb{R}^p$ is the Chebyshev ball of the original set $\mathbb{V}$ defined as the ball centered in the origin, with the radius equal with the Chebyshev radius computed according to (4.9);
- $\bar{\mathbb{B}}^q_{r_q} \subset \mathbb{R}^p$, for any $q < p$, denotes the *q-dimensional* Chebyshev ball in the relative interior of the set $\mathbb{V} \ominus \bar{\mathbb{B}}^p_{r_p} \ominus \ldots \ominus \bar{\mathbb{B}}^{q+1}_{r_{q+1}}$.

Remark 4.7. The set $\mathbb{V} \ominus \bar{\mathbb{B}}^p_{r_p} \ominus \ldots \ominus \bar{\mathbb{B}}^{q+1}_{r_{q+1}}$ with $q = 1, \ldots, p-1$ describes the deflation of $\mathbb{V} \subset \mathbb{R}^p$ into a q*-dimensional* subspace. For $q = 1$, the set $\mathbb{V}$ is degenerated into a point y_c called the *general center* of the cell $\mathbb{V}$.

Example 4.5. Let us consider the green polyhedron $\mathbb{V} \subset \mathbb{R}^3$ in Fig. 4.5. Its general center is determined iteratively according to Remark 4.7. Respectively, the set $\mathbb{V}$ is deflated into:

- a bounded surface (*2-dimensional*, blue) via the Pontryagin difference $\mathbb{V} \ominus \bar{\mathbb{B}}^3_{r_3}$;
- a line segment (*1-dimensional*, yellow) by means of $\mathbb{V} \ominus \bar{\mathbb{B}}^3_{r_3} \ominus \bar{\mathbb{B}}^2_{r_2}$;
- a unique point $\bar{y}$ (red) by means of $\mathbb{V} \ominus \bar{\mathbb{B}}^3_{r_3} \ominus \bar{\mathbb{B}}^2_{r_2} \ominus \bar{\mathbb{B}}^1_{r_1}$, which is the *general center* of $\mathbb{V}$.

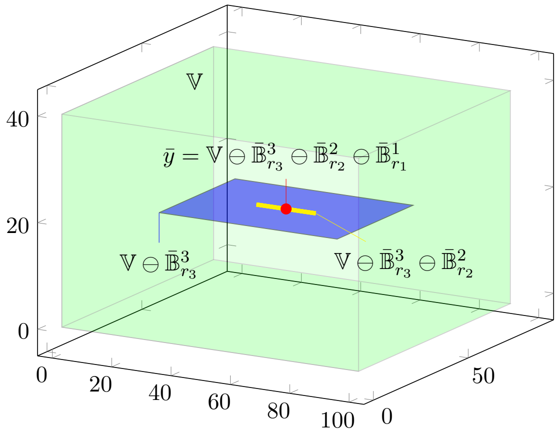
$\mathbb{V}$
$\bar{y} = \mathbb{V} \ominus \bar{\mathbb{B}}^3_{r_3} \ominus \bar{\mathbb{B}}^2_{r_2} \ominus \bar{\mathbb{B}}^1_{r_1}$
$\mathbb{V} \ominus \bar{\mathbb{B}}^3_{r_3}$
$\mathbb{V} \ominus \bar{\mathbb{B}}^3_{r_3} \ominus \bar{\mathbb{B}}^2_{r_2}$
40
20
0
0
20
40
60
80
100
0
50

FIGURE 4.5: Degeneracy of polyhedron $\mathbb{V}$ into the general center $\bar{y}$.

The control objective now is to drive the global Multi-Agent system into a *General Static Configuration* (GSC), by steering at each time instant the agent output towards its associated general center. The computation of a general center is well-posed and unique, based on the local information (Voronoi cell).

The following notation is used to present the dimensional-level of deflation while finding the general center for an i^{th} agent. We use $\mathbb{V}^q_i(k) \subset \mathbb{R}^p$ to denote the deflation of $\mathbb{V}_i(k) \subset \mathbb{R}^p$ in the q*-dimensional* subspace. The set $\mathbb{V}^q_i(k)$ is computed based on the recurrence

$$\mathbb{V}^{q-1}_i(k) = \mathbb{V}^q_i(k) \ominus \bar{\mathbb{B}}^q_{r_q,i}(k) \tag{4.24}$$

with the initial condition

$$\mathbb{V}^p_i(k) = \mathbb{V}_i(k) \tag{4.25}$$

The equations (4.24)-(4.25) imply the uniqueness of the general center with respect to Definition 4.4. It remains to find the appropriate decentralized control to steer each agent towards its general center meanwhile guarantee the λ-contractive of its Voronoi cell with respect to the agent's dynamics and the inner general center. We propose the following definition to describe the set of decentralized control satisfying these objectives.

Proposition 4.7. *The convergence towards a General Static Configuration of the Multi-Agent system* $\mathbf{\Sigma}$ *can be achieved in a decentralized manner by using* $u_i(k) \in \mathcal{U}(y_i(k), \mathbb{V}_i(k))$, *where*

$$\mathcal{U}(y_i, \mathbb{V}_i) = \{u_i \in \mathbb{R}^{m_i} | \exists \lambda \in \mathbb{R}_{[0,1)} \text{ s.t. } C_i(A_i x_i + B_i u_i) \in \bar{y}_i(\mathbb{V}_i) \oplus \lambda(\mathbb{V}_i \oplus \{-\bar{y}_i(\mathbb{V}_i)\})\} \tag{4.26}$$

Sketch of proof. The statement is a generalization of Proposition 4.5, enhanced by the intrinsic uniqueness properties of the general center as introduced in Definition 4.4. The uniqueness follows by the fact that $\bar{\mathbb{B}}^p_{r_p}$ are uniquely defined and by the fact that for $p = 1$ the associated Chebyshev center is unique. This proves the existence of a General Static Configuration.

The set $\mathcal{U}$ characterizes the collection of control actions which ensures the one-step λ-contractiveness (see Definition 4.1) of the cell $\mathbb{V}_i$ with respect to the agent's dynamics (2.1). The closed-loop dynamics consequently ensures the convergence of the agent's output y_i towards the general center $\bar{y}_i$ representing an equilibrium point of the agent (according to Assumption 4), implying the non increasing property of the value function (4.19). This is guaranteed by means of the convergence of the term $\theta_i(k) = r_i^m(k) - r_i^m(k+1)$ (see Remark 4.6) towards zero, leading to the convergence of the minimal distance to a frontier r_i^m towards a stationary value in virtue of Proposition 4.5. Therefore, the stability proof is completed. □

In this section, we give some counterexample showing the non-uniqueness of the Chebyshev center computation and further provide a so-called general center based on deflation principle. The main advantage of this chosen center is that its uniqueness is ensured by deflation construction. At the end of this section, we present the basic requirements to determine the set of decentralized control to steer each agent's output towards its center. The specific design of decentralized control will be detailed in the next section.

4.4 General decentralized control

In the previous section, Proposition 4.7 offers the theoretical framework for the selection of a stabilizing decentralized control for MAS. In order to address the efficiency of the control policies, an optimization-based synthesis can be formulated. At each sampling instant the next convex optimization problem can be solved for the i^{th} agent

$$u_i(k) = \arg\min_{u_i(k)} \|y_i(k+1) - \bar{y}_i(k)\|_Q^2 + \|u_i(k) - \bar{u}_i(k)\|_R^2 \tag{4.27a}$$

$$\text{s.t.:} \quad \bar{x}_i(k) = A_i \bar{x}_i(k) + B_i \bar{u}_i(k) \tag{4.27b}$$

$$\bar{y}_i(k) = C_i \bar{x}_i(k) \tag{4.27c}$$

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) \tag{4.27d}$$

$$y_i(k+1) = C_i x_i(k+1) \tag{4.27e}$$

$$u_i(k) \in \mathcal{U}(y_i(k), \mathbb{V}_i(k)) \tag{4.27f}$$

with $i \in \mathbb{N}_{[1,N]}$. The weighting matrices $Q \in \mathbb{R}^{p_i \times p_i}$, $R \in \mathbb{R}^{m_i \times m_i}$ in the cost function (4.27a) are chosen such as $Q = Q^\top \succeq 0$ and $R = R^\top \succeq 0$, which can be adjusted to offer the balance between the tracking of the center $\bar{y}_i(k)$ and the control effort. The constraints (4.27b) and (4.27c) are used to compute $(\bar{x}_i(k), \bar{u}_i(k))$ from $\bar{y}_i(k)$. The constraints (4.27d)-(4.27e) are employed to predict the one-step forward value of the agent's output $y_i(k+1)$, state $x_i(k+1)$ with respect to the dynamics (2.1) and the current state $x_i(k)$ and output $y_i(k)$. The last constraint (4.27f) means that $u_i(k)$ ensures the λ-contractiveness of $\mathbb{V}_i(k)$.

If the solution of (4.27) cannot ensure the one-step-forward inclusion of the agent's output $y_i(k+1) = C_i(A_i x_i(k) + B_i u_i(k))$ inside of $\mathbb{V}_i(k)$ (e.g. non minimum phase dynamics), the following constrained Model Predictive Control (MPC) can be considered

$$u_i(k) = \arg\min_{u_i} \sum_{l=0}^{N_p-1} \|y_i(k+l+1) - \bar{y}_i(k)\|_Q^2 + \sum_{l=0}^{N_p-1} \|u_i(k+l) - \bar{u}_i(k)\|_R^2 \tag{4.28a}$$

$$\text{s.t.:} \quad \bar{x}_i(k) = A_i \bar{x}_i(k) + B_i \bar{u}_i(k) \tag{4.28b}$$

$$\bar{y}_i(k) = C_i \bar{x}_i(k) \tag{4.28c}$$

$$x_i(k+l+1) = A_i x_i(k+l) + B_i u_i(k+l) \tag{4.28d}$$

$$y_i(k+l+1) = C_i x_i(k+l+1) \tag{4.28e}$$

$$y_i(k+l+1) \in \mathcal{W},\ l \in \mathbb{N}_{[0,N_p-1]} \tag{4.28f}$$

$$y_i(k+N_p) \in \bar{y}_i(k) \oplus \lambda(\mathbb{V}_i \oplus \{-\bar{y}_i(k)\}),\ \text{with } \lambda \in \mathbb{R}_{[0,1)} \tag{4.28g}$$

Similar to the control problem (4.27), the weighting matrices $Q \in \mathbb{R}^{p_i \times p_i}$, $R \in \mathbb{R}^{m_i \times m_i}$ in the cost function (4.28a) are chosen such as $Q = Q^\top \succeq 0$ and $R = R^\top \succeq 0$. The constraints (4.28b) and (4.28c) are used to compute $(\bar{x}_i(k), \bar{u}_i(k))$ from $\bar{y}_i(k)$.

The constraints (4.28d)-(4.28e) are employed to predict the future values of the agent's output $y_i(k+l+1)$, state $x_i(k+l+1)$ and input $u_i(k+l)$ with respect to the dynamics (2.1) and the current state $x_i(k)$ and output $y_i(k)$. We use (4.28f) to ensure that the agent's output do not leave out of $\mathcal{W}$. The condition (4.28g) is considered as a terminal set constraint for the problem (4.28), which implies that the agent's final position $y_i(k+N_p)$ has to be included in an invariant subset inside its Voronoi cell $\mathbb{V}_i(k)$. This implies that the cell $\mathbb{V}_i(k)$ has to be a N-step controlled λ-contractive with respect to the agent's dynamics (2.1) for the center $\bar{y}_i(k)$ (see Definition 4.2). For this reason, it is preferable to choose a sufficiently large prediction horizon N_p to ensure that the constraint (4.28g) is validated for all agents, i.e.

$$N_p = \max\{N_{p_i}\}, \forall i \in \mathcal{N} \tag{4.29}$$

in which N_{p_i} is the shortest prediction horizon such that $y_i(k+N_{p_i}) \in \bar{y}_i(k) \oplus \lambda(\mathbb{V}_i \oplus \{-\bar{y}_i(k)\})$. It can be considered also as the minimum time between two successive times of partition step.

To conclude, this section proposes a novel decentralized control solution for the deployment task of a Multi-Agent system using dynamic Voronoi partition. The originality resides in the fact that the Chebyshev center is replaced by a general center notion which enhances the uniqueness of the tracking point for each agent. We describe also a class of viable control actions at each time instant which ensure the asymptotic convergence towards a central point of each Voronoi cell and ultimately to a static configuration for the Multi-Agent system. As a remarkable feature, the local control at the level of each agent is decentralized and uses only the information of the current state and the associated Voronoi cell[4]. This cell is known to depend on the agent's position and the position of the agents in his neighbor as defined by the non-redundant constraints in (4.5)-(4.6).

In the next section, we will present two numerical scenarios to illustrate the deployment performance of a Multi-Agent system over a bounded region $\mathcal{W}$ by using the basic solution (Section 4.2) and the general solution (Section 4.3.2).

4.5 Numerical illustrations

Let us consider a Multi-Agent system composed of $N = 4$ mobile agents. Each agent has its own continuous-time dynamics

$$\begin{cases} \dot{x}_i = \begin{bmatrix} -\frac{1}{\tau} & 0 & \frac{1}{\tau} & 0 \\ 0 & -\frac{1}{\tau} & 0 & \frac{1}{\tau} \\ 0 & 0 & -\frac{1}{\tau} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} x_i + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{\tau} & 0 \\ 0 & \frac{1}{\tau} \end{bmatrix} u_i \\ y_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_i \end{cases} \tag{4.30}$$

where $x_i \in \mathbb{R}^4$, $u_i \in \mathbb{R}^2$ and $y_i \in \mathbb{R}^2$ refer respectively to the agent state, input and output. Furthermore, the input u_i denotes the accelerator, and x_i is composed of the agent's position and velocity. The choice of the matrix C_i help us to capture the agent's position $y_i \in \mathbb{R}^2$ from the state vector x_i. As seen in the structure of the matrices A_i and B_i, the pure integrator from the velocity to the position and similar from the acceleration to the velocity is replaced by a first-order filter with $\tau = 0.1s$ denoting the time constant. The dynamics (4.30) is discretized according to the zero-order-hold method, with the sampling time chosen as $T_s = 0.01s$, i.e.

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) \\ y_i(k) = C_i x_i(k) \end{cases} \tag{4.31}$$

[4] A Voronoi partition can be obtained in a decentralized manner by considering that each agent construction relies only on its local state and the information of its closest neighbors, to construct its own Voronoi cell.

with $A_i = \begin{bmatrix} 0.9048 & 0 & 0.0905 & 0 \\ 0 & 0.9048 & 0 & 0.0905 \\ 0 & 0 & 0.9048 & 0 \\ 0 & 0 & 0 & 0.9048 \end{bmatrix}$, $B_i = \begin{bmatrix} 0.0047 & 0 \\ 0 & 0.0047 \\ 0.0952 & 0 \\ 0 & 0.0952 \end{bmatrix}$, $C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.
The region $\mathcal{W}$ is defined as a box in $\mathbb{R}^2$. The Voronoi tessellation is employed at each time instant to decompose $\mathcal{W}$ into an union of Voronoi cells, i.e. $\mathcal{W} = \bigcup\limits_{i=1}^{4} \mathbb{V}_i(k)$. Each cell corresponds to one agent's authorized functioning zone, used to design the decentralized control input $u_i(k)$.

In all the considered scenarios, the main objective is to drive the agents toward a static configuration depending only on the choice of the inner target point: Chebyshev center or general center. In Figs. 4.6 and 4.9, we use the green and blue points to denote respectively the agents initial positions $y_i(0)$ and their evolution in time. The centers $\bar{y}_i$ (Chebyshev/general center) have their trajectories described by the red points. The tracking error $\|y_i(k) - \bar{y}_i(k)\|$ are illustrated in Figs. 4.7 and 4.10. Figs. 4.8 and 4.11 show the evolution in time of the value function (4.19) corresponding to each scenario.

We show the value of $y_i(0)$ directly in the figures and also the static Voronoi configuration. The decentralized MPC framework (4.28) is employed with the value of the weighting matrices $Q \in \mathbb{R}^{2\times 2}$, $R \in \mathbb{R}^{2\times 2}$ and the horizon of prediction N_p given for each simulation.

A. Chebyshev Static Configuration convergence

In the first scenario, the working region is

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} -20 \\ -20 \end{bmatrix}, \begin{bmatrix} 20 \\ -20 \end{bmatrix}, \begin{bmatrix} 20 \\ 20 \end{bmatrix}, \begin{bmatrix} -20 \\ 20 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

as presented in Fig. 4.6. The agents use the decentralized MPC framework (4.28) to push the output $y_i(k)$ towards the corresponding Chebyshev center $\bar{y}_i(k)$. The value of the center $\bar{y}_i(k)$ associated to the Voronoi cell $\mathbb{V}_i(k)$ is obtained by solving the optimization problem (4.9). The weighting matrices are $Q = I_2$, $R = 50I_2$ and the prediction horizon is $N_p = 5$. The deployment result is shown in Fig. 4.6. The center tracking errors $\|y_i(k) - \bar{y}_i(k)\|$ and the evolution of the value function $V(\mathcal{Y}(k))$ are plotted respectively in Figs. 4.7-4.8. As result, the tracking errors go asymptotically to zero and thus the MAS approaches a stable CSC.

B. General Static Configuration convergence

In the second scenario, the working region is

$$\mathcal{W} = \text{conv}\left\{ \begin{bmatrix} -8 \\ -1 \end{bmatrix}, \begin{bmatrix} 8 \\ -1 \end{bmatrix}, \begin{bmatrix} 8 \\ 1 \end{bmatrix}, \begin{bmatrix} -8 \\ 1 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

as illustrated in Fig. 4.9(a). The weighting matrices are $Q = 20I_2$, $R = I_2$ and the prediction horizon is $N_p = 5$ for the decentralized MPC framework (4.28). Furthermore, the initial configuration of the four agents is chosen such that the shape of each Voronoi cell in Fig. 4.9(a) is

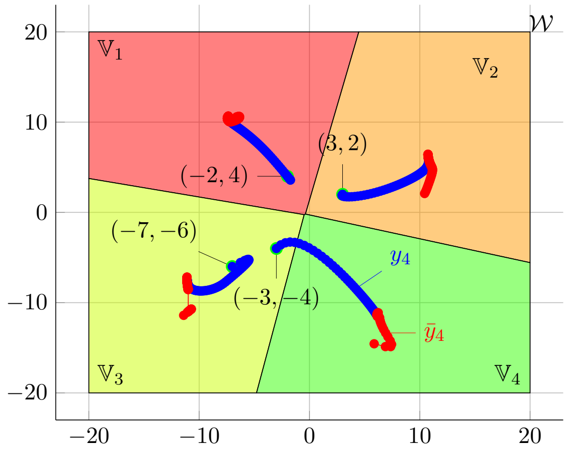

FIGURE 4.6: CSC deployment.

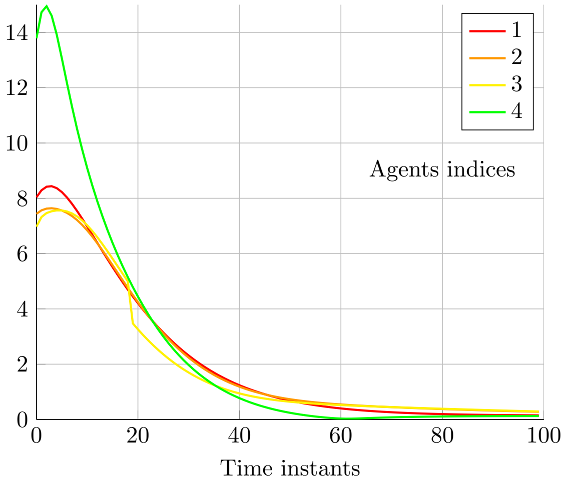

FIGURE 4.7: Agents center tracking error $\|y_i(k) - \bar{y}_i(k)\|$.

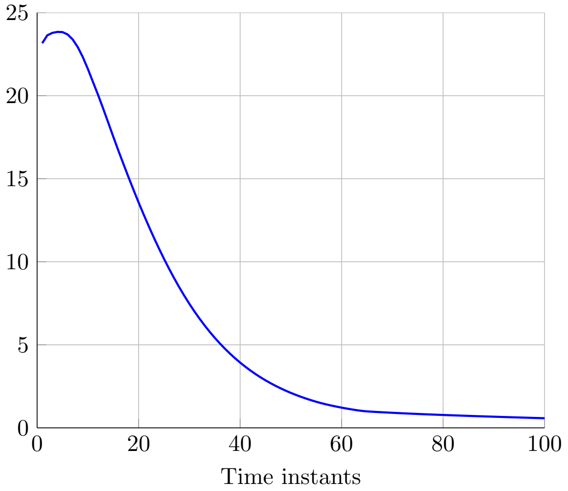

FIGURE 4.8: Evolution of Value function $V(\mathcal{Y}(k))$.

similar to the polyhedron's shape given in Examples 4.3 and 4.4. According to the results provided in these two examples, such configuration of Voronoi partition leads to the non-uniqueness of the problem (4.9) (see Fig. 4.9(a)) for all agents cells.

In order to overcome this computational problem, we replace the Chebyshev center by the general center. With respect to Definition 4.4, the uniqueness of the general center computation is guaranteed by the deflation principle and Pontryagin difference (see Definition 2.6). The deployment resulting from the generalized tracking strategy is shown in Fig. 4.9(b). The general centers $\bar{y}_i$ are determined at each time instant k by following the equation (4.24)-(4.25). The red points denote the motion of each general center $\bar{y}_i$. As mentioned above, the tuple of general centers at each time instant is unique. Concerning the deployment efficiency, each agent converges efficiently to its associated general center. This is illustrated by the curves of the agents' center tracking error $\|y_i(k) - \bar{y}_i(k)\|$ in Fig. 4.10 and the evolution of $V(\mathcal{Y}(k))$ in Fig. 4.11, which confirm the convergence of the entire Multi-Agent system into a stable GSC.

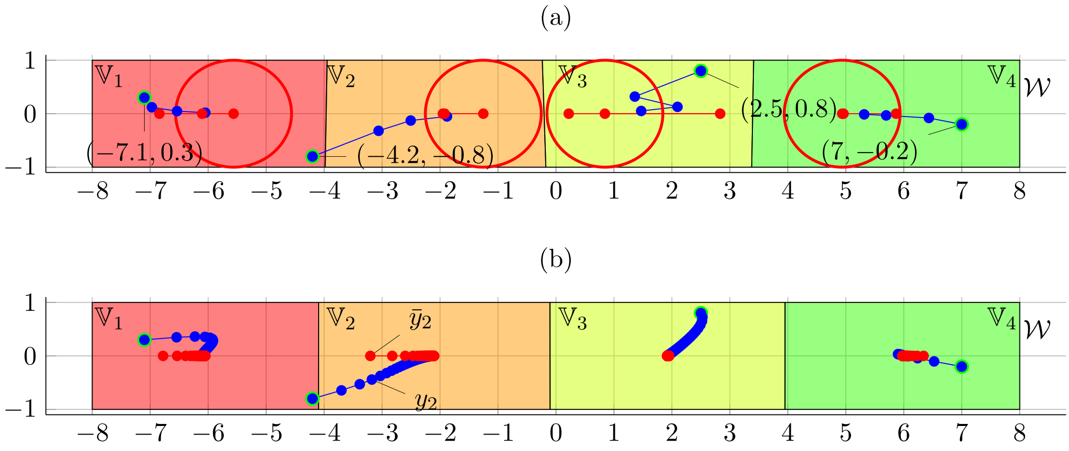

FIGURE 4.9: (a) Non unique CSC. (b) GSC deployment.

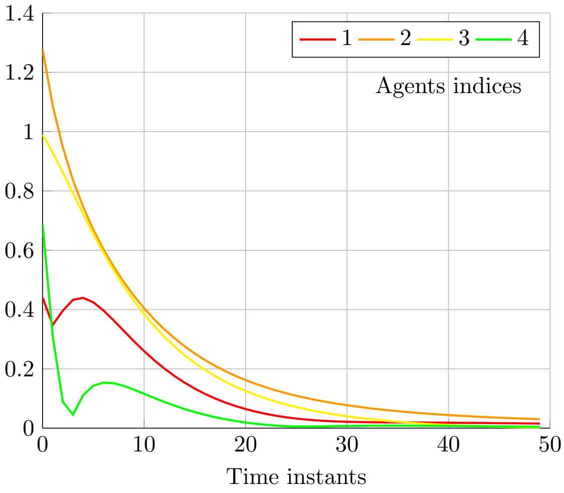

FIGURE 4.10: Agents center tracking error $\|y_i(k) - \bar{y}_i(k)\|$.

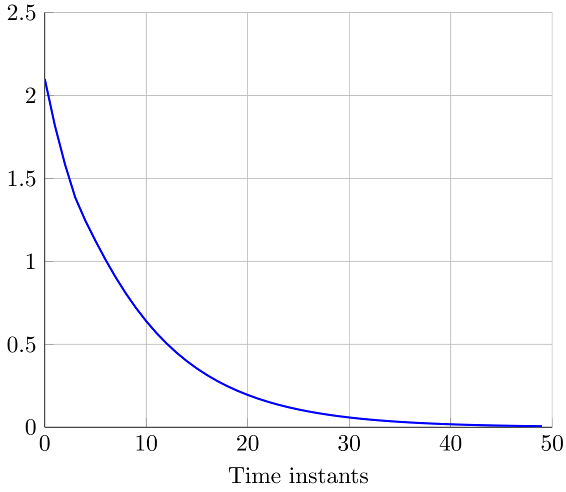

FIGURE 4.11: Evolution of the value function $V(\mathcal{Y}(k))$.

4.6 Conclusion

This chapter provides a novel set-based decentralized control for Multi-Agent self-deployment. The ultimate goal is to maximize the coverage by stabilizing the deployment of a Multi-Agent system around a static configuration over a bounded region. This goal is obtained in a decentralized manner by ensuring the convergence of each agent toward a target point inside its associated Voronoi cell-functioning zone. The computation of the set of agents functioning zones follows the concept of the Voronoi tessellation of the working region. The center of mass is usually chosen as the conventional inner target point and subsequently any decentralized control can be applied to steer the agent into its center of mass. This is the main principles of Lloyd's algorithm to compute an optimal coverage in which each agent coincides with its center of mass. However, the complexity of the center of mass computation is the main drawback of Lloyd's algorithm.

To overcome the computational drawback of center of mass, in this chapter, we replace the center of mass by the Chebyshev center. It allows using set-theoretic notions to formulate the center computation and the decentralized control design. Some counterexamples are given throughout the chapter to prove the non-uniqueness of the Chebyshev center computation and the associated dynamical Voronoi partition.

In order to overcome this problem, we advocate a novel solution based on the choice of a so-called constructed by recursive deflation after Chebyshev radius computation. This central point is subsequently used as the agent's target. For a given bounded convex set, its general center is unique by following the principle of deflation, leading to a stable static configuration. More than that, it is shown that any control input making the agent's cell controlled λ-contractive ensures the convergence to a static configuration. The control synthesis is decentralized according to the local Voronoi cell shape and developed based on the principles of Model Predictive Control framework subject to the λ-contractiveness of the Voronoi cell at the end of the prediction horizon. Beside these choices of the center, we will introduce a new so-called *vertex interpolated center* with its associated decentralized control design in Appendix A. Moreover, our novel decentralized control to drive a Multi-Agent system towards a centroidal Voronoi configuration will be detailed in Appendix B.

The present chapter along with Chapter 3 proposes a decentralized formation controls subject to anti-collision constraints for Multi-Agent dynamical system. The principle consists in enclosing each agent strictly inside its functioning zone such as these zones are non-overlapping. The functioning zone is firstly introduced in Chapter 3 and its method of construction is detailed in this present chapter. In Chapter 5, beside the fault tolerant control, we will detail the design of a set-based fault detection and isolation layer to supervise the number of agents in a formation.

Chapter 5

Formation reconfiguration using MPC techniques

5.1 Introduction

Fault Detection and Isolation (FDI) is basically defined as a supplementary layer to detect and isolate the faults during the functioning of a dynamical system. According to Blanke and Schröder (2006) and Isermann (2005), a *fault* in a dynamical system can be usually understood as a failure/damage in the system's components which change the dynamics and further deviates it from its nominal functioning mode. Many studies in the literature have been conducted on this topic and various results were obtained. Most of these works have the same principle related to *Model-based generation of residuals*. In the context of FDI, a residual is defined as an indicator signal of fault occurrence. It is basically built on the difference between the system's measured information and an estimated signal which is obtained by means of the foreknowledge of the system's model. For the design of residual, various results can be found, such as parity space (Patton et al. (1989), Gertler (1991)), state estimation by using observer(Edwards et al. (2000), Theilliol et al. (2002), Chen and Patton (2012)) or Kalman filter (Willsky (1976), Blanke and Schröder (2006)). Other approach related to FDI design is built on data-based method, such as statistical techniques (Mehra and Peschon (1971), Willsky (1976)) or neural networks (Venkatasubramanian et al. (1990), Altug et al. (1999)).

Basically, a FDI layer is followed by a reconfiguration strategy to fully/partially reconfigure the system once the faults are located. The main principle of this reconfiguration step consists in compensating the faults impact (Xiao-Zheng and Guang-Hong (2009), Seron and De Doná (2009)), or adapting the control action to the change in the system's dynamics by tuning the controller's parameter/structure (Eva Wu et al. (2000), Yu et al. (2005)). In some specific cases of actuators or sensors faults, hardware redundancy can be employed, in order to disconnect the failed part and connect to another component achieving the same task (Theilliol et al. (2002), Franze et al. (2012), Stoican and Olaru (2013) and Rotondo et al. (2014)).

Although FDI study for single dynamical system is widely known and developed in the literature as presented above, just few results relate FDI concept for MAS. We remind that the concept of MAS safety is more general than the definition of safety for single system. More specific, the functioning of MAS is not uniquely impacted by the damage on the agent's components but the collision can also be considered as faults for MAS.

Set theory is proved to be a powerful tool for monitoring by means of set-theoretic methods characterizing the system's functioning. Therefore, set-theoretic tools have been used to design FDI schemes based on the separation between different functioning modes. The faults treated in this framework are principally sensor faults (Stoican et al. (2013), Olaru et al. (2010)) and actuator faults (Seron and De Doná (2009), Franze et al. (2012)) for linear dynamical systems. The reported results on the application of set-theoretic and optimization tools for MAS control, did not integrate the Fault detection and reconfiguration capabilities to the best of our knowledge. Recently an effort has been made to link these two areas of developments in order to develop a comprehensive set-theoretic framework to design fault tolerant control strategies for MAS along the ideas presented in Fagiolini et al. (2007) and Rosich et al. (2014).

The main objective of this chapter is to employ set-theoretic methods to design a layer of fault detection and isolation for a homogeneous Multi-Agent system involving the set-description of their characteristics. This objective consists in supervising the interaction of the agents in the global MAS and detecting if agents from exterior try to integrate the current group of agents. As a field of application of these methodological developments one can consider the platooning problem (see Sheikholeslam and Desoer (1992), Stanković et al. (2000), Sabau et al. (2015), Ploeg et al. (2015), Farokhi and Johansson (2015) and Liang et al. (2016)). Due to the need to preserve the safety of all agents from the collision with the faulty agents inside the formation and with the intruders with respect to the current formation, we will privilege a centralized approach for the design of the FDI layer.

From a structural point of view, the control and monitoring priority is to preserve the formation and thus the design can be considered to be placed at a supervisory level. The main contribution, which is part of Nguyen et al. (2015a), is twofold:

- First, the proposed centralized FDI scheme for Multi-Agent systems is able to detect if an agent is faulty and if this fault falls in a serious category, to eliminate the faulty agent from the team (and automatically reconfigure the formation).
- Second, it will establish a threshold on the safety distances with respect to agents outside the formation in order to detect intruders. The FDI step is subsequently completed by a reconfiguration step to calculate a new optimal configuration for the global system. After finding the optimal formation, a classical control action is designed to steer and keep the MAS into this new formation, with respect to the collision avoidance constraints between the agents.

The design of such centralized FDI scheme relies on the quality of the communication tasks between the agents. It is assumed that the communication graph of the global Multi-Agent system is *fully connected*, i.e. any agent can send its information (e.g. position, speed...) to all agents in the global system and also receive the information from all these agents (see Mesbahi and Egerstedt (2010)). Moreover we assume that there is no degradation in the information exchanged between the agents due to the large disturbance of the environment or due to the delay of communication.

In the following, we will present the necessary prerequisites in Section 5.2, related to employing the notion of robust tube-based safety region (see subsection 2.3.1) of an agent and the collision avoidance constraints description in subsection 2.3.2. After, Section 5.3 formulates the tracking problem for a Multi-Agent system completed by few challenges due to the change of the number of agents in the formation. Section 5.4 and Section 5.5 describe respectively two scenarios of functioning in presence of faults and present a corresponding FDI framework with the associated reconfiguration step. Section 5.6 proposes a numerical simulation to illustrate the performance of the new FDI algorithm for a MAS composed of 3 agents subject to two faulty scenarios. Finally, the present chapter is ended by some concluding remarks and perspectives given in Section 5.7.

5.2 Background in Multi-Agent formation control

In order to elaborate the contribution, in the next subsections, we recall two main elements:

- The centralized optimization-based framework to obtain an optimal formation;
- The control action for the tracking mission for a formation in the fault-free functioning.

5.2.1 Minimal formation

A minimal formation of the MAS system $\mathbf{\Sigma}$ is defined as an ideal configuration where all the considered agents are as close as possible to a common reference, which is the reference of the formation center. This formation is defined as the optimal solution $\bar{\mathbf{x}}^*$ of the following problem:

$$\bar{\mathbf{x}}^* = \arg\min_{\bar{\mathbf{u}}^*} \sum_{i=1}^{N} \|\bar{x}_i\| \tag{5.1a}$$

$$\text{s.t.: } \bar{x}_i - \bar{x}_j \notin (-\mathcal{S}_i) \oplus \mathcal{S}_j, \forall i, j \in \mathbb{N}_{[1,N]}, i \neq j \tag{5.1b}$$

$$\bar{x}_i = A_i \bar{x}_i + B_i \bar{u}_i \tag{5.1c}$$

Here, $\bar{\mathbf{x}}^*$ denotes the collective vector containing the equilibrium optimal states of all agents in the MAS, i.e. $\bar{\mathbf{x}}^* = \left[\bar{x}_1^\top \; \bar{x}_2^\top \ldots \bar{x}_N^\top\right]^\top \in \mathbb{R}^{Nn}$. The state vector $\bar{x}_i$ indicates the displacement between the state of the i^{th} agent and the common reference (which represents the origin after an appropriate change of coordinates). The expression (5.1b) denotes the collision avoidance constraints, while (5.1c) emphasizes that $\bar{x}_i$ is determined as a static equilibrium point. The problem (5.1) is a non convex problem and due to the anti-collision constraint (5.1b), it can be casted in Mixed Integer Programming (MIP) class[1]. A Mixed Integer Programming problem is defined as an optimization problem where some of the decision variables are constrained to be integer values at the optimal solution. In the context of non-convex collision avoidance constraints, we add some supplementary binary variables to translate the non-convex optimization problem into a tractable convex problem. However, the main drawback relates its computational complexity which increases exponentially with the number of binary variables.

Remark 5.1. Solving the problem (5.1) requires the full information of all agents in the global system hence its calculation is centralized and thus it is computed in a centralized way.

The minimal formation can be computed offline or online, according to the number of agents in the Multi-Agent system, with the geometry of their safety region. We will detail next how to preserve this formation during the tracking mission.

5.2.2 Centralized tracking reference

Once the optimal formation is determined via solving the problem (5.1), it will be preserved along the *common reference*[2] x_{ref}. The main purpose of the formation control remains the design of a closed-loop control scheme so that the MAS's states track the common reference which can be interpreted as a feedforward signal:

$$x_{ref}(k+1) = A_{ref}x_{ref}(k) + B_{ref}u_{ref}(k) \tag{5.2}$$

with $A_{ref} = A_i$ for an homogeneous MAS. In this context, each agent has to determine for itself a *target trajectory* to follow, which can be defined as a sum of the common reference and the optimal position of this agent with respect to the origin. More specifically, the target trajectory of uniquely the i^{th} agent is denoted by:

$$\begin{aligned} \breve{x}_i(k) &= x_{ref}(k) + \bar{x}_i^* \\ \breve{u}_i(k) &= u_{ref}(k) + \bar{u}_i^* \end{aligned} \tag{5.3}$$

This trajectory is associated to the dynamical equation:

$$\breve{x}_i(k+1) = A_i\breve{x}_i(k) + B_i\breve{u}_i(k) \tag{5.4}$$

with $(\bar{x}_i^*, \bar{u}_i^*)$ being the solution of solving the problem (5.1), related to the optimal equilibrium position of the i^{th} agent in the minimal formation. Note that the pairs $(\bar{x}_i^*, \bar{u}_i^*)$ obtained by

[1] BNB is the main solver used to solve all MIP problems considered in the thesis. This solver is a quick implementation of a standard *branch & bound* algorithm for Mixed Integer Programming, already packaged in YALMIP Löfberg (2004).

[2] The common reference is the reference of the formation center.

solving (5.1) are always static. In others words, they are used to represent the offset between the time-varying common reference $(x_{ref}(k), u_{ref}(k))$ compatible with the agents' dynamics and the reference of each agent $(\breve{x}_i(k), \breve{u}_i(k))$.

Hence from the equations (5.3), we can derive $\breve{x}_i(k+1) = x_{ref}(k+1) + \bar{x}_i^*$. Using the equation (5.2) and the expression $\bar{x}_i^* = A_i\bar{x}_i^* + B_i\bar{u}_i^*$ with respect to the fact that $(\bar{x}_i^*, \bar{u}_i^*)$ satisfies the constraint (5.1c), we obtain

$$\breve{x}_i(k+1) = A_i\breve{x}_i(k) + B_i\breve{u}_i(k) \tag{5.5}$$

Remark 5.2. If $(\bar{x}_i^*, \bar{u}_i^*)$ is considered as the equilibrium position of the i^{th} agent with respect to the origin, then $(\breve{x}_i, \breve{u}_i)$ is its equilibrium position with respect to the value of the common reference x_{ref}, according to the equations (5.3).

Example 5.1. Consider a MAS composed of $N = 3$ homogeneous agents having the same dynamics with the agent in Example 2.8. The common reference x_{ref} is denoted by a red dashed line in Fig. 5.1. Applying the tube-based construction (see subsection 2.3.1) to construct the agents safety regions. The detail of construction was given in Example 2.8. Solving the problem (5.1), we obtain the tight formation for these 3 agents. By using the equations (5.3), we can determine the target trajectory $\breve{x}_i$ for each agent end further enclose each $\breve{x}_i$ within its safety tube $\mathcal{S}(\breve{x}_i)$. As shown in Fig. 5.1, the tight formation is preserved along the common reference x_{ref}.

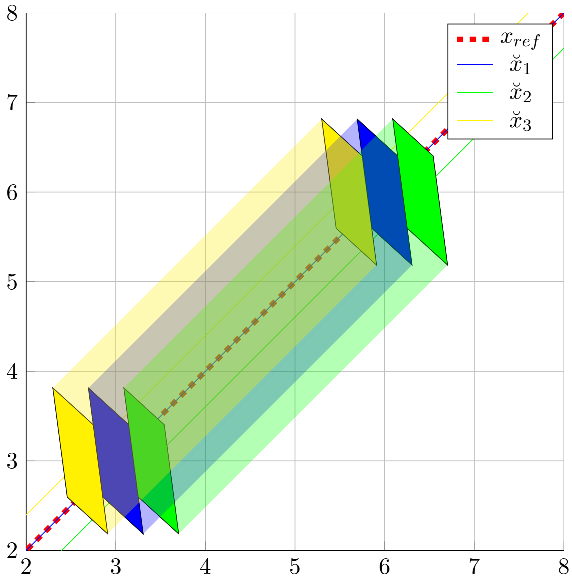

FIGURE 5.1: Formation of 3 homogeneous agents.

After determining the agents target trajectories, the following centralized Model Predictive Control action is designed to steer each agents state x_i towards its corresponding target trajectory $\breve{x}_i$:

$$\mathbf{u}^*(k) = \arg \min_{\mathbf{u}^*(k),\dots,\mathbf{u}^*(k+N_p-1)} \sum_{l=0}^{N_p-1} \Big(\|\mathbf{x}(k+l) - \breve{\mathbf{x}}(k+l)\|_{\mathbf{Q}}^2 + \|\mathbf{u}(k+l) - \breve{\mathbf{u}}(k+l)\|_{\mathbf{R}}^2 \Big) + \|\mathbf{x}(k+N_p) - \breve{\mathbf{x}}(k+N_p)\|_{\mathbf{P}}^2 \tag{5.6a}$$

$$\text{s.t. } \mathbf{x}(k+l+1) = \mathbf{A}\mathbf{x}(k+l) + \mathbf{B}\mathbf{u}(k+l) \tag{5.6b}$$

$$\breve{\mathbf{x}}(k+l+1) = \mathbf{A}\breve{\mathbf{x}}(k+l) + \mathbf{B}\breve{\mathbf{u}}(k+l) \tag{5.6c}$$

$$x_i(k+l+1) - x_j(k+l+1) \notin (-\mathcal{S}_i) \oplus \mathcal{S}_j \tag{5.6d}$$

$$\text{with } l \in \mathbb{N}_{[0,N_p-1]}, \ \forall i,j \in \mathcal{N}, i \neq j$$

with $\mathbf{x} = \left[x_1^\top \, x_2^\top \dots x_N^\top\right]^\top \in \mathbb{R}^{Nn}$, $\mathbf{u} = \left[u_1^\top \, u_2^\top \dots u_N^\top\right]^\top \in \mathbb{R}^{Nm}$ denoting the collective state and input vector of all agents in the MAS. The notations $\breve{\mathbf{x}} = \left[\breve{x}_1^\top \, \breve{x}_2^\top \dots \breve{x}_N^\top\right]^\top \in \mathbb{R}^{Nn}$, $\breve{\mathbf{u}} = \left[\breve{u}_1^\top \, \breve{u}_2^\top \dots \breve{u}_N^\top\right]^\top \in \mathbb{R}^{Nm}$ represent the collective state and input vector of the agents target trajectories. The matrices $\mathbf{A} = \operatorname{diag}\{A_1, A_2, \dots, A_N\}$ and $\mathbf{B} = \operatorname{diag}\{B_1, B_2, \dots, B_N\}$ collect all the matrices corresponding to each agent by juxtaposition. The weighting matrices $\mathbf{Q} \in \mathbb{R}^{Nn\times Nn}$, $\mathbf{R} \in \mathbb{R}^{Nm\times Nm}$ and $\mathbf{P} \in \mathbb{R}^{Nn\times Nn}$ are all symmetric and positive definite.

The constraints (5.6b)-(5.6c) are employed to predict the future values of the agent's state $x_i(k+l+1)$, input $u_i(k+l)$ and also the future values of the target trajectories, with respect to the agents dynamics from the current state $\mathbf{x}(k)$. The last expression (5.6d) is the anti-collision constraint added to the control problem (5.6). Alternative formulations enforcing the stability can be employed by adding terminal constraints to (5.6d). However, in the present work, we consider that these issues are taken into account by adjusting the length of the prediction horizon in order to guarantee the stability via a pseudo-infinite cost function (see Chmielewski and Manousiouthakis (1996)) and additionally the feasible domain (if saturations are to be taken into account which is not the case here). Starting from the basic MPC formulation in (5.6) the focus is on the monitoring and fault detection of the MAS formation (supposed to run on a properly design tracking control mechanism). Note also that the problem (5.6) does not include static input-state limitations. The feasibility of the anti-collision constraint (5.6d) can be handled via reachability analysis or viability theory whenever these constraints are considered as "hard" (see Aubin (2009)) or by the methods proposed earlier in Section 3. Such centralized control structure requires that any agent has to send its information (e.g. position, speed...) to all agents in the global system and also receive the information from all these agents. The computational time for each prediction depends on the dimension of the Multi-Agent system.

Briefly, in this section, we provided the background in Multi-Agent formation control subject to anti-collision constraints using centralized Model Predictive Control approach. Next, Section 5.3 formulates the tracking problem for a Multi-Agent system completed by few challenges due to the change of the number of agents within the formation.

5.3 Problem statement

Given a dynamical Multi-Agent system, the mission safety is defined as the achievement of a common tracking goal while ensuring that the interaction between the agents do not damage the structural organization of the MAS. More precisely, all agents have to track a given reference within a predefined formation and integrate the collision avoidance constraints while fulfilling the mission objective. The framework (5.1) describes a minimal configuration for the system but this predefined configuration does not adapt to the real time evolution of MAS. In case of changing the number of agents in MAS, typically when an agent leaves definitely its team[3], due to a serious fault or due to the operator decision, or when some agents from exterior try to join the current MAS, clearly a fixed configuration is not suitable. Moreover, having agents that leave the formation may have a disastrous impact if a subgroup of agent follows such a faulty element.

Our interest is to reinforce the safety of a Multi-Agent system during a tracking mission by means of a centralized framework for fault monitoring. This requirement becomes challenging whenever the supervision level has to integrate the environmental disturbance and cumbersome the computation load due to the dimension of the aggregated MAS. The most important feature is that this supervision/monitoring framework has to be suitable to (compatible with) the established control objectives. More precisely, the implementation of a fault tolerant supplementary layer can introduce some loss of performance during the operation of the global system.

The theoretic basis is offered by the use of set-theoretic tools. Based on the knowledge of the agents dynamics and their safety region construction, we will present in the next sections the fault detection and isolation layer for the global system. The decision making will be based on an adaptive threshold with respect to the disturbance and furthermore based on the predicted state of the global system $\mathbf{\Sigma}$ (modeled as linear dynamics subject to bounded additive disturbances). The FDI layer for the fault monitoring will enable a reconfiguration step which aims to recover the control structure with respect to the remaining healthy agents in the system. This FDI layer is installed on each agent, based on the communication between this agent and all the remaining agents in the global system. The fault scenarios for the MAS are classified in two notable cases, respectively represented in Examples 5.2 and 5.3.

In the first case, an agent belonging to the $\mathbf{\Sigma}$ can suffer damages on its components which decrease the performance of functioning. This is translated by a switch in the dynamical behavior relative to the common healthy behavior of the other (homogeneous) agents in $\mathbf{\Sigma}$ (and subsequently with an impact in the common prediction model). For this case, the FDI layer has to characterize the behavior of each agent independently allowing to monitor and detect if a change in the nominal prediction model took place and if the anomalies are characterized as serious enough to subsequently isolate this agent.

[3] It can become even adversary with respect to the team but such behavior is not considered here. In the following, all the intruders are considered as cooperative and their inclusion is automatically granted to the formation, subject to reconfiguration.

Example 5.2. Consider an homogeneous MAS of 4 agents which has to track a reference (red dashed line). An agent is denoted by its safety region centered at this agent's nominal state (blue dot). If an agent is healthy, its safety region is colored by yellow, otherwise it is colored by green.

The faulty scenario is shown in Fig. 5.2 via four steps. Their details are given in Table 5.2. The occurrence of fault in step (a) impacts the agent's dynamics and leads to a non-cooperative behavior compared to the remaining healthy agents (step(b)) (represented by a change in color in Fig. 5.2). Note that in step (c), the impact of the fault is expressed in terms of the application of the centralized control action on the faulty dynamics, leading to seriously degrade the tracking performance of the remaining agents. After being detected, the faulty agent is eliminated and subsequently the formation is reconfigured for the remaining agents (step(d)).

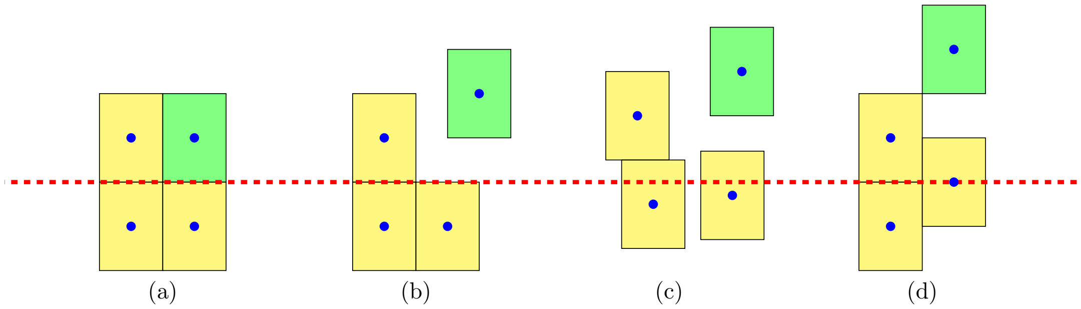

FIGURE 5.2: Outgoing fault detection and reconfiguration mechanism.

TABLE 5.1: Outgoing fault scenaria and FDI mechanism corresponding.

Step (a)	Occurrence of fault impacting the agent's dynamics
Step (b)	Detection of faulty agent
Step (c)	Degradation of the tracking performance
Step (d)	Reconfiguration of the formation by eliminating the faulty agent

In the second case, during the mission, some agents positioned outside of the formation can make maneuvers in order to integrate to the current formation (it is generally the case in a platooning maneuver when a track/car engages into the platoon). This scenario is critical because it can lead to safety loss. A FDI layer which is based on set-theoretic methods is further used as a threshold to detect these *intruders*. In order to guarantee the safety of the formation, this type of fault (or abnormal functioning of the formation) can be decomposed into two phases. First, once detected, the status *healthy* of the incoming agent has to be validated. This validation step needs a validation-time to ensure the integration objective of the intruders. Second, after being validated as healthy, a suitable reconfiguration step will be effectuated. It authorizes the inclusion of a novel index (of the intruder) to the global system $\mathbf{\Sigma}$ and the reconfiguration of the formation at the next iteration. The new formation has to be a set of optimal positions to which the current agents and the new ones will converge while fulfilling the anti-collision constraints. Four main steps of this FDI mechanism designed for the incoming fault case are given in Example 5.3.

Example 5.3. Consider an homogeneous MAS of 4 agents which has to track a reference (red dashed line). An agent is denoted by its safety region centered at this agent's nominal state (blue dot). If an agent is taken into account in the current MAS formation, its safety region is colored by yellow, otherwise it is colored by green.

The faulty scenario is shown in Fig. 5.3 via four steps. We detail the role of each step in Table 5.3. In step (a), the intruder is detected and immediately considered as an adversary intruder. As long as its integration is still not accepted, the formation has to avoid the collision with it thus the tracking performance can be impacted (step(b)). If the integration of the intruder is accepted, the current formation is reconfigured by taking into account this new agent in the centralized computation (step(c)) and further continue the tracking operation (step(d)).

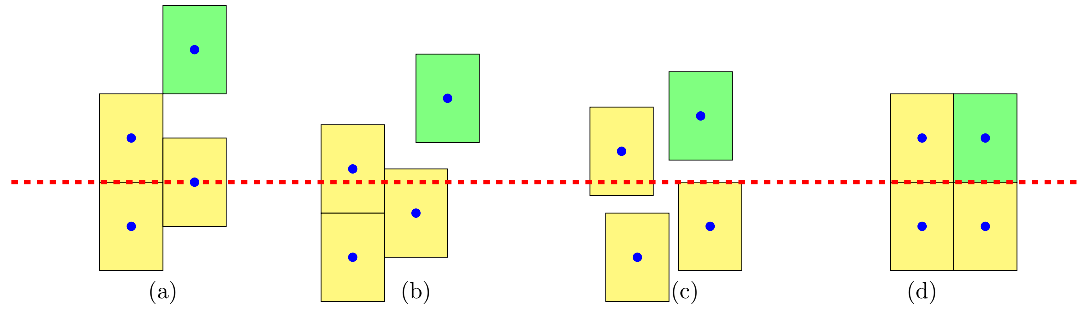

Figure 5.3: Incoming fault detection and reconfiguration mechanism.

Table 5.2: Incoming fault scenaria and FDI mechanism corresponding.

Step (a)	Detection of intruder outside the current formation
Step (b)	Protection of the formation from the collision with the intruder
Step (c)	Reconfiguration of the formation in case of accepted integration
Step (d)	Continuity of the tracking operation

These two scenarii will be described in the next two sections.

5.4 Outgoing-agent case-study

5.4.1 Fault Detection and Isolation for outgoing agents

In this section, the FDI layer is designed in order to detect and eliminate the faulty agents from the current formation. After the elimination step, the formation will be reconfigured based on the healthy (remaining) subset of agents. A certain time window will be imposed from the detection stage to validate the faulty status of an outgoing agent and engage the reconfiguration status. Two levels of faults are considered:

- *Quarantined Faulty Agent:* The agent suffers anomalies in its dynamic behavior relative to the healthy behavior of its neighbors. These anomalies make the agent's behavior different from its nominal (predicted) dynamics, but this still cannot prove that this agent is faulty, because the anomalies may be issued from environmental disturbance or local decisions made at the control level, see the problem (5.6). For this reason, we need to set a certain time window of length N_m to validate the fault. If it is maintained more than N_m time steps, this agent will be certified as faulty.
- *Certified Faulty Agent:* This case is similar to the previous case from the detection of the mis-functioning but the impact of the decision is different in the reconfiguration of the formation. The certification is done whenever the system in quarantine presents a state which is "largely" different from the remaining agents, practically outside of the current formation envelope. In this case, the faulty status is reported to the central unit which computes a novel optimal formation by eliminating the faulty agent.

In the sequel, these two cases are considered.

5.4.1.1 Quarantined Faulty Agent Detection

In order to determine the functioning mode (Healthy or Faulty) of an agent, a set of N residuals will be used, one for each agent. Each residual is defined as:

$$r_i(k) = \tilde{x}_i(k) - \check{x}_i(k), \text{ with } i \in \mathcal{N} \tag{5.7}$$

with $\tilde{x}_i(k)$ and $\check{x}_i(k)$ denoting the real state (see equation (2.6)) and the one-step predictable state of the i^{th} agent, respectively. The value of $\check{x}_i(k)$ is obtained by using the nominal dynamics (2.1) and the last available state $x_i(k-1)$ i.e.:

$$\check{x}_i(k) = A_i \check{x}_i(k-1) + B_i u_i^*(k-1), \text{ with } i \in \mathcal{N} \tag{5.8}$$

where $u_i^*(k-1)$ is the control action of the i^{th} element of the optimal solution of (5.6) at time instant $k-1$.

If there is no fault, after a transition when the tracking error converges towards the tight configuration, then $r_i(k) \in \mathcal{S}_i$. Hence, the safety region $\mathcal{S}_i$ is also the set $\mathcal{R}_i^H$ which characterizes the *Healthy functioning* of the i^{th} agent:

$$\mathcal{R}_i^H = \mathcal{S}_i, \text{ with } i \in \mathcal{N} \tag{5.9}$$

We consider a set denoted $\mathcal{R}_i^F$ to characterize the *Faulty functioning* and $\mathcal{R}_i^{H\to F}$ to characterize the Healthy-to-Faulty transition functioning. These sets can be used to characterize the detectability condition based on set separation

$$\begin{cases} \mathcal{R}_i^H \cap \mathcal{R}_i^{H\to F} = \emptyset \\ \mathcal{R}_i^H \cap \mathcal{R}_i^F = \emptyset \end{cases}, \text{ with } i \in \mathcal{N} \tag{5.10}$$

If one of these separations does not hold, then the FDI mechanism will not be able to ensure the one-step detection but it can engage a monitoring procedure (see Stoican and Olaru (2013)).

In the present framework, we consider the case of critical faults, like leaving the formation due to serious faults. The fault isolation/identification is not considered here, mainly due to the fact that a precise signature has not been pre-imposed. This means that once the fault occurs, it will be detected if the residual $r_i(k)$ jumps out of $\mathcal{R}_i^H$ and transits to $\mathcal{R}_i^F$. The condition of Fault detection and isolation (5.10) is thus simplified[4] to

$$\mathcal{R}_i^H \cap \mathcal{R}_i^F = \emptyset, \text{ with } i \in \mathcal{N} \tag{5.11}$$

The candidate set which satisfies the condition (5.11) is simply defined as the complement set[5] of $\mathcal{R}_i^H$, i.e.

$$\mathcal{R}_i^F = C(\mathcal{R}_i^H) \tag{5.12}$$

This criterion is based on the invariance properties of the safety regions which is preserved under nominal functioning in presence of modeled additive disturbances. It has to be mentioned that the presence of such a fault will trigger a collision avoidance mechanism to its neighbors as long as their local safety regions are not respected. All these mechanisms are in place at the local level and the global system is functioning in a degraded mode.

The detection of the quarantined faulty agent has the form of a set inclusion

$$r_i(k) \notin \mathcal{R}_i^H \tag{5.13}$$

Therefore, the condition to associate the quarantined faulty status to a healthy agent is that its residual r_i is inside $\mathcal{R}_i^H$ at the previous instant $k-1$ and subsequently it falls outside $\mathcal{R}_i^H$ at

[4] Note that a priori, the instantaneous fault detection is completed via the separation between $\mathcal{R}_i^H$ and $\mathcal{R}_i^{H\to F}$, i.e. $\mathcal{R}_i^H \cap \mathcal{R}_i^{H\to F} = \emptyset$ but this separation could not be maintained over a large transition window. This case however is not critical in the present MAS framework because there always exists a permanent tracking error imposed by the centralized control subject to anti-collision constraints.

[5] The candidate set is normally chosen as $\mathcal{R}_i^{H\to F} = C(\mathcal{R}_i^H)$ with respect to the instantaneous fault detection $\mathcal{R}_i^H \cap \mathcal{R}_i^{H\to F} = \emptyset$.

the current instant k, i.e.

$$\begin{cases} r_i(k-1) \in \mathcal{R}_i^H \\ r_i(k) \notin \mathcal{R}_i^H \end{cases} \tag{5.14}$$

If r_i is back to $\mathcal{R}_i^H$ whenever it is previously outside $\mathcal{R}_i^H$, the agent is certified healthy, i.e.

$$\begin{cases} r_i(k-1) \notin \mathcal{R}_i^H \\ r_i(k) \in \mathcal{R}_i^H \end{cases} \tag{5.15}$$

5.4.1.2 Faulty agent certification

The previous FDI scheme is used to detect and quarantine an agent which exhibits a fault. With respect to the conditions above, we can confirm that a healthy agent is certainly faulty by checking the general conditions (5.16), in which the set inclusion (5.13) is examined during a time window of length N_m. At the instant $k - N_m - 1$, its residual r_i is inside $\mathcal{R}_i^H$, but after it stays outside $\mathcal{R}_i^H$ during N_m steps until the current moment k, thus it is certainly faulty

$$\begin{cases} r_i(k - N_m - 1) \in \mathcal{R}_i^H \\ r_i(l) \notin \mathcal{R}_i^H, \text{ with } \forall l \in \mathbb{N}_{[k-N_m,k]} \end{cases} \tag{5.16}$$

Moreover, we propose here another set construction to detect whenever the fault of an individual agent becomes critical. The threshold set for this case is parameterized by a set

$$\check{\mathcal{S}}(x_{ref}(k)) = \operatorname{conv}\left(\bigcup \mathcal{S}(\check{x}_i(k))\right), \forall i \in \mathcal{N}_R \tag{5.17}$$

Here x_{ref} is the common reference of the global system $\mathbf{\Sigma}$ (see subsection 5.2.2) and $\check{x}_i(k)$ is the one-step predicted state of the i^{th} agent. We recall that $\mathcal{N}_R$ denotes the indices set of the remaining agents in $\mathbf{\Sigma}$. More precisely, the set (5.17) describes an envelope of formation evolving under healthy behavior which take the form of a convex set prameterized by the evolving-center of the formation x_{ref}.

This threshold set $\check{\mathcal{S}}(x_{ref}(k))$ is defined as the convex hull of the one-step predicted position of all agents around the reference. This description is similar to obtaining a tube-based construction centered by the common reference $x_{ref}(k)$, i.e.

$$\check{\mathcal{S}}(x_{ref}(k)) = \{x_{ref}(k)\} \oplus \check{\mathcal{S}} \tag{5.18}$$

This set $\check{\mathcal{S}}(x_{ref}(k))$ can be called the *elimination tube set* for the formation $\mathbf{\Sigma}$. The residual and the detection of the critical form take simply the form of a set inclusion

$$x_i(k) \notin \check{\mathcal{S}}(x_{ref}(k)) \tag{5.19}$$

Example 5.4 illustrates these faulty situations of the outgoing-agent case-study.

Example 5.4. We illustrate in Fig. 5.4 two faulty cases of the outgoing-agent case-study for a group of 5 agents. These 5 agents are at the initial stage in a minimal formation centered by the common reference x_{ref} (red dot) as shown in Fig. 5.4. Their predicted states $\check{x}_i$ are presented by the blue dots and their real states $\tilde{x}_i$ are denoted by the black dots. The common safety region of the agents is

$$\mathcal{S} = \text{conv}\left\{ \begin{bmatrix} 0.125 \\ 0.714 \end{bmatrix}, \begin{bmatrix} -0.125 \\ 0.714 \end{bmatrix}, \begin{bmatrix} -0.12 \\ 1.286 \end{bmatrix}, \begin{bmatrix} 0.125 \\ 1.286 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

centered at the predicted state $\check{x}_i$. The agents one-step predictable collective state is

$$\check{\mathbf{x}} = \begin{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix}^\top, \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}^\top, \begin{bmatrix} 1 \\ -1 \end{bmatrix}^\top, \begin{bmatrix} 0 \\ -1.5 \end{bmatrix}^\top, \begin{bmatrix} -1 \\ -1 \end{bmatrix}^\top \end{bmatrix}^\top$$

The agents real collective state is

$$\tilde{\mathbf{x}} = \begin{bmatrix} \begin{bmatrix} -0.9 \\ 1.2 \end{bmatrix}^\top, \begin{bmatrix} 1.5 \\ 0.2 \end{bmatrix}^\top, \begin{bmatrix} 1 \\ -0.8 \end{bmatrix}^\top, \begin{bmatrix} 0 \\ -1 \end{bmatrix}^\top, \begin{bmatrix} -2 \\ 1 \end{bmatrix}^\top \end{bmatrix}^\top$$

We construct the elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$ using the equation (5.17) and denote it by the black lined set. The agent A has its residual $r_A = \tilde{x}_A - \check{x}_A = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$ (see the equation (5.7)), thus $r_A \notin \mathcal{R}_A^H$, with $\mathcal{R}_A^H = \mathcal{S}$ according to the equation (5.9). As a consequence, this agent A is quarantined. Another agent is certified faulty (the B point) because $\tilde{x}_B = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \notin \check{\mathcal{S}}(x_{ref})$ as shown in Fig. 5.4.

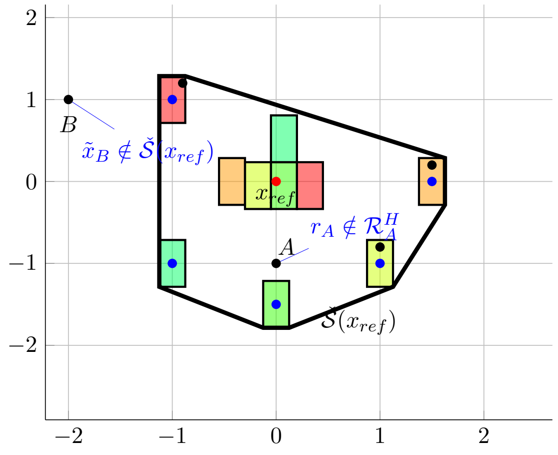

FIGURE 5.4: Faulty cases in the interior of the formation.

In the sequel, we will detail our reconfiguration solutions for the outgoing-agents case-study.

5.4.2 Reconfiguration - Outgoing-agents case-study

This section proposes a reconfiguration mechanism which is activated when an agent is certified faulty. It builds on the definition of the two faulty cases and the detection mechanisms described in the subsection 5.4.1. This reconfiguration step is performed (and is enabled) after the FDI certification step.

Firstly, the nature of the fault has to be determined. At each iteration k, we check if the residual signal $r_i(k)$ belongs to $\mathcal{R}_i^H$ or not. If $r_i(k) \notin \mathcal{R}_i^H$, the respective agent will be labeled as quarantined faulty agent. After N_m iterations corresponding to the fault monitoring horizon (N_m is used to characterize the time to detect a fault), if $r_i(k+N_m) \notin \mathcal{R}_i^H$, the i^{th} agent is certified faulty and subsequently will be eliminated from the team. This decision is made in order to avoid the functioning of the formation in a degraded mode (with tracking errors and anti-collision mechanism activated).

Otherwise, whenever $\tilde{x}_i(k) \notin \check{\mathcal{S}}(x_{ref}(k))$ and the i^{th} agent is immediately certified faulty, it will be eliminated without passing by the quarantine stage.

After characterizing the fault nature, the reconfiguration layer will be activated. Thus the formation is further reconfigured at the next iteration for the remaining healthy agents. To illustrate these ideas, in the following example, we will apply this reconfiguration mechanism for the Multi-Agent system considered in Example 5.4.

Example 5.5. Consider the homogeneous Multi-Agent system of 5 agents in Example 5.4 with the same notations. We already know that the agent B is certified faulty because its real state $\tilde{x}_B$ is outside the set $\check{\mathcal{S}}(x_{ref})$ (see Fig. 5.4). Hence immediately the reconfiguration takes place for a Multi-Agent system of 4 agents (see Fig. 5.5), by eliminating this faulty agent from the current formation. As a consequence, after the reconfiguration step, the agents one-step predictable collective state becomes

$$\check{\mathbf{x}} = \left[\begin{bmatrix}-1\\1\end{bmatrix}^\top, \begin{bmatrix}1.5\\0\end{bmatrix}^\top, \begin{bmatrix}1\\-1\end{bmatrix}^\top, \begin{bmatrix}0\\-1.5\end{bmatrix}^\top\right]^\top$$

The agents real collective state is

$$\tilde{\mathbf{x}} = \left[\begin{bmatrix}-0.9\\1.2\end{bmatrix}^\top, \begin{bmatrix}1.5\\0.2\end{bmatrix}^\top, \begin{bmatrix}1\\-0.8\end{bmatrix}^\top, \begin{bmatrix}0\\-1\end{bmatrix}^\top\right]^\top$$

The new elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$ and the new formation (of four agents) are illustrated in Fig. 5.5.

Moreover, according to the conclusion in Example 5.4, the agent A is quarantined because its residual r_A is not included inside the set $\mathcal{R}_A^H$. If the quarantined status is maintained for N_m

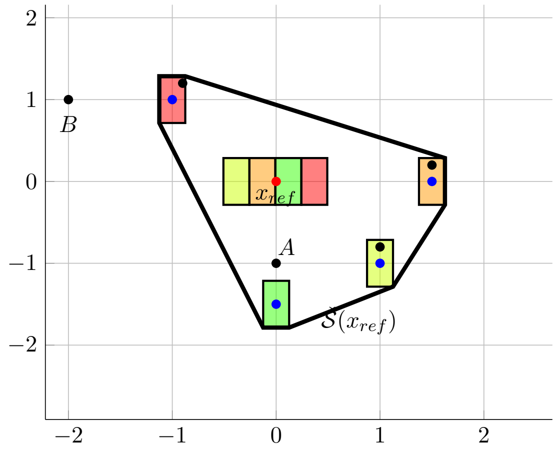

Figure 5.5: Reconfigured formation after elimination of B.

steps then the agent A will be also eliminated and the formation is reconfigured for the three remaining agents

After the reconfiguration step, the agents one-step predictable collective state becomes

$$\check{\mathbf{x}} = \left[\begin{bmatrix} -1 \\ 1 \end{bmatrix}^{\top}, \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}^{\top}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}^{\top} \right]^{\top}$$

The agents real collective state is

$$\tilde{\mathbf{x}} = \left[\begin{bmatrix} -0.9 \\ 1.2 \end{bmatrix}^{\top}, \begin{bmatrix} 1.5 \\ 0.2 \end{bmatrix}^{\top}, \begin{bmatrix} 1 \\ -0.8 \end{bmatrix}^{\top} \right]^{\top}$$

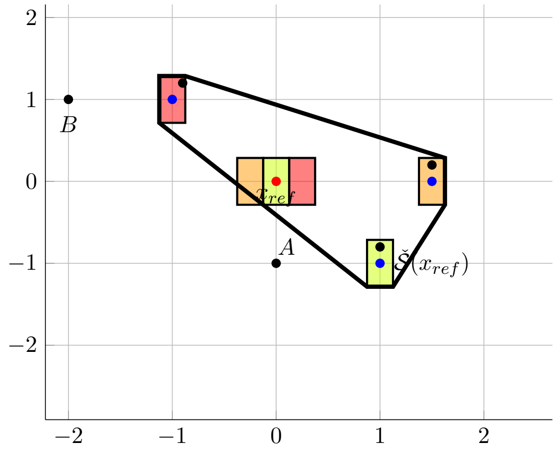

Figure 5.6: Reconfigured formation after elimination of A.

The new elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$ and the new formation (of three agents) is depicted in Fig. 5.6.

The algorithm summarizing our reconfiguration mechanism of outgoing-agents case-study will be presented in the next subsection.

5.4.3 Algorithm for the outgoing-agents scenario

All the above ideas are incorporated in Algorithm 3 for the task assignment of the $\mathcal{N}_R$ healthy subset of agents. This algorithm is executed at each sampling time. A set of timers will be activated in order to count the time steps when $r_i(k) \notin \mathcal{R}_i^H$. Each timer is associated with one agent in $\boldsymbol{\Sigma}$.

The functioning mode of each agent is represented by:

$$status_i = \begin{cases} 1 \text{ if Healthy} \\ 0 \text{ if Faulty} \end{cases}$$

The status (activated/deactivated) of each timer is described by the corresponding element in the vector $timer = [timer_1 \; timer_2 \ldots timer_N]^\top$, with

$$timer_i = \begin{cases} 1, & \text{if activated} \\ 0, & \text{if deactivated} \end{cases}$$

At each sampling time, the current state of all agents in $\boldsymbol{\Sigma}$ will be collected and also their residual sets $\mathcal{R}_i^H$. A loop of calculation is activated for all agents (line 2-33). For each agent, if its real state $\tilde{x}_i(k)$ is not included inside the elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$ then it will be eliminated immediately, else the verification necessary for the quarantine status is activated (line 3-6). In this case, its residual signal $r_i(k) = \tilde{x}_i(k) - \check{x}_i(k)$ is calculated (line 7) and the corresponding timer will be activated (line 8-31). During the activation of its timer, if its residual signal $r_i(k)$ is inside the Healthy functioning mode set $\mathcal{R}_i^H$ then the agent is healthy and its timer is deactivated (line 26-30). Else after N_m sampling time, if its residual signal $r_i(k)$ is outside the Healthy functioning mode set $\mathcal{R}_i^H$ this agent will be certified faulty and it will be eliminated (line 10-12). The reconfiguration step is activated at the end of the loop and it considers the remaining healthy agents (line 34).

We presented in this section our FDI framework followed by a appropriate reconfiguration step for the case of outgoing-agents. In the next subsection, we will consider the fault case where some new agents try to integrate the current formation.

Algorithm 3: Task assignment reconfiguration in the case of the elimination of a faulty agent

```
Input  : Current state $x(k)$, residual set $\mathcal{R}_i^H$
Output: Minimal reconfigured formation at sample time $k$
- construct $\check{\mathcal{S}}(x_{ref})$ for $\mathcal{N}_R$;
for $i \in \mathcal{N}_R$ do
    if $\tilde{x}_i(k) \notin \check{\mathcal{S}}(x_{ref}(k))$ then
        $status_i := 0$;
        $\mathcal{N}_R := \mathcal{N}_R \setminus \{i\}$;
    else
        - calculate residual $r_i(k) = \tilde{x}_i(k) - \check{x}_i(k)$;
        if $timer_i = 1$ then
            if $r_i(k) \notin \mathcal{R}_i^H$ then
                if $t = N_m$ then
                    $status_i := 0$;
                    $\mathcal{N}_E := \mathcal{N}_E \cup \{i\}$;
                else
                    $t := t + 1$;
                end
            else
                $timer_i := 0$;
                $status_i := 1$;
                $t := 0$;
            end
        else
            if $r_i(k) \notin \mathcal{R}_i^H$ then
                $timer_i := 1$;
                $status_i := 1$;
                $t := 1$;
            else
                $timer_i := 0$;
                $status_i := 1$;
                $t := 0$;
            end
        end
    end
end
- solve (5.1) for $\mathcal{N}_R$;
```

5.5 Incoming-agent case-study

Section 5.4 presented our FDI framework to detect and isolate the faulty agent in the interior of the formation. Here we present another FDI framework to detect the agents from exterior which aims to integrate the formation, and further to provide the appropriate reconfiguration mechanism.

5.5.1 Detection - Incoming-agent

Apart from the faulty agent detection and isolation, we propose here a FDI scheme to preserve the MAS safety when some new agents coming from the exterior of the current formation join the system. The main purpose is to detect whether/when an agent tries to join the formation while guaranteeing the safety of the global system during the integration process.

For this purpose of detection of intrusion, based on the elimination tube set $\check{\mathcal{S}}$ as defined in (5.17), we introduce a new set $\mathcal{T}$, called *intruder detection tube* which encircles the one-step forward predicted formation, e.g.:

$$\mathcal{T}(x_{ref}(k)) = \{x_{ref}(k)\} \oplus \alpha\check{\mathcal{S}} \tag{5.20}$$

This set $\mathcal{T}$ is constructed to bound the elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$ by scaling-out it with respect to the common reference x_{ref}. In the equation (5.20), the scalar $\alpha > 1$ is a scaling factor which accounts for the *visibility region* in the neighborhood of the formation. Its value can be adjusted in order to enlarge or reduce the scope of detection. Similar to the description of $\check{\mathcal{S}}(x_{ref}(k))$, the set $\mathcal{T}(x_{ref}(k))$ represents a tube set centered in the common reference $x_{ref}(k)$. The residual is defined as the distance with respect to the set and the detection mechanism is simply implemented by the set inclusion

$$\tilde{x}_i(k) \in \mathcal{T}(x_{ref}(k)) \tag{5.21}$$

This set is used to detect if some agents from exterior try to join the current formation of $\mathbf{\Sigma}$. If this integration effort is accepted by the supervision decision level, the following reconfiguration step has to be activated.

5.5.2 Reconfiguration - Incoming-agent case-study

Let us consider an i^{th} agent as not taken into account by $\mathbf{\Sigma}$, i.e. $i \notin \mathcal{N}_R$. When this agent approaches the formation of $\mathbf{\Sigma}$, if the condition (5.21) is validated, a timer will be activated. After N_m iterations (for brevity we take the same monitoring horizon in the case of incoming or outgoing agents, although a different length can be adopted if necessary), if the validation of (5.21) does not change, then the integration is accepted. The conditions to accept the integration of an incoming agent are formulated as

$$\begin{cases} \tilde{x}_i(k - N_m - 1) \notin \mathcal{T}(x_{ref}(k - N_m - 1)) \\ \tilde{x}_i(l) \in \mathcal{T}(x_{ref}(l)), \text{ with } \forall l \in \mathbb{N}_{[k-N_m,k]} \end{cases} \tag{5.22}$$

with $\tilde{x}_i$ denoting the agent's real state. For the reconfiguration mechanism, the index of this agent will be taken into account in the global task-allocation and the minimal formation, i.e. $\mathcal{N}_R = \{i\} \cup \mathcal{N}_R$ and a new configuration will be generated for the new subset $\mathcal{N}_R$. A natural question can rise about the need of a monitoring window for the incoming agents. The reason for its inclusion resides in the undesired effects of coupling the outgoing and the incoming agents monitoring. Indeed, in the case of an outgoing agent, its fault can be considered concomitantly as a potential incoming behavior and the status of the respective agent will be indiscernible.

Here, in order to improve the safety of the integration process, the new formation obtained from solving (5.1) has to be an optimal formation but admissible from the current position of all the agents in $\mathcal{N}_R$.

We illustrate in Example 5.6 the detection of incoming agents presented above for a group of 3 agents.

Example 5.6. Consider a homogeneous Multi-Agent system of 3 agents which are in a minimal formation centered by the common reference x_{ref} (red dot) as illustrated in Fig. 5.7. Their predicted states $\check{x}_i$ are presented by the blue dots and their real states $\tilde{x}_i$ are denoted by the black dots. The common safety region of the agents is

$$\mathcal{S} = \text{conv}\left\{ \begin{bmatrix} 0.125 \\ 0.286 \end{bmatrix}, \begin{bmatrix} -0.125 \\ 0.286 \end{bmatrix}, \begin{bmatrix} -0.125 \\ -0.286 \end{bmatrix}, \begin{bmatrix} 0.125 \\ -0.286 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

centered at the predicted state $\check{x}_i$. The agents one-step predictable collective state is

$$\check{\mathbf{x}} = \begin{bmatrix} \begin{bmatrix} 1 \\ -0.4 \end{bmatrix}^\top, \begin{bmatrix} -1 \\ -0.4 \end{bmatrix}^\top, \begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top \end{bmatrix}^\top$$

The agents real collective state is

$$\tilde{\mathbf{x}} = \begin{bmatrix} \begin{bmatrix} -1 \\ -0.2 \end{bmatrix}^\top, \begin{bmatrix} 1 \\ -0.6 \end{bmatrix}^\top, \begin{bmatrix} -0.1 \\ 1.2 \end{bmatrix}^\top \end{bmatrix}^\top$$

We construct the elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$ using the equation (5.17) and denote it by the black lined set.

The intruder detection tube $\mathcal{T}(x_{ref}(k))$ (bounded by dash line) is obtained by scaling-out the set $\check{\mathcal{S}}(x_{ref}(k))$ with the chosen scale factor $\alpha = 1.5$. One agent whose its real state is $\tilde{x}_C = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$ is trying to join the current group.

When its integration is accepted by means of the validation of the conditions (5.22), the formation is reconfigured for the four actual agents, thus a new formation is shown in Fig. 5.8.

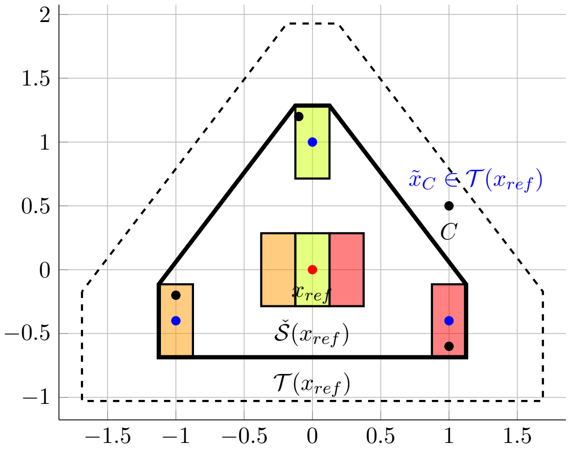

FIGURE 5.7: Detection of an agent outside of the current formation.

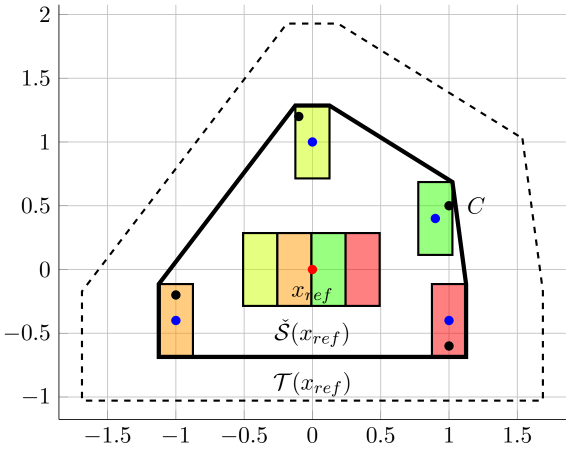

FIGURE 5.8: Reconfigured formation after integration accepted.

As a consequence, after the reconfiguration step, the agents one-step predictable collective state becomes

$$\check{\mathbf{x}} = \left[\begin{bmatrix} 1 \\ -0.4 \end{bmatrix}^\top, \begin{bmatrix} -1 \\ -0.4 \end{bmatrix}^\top, \begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top, \begin{bmatrix} 0.9 \\ 0.4 \end{bmatrix}^\top \right]^\top$$

The agents real collective state is

$$\tilde{\mathbf{x}} = \left[\begin{bmatrix} -1 \\ -0.2 \end{bmatrix}^\top, \begin{bmatrix} 1 \\ -0.6 \end{bmatrix}^\top, \begin{bmatrix} -0.1 \\ 1.2 \end{bmatrix}^\top, \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}^\top \right]^\top$$

The new elimination tube set $\check{\mathcal{S}}(x_{ref}(k))$, the intruder detection tube $\mathcal{T}(x_{ref}(k))$ and the new intruder detection tube $\mathcal{T}(x_{ref}(k))$ are illustrated in Fig. 5.8.

The algorithm summarizing our reconfiguration mechanism of incoming-agents case-study will be presented in the next subsection.

5.5.3 Algorithm for the incoming-agent scenario

The main ideas of the monitoring and reconfiguration are resumed in Algorithm 4. We reuse the notation concerning the timer in section 5.4.3. It is important to note that such a timer is activated for an unique agent from outside and there is no confusion between the incoming and outgoing agents. Moreover, we need to consider the following evaluation of the status of the agent which approaches the formation:

$$status_i = \begin{cases} 1 \text{ if accepted} \\ 0 \text{ if denied} \end{cases}$$

This means that the integration of an exterior agent is accepted if and only if its status binary variable is 1, otherwise it is rejected.

At each sampling time, the current real state of the intruder-agent $\tilde{x}_i(k)$ will be collected and also the elimination tube set $\check{\mathcal{S}}(x_{ref})$, the intruder detection tube $\mathcal{T}(x_{ref})$ and the indices set of the remaining healthy agents $\mathcal{N}_R$ of the current formation. If $\tilde{x}_i(k) \in \mathcal{T}(x_{ref}(k))$ and there is no timer activated for its integration, then the activation will take place (line 15-18). During the activation of its timer, if $\tilde{x}_i(k) \notin \mathcal{T}(x_{ref}(k))$ then the timer is deactivated and the integration request is denied (line 19-23). Else after N_m sampling times, if $\tilde{x}_i(k) \in \mathcal{T}(x_{ref}(k))$ the integration request is accepted (line 2-5). The reconfiguration step is activated at the end of the Algorithm for the healthy agents in the new subset $\mathcal{N}_R$ (line 25).

Our FDI framework with its reconfiguration step for the case of incoming agents are provided in this section. We will show some numerical simulations in the next section to illustrate the performance of our FDI-based propositions.

5.6 Illustrative example

In this section, a numerical example is presented in order to illustrate the results obtained by applying Algorithms 3 and 4 on a Multi-Agent system $\mathbf{\Sigma}$ composed of $N = 3$ homogeneous agents. The common dynamics is

$$x_i(k+1) = \begin{bmatrix} -0.2 & 0.5 \\ 0.2 & 0.71 \end{bmatrix} x_i(k) + \begin{bmatrix} 0.71 & 0 \\ 0 & 0.22 \end{bmatrix} u_i(k) + w_i, \ i \in \{1, 2, 3\}$$

The disturbance is bounded, i.e. $|w_i| \leq \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$. We employ the control action (2.7) to stabilize the agent's dynamics, with the feedback gain is obtained by using the pole placement technique

Algorithm 4: Task assignment reconfiguration in case of integration of agent from exterior

```
Input  : $\tilde{x}_i(k)$,$\check{\mathcal{S}}(x_{ref})$, $\mathcal{T}(x_{ref})$, $\mathcal{N}_R$
Output: Minimal reconfigured formation at sample time $k$
if $timer_i = 1$ then
    if $\tilde{x}_i(k) \in \mathcal{T}(x_{ref}(k))$ then
        if $t = N_m$ then
            $status_i := 1$;
            $\mathcal{N}_R := \mathcal{N}_R \cup \{i\}$;
        else
            $t := t + 1$;
        end
    else
        $timer_i := 0$;
        $status_i := 0$;
        $t := 0$;
    end
else
    if $\tilde{x}_i(k) \in \mathcal{T}(x_{ref}(k))$ then
        $timer_i := 1$;
        $status_i := 0$;
        $t := 1$;
    else
        $timer_i := 0$;
        $status_i := 0$;
        $t := 0$;
    end
end
- solve (5.1) for $\mathcal{N}_R$;
```

(0.2 and 0.5 as the chosen pole). The safety region is thus obtained by using Lemma 2.2 (see the details in subsection 2.3.1). The $\mathcal{H}$-representation of the safety region is thus

$$\mathcal{S} = \text{conv}\left\{ \begin{bmatrix} 0.125 \\ 0.286 \end{bmatrix}, \begin{bmatrix} -0.125 \\ 0.286 \end{bmatrix}, \begin{bmatrix} -0.125 \\ -0.286 \end{bmatrix}, \begin{bmatrix} 0.125 \\ -0.286 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

For the MPC controller used in (5.6), the weighting matrices are $\mathbf{P} = 10I_{Nn}$, $\mathbf{Q} = 100I_{Nn}$ and $\mathbf{R} = I_{Nm}$. The prediction horizon is $N_p = 3$ and the monitoring horizon chosen to validate the elimination of a faulty agent from the formation and also the integration of an exterior agent in the current formation is $N_m = 3$.

The feedforward common reference is generated by a MPC reference controller, including an integral tracking error as cost function. The weighting matrices chosen are $Q_r = 10I_n$, $R_r =$

$0.01I_m$. This leads to the following MPC framework

$$u^*_{ref}(k) = \arg \min_{u^*_{ref}(k),\ldots,u^*_{ref}(k+N_p-1)} \sum_{l=k}^{k+N_p} \|x_{ref}(l) - r(k)\|^2_{Q_r} + \sum_{o=k+1}^{k+N_p-1} \|u_{ref}(o) - u_{ref}(o-1)\|^2_{R_r} \tag{5.23a}$$

$$\text{s.t. } x_{ref}(l+1) = A_{ref}x_{ref}(l) + B_{ref}u_{ref}(l),\ l \in \mathbb{N}_{[k,k+N_p]} \tag{5.23b}$$

We recall here that $\mathbf{\Sigma}$ is homogeneous. Thus the model dynamics implemented in this MPC layer is the common dynamics of all agents, hence $A_{ref} = A_i$ with $i \in \{1, 2, 3\}$.

As illustrated in Fig. 5.9, the global Multi-Agent system $\mathbf{\Sigma}$ pursuits a periodic trajectory illustrated by the black line. At the beginning (the A point), $\mathbf{\Sigma}$ is composed of 3 agents, with $\mathcal{N}_R = \{1, 2, 3\}$. The task assignment gives the initial admissible optimal formation for $\mathbf{\Sigma}$ with respect to the origin

$$\bar{\mathbf{x}} = \begin{bmatrix} \begin{bmatrix} -0.25 \\ 0 \end{bmatrix}^\top, \begin{bmatrix} 0 \\ 0 \end{bmatrix}^\top, \begin{bmatrix} 0.25 \\ 1 \end{bmatrix}^\top \end{bmatrix}^\top$$

This formation is preserved with the common reference x_{ref} and the trajectories evolve in a tube centered at the $x_{ref}(k)$ with its associated elimination tube set $\breve{\mathcal{S}}(x_{ref}(k))$ and its intruder detection tube $\mathcal{T}(x_{ref}(k))$. The intruder detection tube $\mathcal{T}^*(x_{ref}(k))$ for these three agents is covered by yellow in Fig. 5.9. The red lines present the set of reference trajectory $\breve{x}_i$ of the agents.

At $k = 13$ (the B point), the 3^{rd} agent is subject to an actuator fault, and then it stops. The remaining agents continue to track their reference trajectory with respect to the last configuration. When $\tilde{x}_3$ does not belong to the elimination tube set $\breve{\mathcal{S}}(x_{ref})$ (the C point), the 3^{rd} agent is certified faulty and then the formation is reconfigured for the two remaining agents. The optimal position of these agents in the new formation determined with respect to the origin are

$$\bar{\mathbf{x}} = \begin{bmatrix} \begin{bmatrix} -0.125 \\ 0 \end{bmatrix}^\top, \begin{bmatrix} 0.125 \\ 0 \end{bmatrix}^\top \end{bmatrix}^\top$$

The new MAS is denoted by $\mathbf{\Sigma}^*$, with $\mathcal{N}_R = \{1, 2\}$. The elimination tube set $\breve{\mathcal{S}}^*(x_{ref}(k))$ and the intruder detection tube $\mathcal{T}^*(x_{ref}(k))$ are recalculated for $\mathbf{\Sigma}^*$. The new intruder detection tube $\mathcal{T}^*(x_{ref}(k))$ for two agents is covered by green in Fig. 5.9.

At $k = 25$ (the D point), the 4^{th} agent from outside exhibits an incoming trajectory with respect to $\mathbf{\Sigma}^*$. When its real state $\tilde{x}_4(k)$ is inside the tube $\mathcal{T}^*(x_{ref}(k))$ for more than N_m time steps (the E point), the integration of the 4^{th} agent is accepted. Thus $\mathcal{N}_R = \{1, 2, 4\}$ and the formation will be reconfigured for 3 healthy agents. The optimal position of the agents in the

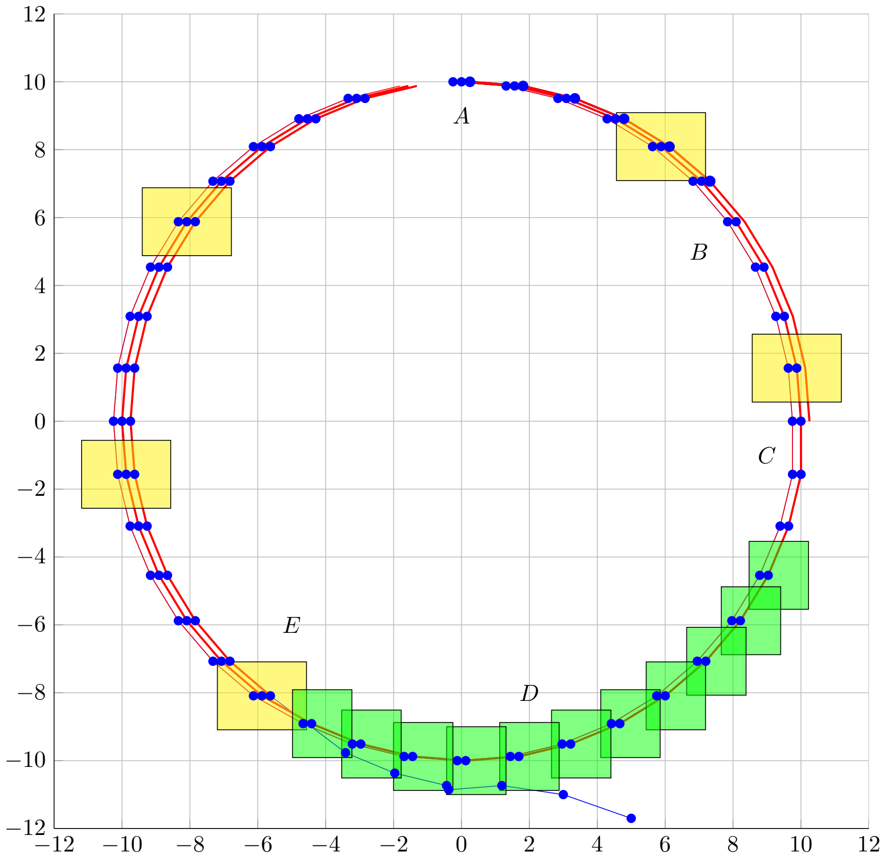

FIGURE 5.9: Illustrative example.

new formation determined with respect to the origin are

$$\bar{\mathbf{x}} = \left[\begin{bmatrix} -0.25 \\ 0 \end{bmatrix}^\top, \begin{bmatrix} 0 \\ 0 \end{bmatrix}^\top, \begin{bmatrix} 0.25 \\ 1 \end{bmatrix}^\top \right]^\top$$

The elimination tube set $\breve{\mathcal{S}}(x_{ref}(k))$ and the intruder detection tube $\mathcal{T}(x_{ref}(k))$ are recalculated again. Thus, we find the intruder detection tube $\mathcal{T}(x_{ref}(k))$ (covered by yellow) for three agents.

5.7 Conclusion

This chapter uses set-theoretic methods as a basic tool to design a supervision layer in order to maintain the functioning of a global Multi-Agent system under two particular faulty situations: elimination of a faulty agent from the formation and addition of an external agent to the current formation. The main objective is to supervise the functioning of the agents such that the damages issued from the collision during the operation can be predicted and avoided. We recall that the concept of Multi-Agent system Fault detection and isolation is more general than the classical concepts for simple system found in the literature. Specifically, our proposed set-theoretic FDI aims to protect the safety of formation. Fortunately, theoretical results concerning set-based FDI depicted in the literature can still be used selectively in this work, but some simplifying assumptions are supposed to be admissible. In order to serve the fault detection purpose, set-theoretics tools are employed to construct off-line the safety region of the agents and further to combine them on-line in order to build a threshold set for fault detection and isolation. The main contribution follows the philosophy of set separation, with respect to the foreknowledge of faulty and healthy functioning mode of each agent. Moreover, the off-line construction of agent's safety region in polyhedral format allows using simple algebraic set operations to construct the centralized safety threshold set for the formation.

In conclusion, we presented three main topics of the thesis related to using set-theoretic method in design of fault detection tools and decentralized optimization-based control for Multi-Agent dynamical system. More precisely, we detailed our decentralized formation control subject to anti-collision constraints in Chapter 3. Chapter 4 provides our decentralized Voronoi-based deployment, solving the problem of non-overlapping functioning zones construction of Chapter 3. Finally, beside the fault-tolerant controls mentioned in the two last chapter, we presents a set-based fault detection and isolation framework in Chapter 5. Next, we will present some concluding remarks and the perspectives for the overall results obtained in three chapters 3, 4 and 5.

Chapter 6

Concluding remarks and future directions

6.1 Conclusion

The present manuscript intended to develop set-theoretic based fault tolerant control design related to fault monitoring and collision free management of dynamical Multi-Agent systems. The ultimate goal is to protect the mission safety from risks of collision despite the possible faults. As already known in the control literature, regulating a group of cooperative mobile agents is obviously not new but the free collision guarantee still remains a challenging problem of MAS. Applying set theory in collision avoidance control is in the presence of faults one contribution to Multi-Agent system Fault-tolerant control design. The novelty was built upon the controlled invariance principle and the set separation thus ensuring the exact detection, isolation of the faulty agents and further the fault-tolerant control performance. It is worth to mention that set-theoretic methods were present in various applications but still remain an open direction for MAS FTC. Few results related to set-based MAS FTC and also FDI can be found in the literature.

With respect to the previous results in the literature, agents having linear time-invariant dynamics with bounded additive disturbances are supposed to observe a uniform distribution in a bounded set. Therefore, the pioneering polyhedron construction method of Kofman et al. (2007) is used as basic tool to build the safety region around the nominal position of each agent, due to its low complexity in terms of mathematical formulation and to the advantages of set interpretations. By this way, the nominal position along with a bounded polyhedral safety region becomes the key characteristics of an agent throughout the manuscript.

The results related to the FDI layer to guarantee the formation safety were presented in Chapter 5. By assuming the foreknowledge of agent's dynamics under the impact of bounded additive disturbances, we believe that the threshold sets characterizing the healthy/faulty behavior of each agent can be constructed by following low complexity set construction method. To be more specific, the threshold set is also the safety region centered at or parameterized by the predicted agent's position. After, a model-based residual signal is employed to determine in which mode the agent behaves (healthy, faulty or quarantined). The threshold set of the entire system is built on the convex hull of all agents predicted threshold set, offering a centralized supervision layer to protect the formation from the damages issued from collision. We focus on the critical cases where the faults are abrupt and thus the Healthy-to-Faulty and Faulty-to-Healthy transitions are omitted. Two typical faulty cases considered relate to detecting faulty agents inside and outside the formation: elimination of a faulty agent from the formation and addition of an external agent to the current formation. The recovery step after determining the faults consists in reconfigure the centralized formation control by taking into account the healthy agents after eliminating the faulty agents.

Being acknowledged as a better solution to overcome the computational inconvenience of centralized approach, the biggest advantage of decentralized control related to FTC concepts is the plug-and-play ability. In Chapter 3, we propose a novel decentralized framework by allocating each agent uniquely in one functioning zone. Each zone is built on the information exchanged between the resided agent with its closest neighbors and provides the local data for the control computation. The goal consists in enforcing the controlled invariance of the safety functioning zone. Another relevant advantage is its ability to handle the local feasibility, which is considered novel compared to set-theoretic based decentralized control in the literature. We consider the cases when there exists a local control action to keep each agent in its strict functioning zone. The feasibility of this determination however is limited in a subset of the functioning zone, assimilated to a controlled invariant set via linear feedback. For the points which are not covered by this set, we propose a control strategy to drive the agent state towards the region where the local linear control action is feasible.

Other more complicated and even hazardous application of dynamical MAS management is the deployment. Overall, such cooperative task can be stabilized by using the classical Lloyd's algorithm, which can be considered as the basic principle of decentralized control. Voronoi partitioning is run at each sampling time according to the agents current position and the local control is computed to ensure the convergence of each agent toward a target point inside its associated Voronoi cell-functioning zone. It remains to select the target center. We first choose the Chebyshev center as the basic solution. A less conservative solution based on a so-called general center is further advocated to overcome the inconvenience related to the non-uniqueness of the Chebyshev center computation. The concept of these centers and also the design of decentralized MPC are detailed in Chapter 4. Furthermore, another contribution in revisiting the Voronoi partition based decentralized coverage consists in considering a different type of target point, such as vertices interpolated center (in Appendix A) and center of mass (in Appendix B.) In both of these two cases, we provide a decentralized optimization-based control framework and we further give a deep analysis concerning the stability, followed by numerical simulations to illustrate the coverage performance.

6.2 Future directions

In the context of set-theoretic methods for control application in Multi-Agent system, some well-known hard problems related to the computational aspect have to be considered in future work. Considering simplifying assumptions without loss of generality, allows us to avoid discussing the computational aspect. Beside computational problem, the results presented above still offer many points to be considered as future directions. The formation safety guarantee management remains a very large topic covering fault tolerant concepts. For this reason, we list below some typical characteristic perspectives related to each of our main results presented above.

We emphasize that the problem mentioned in the main contribution was already simplified by considering a series of assumptions. In particular, all of the above results are presented based on the nominal dynamics of the global Multi-Agent system. For instance, the impact of multiplicative disturbances is not considered, which are various in the context of Multi-Agent system. There can be the disturbances issued from the working environment which cause delays in the communication channel or more seriously degradation of the information exchanged between the agents. These can lead to the deterioration in the communication graph of the Multi-Agent system. In this context, the robustness of the proposed framework seems interesting to study and specifically the relationship with the fault detection decision making. An interesting issue is to robustify the functioning of the global system by compensating the impact of these disturbances.

We believe that the proposed set-based centralized FDI, including the fault detection-isolation layer and the reconfiguration mechanism, can be developed in a decentralized manner. Moreover, the link between the FDI layer and the control action applied to the global system needs to be further clarified and thus the theoretic aspects have to be developed in order to offer a clear picture on the severity of collision constraints after reconfiguration. Another research direction which considers fault tolerant control for a heterogeneous system in a wider sense will be welcomed. This case is interesting and challenging because it is related to the constraints on the dynamics of each agent.

Concerning the decentralized FTC design, our solution depends strictly on the geometry of the agent's functioning zone. With respect to the polyhedral description, the choice of these hyperplanes forming the functioning zone should be careful made due to the impact on the control feasibility. This leads to an interesting research direction related to determining a partition of the working space such that it is controlled invariant with the set of local feedback controls predefined between each follower and the leader. The expected partition should ensure that each agent with its corresponding safety region is included strictly inside its associated functioning zone. Furthermore, it would be interesting to compare the proposed solutions with other existing approaches (such as MIP approach), in terms of computational time, rate of convergence, etc.

In the proposed solutions related to the self-deployment operation, the main idea consists of providing the decentralized control policies to enhance the coverage of a group of cooperative homogeneous mobile agents over a bounded region. Similar to the decentralized formation framework above, we have the same problem concerning the partition. We choose the Voronoi tessellation according to the previous results in the literature. However, the geometry of agent's safety region is totally omitted due to the fact that the additive disturbance in the agent's dynamics is not taken into account in the classical Voronoi tessellation construction. Future work can focus on developing a generalized Voronoi partition based on the current agents positions and the shape of their corresponding safety region, meanwhile suitable to the set of the predetermined local control laws.

Moreover, the results proposed in this thesis are based on the assumption that the entire working region is controlled invariant with respect to the agent LTI dynamics. Future work may focus on the case where the controlled invariance cannot be guaranteed for the entire working region. In this case, one solution related to combining the decentralized controls of Chapter 3 with the Voronoi-based deployment of Chapter 4 can be considered.

One of the most challenging perspective consists in taking into account the prediction of the Voronoi partition, which is clearly a hard problem to solve because of the complexity to find an explicit equation characterizing the partition's evolution.

Another interesting topics to be addressed relate the ingredients of the optimization-based control with respect to the constraints other than collision avoidance, such as energy-efficiency or limitation of traveling time. These points need to be discussed in future work because they relate the computational aspect of optimization-based framework.

It is worth to note that the theoretical aspects of our contributions are well developed in the thesis. In future work, we should focus on applying these theoretic results in some real systems such as a group of multi-vehicles (platoon systems composed of Unmanned Aerial Vehicles, Autonomous Underwater Vehicles or trucks), multi-robots, etc. The goal is to validate the efficiency and further evaluate the performance of our proposed solutions.

Appendix A

Voronoi based decentralized coverage problem using the vertex interpolation

In the context of the Multi-Agent system deployment, as a complement of the results presented in Chapter 4, the aim of this appendix is to revisit the coverage control problem by using a so-called *vertex interpolated center* as the inner target point. The control design follows the well-known principle of Voronoi-based decentralized control used in the literature (see the works in Cortes et al. (2002), Schwager et al. (2009), Bakolas and Tsiotras (2010) and our results in Chapter 4). More precisely, the safety constraints are satisfied by means of keeping each agent strictly inside a region called functioning zone. These zones are the results of a real time partition step based on the current state information (e.g. position) of the agents. The Voronoi algorithm is employed in this step to decompose the working space into a union of Voronoi cells. After, each agent is associated with one cell which will provide the vertices used to design the control action in order to make each agent converging to a fixed point inside its Voronoi cell. This control design step (based on both the agent's position and the vertices of its Voronoi cell containing the agent) by using the classical framework of optimal control and Model Predictive Control (MPC) is the main novelty of this appendix.

The results proposed in this appendix have been published in Nguyen and Maniu (2016). Due to laborious mathematical formulations related to the control laws computation and their stability proofs, we prefer to formulate the proposed results as an appendix, in order to simplify the reading of the manuscript.

We will present the details of the decentralized optimal control using the vertex interpolation in the next section.

A.1 Voronoi-based optimal control design

In this section, we propose a revisited decentralized control based on the optimal control design approach, using the local information (i.e. the agent position and the vertices of the Voronoi cell associated to the agent) received after the *partition step*. This control design is regarded as solving an optimization problem over an infinite horizon. Firstly, the explicit computation of the optimal solution is presented, then the stability proof is detailed by means of a resulting positive decreasing value function.

Remark A.1. In the sequel, the agent's index is neglected to simplify the notation.

Let us consider an agent represented by its discrete-time dynamics equation

$$x(k+1) = Ax(k) + Bu(k) \tag{A.1}$$

The matrix B is assumed to be full row rank, according to Assumption 4 in Chapter 4.

The Voronoi partition is illustrated in Fig. A.1, with the red points representing the considered agents. The Voronoi cell associated with agent x is denoted by $\mathbb{V}$ (see the green cell delimited by a dash-line in Fig. A.1). The notations $\mathcal{V}$ and $\mathcal{N}_{\mathcal{V}}$ (with $\mathcal{N}_{\mathcal{V}} = \{1, \ldots, 5\}$ in Fig. A.1) represent respectively the set of its vertices v_i and the indices set of these vertices. According to Assumption 4, each vertex is considered as an equilibrium point (v_i, w_i) of (A.1) such as $(I - A)v_i = Bw_i$.

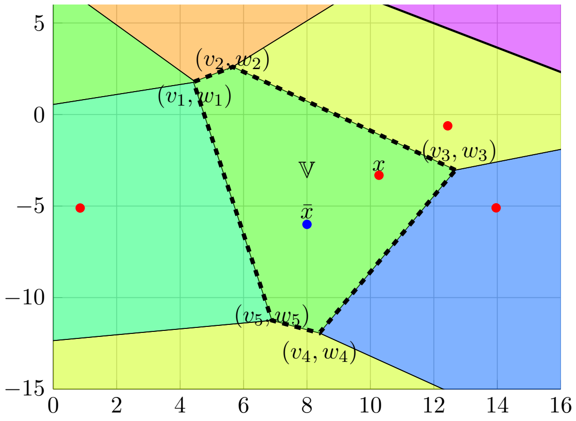

FIGURE A.1: Voronoi cell of agent x and its target fixed point $\bar{x}$.

The formulation of decentralized optimal control using vertex interpolation will be provided in the next subsection.

A.1.1 Optimal control solution

The optimal control of (A.1) is the solution of the following optimization control problem

$$\begin{aligned} &\min_{u} \sum_{k=0}^{\infty} L(x(k), u(k)) \\ \text{s.t.: } & x(k+1) = Ax(k) + Bu(k) \end{aligned} \tag{A.2}$$

with

$$L(x(k), u(k)) = \sum_{i,j \in \mathcal{N}_{\mathcal{V}}} \Big((x(k) - v_i)^\top Q_{ij} (x(k) - v_j) + (u(k) - w_i)^\top R_{ij} (u(k) - w_j) \Big) \tag{A.3}$$

indicating the running cost.

Moreover, by denoting[1] $\breve{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_{|\mathcal{V}|} \end{bmatrix}$ and $\breve{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_{|\mathcal{V}|} \end{bmatrix}$, with $|\mathcal{V}|$ being the cardinality of the set $\mathcal{V}$, we can define the augmented state $\breve{z}$ and input $\breve{t}$ as

$$\breve{z} = \begin{bmatrix} x - v_1 \\ \vdots \\ x - v_{|\mathcal{V}|} \end{bmatrix} = \breve{M}x - \breve{v} \in \mathbb{R}^{|\mathcal{V}|n} \tag{A.4a}$$

$$\breve{t} = \begin{bmatrix} u - w_1 \\ \vdots \\ u - w_{|\mathcal{V}|} \end{bmatrix} = \breve{N}u - \breve{w} \in \mathbb{R}^{|\mathcal{V}|m} \tag{A.4b}$$

with $\breve{M} = 1_{|\mathcal{V}|} \otimes I_n$ and $\breve{N} = 1_{|\mathcal{V}|} \otimes I_m$. The notation $1_{|\mathcal{V}|} \in \mathbb{R}^{|\mathcal{V}|}$ is used for the column vector whose elements are 1.

Then the extended dynamics can be derived

$$\breve{z}(k+1) = \breve{A}\breve{z}(k) + \breve{B}\breve{t}(k) \tag{A.5}$$

with $\breve{A} = I_{|\mathcal{V}|} \otimes A$ and $\breve{B} = I_{|\mathcal{V}|} \otimes B$.

Let us consider the additional notations $\breve{Q} = [Q_{ij}]$, $\breve{R} = [R_{ij}]$ with $\forall i, j \in \mathcal{N}_{\mathcal{V}}$ such that $\breve{Q}$ and $\breve{R}$ are symmetric positive definite matrices, i.e. $\breve{Q}^\top = \breve{Q} \succ 0$ and $\breve{R}^\top = \breve{R} \succ 0$.

With these notations, the running cost (A.3) can be rewritten as

$$L(\breve{z}(k), \breve{t}(k)) = \breve{z}(k)^\top \breve{Q} \breve{z}(k) + \breve{t}(k)^\top \breve{R} \breve{t}(k) \tag{A.6}$$

[1] The $\breve{}$ symbol is used to refer to all the vertices concomitantly.

Theorem A.1. *The solution of the optimal control problem* (A.2) *for the dynamics* (A.5) *is*

$$\theta u(k) = \breve{G}\breve{w} + \breve{K}\breve{z}(k) \tag{A.7}$$

by considering the notations

$$\Phi = \breve{R} + \breve{B}^\top \breve{P}\breve{B} \tag{A.8a}$$
$$\theta = \breve{N}^\top \Phi \breve{N} \tag{A.8b}$$
$$\breve{G} = \breve{N}^\top \Phi \tag{A.8c}$$
$$\breve{K} = -\breve{N}^\top \breve{B}^\top \breve{P}\breve{A} \tag{A.8d}$$

and

$$V(\breve{z}(k)) = \breve{z}^\top(k)\breve{P}\breve{z}(k) \tag{A.9}$$

which is considered as the cost-to-go of the control problem (A.2)*, with* $\breve{P} = [P_{ij}]$*,* $\forall i, j \in \mathcal{N}_\mathcal{V}$ *and* $\breve{P}^\top = \breve{P} \succ 0$.

Proof. According to Bellman principle of optimality (see Bellman (1954), Kirk (2012), Sanchez and Ornelas-Tellez (2013)), the optimal control solution of (A.2) has to ensure

$$V(x(k)) = \min_{u(k)}\{L(x(k), u(k)) + V(x(k+1))\} \tag{A.10}$$

which leads to the following Hamiltonian

$$H(x(k), u(k)) = L(x(k), u(k)) + V(x(k+1)) - V(x(k)) \tag{A.11}$$

Using (A.6) and (A.9), it is inferred that $H(\breve{z}(k), \breve{t}(k)) = L(\breve{z}(k), \breve{t}(k)) + V(\breve{z}(k+1)) - V(\breve{z}(k))$ yielding $H(\breve{z}(k), \breve{t}(k)) = \breve{z}(k)^\top \breve{Q}\breve{z}(k) + \breve{t}(k)^\top \breve{R}\breve{t}(k) + \breve{z}(k+1)^\top \breve{P}\breve{z}(k+1) - \breve{z}(k)^\top \breve{P}\breve{z}(k)$.

Based on the notations (A.4) and the dynamics (A.5), this can be detailed as $H(\breve{z}(k), \breve{t}(k)) = \breve{z}(k)^\top \breve{Q}\breve{z}(k) + (\breve{N}u(k) - \breve{w})^\top \breve{R}(\breve{N}u(k) - \breve{w}) + (\breve{A}\breve{z}(k) + \breve{B}\breve{t}(k))^\top \breve{P}(\breve{A}\breve{z}(k) + \breve{B}\breve{t}(k)) - \breve{z}(k)^\top \breve{P}\breve{z}(k)$.

Solving $\dfrac{\partial H}{\partial u} = 0$ leads to $2\breve{N}^\top \breve{R}\breve{N}u(k) - 2\breve{N}^\top \breve{R}\breve{w} + 2\breve{N}^\top \breve{B}^\top \breve{P}\breve{B}\breve{N}u(k) + 2\breve{N}^\top \breve{B}^\top \breve{P}(\breve{A}\breve{z}(k) - \breve{B}\breve{w}) = 0$ or equivalently $[\breve{N}^\top(\breve{R} + \breve{B}^\top \breve{P}\breve{B})\breve{N}]u(k) = \breve{N}^\top(\breve{R} + \breve{B}^\top \breve{P}\breve{B})\breve{w} - \breve{N}^\top \breve{B}^\top \breve{P}\breve{A}\breve{z}(k)$. Considering the notation (A.8), it leads to $\theta u(k) = \breve{G}\breve{w} + \breve{K}\breve{z}(k)$, with θ invertible. □

Remark A.2. The optimal solution (A.7) is unique by means of the convexity of the cost function in the optimization problem (A.2).

We have shown above the explicit optimal control using vertex interpolation. In the next subsection, we will discuss the stability of this control solution by means of the Bellman principle of optimality.

A.1.2 Stability analysis

Here we analyze the stability of the closed-loop agent's dynamics with respect to the optimal solution (A.7).

According to the Bellman principle of optimality, the optimal solution (A.7) has to ensure $H(\breve{z}(k),\breve{t}(k))=0$. It implies $L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=V(\breve{z}(k))$. In order to guarantee the Lyapunov stability, a sufficient condition is $V(\breve{z}(k+1))-V(\breve{z}(k))=-L(\breve{z}(k),\breve{t}(k))\leq 0$.

Starting from $L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=\breve{z}(k)^\top\breve{Q}\breve{z}(k)+\breve{t}(k)^\top\breve{R}\breve{t}(k)+\breve{z}(k+1)^\top\breve{P}\breve{z}(k+1)$, using (A.4b), it can be derived that

$L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=\breve{z}(k)^\top\breve{Q}\breve{z}(k)+(\breve{N}u(k)-\breve{w})^\top\breve{R}(\breve{N}u(k)-\breve{w})+(\breve{A}\breve{z}(k)+\breve{B}\breve{N}u(k)-\breve{B}\breve{w})^\top\breve{P}(\breve{A}\breve{z}(k)+\breve{B}\breve{N}u(k)-\breve{B}\breve{w})$.

Using the notations (A.8a) and (A.8b), this becomes

$L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=\breve{z}(k)^\top(\breve{Q}+\breve{A}^\top\breve{P}\breve{A})\breve{z}(k)+\breve{w}^\top\Phi\breve{w}+u(k)^\top\theta u(k)-2(\breve{w}^\top\breve{R}-(\breve{A}\breve{z}(k)-\breve{B}\breve{w})^\top\breve{P}\breve{B})\breve{N}u(k)-2\breve{w}^\top\breve{B}^\top\breve{P}\breve{A}\breve{z}(k)$.

Using (A.8c) and (A.8d), the previous scalar sum is equal to $L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=\breve{z}(k)^\top(\breve{Q}+\breve{A}^\top\breve{P}\breve{A})\breve{z}(k)+\breve{w}^\top\Phi\breve{w}+u(k)^\top\theta u(k)-2u(k)^\top(\breve{G}\breve{w}+\breve{K}\breve{z}(k))-2\breve{w}^\top\breve{B}^\top\breve{P}\breve{A}\breve{z}(k)$.

For θ inversible, replace $u(k)$ from (A.7) to obtain $L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=\breve{z}^\top(k)(\breve{Q}+\breve{A}^\top\breve{P}\breve{A})\breve{z}(k)+\breve{w}^\top\Phi\breve{w}-(\breve{G}\breve{w}+\breve{K}\breve{z}(k))^\top\theta^{-\top}\theta\theta^{-1}(\breve{G}\breve{w}+\breve{K}\breve{z}(k))-2\breve{w}^\top\breve{B}^\top\breve{P}\breve{A}\breve{z}(k)$.

Regrouping the terms in $\breve{z}(k)$ and $\breve{w}$, we obtain

$L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=\breve{z}^\top(k)(\breve{Q}+\breve{A}^\top\breve{P}\breve{A}-\breve{K}^\top\theta^{-\top}\breve{K})\breve{z}(k)+\breve{w}^\top(\Phi-\breve{G}^\top\theta^{-\top}\breve{G})\breve{w}-2\breve{w}^\top(\breve{B}^\top\breve{P}\breve{A}+\breve{G}^\top\theta^{-\top}\breve{K})\breve{z}(k)$.

However $L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))=V(\breve{z}(k))$ requires that the two terms $\breve{w}^\top(\Phi-\breve{G}^\top\theta^{-\top}\breve{G})\breve{w}$ and $\breve{w}^\top(\breve{B}^\top\breve{P}\breve{A}+\breve{G}^\top\theta^{-\top}\breve{K})$ have to vanish. For a non zero vector $\breve{w}$, it is difficult to impose $\breve{w}^\top(\Phi-\breve{G}^\top\theta^{-\top}\breve{G})\breve{w}=0$. A lightened condition $\breve{w}^\top(\Phi-\breve{G}^\top\theta^{-\top}\breve{G})\breve{w}\leq 0$ together with $\breve{w}^\top(\breve{B}^\top\breve{P}\breve{A}+\breve{G}^\top\theta^{-\top}\breve{K})=0$ are further considered, leading to

$$L(\breve{z}(k),\breve{t}(k))+V(\breve{z}(k+1))\leq V(\breve{z}(k))$$

The conditions to obtain this expression will be provided in the following theorem.

Theorem A.2. *If there exists a vector $\tilde{w}\in\mathbb{R}^m$ such that the following expressions hold*

$$\breve{G}\breve{w}=\breve{G}\breve{N}\tilde{w} \tag{A.12a}$$

$$\breve{w}^\top\breve{B}^\top\breve{P}\breve{A}=-\tilde{w}^\top\breve{K} \tag{A.12b}$$

$$\begin{bmatrix}\Phi & \breve{G}^\top\\ \breve{G} & \theta\end{bmatrix}\preceq 0 \tag{A.12c}$$

then the following expressions are verified

$$\breve{w}^\top(\Phi - \breve{G}^\top \theta^{-\top} \breve{G})\breve{w} \leq 0 \tag{A.13a}$$

$$\breve{w}^\top(\breve{B}^\top \breve{P}\breve{A} + \breve{G}^\top \theta^{-\top} \breve{K}) = 0 \tag{A.13b}$$

Proof. Applying the Schur complement in (A.12c) leads to $\Phi - \breve{G}^\top \theta^{-\top} \breve{G} \preceq 0$. Multiplying left and right by the non zero vectors $\breve{w}^\top$ and $\breve{w}$ leads to $\breve{w}^\top(\Phi - \breve{G}^\top \theta^{-\top} \breve{G})\breve{w} \leq 0$.

Using (A.12a), the expression $\breve{w}^\top(\breve{B}^\top \breve{P}\breve{A} + \breve{G}^\top \theta^{-\top} \breve{K}) = \breve{w}^\top \breve{B}^\top \breve{P}\breve{A} + \breve{w}^\top \breve{G}^\top \theta^{-\top} \breve{K}$ becomes $\breve{w}^\top \breve{B}^\top \breve{P}\breve{A} + \tilde{w}^\top \breve{N}^\top \breve{G}^\top \theta^{-\top} \breve{K}$. From (A.8b) and (A.8c), it results in $\breve{G}\breve{N} = \theta$. This allows us to obtain the following result $\breve{w}^\top(\breve{B}^\top \breve{P}\breve{A} + \breve{G}^\top \theta^{-\top} \breve{K}) = \breve{w}^\top \breve{B}^\top \breve{P}\breve{A} + \tilde{w}^\top \theta^\top \theta^{-\top} \breve{K}$. Using (A.12b), this leads to $\breve{w}^\top(\breve{B}^\top \breve{P}\breve{A} + \breve{G}^\top \theta^{-\top} \breve{K}) = 0$.

□

Remark A.3. Validating the conditions (A.12) leads to a classical Riccati equation

$$\breve{Q} + \breve{A}^\top \breve{P}\breve{A} - \breve{K}^\top \theta^{-\top} \breve{K} = \breve{P} \tag{A.14}$$

Following Theorem A.2, if the solution of the equations (A.12) is feasible then there exists a vector $\tilde{w}$ to guarantee $L(\breve{z}(k), \breve{t}(k)) + V(\breve{z}(k+1)) \leq V(\breve{z}(k))$. However, a feasible vector $\tilde{w}$ depends on the choice of the weighting matrices $\breve{P}$, $\breve{Q}$ and $\breve{R}$, which will be illustrated via the following proposition.

Proposition A.3. *Consider the matrices $R \in \mathbb{R}^{m\times m}$, $P \in \mathbb{R}^{n\times n}$ such as $R = R^\top \succ 0$, $P = P^\top \succ 0$, and a weighting structure matrix $\Delta \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. Hence the weighting matrices $\breve{R}$ and $\breve{P}$ can be chosen such that $\breve{R} = \Delta \otimes R$ and $\breve{P} = \Delta \otimes P$. The matrix $\breve{Q}$ can be obtained by solving the Riccati equation (A.14).*

Proof. Consider a weighting structure matrix $\Delta = [\Delta_1 \ldots \Delta_{|\mathcal{V}|}]$, with Δ_i denoting the i^{th} column of Δ. The equation (A.12a) is rewritten as $\breve{N}^\top \Phi \breve{w} = \breve{N}^\top \Phi \breve{N} \tilde{w}$. Due to (A.8a), we get $\Phi = \breve{R} + \breve{B}^\top \breve{P}\breve{B} = \Delta \otimes (R + B^\top PB)$. Hence (A.12a) becomes $\breve{N}^\top(\Delta \otimes (R + B^\top PB))\breve{w} = \breve{N}^\top(\Delta \otimes (R + B^\top PB))\breve{N}\tilde{w}$. From the definition of $\breve{N}$, it is infered that $N^\top = [I_m \quad \ldots \quad I_m] \in \mathbb{R}^{m\times(m|\mathcal{V}|)}$. Rewriting $\Delta = [\delta_{ij}], \forall i, j \in \{1, \ldots, |\mathcal{V}|\}$ and denoting $\tilde{R} = R + B^\top PB$ leads to

$$\breve{N}^\top(\Delta \otimes (R + B^\top PB)) = \begin{bmatrix} I_m & \ldots & I_m \end{bmatrix} \begin{bmatrix} \delta_{11}\tilde{R} & \ldots & \delta_{1|\mathcal{V}|}\tilde{R} \\ \vdots & \ddots & \vdots \\ \delta_{|\mathcal{V}|1}\tilde{R} & \ldots & \delta_{|\mathcal{V}||\mathcal{V}|}\tilde{R} \end{bmatrix}.$$

Then, we get $\left[\sum_{j=1}^{|\mathcal{V}|} \delta_{j,1}\tilde{R} \quad \ldots \quad \sum_{j=1}^{|\mathcal{V}|} \delta_{j,|\mathcal{V}|}\tilde{R}\right]\breve{w} = \left[\sum_{j=1}^{|\mathcal{V}|} \delta_{j,1}\tilde{R} \quad \ldots \quad \sum_{j=1}^{|\mathcal{V}|} \delta_{j,|\mathcal{V}|}\tilde{R}\right] \begin{bmatrix} I_m & \ldots & I_m \end{bmatrix}^\top \tilde{w}.$

It is possible to write $\sum_{i=1}^{|\mathcal{V}|} \sigma(\Delta_i)\tilde{R}w_i = \sigma(\Delta)\tilde{R}\tilde{w}$, with the scalars $\sigma(\Delta_i) = \sum_{j=1}^{|\mathcal{V}|} \delta_{j,i}$. Then $\sigma(\Delta)$ is the sum of all the elements of Δ. After symplifying $\tilde{R}$, we obtain $\sum_{i=1}^{|\mathcal{V}|} \sigma(\Delta_i)w_i = \sigma(\Delta)\tilde{w}$, which is further equivalent to $\tilde{w} = \sum_{i=1}^{|\mathcal{V}|}(\sigma(\Delta))^{-1}\sigma(\Delta_i)w_i$. Similar results will be obtained for (A.12b). In (A.12c), because all matrices Φ, $\breve{G}$ and θ depend on Δ, we can obtain a value for Δ by solving the feasibility problem (A.12c) with the decision variable Δ. Therefore, we can conclude that all conditions (A.12) are fulfilled proving a choice for $\tilde{w}$. □

From now, using Theorem A.2 with Proposition A.3 ensures the closed-loop stability of (A.5) with respect to the optimal control solution (A.7). We can see in the sequel that it implies also the closed-loop stability of (A.1) by proving the equivalence between the Lyapunov stability conditions of these two dynamics.

Replacing (A.4b) in (A.5) and then using (A.7) yields

$$\breve{z}(k+1) = (\breve{A} + \breve{B}\breve{N}\theta^{-1}\breve{K})\breve{z}(k) + \breve{B}\breve{N}\theta^{-1}\breve{G}\breve{w} - \breve{B}\breve{w} \tag{A.15}$$

with the stability proof being

$$(\breve{A} + \breve{B}\breve{N}\theta^{-1}\breve{K})^\top \breve{P}(\breve{A} + \breve{B}\breve{N}\theta^{-1}\breve{K}) \preceq \breve{P} \tag{A.16}$$

Furthermore, substituting (A.7) in the dynamics equation (A.1) and using (A.4a) to replace $\breve{z}$ by x, we get

$$x(k+1) = (A + B\theta^{-1}\breve{K}\breve{M})x(k) + B\theta^{-1}(\breve{G}\breve{w} - \breve{K}\breve{v}) \tag{A.17}$$

which leads to the Lyapunov stability condition of (A.1)

$$(A + B\theta^{-1}\breve{K}\breve{M})^\top P(A + B\theta^{-1}\breve{K}\breve{M}) \preceq P \tag{A.18}$$

Now consider the two dynamics equation (A.1) and (A.5) with the stability conditions (A.18) and (A.16), respectively. The following theorem shows the equivalence of the stability of these two dynamics.

Theorem A.4. *The expression* (A.18) *is verified if and only if* (A.16) *is verified.*

Proof. Consider (A.16). Left and right multiplying (A.16) respectively with the matrices $\breve{M}^\top$ and $\breve{M}$, we obtain $\breve{M}^\top(\breve{A} + \breve{B}\breve{N}\theta^{-1}\breve{K})^\top \breve{P}(\breve{A} + \breve{B}\breve{N}\theta^{-1}\breve{K})\breve{M} \preceq \breve{M}^\top\breve{P}\breve{M}$. From the definitions of $\breve{A}$ and $\breve{M}$ it is inferred[2] that $\breve{A}\breve{M} = \breve{M}A$ and $\breve{B}\breve{N} = \breve{M}B$. Thus we get $\breve{M}^\top(\breve{A} +$

[2] Using the definitions of $\breve{A}$ and $\breve{M}$, notice that $\breve{A}\breve{M} = \breve{M}A$ is equivalent to $\begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} \begin{bmatrix} I_n \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} I_n \\ \vdots \\ I_n \end{bmatrix} A$. In a similar way, the following expression is verified $\breve{B}\breve{N} = \breve{M}B$.

$\breve{B}\breve{N}\theta^{-1}\breve{K})^\top\breve{P}(\breve{A}+\breve{B}\breve{N}\theta^{-1}\breve{K})\breve{M} = (A+B\theta^{-1}\breve{K}\breve{M})^\top\breve{M}^\top\breve{P}\breve{M}(A+B\theta^{-1}\breve{K}\breve{M})$, which leads to $(A+B\theta^{-1}\breve{K}\breve{M})^\top\breve{M}^\top\breve{P}\breve{M}(A+B\theta^{-1}\breve{K}\breve{M}) \preceq \breve{M}^\top\breve{P}\breve{M}$. By choosing a matrix $P = \breve{M}^\top\breve{P}\breve{M}$, we get the condition (A.18).

Considering (A.18) in order to prove (A.16) can be done in a similar way. □

The previuos theorem illustrated the equivalence between the stability of systems (A.1) and (A.5). The next step is to find the equilibrium point the agent's state $x(k)$ converges to. The following theorem is used to determine this equilibrium point.

Proposition A.5. *If the dynamics equation* (A.1) *is stabilized with respect to the control law* (A.7)*, the state $x(k)$ and the control input $u(k)$ of* (A.1) *converge asymptotically towards an equilibrium point $(\bar{x},\bar{u})$ determined as:*

$$(I-A-B\theta^{-1}\breve{K}\breve{M})\bar{x} = B\theta^{-1}(\breve{G}\breve{w}-\breve{K}\breve{v}) \tag{A.19a}$$

$$(I-A)\bar{x} = B\bar{u} \tag{A.19b}$$

In fact, finding the equilibrium points for the systems (A.17) and (A.1) leads to the expressions (A.19).

We prove above that the stability of (A.1) is equivalent to the stability of (A.5). Additionally it is certain that the agent's state and input converge to the equilibrium point $(\bar{x},\bar{u})$ determined via (A.19). However until now the role of the vector $\tilde{w}$ introduced in Theorem A.2 is still not clarified. We need to know if there exist some relations between the equilibrium point $(\bar{x},\bar{u})$ and the vector $\tilde{w}$.

Remark A.4. By using (A.12a), the right-hand side of the expression (A.19a) becomes $B\theta^{-1}\breve{G}\breve{N}\tilde{w}-B\theta^{-1}\breve{K}\breve{v}$. Using (A.8b) and (A.8c) leads to $\theta = \breve{G}\breve{N}$. Thus we get $(I-A-B\theta^{-1}\breve{K}\breve{M})\bar{x} = B\tilde{w}-B\theta^{-1}\breve{K}\breve{v}$. Substituting (A.19b) in the left-hand side of (A.19a) leads to $B\bar{u}-B\theta^{-1}\breve{K}\breve{M}\bar{x} = B\tilde{w}-B\theta^{-1}\breve{K}\breve{v}$. After term by term identification on the two sides of this result, we find

$$\breve{K}\breve{M}\bar{x} = \breve{K}\breve{v} \tag{A.20a}$$

$$\bar{u} = \tilde{w} \tag{A.20b}$$

The following proposition illustrates the location of the fixed point $(\bar{x},\bar{u})$ of the agent relative to its Voronoi cell $\mathbb{V}$.

Proposition A.6. *The fixed point $(\bar{x}$,$\bar{u})$ determined by solving* (A.19) *belongs to the interior of $\mathbb{V}$, depending on the choice of Δ.*

Proof. Replacing the notation (A.8d) into (A.20a) yields $\breve{N}^\top\breve{B}^\top\breve{P}\breve{A}\breve{M}\bar{x} = \breve{N}^\top\breve{B}^\top\breve{P}\breve{A}\breve{v}$. Using Proposition A.3, one has $\breve{B}^\top\breve{P}\breve{A} = \Delta\otimes(B^\top PA)$, thus the following expression holds $\breve{N}^\top(\Delta\otimes(B^\top PA))\breve{M}\bar{x} = \breve{N}^\top(\Delta\otimes(B^\top PA))\breve{v}$. Similar to the proof of Proposition A.3, we get $\sigma(\Delta)\bar{x} = \sum_{i=1}^{|\mathcal{V}|} \sigma(\Delta_i)v_i$ then $\bar{x} = \sum_{i=1}^{|\mathcal{V}|} (\sigma(\Delta))^{-1}\sigma(\Delta_i)v_i$. Due to $\sum_{i=1}^{|\mathcal{V}|} \sigma(\Delta_i) = \sigma(\Delta)$, we obtain $\sum_{i=1}^{|\mathcal{V}|} (\sigma(\Delta))^{-1}\sigma(\Delta_i) = 1$ and finally we get $\bar{x} \in int(\mathbb{V})$. □

To conclude, the stability of (A.1) subject to the optimal solution (A.7) is proved in this section. This control ensures the convergence of the agent's state x towards the fixed point $\bar{x}$ inside of its Voronoi cell $\mathbb{V}$, according to Proposition A.6. In the next section, these results will be extended within the Model Predictive Control context.

A.2 Voronoi-based MPC control design

The features presented in the previous section can be extended over a finite horizon N_p. Solving an unconstrained Model Predictive Control problem is similar to solving an optimal control problem over a finite horizon. We will show in this section the explicit solution of this control problem (see subsection A.2.1) and further prove that the convergence can be built similarly as the optimal control approach (see subsection A.2.2).

For brevity, we use $k = 0$ to denote the current instant time.

A.2.1 Explicit control solution

The Voronoi-based MPC framework for the agent having the dynamics (A.1) is

$$\begin{aligned} u(0) = \arg\min_{u(k)} \sum_{k=0}^{N_p-1} L(x(k), u(k)) + V(x(N_p)) \\ \text{s.t: } x(k+1) = Ax(k) + Bu(k) \end{aligned} \tag{A.21}$$

The cost function in the MPC framework (A.21) is

$$J = \sum_{k=0}^{N_p-1} L(x(k), u(k)) + V(x(N_p))$$

with the running cost $L(x(k), u(k))$ defined in (A.3) and the terminal cost

$$V(x(N_p)) = \sum_{i,j\in\mathcal{N}_\mathcal{V}} (x(N_p) - v_i)^\top Q_{ij}(x(N_p) - v_j)$$

with respect to the expression (A.9). Notice that the vertices of the Voronoi cell of the considered agent are included in both the running cost $L(x(k), u(k))$ and the terminal cost function $V(x(k))$.

Here $V(x(N_p))$ denotes the final state penalty expected to be reached at the end of the prediction horizon.

By reusing the notations $\breve{P}, \breve{Q}, \breve{R}, \breve{A}, \breve{B}, \breve{M}, \breve{N}, \breve{v}, \breve{w}$ and the extended dynamics equation (A.5) of the previous section, we can derive the following result by a recurrent construction over the prediction horizon N_p

$$\mathbf{z} = \mathbf{A}\breve{z}(0) + \mathbf{B}(\mathbf{NU} - \mathbf{w}) \tag{A.22}$$

with $\mathbf{z} = \begin{bmatrix} \breve{z}(1) \\ \vdots \\ \breve{z}(N_p) \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} u(0) \\ \vdots \\ u(N_p-1) \end{bmatrix}$, $\mathbf{w} = \begin{bmatrix} \breve{w} \\ \vdots \\ \breve{w} \end{bmatrix}$, $\mathbf{A} = \begin{bmatrix} \breve{A} \\ \vdots \\ \breve{A}^{N_p} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} \breve{B} & & \\ \vdots & \ddots & \\ \breve{A}^{N_p-1}\breve{B} & \dots & \breve{B} \end{bmatrix}$,

$\mathbf{N} = \begin{bmatrix} \breve{N} & & \\ & \ddots & \\ & & \breve{N} \end{bmatrix}$.

Notice that the bold characters collect the information related to the prediction horizon N_p. These notations are proposed in order to preserve the coherence of the presentation with the optimal control section (e.g. the system dynamics (A.5) with the notations (A.4b)).

Using these notations, the cost function is rewritten as

$$\begin{aligned} J = \breve{z}(0)^\top \breve{Q}\breve{z}(0) + (\mathbf{A}\breve{z}(0) + \mathbf{BNU} - \mathbf{Bw})^\top \mathbf{P}(\mathbf{A}\breve{z}(0) + \mathbf{BNU} - \mathbf{Bw}) \\ + \mathbf{U}^\top\mathbf{N}^\top\mathbf{RNU} - 2\mathbf{U}^\top\mathbf{N}^\top\mathbf{Rw} + \mathbf{w}^\top\mathbf{Rw} \end{aligned} \tag{A.23}$$

with $\mathbf{P} = \begin{bmatrix} \breve{Q} & & & \\ & \ddots & & \\ & & \breve{Q} & \\ & & & \breve{P} \end{bmatrix}$, $\mathbf{R} = \begin{bmatrix} \breve{R} & & \\ & \ddots & \\ & & \breve{R} \end{bmatrix}$.

In order to simplify the presentation, consider the notation

$$\mathbf{\Phi} = \mathbf{R} + \mathbf{B}^\top\mathbf{PB} \tag{A.24a}$$

$$\mathbf{\Theta} = \mathbf{N}^\top\mathbf{\Phi N} \tag{A.24b}$$

$$\mathbf{G} = \mathbf{N}^\top\mathbf{\Phi} \tag{A.24c}$$

$$\mathbf{K} = -\mathbf{N}^\top\mathbf{B}^\top\mathbf{PA} \tag{A.24d}$$

The notations above help us to formulate the explicit solution of the control problem (A.21) in the following theorem.

Theorem A.7. *Consider the optimization control problem* (A.21)*. The optimal control solution over the prediction horizon N_p of this problem is*

$$\mathbf{\Theta U} = \mathbf{Gw} + \mathbf{K}\breve{z}(0) \tag{A.25}$$

Proof. Solving $\frac{\partial J}{\partial \mathbf{U}} = 0$ in a similar way to the optimal control case (see the proof of Theorem 1), we get $[\mathbf{N}^\top(\mathbf{R}+\mathbf{B}^\top\mathbf{P}\mathbf{B})\mathbf{N}]\mathbf{U} = \mathbf{N}^\top(\mathbf{R}+\mathbf{B}^\top\mathbf{P}\mathbf{B})\mathbf{w} - \mathbf{N}^\top\mathbf{B}^\top\mathbf{P}\mathbf{A}\breve{z}(0)$, and thus $\mathbf{\Theta}\mathbf{U} = \mathbf{G}\mathbf{w} + \mathbf{K}\breve{z}(0)$, with $\mathbf{\Theta}$ invertible. □

We have shown above the explicit MPC control using vertex interpolation. Similar to the coptimal control case, in the next subsection, we will discuss the stability of this control solution by means of Bellman principle of optimality.

A.2.2 Stability analysis

Replacing the variable $\mathbf{U}$ in the cost function (A.23) by $\mathbf{U} = \mathbf{\Theta}^{-1}(\mathbf{G}\mathbf{w} + \mathbf{K}\breve{z}(0))$ derived from (A.25), we get

$$\begin{aligned} J = \breve{z}(0)^\top(\breve{Q} + \mathbf{A}^\top\mathbf{P}\mathbf{A} - \mathbf{K}^\top\mathbf{\Theta}^{-\top}\mathbf{K})\breve{z}(0) \\ +\mathbf{w}^\top(\mathbf{\Phi} - \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{G})\mathbf{w} - 2\mathbf{w}^\top(\mathbf{B}^\top\mathbf{P}\mathbf{A} + \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{K})\breve{z}(0) \end{aligned} \tag{A.26}$$

In a similar way as the optimal control approach, the control (A.25) guarantees $J \le J^*$, with $J^* = \breve{z}(0)^\top\breve{P}\breve{z}(0)$ if and only if $\mathbf{w}^\top(\mathbf{\Phi} - \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{G})\mathbf{w} \le 0$ and $\mathbf{w}^\top(\mathbf{B}^\top\mathbf{P}\mathbf{A} + \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{K}) = 0$. This result is further formulated via the Theorem A.8.

Theorem A.8. *If there exists a vector* $\mathbf{w} \in \mathbb{R}^{mN_p}$ *which respects the conditions*

$$\mathbf{G}\mathbf{w} = \mathbf{G}\mathbf{N}\tilde{\mathbf{w}} \tag{A.27a}$$

$$\mathbf{w}^\top\mathbf{B}^\top\mathbf{P}\mathbf{A} = -\tilde{\mathbf{w}}^\top\mathbf{K} \tag{A.27b}$$

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{G}^\top \\ \mathbf{G} & \mathbf{\Theta} \end{bmatrix} \preceq 0 \tag{A.27c}$$

then the following expressions are verified

$$\begin{aligned} \mathbf{w}^\top(\mathbf{\Phi} - \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{G})\mathbf{w} \le 0 \\ \mathbf{w}^\top(\mathbf{B}^\top\mathbf{P}\mathbf{A} + \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{K}) = 0 \end{aligned}$$

Proof. The proof is identical to the optimal control case (see Theorem A.2), by considering the expression (A.27c) to get $\mathbf{w}^\top(\mathbf{\Phi} - \mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{G})\mathbf{w} \le 0$, then using (A.27a) and (A.27b) to obtain $\mathbf{w}^\top\mathbf{B}^\top\mathbf{P}\mathbf{A} = -\mathbf{w}^\top\mathbf{G}^\top\mathbf{\Theta}^{-\top}\mathbf{K}$. □

Applying Theorem A.8 helps us to keep $J \le J^*$ and therefore the stability is guaranteed at the end of the prediction horizon, i.e. $\sum_{k=0}^{N_p-1} L(x(k), u(k)) + V(x(N_p)) \le V(x(0))$ leads to $V(x(N_p)) - V(x(0)) \le -\sum_{k=0}^{N_p-1} L(x(k), u(k))$. In the following, we will use the weighting matrices

$\breve{P}$, $\breve{Q}$, $\breve{R}$ from the previous section in order to guarantee $V(x(k+1))-V(x(k)) \leq -L(x(k), u(k))$. Rewriting this over the prediction horizon N_p, we derive that

$$\begin{aligned} V(x(1)) - V(x(0)) &\leq -L(x(0), u(0)) \\ V(x(2)) - V(x(1)) &\leq -L(x(1), u(1)) \\ &\vdots \\ V(x(N_p)) - V(x(N_p-1)) &\leq -L(x(N_p-1), u(N_p-1)) \end{aligned}$$

By taking the sum of these inequalities above, we get

$$V(x(N_p)) - V(x(0)) \leq -\sum_{k=0}^{N_p-1} L(x(k), u(k))$$

In other words, the stability of the MPC problem (A.21) is covered by the stability of the optimal control problem (A.2). Therefore the results obtained in subsection A.2.2 allows us to validate the MPC stability in the unconstrained case.

Remark A.5. An advantage of MPC is its constraint handling ability. Hence, we can modify (A.21) by considering, for instance, the constraints $u(k) \in \mathbb{U}$ on the control action, i.e.

$$\begin{aligned} u(0) = \arg\min_{u(k)} & \textstyle\sum_{k=0}^{N_p-1} L(x(k), u(k)) + V(x(N_p)) \\ \text{s.t.: } & x(k+1) = Ax(k) + Bu(k) \\ & u(k) \in \mathbb{U} \end{aligned} \tag{A.28}$$

with the set $\mathbb{U}$ assumed to be bounded.

To conclude, in this section we have revisited the MPC framework by using the vertices information to drive the agents towards the interior of their respective Voronoi cell. The explicit control solution in the unconstrained case with its associated stability proof were given. We considered also the case where constraints on the control input are added into the MPC framework. However, the constrained MPC case analysis is still under development.

We will give in the next section some numerical simulations to prove the performance of the proposed decentralized control laws.

A.3 Simulation results

The previous results in Section A.1 and A.2 are demonstrated by means of numerical simulations provided in this section. Let us consider the Multi-Agent System $\mathbf{\Sigma}$ composed of $N_a = 15$ homogeneous agents. These agents have the first-time order dynamics equation

$$x_i(k+1) = x_i(k) + T_s u_i(k), \;\; i \in \mathcal{N} \tag{A.29}$$

with $x_i, u_i \in \mathbb{R}^2$ the position and the speed of the i^{th} agent and $T_s = 0.1$ the sampling period. In particular, the equation characterizing the fixed point of (A.29) is $(I - I)x_i = T_s u_i$, hence all positions in $\mathbb{R}^2$ can be a fixed point of (A.29) if the speed at this point is zero. Regarding the deployment task, the Multi Agent System $\mathbf{\Sigma}$ has to realize the surveillance task over a region

$$\mathcal{X} = \text{conv}\left\{ \begin{bmatrix} -20 \\ -20 \end{bmatrix}, \begin{bmatrix} 20 \\ -20 \end{bmatrix}, \begin{bmatrix} 20 \\ 20 \end{bmatrix}, \begin{bmatrix} -20 \\ 20 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

At each time instant, the region $\mathcal{X}$ is partitioned upon the Voronoi algorithm based on the current measured positions $x_i(k)$ of the agents, i.e. $\mathcal{X} = \mathbb{V} = \bigcup\limits_{i=1}^{N} \mathbb{V}_i(k)$. We can consider that each agent construction relies only on the local information as its position and the location of its closest neighbors, to construct its own Voronoi cell $\mathbb{V}_i(k)$. This local information is employed also to design the decentralized control action at each time instant. We will study respectively the performance of the control design (A.2) and (A.21) via the movement of the agents over $\mathcal{X}$ and also the tracking distance $||x_i - \bar{x}_i||$. We use the red colored circles to denote the initial positions $x_i(0)$ of the agents (see Figs. A.2 and A.4). In Fig. A.2, the green colored stars represent the evolution of each equilibrium fixed point $\bar{x}_i(k)$ and the blue colored circle-lines represent the evolution $x_i(k)$ of the agents.

The next step consists in finding the matrices P, Q, R. We can choose for instance $R = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}$. The matrix $P = \begin{bmatrix} 9.84 & 1.24 \\ 1.24 & 8.74 \end{bmatrix}$ is found such that $(I + T_s K)^\top P (I + T_s K) \preceq P$, with the feedback gain K obtained by pole placement techniques (with 0.8 and 0.5 as stable poles). The matrix $Q = \begin{bmatrix} 2.74 & -0.26 \\ -0.26 & 4.06 \end{bmatrix}$ is computed by substituting P and R in a Riccati equation

$$Q = \breve{P} - A^\top P A + K^\top (R + B^\top P B)^{-\top} K$$

From P, Q, R, the matrices $\breve{Q}$, $\breve{R}$ and $\breve{P}$ are obtained and are further used in the optimal control problem (A.2). Here we choose $\Delta = I_{|\mathcal{V}|}$. This choice of Δ means that we want to drive $x_i(k)$ towards the centroid of the $\mathbb{V}_i(k)$.

In the first scenario, the decentralized optimal control (A.2) is used. The results are illustrated in Fig. A.2. All the agents from their initial positions deploy $\mathcal{X}$ and quickly obtain the optimal coverage. This is shown in Fig. A.3 where the tracking errors $||x_i - \bar{x}_i||$ drop to zero after a certain time.

We get similar results in the second scenario, using the unconstrained MPC control (A.21). In this case, we use the same matrices P, Q, R and Δ of the first scenario. The prediction horizon is $N_p = 3$. As shown in Figs. A.4 and A.5, the optimal coverage is obtained as soon as the tracking errors converge to zero.

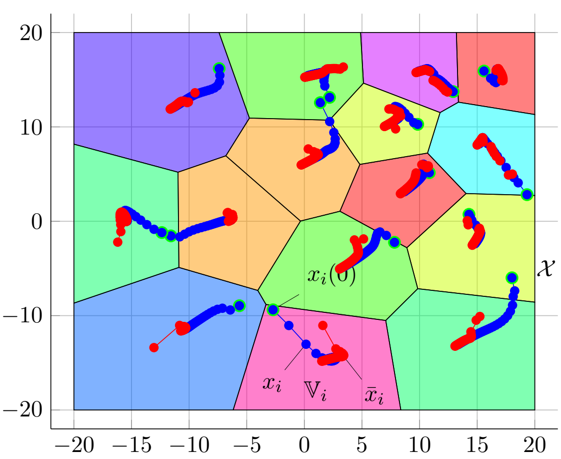

FIGURE A.2: Coverage by using optimal control approach.

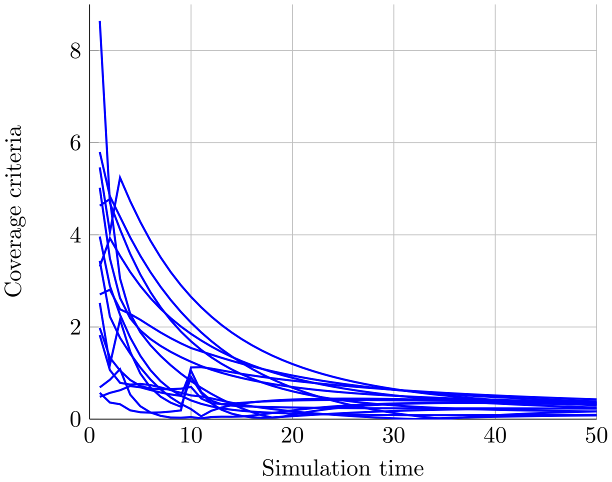

FIGURE A.3: Coverage criteria by using optimal control approach.

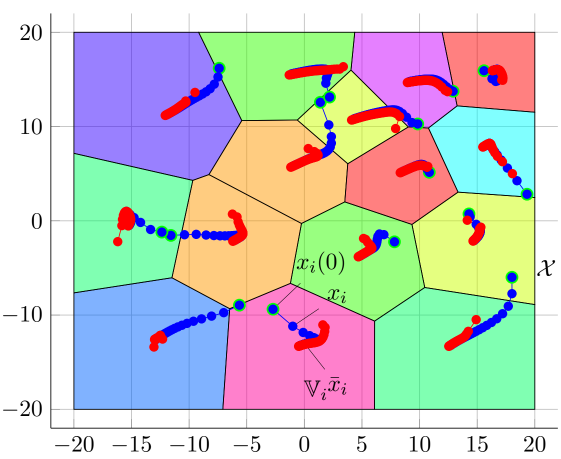

FIGURE A.4: Coverage by using unconstrained MPC approach.

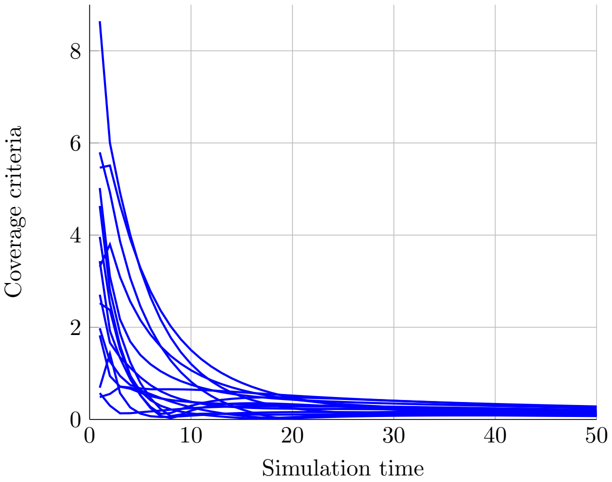

FIGURE A.5: Coverage criteria by using unconstrained MPC approach.

In the last scenario, the constraint $u(k) \in \mathbb{U}$ is added in the control computation (see (A.28)), with the set $\mathbb{U}$ defined as

$$\mathbb{U} = \left\{ u \in \mathbb{R}^2 \;\middle|\; \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} \right\}$$

This constraint represents the limited speed authorized by the actuators. The agents still converge to a coverage configuration (see Fig. A.6) (we recall here that these configurations are not unique, according to Du et al. (1999)) but longer than the second scenario (see Fig. A.6). This is logical due to the limitation of the speed.

In all scenarios, sometimes there are some ”jumps” representing the discontinuities due to the variation of the vertices defining the Voronoi cell. This problem is trivial because the number

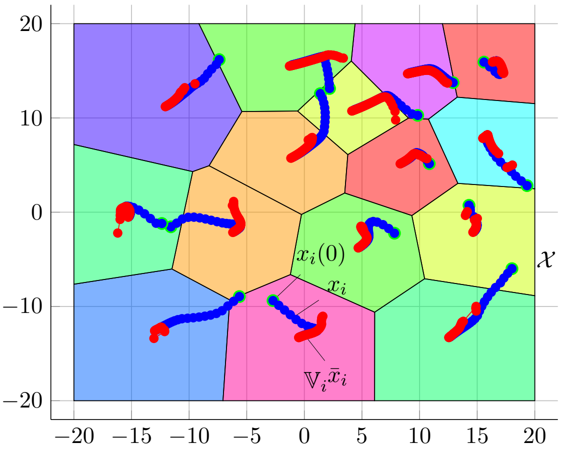

FIGURE A.6: Coverage by using constrained MPC approach.

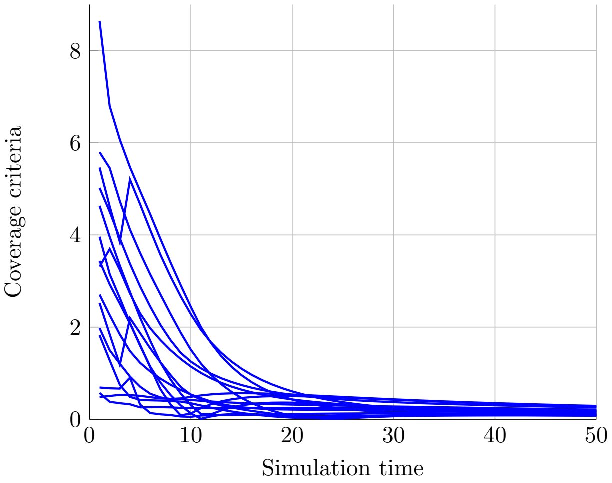

FIGURE A.7: Coverage criteria by using constrained MPC approach.

of neighbors taken into account in the construction of the Voronoi cell can change and also can cause the discontinuities in the evolution of the centroid.

A.4 Conclusion

The well-known Voronoi-based decentralized coverage problem is already revisited in Chapter 4 by choosing the Chebyshev center and further a so-called generalized center as the inner target point for each agent. The main idea consists of providing the decentralized control policies to enhance the coverage of a group of cooperative homogeneous mobile agents over a bounded region. The classical Voronoi algorithm is employed to partition the whole region in real time. This partition step is based on the positions of the agents and subsequently gives the necessary information to design the decentralized control action for each agent. In the present appendix, we consider a so-called vertex interpolated center and develop the decentralized control appropriate to drive the Multi-Agent system towards an optimal configuration associated with the new center. The control design is based on respectively the optimal control approach and further the MPC approach. We also provide the stability proofs corresponding to each approach. Both proposed approaches use the current state of the agent and the vertices of its Voronoi cell to compute the decentralized control.

Most of the works related to Voronoi-based decentralized coverage problem choose the center of mass of the Voronoi cell as the inner target point. The optimal coverage over the deployed region associated with this choice is thus obtained when each agent's output coincides with its center of mass. Such configuration known as centroidal Voronoi configuration can be achieved by driving each agent individually towards its center of mass, according to the principle of the Lloyd's algorithm. Notice that, in the open literature, most of the results are developed for continuous-time system. In Appendix B, we will revisit the decentralized control to drive a

Multi-Agent system towards a centroidal Voronoi configuration in the context of discrete-time dynamics.

Appendix B

Voronoi based decentralized coverage problem using the center of mass

The Multi-Agent system coverage is a well-known control problem in the literature (see Cortes et al. (2002), Schwager et al. (2009) and Moarref and Rodrigues (2014)). We have revisited this problem by choosing the inner point towards which each agent is driven, such as Chebyshev center, generalized center (these two kinds of center are introduced in Chapter 4) and a so-called vertex interpolated center (see Appendix A). It is worth to mention that the conventional inner target chosen for Voronoi-based deployment is the center of mass. Therefore, the decentralized control is design such that each agent goes to its corresponding center of mass, and thus the Multi-Agent system reaches a centroidal Voronoi configuration over a deployed region. However, most of the results obtained are developed for continuous-time systems.

In this context, the aim of the present appendix is to extend the optimal formulation-based decentralized control in Moarref and Rodrigues (2014) for discrete-time systems, which can be used in several applications, typically in the context of mobile sensor network Cortes et al. (2002), multi-robots coverage Schwager et al. (2009).

The main novelty is to use the prediction of the dynamic Voronoi partition in order to maximize the coverage quality by driving the Multi-Agent system to a centroidal Voronoi configuration. Similar to Appendix A, due to laborious mathematical formulations, the proposed results are formualted as an appendix. This work has been published in Nguyen et al. (2016a).

We will start by formulating the problem in Section B.1.

B.1 Problem formulation

In this section, we will introduce respectively the class of agent's dynamics in subsection B.1.1. This is followed by the details of the constraints on the agent's environment in subsection B.1.2

and the overall of decentralized Voronoi coverage problem of Moarref and Rodrigues (2014) in subsection B.1.3.

B.1.1 System description

Consider the Multi-Agent system (denoted by $\mathbf{\Sigma}$) composed of N mobile agents. The indices set is $\mathcal{N} = \{1, \dots, N\}$. Each agent has its own dynamics

$$\dot{x}_i = u_i, \text{with } i \in \mathcal{N} \tag{B.1}$$

which has the discrete-time zero-order-hold form

$$x_i(k+1) = x_i(k) + T_s u_i(k), \text{with } i \in \mathcal{N} \tag{B.2}$$

where $x_i \in \mathbb{R}^n$ is the state-space vector, $u_i \in \mathbb{R}^n$ is the input vector[1] and T_s is the sampling time. We use $x = [x_1^\top \dots x_N^\top]^\top \in \mathbb{R}^{Nn}$ and $u = [u_1^\top \dots u_N^\top]^\top \in \mathbb{R}^{Nn}$ to denote respectively the collective state and input of the system $\mathbf{\Sigma}$.

B.1.2 Constraints on the agents' environment

Assume that the common working space $\mathcal{X} \subset \mathbb{R}^n$ is convex and bounded, represented by a polytope. A partition $\mathbb{V}(x_1, \dots, x_N)$ of $\mathcal{X}$ is

$$\mathcal{X} = \bigcup_{i=1}^{N} \mathbb{V}_i, \quad \mathbb{V}_i \cap \mathbb{V}_j = \emptyset, \quad \forall i, j \in \mathcal{N} \tag{B.3}$$

A natural mathematical definition of such a decomposition is provided by the Voronoi partition, which characterizes the *neighborhood* $\mathbb{V}_i(x_i)$ as

$$\mathbb{V}_i = \{x \in \mathcal{X} |\ \|x_i - x\| \leq \|x_j - x\|,\ \forall j \neq i\} \tag{B.4}$$

It is worth to be mentioned that each set $\mathbb{V}_i$ is a polytope as a consequence of the boundedness of $\mathcal{X}$ and the structure of the constraints in (B.4). Using the available state measurement of the Multi-Agent system $\mathbf{\Sigma}$ at the time instant k, the geometric formulation (2.10) leads to a time-varying partition.

[1] The dimension choice of u_i follows the chosen dynamics (B.1). The results can be extended for $u_i \in \mathbb{R}^m$, with $m \neq n$.

B.1.3 Coverage control

The ultimate goal of the deployment is to maximize the coverage. The density function[2] $\phi : \mathcal{X} \to \mathbb{R}_+$ denotes the priority of coverage at a point $q \in \mathcal{X}$. A candidate Lyapunov function $V(x)$ (see Cortes et al. (2002), Cortes et al. (2005)) is defined as

$$V(x) = \sum_{i=1}^{N} p_i \int_{\mathbb{V}_i} \|x_i - q\|^2 \phi(q)dq \tag{B.5}$$

with the positive scalar p_i denoting a weighting coefficient for $V(x)$. The optimal (maximized) coverage is achieved if $V(x)$ reaches its minimum.

Its local minimum points are obtained by solving $\frac{\partial V}{\partial x_i} = 0$ with the partial derivative of $V(x)$ with respect to x_i being

$$\begin{aligned}\frac{\partial V}{\partial x_i} &= 2p_i \int_{\mathbb{V}_i} (x_i - q)^\top \phi(q)dq \\ &= 2p_i \left(\int_{\mathbb{V}_i} \phi(q)dq\right) \left(x_i - \frac{\int_{\mathbb{V}_i} q\phi(q)dq}{\int_{\mathbb{V}_i} \phi(q)dq}\right)^\top \\ &= 2p_i M_{\mathbb{V}_i}(x_i - C_{M_{\mathbb{V}_i}})^\top\end{aligned} \tag{B.6}$$

where the *mass* $M_{\mathbb{V}_i}$ and the *center of mass* $C_{M_{\mathbb{V}_i}}$ of the Voronoi cell $\mathbb{V}_i$ are respectively defined as in Moarref and Rodrigues (2014)

$$M_{\mathbb{V}_i} = \int_{\mathbb{V}_i} \phi(q)dq \tag{B.7}$$

$$C_{M_{\mathbb{V}_i}} = \frac{\int_{\mathbb{V}_i} q\phi(q)dq}{\int_{\mathbb{V}_i} \phi(q)dq} \tag{B.8}$$

Solving $\frac{\partial V}{\partial x_i} = 0, \forall i \in \mathcal{N}$ leads to an optimal configuration where $x_i = C_{M_{\mathbb{V}_i}}$. Such optimal configuration is called *Centroidal Voronoi Configuration* (CVC). Using the LaSalle's invariance principle (see Khalil and Grizzle (1996)), it can be proved that the agent's local control $u_i = k_i(x_i - C_{M_{\mathbb{V}_i}})$, with $k_i < 0$ can lead to the convergence of the entire MAS to a CVC.

The results obtained for the continuous-time MAS will be recalled in the next section.

B.2 Continuous-time decentralized optimal control

This section recalls the main results given in Moarref and Rodrigues (2014), applied for a MAS whose agents dynamics are characterized by the continuous-time equations (B.1). The

[2] The function ϕ is continuously differentiable over $\mathcal{X}$.

decentralized control is obtained locally by solving the following optimization control problem

$$\inf_{u_i, i\in\mathcal{N}} \int_0^\infty L(x,u)d\tau \text{ s.t.: } \dot{x}_i = u_i \tag{B.9}$$

where $L(x,u) = \sum_{i=1}^{N} \left(s_i \left\| \int_{\mathbb{V}_i} (x_i - q)\phi(q)dq \right\|^2 + r_i \left\| u_i \right\|^2 \right)$ denotes the running cost. The scalars $s_i > 0$, $r_i > 0$ represent the weighting coefficients. The Hamilton-Jacobi-Bellman (HJB) equation of (B.9) with the dynamics (B.1) is

$$\inf_{u_i, i\in\mathcal{N}} H\left(L(x,u), \tfrac{\partial V}{\partial x}\right) = 0 \tag{B.10}$$

with the Hamiltonian

$$H = L(x,u) + \tfrac{\partial V}{\partial x}\dot{x} \quad = \sum_{i=1}^{N} \left(s_i \left\| \int_{\mathbb{V}_i} (x_i - q)\phi(q)dq \right\|^2 + r_i \left\| u_i \right\|^2 + \frac{\partial V}{\partial x_i}\dot{x}_i \right)$$

associated with a value function $V = V(x)$

$$V(x) = \sum_{i=1}^{N} \sqrt{s_i r_i} \int_{\mathbb{V}_i} \left\| x_i - q \right\|^2 \phi(q)dq \tag{B.11}$$

with $s_i r_i = s_j r_j = p^2$, $\forall i,j \in \mathcal{N}$ and p denoting a constant positive scalar. By solving $\dfrac{\partial H}{\partial u_i} = 0$, we obtain the *continuous-time decentralized optimal control* (CDOC)

$$u_i = -\sqrt{\frac{s_i}{r_i}} \int_{\mathbb{V}_i} (x_i - q)\phi(q)dq, \quad i \in \mathcal{N} \tag{B.12}$$

Furthermore, the solution (B.12) satisfies the LaSalle's invariance principle and thus ensures the Lyapunov convergence of the MAS into a CVC by means of $\dot{V} = -L \leq 0$.

In the following, we propose a decentralized optimal control technique for Multi-Agent system Voronoi-based deployment.

B.3 Discrete-time decentralized optimal control

This section proposes an optimal control for the discrete-time system's dynamics (B.2), based on the approach developed in Moarref and Rodrigues (2014). For brevity, in the sequel we use $\mathbb{V}_i^+ = \mathbb{V}_i(x_i(k+1))$ to denote the Voronoi cell of $x_i(k+1)$, i.e. the i-th agent at time $k+1$. The mass and center of mass of the Voronoi cell $\mathbb{V}_i^+$ are respectively $M_{\mathbb{V}_i^+}$ and $C_{M_{\mathbb{V}_i^+}}$.

Transposing the previous work Moarref and Rodrigues (2014) to the discrete-time dynamics (B.2) yields the following optimization problem

$$\begin{aligned} &\min_{u_i, i\in\mathcal{N}} \sum_{k=0}^{\infty} L(x(k), u(k)) \\ &\text{s.t: } x_i(k+1) = x_i(k) + T_s u_i(k) \end{aligned} \tag{B.13}$$

with $L(x(k), u(k)) = \sum_{i=1}^{N} \left(s_i \left\| \int_{\mathbb{V}_i} (x_i(k) - q)\phi(q)dq \right\|^2 + r_i \left\| u_i(k) \right\|^2 \right)$ indicating the running cost.

Theorem B.1. *A discrete-time decentralized stabilizing suboptimal control of the problem* (B.13) *is*

$$u_i(k) = -\frac{p_i}{r_i + p_i T_s M_{\mathbb{V}_i^+}} \int_{\mathbb{V}_i^+} (x_i(k) - q)\phi(q)dq \tag{B.14}$$

by considering the cost-to-go function

$$V(x(k)) = \sum_{i=1}^{N} p_i \int_{\mathbb{V}_i} \left\| x_i(k) - q \right\|^2 \phi(q)dq \tag{B.15}$$

with $p_i = \sqrt{s_i r_i}$ *and* $p_i = p_j = p > 0, \ \forall i, j \in \mathcal{N}$.

Proof. The discrete-time Hamiltonian (see Bellman (1954)) of (B.13) is

$$H(k) = L(x(k), u(k)) + \frac{\Delta V(k)}{T_s} \tag{B.16}$$

with $\Delta V(k) = V(x(k+1)) - V(x(k))$. By using (B.15), we can express $\Delta V(k)$ as $\Delta V(k) = \sum_{i=1}^{N} p \int_{\mathbb{V}_i^+} \left\| x_i(k+1) - q \right\|^2 \phi(q)dq - \sum_{i=1}^{N} p \int_{\mathbb{V}_i} \left\| x_i(k) - q \right\|^2 \phi(q)dq$. Using the expressions (B.6)-(B.8), this can be rewritten in terms of the mass and the center of mass such as $\Delta V(k) = \sum_{i=1}^{N} p \Big(||x_i(k+1)||^2 M_{\mathbb{V}_i^+} + \int_{\mathbb{V}_i^+} ||q||^2 \phi(q)dq - 2x_i^\top(k+1) M_{\mathbb{V}_i^+} C_{M_{\mathbb{V}_i^+}} \Big) - \sum_{i=1}^{N} p \Big(||x_i(k)||^2 M_{\mathbb{V}_i} - 2x_i^\top(k) M_{\mathbb{V}_i} C_{M_{\mathbb{V}_i}} + \int_{\mathbb{V}_i} ||q||^2 \phi(q)dq \Big)$. By following the definition (2.10)-(B.4), we have

$$\sum_{i=1}^{N} \int_{\mathbb{V}_i} ||q||^2 \phi(q)dq = \int_{\bigcup_{i=1}^{N} \mathbb{V}_i} ||q||^2 \phi(q)dq = \int_{\mathcal{X}} ||q||^2 \phi(q)dq$$

and this leads to

$$\sum_{i=1}^{N} \int_{\mathbb{V}_i^+} ||q||^2 \phi(q)dq - \sum_{i=1}^{N} \int_{\mathbb{V}_i} ||q||^2 \phi(q)dq = 0$$

This allows us to rewrite $\Delta V(k)$ as follows

$$\Delta V(k) = \sum_{i=1}^{N} p\Big(||x_i(k+1)||^2 - 2x_i^\top(k+1) C_{M_{\mathbb{V}_i^+}} \Big) M_{\mathbb{V}_i^+} - \sum_{i=1}^{N} pM_{\mathbb{V}_i}\Big(||x_i(k)||^2 - 2x_i^\top(k) C_{M_{\mathbb{V}_i}} \Big) \tag{B.17}$$

Differentiating $H(k)$ with respect to $u_i(k)$, we obtain

$$\frac{\partial H(k)}{\partial u_i(k)} = 2r_i u_i^\top(k) + \frac{1}{T_s}\frac{\partial \Delta V(k)}{\partial u_i(k)} \tag{B.18}$$

where the partial derivative of $\Delta V(k)$ with respect to $u_i(k)$ can be obtained by differentiating (B.17) by $u_i(k)$ (using the dynamics (B.2)), i.e.

$$\tfrac{\partial \Delta V(k)}{\partial u_i(k)} = 2pT_s M_{\mathbb{V}_i^+}\Big(x_i(k+1) - C_{M_{\mathbb{V}_i^+}} \Big)^\top + T_s\theta_i \tag{B.19}$$

where
$\theta_i = \frac{p}{T_s}\Big(||x_i(k+1)||^2 \frac{\partial M_{\mathbb{V}_i^+}}{\partial u_i(k)} - 2x_i^\top(k+1) \frac{\partial (M_{\mathbb{V}_i^+} C_{M_{\mathbb{V}_i^+}})}{\partial u_i(k)} \Big) + \frac{1}{T_s} \sum_{j\in\mathcal{N}_i} p\Big(||x_j(k+1)||^2 \frac{\partial M_{\mathbb{V}_j^+}}{\partial u_i(k)} - 2x_j^\top(k+1) \frac{\partial (M_{\mathbb{V}_j^+} CM_{\mathbb{V}_j^+})}{\partial u_i(k)} \Big)$
collects all partial derivatives of the mass and center of mass with respect to $u_i(k)$.

Replacing (B.19) in (B.18), we get $\frac{\partial H(k)}{\partial u_i(k)} = 2r_i u_i^\top(k) + 2px_i^\top(k+1) M_{\mathbb{V}_i^+} - 2pM_{\mathbb{V}_i^+} C^\top_{M_{\mathbb{V}_i^+}} + \theta_i$. Using the discrete-time dynamics (B.2) and regrouping the terms related to the control $u_i(k)$, the following expression is obtained

$$\frac{\partial H(k)}{\partial u_i(k)} = 2(r_i + pT_s M_{\mathbb{V}_i^+}) u_i^\top(k) + 2pM_{\mathbb{V}_i^+}(x_i(k) - C_{M_{\mathbb{V}_i^+}})^\top + \theta_i \tag{B.20}$$

A suboptimal solution $u_i(k)$ can be obtained by solving $(r_i + pT_s M_{\mathbb{V}_i^+}) u_i^\top(k) + pM_{\mathbb{V}_i^+}(x_i(k) - C_{M_{\mathbb{V}_i^+}})^\top = 0$, i.e.

$$u_i(k) = -\frac{pM_{\mathbb{V}_i^+}}{r_i + pT_s M_{\mathbb{V}_i^+}} (x_i(k) - C_{M_{\mathbb{V}_i^+}}) \tag{B.21}$$

which is also the control solution (B.14) and the proof of the main claim is completed.

With respect to the local stability of the i^{th} agent in case of static Voronoi partition, by replacing (B.14) in (B.2), the closed-loop dynamics becomes
$x_i(k+1) = \frac{r_i}{r_i + p_i T_s M_{\mathbb{V}_i^+}} x_i(k) + \frac{p_i T_s M_{\mathbb{V}_i^+}}{r_i + p_i T_s M_{\mathbb{V}_i^+}} C_{M_{\mathbb{V}_i^+}}.$

Obviously, the following expression holds

$$\frac{r_i}{r_i + p_i T_s M_{\mathbb{V}_i^+}} \leq 1$$

because $r_i, p_i, T_s, M_{\mathbb{V}_i^+} > 0$. □

Remark B.1. In Theorem B.1, the general case $p_i = p_j = p$ was considered. Notice that we can choose $p = 1$ in order to keep the consistency with the results proposed in Moarref and Rodrigues (2014).

Remark B.2. The computation of the control (B.14) requires the predicted mass $M_{\mathbb{V}_i^+}$ and the predicted center of mass $C_{M_{\mathbb{V}_i^+}}$ of the Voronoi cell $\mathbb{V}_i^+$. This is considered as the main difference by comparison with the continuous-time case.

Lemma B.2. *The suboptimal solution* (B.21) *can be considered as a discrete-time approximation of the continuous-time decentralized optimal control of the problem* (B.13) *because the terms* θ_i *satisfy*

$$\sum_{i=1}^{N} \int \theta_i du_i(k) = 0 \tag{B.22}$$

Proof. Using the control solution (B.21) translates the expression (B.20) into $\dfrac{\partial H(k)}{\partial u_i(k)} = \theta_i$. According to the definition of total derivative of multi-variable function, we can integrate $\dfrac{\partial H(k)}{\partial u_i(k)}$ with respect to all $u_i(k)$ and thus we obtain

$$H(k) = \sum_{i=1}^{N} \int \theta_i du_i(k) \tag{B.23}$$

Consider the mass conservation law (see Cortes et al. (2005)), i.e.

$$\frac{\partial}{\partial x_i} \int_{\mathbb{V}_i} \|x_i - q\|^2 \phi(q) dq = \int_{\mathbb{V}_i} \frac{\partial}{\partial x_i} \|x_i - q\|^2 \phi(q) dq + \int_{\partial \mathbb{V}_i} \|x_i - \gamma\|^2 \phi(\gamma) n^\top(\gamma) \frac{\partial \gamma}{\partial x_i} d\gamma \tag{B.24}$$

where $\partial \mathbb{V}_i$ is the boundary of the set $\mathbb{V}_i$, i.e. $\partial \mathbb{V}_i = \bigcup_{\forall j \in \mathcal{N}_i} (\mathbb{V}_i \cap \mathbb{V}_j)$. Here, $n(\gamma)$ is the unit outward normal to $\partial \mathbb{V}_i$ which is parameterized by the scalar γ. Similar to (B.24), for the function (B.15), we obtain

$$\frac{\partial V(x)}{\partial x_i} = p \frac{\partial}{\partial x_i} \int_{\mathbb{V}_i} \|x_i - q\|^2 \phi(q) dq$$

and derive

$$\frac{\partial V(x)}{\partial x_i} = 2p \int_{\mathbb{V}_i} (x_i - q)^\top \phi(q) dq + p \int_{\partial \mathbb{V}_i} \|x_i - \gamma\|^2 \phi(\gamma) n^\top(\gamma) \frac{\partial \gamma}{\partial x_i} d\gamma$$

Based on (B.6), the discrete-time form at the time instant $k+1$ is further obtained

$$\frac{\partial V(x(k+1))}{\partial x_i(k+1)} = 2pM_{\mathbb{V}_i^+} \left(x_i(k+1) - C_{M_{\mathbb{V}_i^+}} \right)^\top + p \int_{\partial \mathbb{V}_i^+} \|x_i(k+1) - \gamma\|^2 \phi(\gamma) n^\top(\gamma) \frac{\partial \gamma}{\partial x_i} d\gamma \tag{B.25}$$

Using the dynamics (B.2), we have additionally

$$\frac{\partial \Delta V}{\partial u_i(k)} = \frac{\partial V(x(k+1))}{\partial x_i(k+1)} \frac{\partial x_i(k+1)}{\partial u_i(k)} = \frac{\partial V(x(k+1))}{\partial x_i(k+1)} T_s \tag{B.26}$$

By substituting (B.25) in (B.26), we obtain

$$\frac{\partial \Delta V}{\partial u_i(k)} = 2pM_{\mathbb{V}_i^+} T_s \Big(x_i(k+1) - C_{M_{\mathbb{V}_i^+}} \Big)^\top + T_s p \int_{\partial \mathbb{V}_i^+} \|x_i(k+1) - \gamma\|^2 \phi(\gamma) n^\top(\gamma) \frac{\partial \gamma}{\partial x_i} d\gamma \tag{B.27}$$

Subsequently, from (B.19) and (B.27), we get another equation characterizing θ_i

$$\theta_i = p \int_{\partial \mathbb{V}_i^+} \|x_i(k+1) - \gamma\|^2 \phi(\gamma) n^\top(\gamma) \frac{\partial \gamma}{\partial x_i} d\gamma \tag{B.28}$$

Furthermore, the authors of Moarref and Rodrigues (2014) proved that

$$\int_{\partial \mathbb{V}_i} \|x_i - \gamma\|^2 \phi(\gamma) n^\top(\gamma) \frac{\partial \gamma}{\partial x_i} d\gamma = \sum_{j \in \mathcal{N}_i} \int_{\mathbb{V}_i \cap \mathbb{V}_j} \|x_i - \gamma_{ij}\|^2 \phi(\gamma_{ij}) n^\top(\gamma_{ij}) \frac{\partial \gamma_{ij}}{\partial x_i} d\gamma_{ij} \tag{B.29}$$

which represents the mass variation through the boundary of $\mathbb{V}_i$ as a collection of mass flow through each facet $\mathbb{V}_i \cap \mathbb{V}_j$ defining $\partial \mathbb{V}_i$. Therefore, we obtain

$$\theta_i = p \sum_{j \in \mathcal{N}_i} \int_{\mathbb{V}_i^+ \cap \mathbb{V}_j^+} \|x_i(k+1) - \gamma_{ij}\|^2 \phi(\gamma_{ij}) n^\top(\gamma_{ij}) \frac{\partial \gamma_{ij}}{\partial x_i} d\gamma_{ij} \tag{B.30}$$

Additionally, we have $\displaystyle\sum_{i=1}^{N} \int \theta_i du_i(k) = \sum_{i=1}^{N} \int \frac{\theta_i}{T_s} dx_i(k+1)$ with

$$\sum_{i=1}^{N} \int \frac{\theta_i}{T_s} dx_i(k+1) = \sum_{i=1}^{N} \frac{p_i}{T_s} \int \Big(\sum_{j \in \mathcal{N}_i} \int_{\mathbb{V}_i^+ \cap \mathbb{V}_j^+} \|x_i(k+1) - \gamma_{ij}\|^2 \phi(\gamma_{ij}) n^\top(\gamma_{ij}) \frac{\partial \gamma_{ij}}{\partial x_i} d\gamma_{ij} \Big) dx_i(k+1)$$

Since

$$\|x_i - \gamma_{ij}\| = \|x_j - \gamma_{ji}\|, \frac{\partial \gamma_{ij}}{\partial x_i} = \frac{\partial \gamma_{ji}}{\partial x_j}, n(\gamma_{ij}) = -n(\gamma_{ji})$$

it is possible to write[3]

$$\int_{\mathbb{V}_i \cap \mathbb{V}_j} \|x_i - \gamma_{ij}\|^2 \phi(\gamma_{ij}) n^\top(\gamma_{ij}) \frac{\partial \gamma_{ij}}{\partial x_i} d\gamma_{ij} = -\int_{\mathbb{V}_j \cap \mathbb{V}_i} \|x_j - \gamma_{ji}\|^2 \phi(\gamma_{ji}) n^\top(\gamma_{ji}) \frac{\partial \gamma_{ji}}{\partial x_j} d\gamma_{ji} \tag{B.31}$$

[3] From the point of view of mass conservation, we consider the case when $\mathcal{X} = \mathbb{V}_i \cup \mathbb{V}_j$. If one part of the mass belonging to the $\mathbb{V}_i$ cell passes through the facet $\mathbb{V}_i \cap \mathbb{V}_j$ into the $\mathbb{V}_j$ cell, then $\mathbb{V}_i$ loses it but the cell $\mathbb{V}_j$ gets it. In general, the mass over $\mathcal{X}$ is conserved.

and thus the following integral vanishes

$$\sum_{i=1}^{N} p \int \Big(\sum_{j\in\mathcal{N}_i} \int_{\mathbb{V}_i\cap\mathbb{V}_j} \|x_i - \gamma_{ij}\|^2 \phi(\gamma_{ij}) n^\top(\gamma_{ij}) \frac{\partial \gamma_{ij}}{\partial x_i} d\gamma_{ij} \Big) dx_i$$

This result leads to $\sum_{i=1}^{N} \int \theta_i du_i(k) = 0$ and thus, the equation (B.23) becomes $H(k) = 0$, proving that the control (B.21) can be considered as a discrete-time approximation of the optimal control solution of the problem (B.13). □

We have found a discrete-time approximation of CDOC for the optimization problem (B.13). In the next section, we will show the consistence between this solution and the continuous-time optimal control proposed in Moarref and Rodrigues (2014).

B.4 Equivalence between discrete-time approximation of CDOC and CDOC-Stability proof

The previous section presents the proposed discrete-time approximation of CDOC (B.14). It will be proved to converge to the CDOC of Moarref and Rodrigues (2014).

Theorem B.3. *If the sampling time T_s goes to zero, then the next three statements are true:*
i. The discretized equation (B.2) *approaches the continuous-time dynamics* (B.1)*;*
ii. The discrete-time approximation of CDOC (B.14) *approaches CDOC* (B.12)*;*
iii. The HJB equation $\min_u H = 0$ is ensured.

Proof. i. The first statement is obvious.
ii. Consider the discrete-time control solution (B.14). If $T_s \to 0$, this solution (B.14) approaches the limit value $u_i(t) = -\frac{p}{r_i} \int_{\mathbb{V}_i} (x_i(t) - q)\phi(q)dq = -\sqrt{\frac{s_i}{r_i}} \int_{\mathbb{V}_i} (x_i(t) - q)\phi(q)dq$ which is exactly the CDOC solution, for $p^2 = s_i r_i$ (see (B.6) and (B.11)).
iii. According to the definition of the total derivative of a multivariable function, it is inferred that

$$\Delta V(k) = \sum_{i=1}^{N} \int \frac{\partial \Delta V(k)}{\partial u_i(k)} du_i(k) \tag{B.32}$$

Replacing $\Delta V(k)$ in the Hamiltonian (B.16) leads to $H(k) = \sum_{i=1}^{N} \Big(s_i \Big\| \int_{\mathbb{V}_i} (x_i(k) - q)\phi(q)dq \Big\|^2 + r_i \|u_i(k)\|^2 \Big) + \frac{1}{T_s} \sum_{i=1}^{N} \int \frac{\partial \Delta V(k)}{\partial u_i(k)} du_i(k)$. Substitute $\frac{\partial \Delta V(k)}{\partial u_i(k)}$ with (B.19) and use (B.22) to obtain

$$\begin{aligned} H(k) &= \sum_{i=1}^{N}\left(s_i\left\|\int_{\mathbb{V}_i}(x_i(k)-q)\phi(q)dq\right\|^2 + r_i\left\|u_i(k)\right\|^2\right) \\ &+\sum_{i=1}^{N}2p\int\left(M_{\mathbb{V}_i^+}x_i(k+1)-M_{\mathbb{V}_i^+}C_{M_{\mathbb{V}_i^+}}\right)^{\top}du_i(k) \end{aligned} \tag{B.33}$$

and thus

$$\begin{aligned} H(k) &= \sum_{i=1}^{N}\left(s_i\left\|\int_{\mathbb{V}_i}(x_i(k)-q)\phi(q)dq\right\|^2 + r_i\left\|u_i(k)\right\|^2\right) \\ &+\sum_{i=1}^{N}2pM_{\mathbb{V}_i^+}\left(\int x_i^{\top}(k+1)du_i(k)-C_{M_{\mathbb{V}_i^+}}^{\top}\int du_i(k)\right) \end{aligned} \tag{B.34}$$

Using the dynamics (B.2), this can be rewritten as

$$\begin{aligned} H(k) &= \sum_{i=1}^{N}\left(s_i\left\|\int_{\mathbb{V}_i}(x_i(k)-q)\phi(q)dq\right\|^2 + r_i\left\|u_i(k)\right\|^2\right) \\ &+\sum_{i=1}^{N}2pM_{\mathbb{V}_i^+}\left(T_s\frac{||u_i(k)||^2}{2}+(x_i(k)-C_{M_{\mathbb{V}_i^+}})^{\top}u_i(k)\right) \end{aligned} \tag{B.35}$$

Regrouping the terms in $||u_i(k)||^2$, the expression (B.35) becomes

$$H(k)=\sum_{i=1}^{N}\left(s_i\left\|\int_{\mathbb{V}_i}(x_i(k)-q)\phi(q)dq\right\|^2+(r_i+pT_sM_{\mathbb{V}_i^+})\left\|u_i(k)\right\|^2+2pM_{\mathbb{V}_i^+}(x_i(k)-C_{M_{\mathbb{V}_i^+}})^{\top}u_i(k)\right)$$

Replacing $u_i(k)$ by the solution (B.14), this is equivalent to

$$H(k)=\sum_{i=1}^{N}\left(s_i\left\|\int_{\mathbb{V}_i}(x_i(k)-q)\phi(q)dq\right\|^2-\frac{p^2}{r_i+pT_sM_{\mathbb{V}_i^+}}\left\|\int_{\mathbb{V}_i^+}(x_i(k)-q)\phi(q)dq\right\|^2\right)$$

When $T_s \to 0$, using $C_{M_{\mathbb{V}_i^+}} \to C_{M_{\mathbb{V}_i}}$ and $M_{\mathbb{V}_i^+} \to M_{\mathbb{V}_i}$, the control law (B.14) becomes

$$u_i(t) = -\frac{p}{r_i}\int_{\mathbb{V}_i}(x_i(t)-q)\phi(q)dq \tag{B.36}$$

which is further used to find a simplified form of $\min_u H$

$$\min_u H = \sum_{i=1}^{N}\left(s_i-\frac{p^2}{r_i}\right)\left\|\int_{\mathbb{V}_i}(x_i(k)-q)\phi(q)dq\right\|^2 \tag{B.37}$$

Using $p^2 = s_i r_i$, we conclude that if $T_s \to 0$ then $\min_u H \to 0$. The proof is thus completed. □

Briefly, we have proved that the discrete-time approximation of CDOC (B.14) approaches CDOC (B.12) when $T_s \to 0$, and further leads to the stability in the sense of the HJB equation. In the next section, we will provide some numerical simulations to illustrate the performance of our decentralized control solution.

B.5 Numerical example

We consider a Multi-Agent system $\mathbf{\Sigma}$ composed of $N = 7$ homogeneous agents having the common dynamics (B.2), with the sampling time $T_s = 0.01$s, $x_i \in \mathbb{R}^2$ and $u_i \in \mathbb{R}^2$ denoting respectively the agents position and speed.

The agents are deployed within a bounded region

$$\mathcal{X} = \text{conv}\left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 6 \end{bmatrix}, \begin{bmatrix} 6 \\ 6 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \end{bmatrix} \right\} \subset \mathbb{R}^2$$

as illustrated in Fig. B.1. The density function over $\mathcal{X}$ is uniform, i.e. $\phi(q) = 1, \forall q \in \mathcal{X}$.

In Figs. B.1 and B.3, the blue lines represent the motion of the agents in the considered discrete-time case. The agents initial positions $x_i(0)$ are marked by the red points (see the zoom shown in Figs. B.1 and B.3). The last configuration of the entire MAS is shown in each figure, with the green dots denoting the last positions of the agents. The evolution of the coverage criterion $M_{\mathbb{V}_i}\|x_i - C_{M_{\mathbb{V}_i}}\|$ for each agent is shown in Figs. B.2 and B.4.

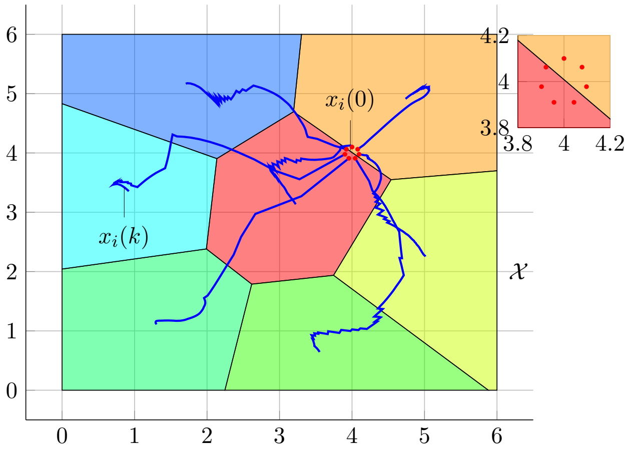

FIGURE B.1: Coverage of $\mathcal{X}$ with $s_i = 10$ and $r_i = 1$.

Two scenarios are considered. In the first scenario, we apply the decentralized control (B.14) with the weighting coefficients $s_i = 10$ and $r_i = 1$. The deployment result is shown in Fig. B.1 with the evolution of the agents and also the Voronoi partition obtained after 100 sampling periods. The coverage criterion curves in Fig. B.2 drop asymptotically to zero, proving that the entire MAS $\mathbf{\Sigma}$ is close to a CVC.

In the second scenario, the weighting coefficients are $s_i = 100$ and $r_i = 1$. The MAS converges into a CVC different from the previous scenario (notice that a CVC is not not unique Du et al.

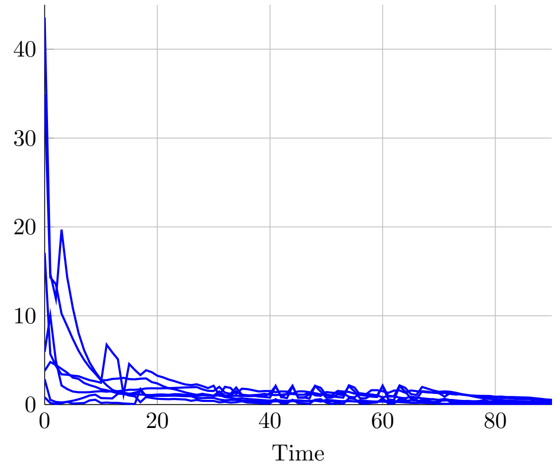

FIGURE B.2: Coverage criterion of N agents with $s_i = 10$ and $r_i = 1$.

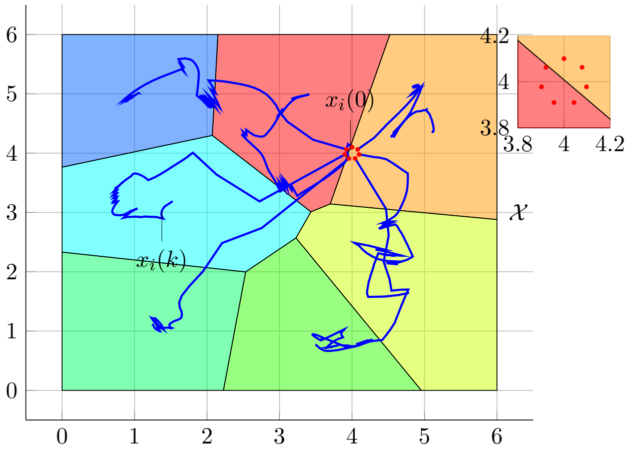

FIGURE B.3: Coverage of $\mathcal{X}$ with $s_i = 100$ and $r_i = 1$.

(1999)) with a faster convergence rate. The drop to zero of the coverage criterion proves that $\mathbf{\Sigma}$ approaches the CVC (see Fig B.2), although there are some discontinuities due to the abrupt change of the shape of the dynamic Voronoi cells.

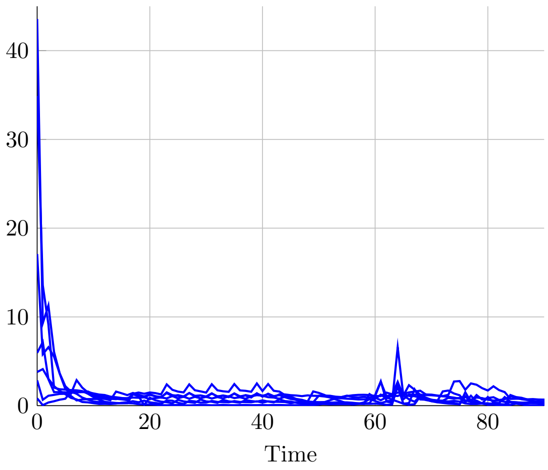

FIGURE B.4: Coverage criterion of N agents with $s_i = 100$ and $r_i = 1$.

B.6 Conclusion

This appendix provides a novel discrete-time decentralized control for the Multi-Agent Voronoi-based coverage/deployment problem, by following the optimal control framework. Similar to the continuous-time case, the discrete-time solution is also spatially distributed over Delaunay graphs. One of the contributions consists of taking into account the prediction of the Voronoi partition. However, finding an explicit equation to predict this partition appears to be complicated and it is part of our current research work. Some simulations exhibit oscillations/discontinuities in the evolution of the centroids which need to be analyzed in the future with respect to the sampling time and the topology of the partition. Another interesting topic to be addressed in future work relates to the choice of the weighting coefficients for the energy-efficiency problem. Extending the control solutions for the case of finite horizon is part of current work.

The propositions given in this appendix along with the results provided in Appendix A and Chapter 4 can considered as our contributions to the Voronoi-based coverage problem. The main achievement consists in improving the classical Lloyd's algorithm by choosing different kinds of inner target points and further developing the appropriate control laws.

List of Figures

List of Tables

Bibliography

Farnaz Adib Yaghmaie, Frank L Lewis, and Rong Su. Output regulation of heterogeneous linear multi-agent systems with differential graphical game. *International Journal of Robust and Nonlinear Control*, 2015.

Sinan Altug, Mo-Yuen Chen, and H Joel Trussell. Fuzzy inference systems implemented on neural architectures for motor fault detection and diagnosis. *IEEE transactions on industrial electronics*, 46(6):1069–1079, 1999.

Gianluca Antonelli, Filippo Arrichiello, Fabrizio Caccavale, and Armando Marino. A decentralized controller-observer scheme for multi-agent weighted centroid tracking. *Automatic Control, IEEE Transactions on*, 58(5):1310–1316, 2013.

Jean-Pierre Aubin. Viability theory. systems & control: Foundations & applications. *Birkhäuser, Boston. doi*, 10(1007):978–0, 1991.

Jean-Pierre Aubin. *Viability theory*. Springer Science & Business Media, 2009.

Jean-Pierre Aubin and Patrick Saint-Pierre. An introduction to viability theory and management of renewable resources. *Decision Making and Risk Management in Sustainability Science*, pages 43–80, 2007.

Franz Aurenhammer. Voronoi diagramsa survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

Efstathios Bakolas and Panagiotis Tsiotras. The Zermelo-Voronoi diagram: A dynamic partition problem. *Automatica*, 46(12):2059–2067, 2010.

Efstathios Bakolas and Panagiotis Tsiotras. Optimal partitioning for spatiotemporal coverage in a drift field. *Automatica*, 49(7):2064–2073, 2013.

Laura E Barnes, Mary Anne Fields, and Kimon P Valavanis. Swarm formation control utilizing elliptical surfaces and limiting functions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(6):1434–1445, 2009.

Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 57–64. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

Dario Bauso, Laura Giarre, and Raffaele Pesenti. Consensus in noncooperative dynamic games: A multiretailer inventory application. *Automatic Control, IEEE Transactions on*, 53(4):998–1003, 2008.

Richard Bellman. Dynamic programming and a new formalism in the calculus of variations. *Proceedings of the national academy of sciences*, 40(4):231–235, 1954.

Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

Alberto Bemporad and Claudio Rocchi. Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 7488–7493. IEEE, 2011.

Ricardo Bencatel, Mariam Faied, Joao Sousa, and Anouck R Girard. Formation control with collision avoidance. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 591–596. IEEE, 2011.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

George Bitsoris and Eliana Gravalou. Design techniques for the control of discrete-time systems subject to state and control constraints. *Automatic Control, IEEE Transactions on*, 44(5): 1057–1061, 1999.

George Bitsoris and Sorin Olaru. Further Results on the Linear Constrained Regulation Problem. In *Mediterranean Control Conference*, Greece, June 2013.

Georges Bitsoris. On the positive invariance of polyhedral sets for discrete-time systems. *Systems & control letters*, 11(3):243–248, 1988a.

Georges Bitsoris. On the positive invariance of polyhedral sets for discrete-time systems. *Systems & Control Letters*, 11(3), 1988b.

Georges Bitsoris. Positively invariant polyhedral sets of discrete-time linear systems. *International Journal of Control*, 47(6):1713–1726, 1988c.

Franco Blanchini. Nonquadratic lyapunov functions for robust control. *Automatica*, 31(3): 451–461, 1995.

Franco Blanchini and Stefano Miani. *Set-theoretic methods in control.* Birkhauser, 2007.

Mogens Blanke and Jochen Schröder. *Diagnosis and fault-tolerant control*, volume 691. Springer, 2006.

Jean-Daniel Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics (TOG)*, 3(4):266–286, 1984.

Francesco Borrelli, Tamás Keviczky, and Gary J Balas. Collision-free uav formation flight using decentralized optimization and invariant sets. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 1099–1104. IEEE, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

Robert L Brabenec. *Resources for the study of real analysis.* MAA, 2004.

Eduardo F Camacho and Carlos Bordons. *Model predictive control.* Springer, 2013.

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *Industrial Informatics, IEEE Transactions on*, 9(1):427–438, 2013.

Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 2012.

Donald Chmielewski and V Manousiouthakis. On constrained infinite-time linear quadratic optimal control. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, volume 2, pages 1319–1324. IEEE, 1996.

Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. Leader–follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44(5): 1343–1349, 2008.

Jorge Cortes, Sonia Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(04):691–719, 2005.

Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1327–1332. IEEE, 2002.

Daniel Crevier. *AI: The tumultuous history of the search for artificial intelligence.* Basic Books, Inc., 1993.

Philippe M Cury, Christian Mullon, Serge M Garcia, and Lynne J Shannon. Viability theory for an ecosystem approach to fisheries. *ICES Journal of Marine Science: Journal du Conseil*, 62(3):577–584, 2005.

Lucas Barcelos de Oliveira and Eduardo Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1):120–139, 2010.

Dimos V Dimarogonas, Emilio Frazzoli, and Karl H Johansson. Distributed event-triggered control for multi-agent systems. *Automatic Control, IEEE Transactions on*, 57(5):1291–1297, 2012.

Dimos V Dimarogonas, Panagiotis Tsiotras, and Kostas J Kyriakopoulos. Leader–follower cooperative attitude control of multiple rigid bodies. *Systems & Control Letters*, 58(6):429–435, 2009.

Aris L Dimeas and Nikos D Hatziargyriou. Operation of a multiagent system for microgrid control. *Power Systems, IEEE Transactions on*, 20(3):1447–1455, 2005.

G Lejeune Dirichlet. Über die reduction der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *Journal für die reine und angewandte Mathematik*, 40:209–227, 1850.

Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

Matthew G Earl and Raffaello D Andrea. Modeling and control of a multi-agent system using mixed integer linear programming. In *IEEE Conference on Decision and Control*, volume 1, pages 107–111. IEEE; 1998, 2001.

Christopher Edwards, Sarah K Spurgeon, and Ron J Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36(4):541–553, 2000.

Magnus B Egerstedt and Xiaoming Hu. Formation constrained multi-agent control. 2001.

N Eva Wu, Youmin Zhang, and Kemin Zhou. Detection, estimation, and accommodation of loss of control effectiveness. *International Journal of Adaptive Control and Signal Processing*, 14(7):775–795, 2000.

Adriano Fagiolini, Gianni Valenti, Lucia Pallottino, Gianluca Dini, and Antonio Bicchi. Decentralized intrusion detection for secure cooperative multi-agent systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 1553–1558. IEEE, 2007.

F. Farokhi and Karl H. Johansson. A study of truck platooning incentives using a congestion game. *Transactions on Intelligent Transportation Systems*, pages 16:2, 581–595, 2015.

J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *Automatic Control, IEEE Transactions on*, 49(9):1465–1476, 2004.

Tse-yun Feng. A survey of interconnection networks. *Computer*, 14(12):12–27, 1981.

Fernando ACC Fontes, Dalila BMM Fontes, and Amélia CD Caldeira. Model predictive control of vehicle formations. In *Optimization and Cooperative Control Strategies*, pages 371–384. Springer, 2009.

Giuseppe Franze, Francesco Tedesco, and Domenico Famularo. Actuator fault tolerant control: a set-theoretic approach. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1822–1827. IEEE, 2012.

Komei Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272, 2004.

Yan Gao, John Lygeros, Marc Quincampoix, and Nicolas Seube. On the control of uncertain impulsive systems: approximate stabilization and controlled invariance. *International Journal of Control*, 77(16):1393–1407, 2004.

Janos Gertler. Analytical redundancy methods in fault detection and isolation. In *Preprints of IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS91*, pages 9–21, 1991.

E.G. Gilbert and Kok Tin Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE TAC*, 36(9), Sep 1991.

A Grancharova and TA Johansen. Distributed mpc of interconnected nonlinear systems by dynamic dual decomposition. In *Distributed Model Predictive Control Made Easy*, pages 293–308. Springer, 2014.

Alexandra Grancharova and Tor A Johansen. Distributed quasi-nonlinear model predictive control by dual decomposition. In *Proceedings of the 18th IFAC World Congress, Milano, Italy*, pages 1429–1434, 2011.

Alexandra Grancharova and Sorin Olaru. An approach to distributed robust model predictive control of discrete-time polytopic systems. In *IFAC WORLD CONGRESS 2014*, volume 19, pages 2576–2581, 2014.

P-O Gutman and Michael Cwikel. Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states. *IEEE TAC*, 31(4):373–376, 1986.

M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. http://control.ee.ethz.ch/~mpt.

H. Hindi. A tutorial on convex optimization. In *IEEE American Control Conference*, volume 4, pages 3252–3265, June 2004.

Rolf Isermann. Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in control*, 29(1):71–85, 2005.

Tor A Johansen and Camilla Storaa. Energy-based control of a distributed solar collector field. *Automatica*, 38(7):1191–1199, 2002.

Michael Jünger, Thomas M Liebling, Denis Naddef, George L Nemhauser, William R Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A Wolsey. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-art.* Springer Science & Business Media, 2009.

Evangelos L Karfopoulos and Nikos D Hatziargyriou. A multi-agent system for controlled charging of a large population of electric vehicles. *Power Systems, IEEE Transactions on*, 28 (2):1196–1204, 2013.

Pia L Kempker, André CM Ran, and Jan H Van Schuppen. A formation flying algorithm for autonomous underwater vehicles. In *CDC-ECE*, pages 1293–1298, 2011.

Pia L Kempker, André CM Ran, and Jan H van Schuppen. Controllability and observability of coordinated linear systems. *Linear Algebra and its Applications*, 437(1):121–167, 2012.

Tamás Keviczky, Francesco Borrelli, Kingsley Fregene, Datta Godbole, and Gary J Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *Control Systems Technology, IEEE Transactions on*, 16(1):19–33, 2008.

Tamás Keviczky and Karl Henrik Johansson. A study on distributed model predictive consensus. *arXiv preprint arXiv:0802.4450*, 2008.

Hassan K Khalil and JW Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

O Khatib, Kazu Yokoi, K Chang, Diego Ruspini, Robert Holmberg, and Arancha Casal. Coordination and decentralized cooperation of multiple mobile manipulators. *Journal of Robotic Systems*, 13(11):755–764, 1996.

Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.

Daniel E Koditschek. Task encoding: Toward a scientific paradigm for robot planning and control. *Robotics and autonomous systems*, 9(1):5–39, 1992.

Ernesto Kofman, Hernan Haimovich, and Marıa M Seron. A systematic method to obtain ultimate bounds for perturbed systems. *International Journal of Control*, 80(2):167–178, 2007.

Mayuresh V Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

Alexander B Kurzhanski and Pravin Varaiya. Ellipsoidal techniques for reachability analysis: internal approximation. *Systems & control letters*, 41(3):201–211, 2000.

Andrew Kwok and Sonia Martinez. Deployment algorithms for a power-constrained mobile sensor network. *International Journal of Robust and Nonlinear Control*, 20(7):745–763, 2010.

Gerardo Lafferriere, J Caughman, and A Williams. Graph theoretic methods in the stability of vehicle formations. In *IEEE American Control Conference*, volume 4, pages 3729–3734. IEEE, 2004.

Gerardo Lafferriere, Alan Williams, J Caughman, and JJP Veerman. Decentralized control of vehicle formations. *Systems & control letters*, 54(9):899–910, 2005.

Vangipuram Lakshmikantham, Vladimir M Matrosov, and Seenith Sivasundaram. *Vector Lyapunov functions and stability analysis of nonlinear systems*. Springer, 1991.

Vladimir B Larin. About the inverse problem of optimal control. *Journal of Applied and Computational Mathematics*, 2(2):90–97, 2003.

Kuo-Yun Liang, Sebastian van de Hoef, Håkan Terelius, Valerio Turri, Bart Besselink, Jonas Mårtensson, and Karl H Johansson. Networked control challenges in collaborative road freight transport. *European Journal of Control*, 2016.

Stuart P Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

J. Löfberg. Yalmip: A toolbox for modeling and optimization in matlabs. In *Proceedings of the IEEE CACDS Conference, Taipei*, pages 284–289, 2004.

Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

JM Maestre, D Muñoz De La Peña, and EF Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153–176, 2011.

José M Maestre and Rudy R Negenborn. *Distributed model predictive control made easy*, volume 69. Springer, 2014.

Lalo Magni and Riccardo Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42(7):1231–1236, 2006.

David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

Pamela McCorduck. Machines who think. 2004.

Raman K Mehra and J Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5):637–640, 1971.

Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

Nader Meskin and Khashayar Khorasani. Actuator fault detection and isolation for a network of unmanned vehicles. *Automatic Control, IEEE Transactions on*, 54(4):835–840, 2009a.

Nader Meskin and Khashayar Khorasani. Fault detection and isolation of discrete-time markovian jump linear systems with application to a network of multi-agent systems having imperfect communication channels. *Automatica*, 45(9):2032–2040, 2009b.

Nader Meskin and Khashayar Khorasani. A geometric approach to fault detection and isolation of continuous-time markovian jump linear systems. *IEEE Transactions on Automatic Control*, 55(6):1343–1357, 2010.

Nader Meskin and Khashayar Khorasani. *Fault detection and isolation: Multi-vehicle unmanned systems*. Springer Science & Business Media, 2011.

Nader Meskin, Khashayar Khorasani, and Camille Alain Rabbath. A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances. *Control Systems Technology, IEEE Transactions on*, 18(6):1422–1429, 2010.

Dejan Milutinovic and Pedro Lima. Modeling and optimal centralized control of a large-size robotic population. *Robotics, IEEE Transactions on*, 22(6):1280–1285, 2006.

Miad Moarref and Luis Rodrigues. An optimal control approach to decentralized energy-efficient coverage problems. In *IFAC World Congress, Cape Town, South Africa*, 2014.

Theodore S Motzkin, Howard Raiffa, Gerald L Thompson, and Robert M Thrall. The double description method. 1953.

Reza Olfati Saber Richard M Murray. Consensus protocols for networks of dynamic agents. In *Proceedings of the 2003 American Controls Conference*, 2003.

Richard M Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

Rudy R Negenborn, Bart De Schutter, and J Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, 2008.

Sergey G Nersesov, Parham Ghorbanian, and Amir G Aghdam. Stabilization of sets with application to multi-vehicle coordinated motion. *Automatica*, 46(9):1419–1427, 2010.

Sergey G Nersesov and Wassim M Haddad. On the stability and control of nonlinear dynamical systems via vector lyapunov functions. *Automatic Control, IEEE Transactions on*, 51(2): 203–215, 2006.

Hoai-Nam Nguyen, Per-Olof Gutman, Sorin Olaru, and Morten Hovd. Implicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints. *Automatica*, 49(9):2754 – 2759, 2013. ISSN 0005-1098.

Minh Tri Nguyen and Cristina Stoica Maniu. Voronoi based decentralized coverage problem: from optimal control to model predictive control. In *21st Mediterranean Conference on Control & Automation (MED)*, 2016.

Minh Tri Nguyen, Cristina Stoica Maniu, Sorin Olaru, and Alexandra Grancharova. About formation reconfiguration for multi-agent dynamical systems. In *AI'14*, pages 141–144, 2014a.

Minh Tri Nguyen, Cristina Stoica Maniu, Sorin Olaru, and Alexandra Grancharova. Formation reconfiguration using model predictive control techniques for multi-agent dynamical systems. In *Developments in Model-Based Optimization and Control*, pages 183–205. Springer, 2015a.

Minh Tri Nguyen, Luis Rodrigues, Cristina Stoica Maniu, and Sorin Olaru. Discretized optimal control approach for dynamic multi-agent decentralized coverage. In *IEEE Multiconference on Systems and Control*, 2016a.

Minh Tri Nguyen, Cristina Stoica Maniu, and Sorin Olaru. Control invariant partition for heterogeneous multi-agent dynamical systems. In *ICSTCC*, pages 354–359. IEEE, 2015b.

Minh Tri Nguyen, Cristina Stoica Maniu, and Sorin Olaru. Decentralized constructive collision avoidance for multi-agent dynamical systems. In *Control Conference (ECC), 2016 European*, pages 812–817, 2016b.

Minh Tri Nguyen, Cristina Stoica Maniu, and Sorin Olaru. Decentralized control for dynamic multi-agent system self-deployment. In *submitted to Decision and Control (CDC), 2016 55th IEEE Conference on*, 2016c.

Minh Tri Nguyen, Cristina Stoica Maniu, Sorin Olaru, and Alexandra Grancharova. Fault tolerant predictive control for multi-agent dynamical systems: formation reconfiguration using set-theoretic approach. In *CODIT*, pages 417–422. IEEE, 2014b.

Hassan Noura, Didier Theilliol, Jean-Christophe Ponsart, and Abbas Chamseddine. *Fault-tolerant control systems: Design and practical applications.* Springer Science & Business Media, 2009.

Carlos Ocampo-Martinez, Vicenç Puig, Gabriela Cembrano, and Joseba Quevedo. Application of predictive control strategies to the management of complex networks in the urban water cycle. *IEEE CONTROL SYSTEMS MAGAZINE*, 33(1):15–41, 2013.

Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.

Sorin Olaru, Jose Adrian De Doná, MM Seron, and Florin Stoican. Positive invariant sets for fault tolerant multisensor control schemes. *International Journal of Control*, 83(12):2622–2640, 2010.

Sorin Olaru, Alexandra Grancharova, and Fernando Lobo Pereira. *Developments in Model-Based Optimization and Control: Distributed Control and Industrial Applications*, volume 464. Springer, 2015.

Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *Automatic Control, IEEE Transactions on*, 51(3):401–420, 2006.

Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.

Ron J Patton, Paul M Frank, and Robert N Clarke. *Fault diagnosis in dynamic systems: theory and application.* Prentice-Hall, Inc., 1989.

Manisa Pipattanasomporn, Hassan Feroze, and S Rahman. Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pages 1–8. IEEE, 2009.

Jeroen Ploeg, Nathan van de Wouw, and Henk Nijmeijer. Fault tolerance of cooperative vehicle platoons subject to communication delay. *IFAC-PapersOnLine*, 48(12):352–357, 2015.

I. Prodan, S. Olaru, C. Stoica, and S. Niculescu. Predictive control for tight group formation of multi-agent systems. In *Proc. of IFAC World Congress*, Milan, Italy, 2011.

Ionela Prodan. *Constrained control of dynamical Multi-Agent systems.* PhD thesis, Supelec, 2012.

Ionela Prodan, Florin Stoican, Sorin Olaru, and Silviu-Iulian Niculescu. Enhancements on the hyperplanes arrangements in mixed-integer programming techniques. *Journal of Optimization Theory and Applications*, 154(2):549–572, 2012.

Ionela Prodan, Florin Stoican, Sorin Olaru, and Silviu-Iulian Niculescu. *Mixed-integer representations in control design: Mathematical foundations and applications.* Springer, 2015.

Vicenç Puig. Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies. *International Journal of Applied Mathematics and Computer Science*, 20(4):619–635, 2010.

SV Raković and David Q Mayne. Robust time optimal obstacle avoidance problem for constrained discrete time systems. In *44th IEEE Conference on Decision and Control and European Control Conference*, pages 981–986. IEEE, 2005.

James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design.* Nob Hill Pub., 2009.

Stefano Riverso, Marcello Farina, and Giancarlo Ferrari-Trecate. Plug-and-play decentralized model predictive control for linear systems. *Automatic Control, IEEE Transactions on*, 58 (10):2608–2614, 2013.

Albert Rosich, Holger Voos, and Mohamed Darouach. Cyber-attack detection based on controlled invariant sets. In *ECC*, pages 2176–2181. IEEE, 2014.

Damiano Rotondo, Fatiha Nejjari, and Vicenç Puig. A virtual actuator and sensor approach for fault tolerant control of lpv systems. *Journal of Process Control*, 24(3):203–222, 2014.

AM Rubinov and AA Yagubov. The space of star-shaped sets and its applications in nonsmooth optimization. In *Quasidifferential Calculus*, pages 176–202. Springer, 1986.

Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

Serban Sabau, Cristian Oara, Sean Warnick, and Ali Jadbabaie. Optimal distributed control for platooning via sparse coprime factorizations. *arXiv preprint arXiv:1510.08915*, 2015.

Reza Olfati Saber and Richard M Murray. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 2022–2028. IEEE, 2003.

Edgar N Sanchez and Fernando Ornelas-Tellez. *Discrete-time inverse optimal control for nonlinear systems.* CRC Press, 2013.

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

Rolf Schneider. *Convex bodies: the Brunn–Minkowski theory.* Number 151. Cambridge University Press, 2013.

Alexander Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

Mac Schwager, Jean-Jacques E Slotine, and Daniela Rus. Unifying geometric, probabilistic, and potential field approaches to multi-robot coverage control. In *ISRR*, pages 21–38. Springer, 2009.

Elham Semsar-Kazerooni and Khashayar Khorasani. Multi-agent team cooperation: A game theory approach. *Automatica*, 45(10):2205–2213, 2009.

María M Seron and José A De Doná. Fault tolerant control using virtual actuators and invariant-set based fault detection and identification. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 7801–7806. IEEE, 2009.

Jeff S Shamma. *Cooperative control of distributed multi-agent systems.* Wiley Online Library, 2007.

Shahab Sheikholeslam and Charles A Desoer. Control of interconnected nonlinear dynamical systems: The platoon problem. *IEEE Transactions on Automatic Control*, 37(6):806–810, 1992.

Dragoslav D Siljak. *Decentralized control of complex systems.* Courier Corporation, 2011.

Yuan Song, Bing Wang, Zhijie Shi, Krishna R Pattipati, and Swastik Gupta. Distributed algorithms for energy-efficient even self-deployment in mobile sensor networks. *Mobile Computing, IEEE Transactions on*, 13(5):1035–1047, 2014.

Srdjan Stanković, Nemanja Ilić, Željko Djurović, Miloš Stanković, and Karl Henrik Johansson. Consensus based overlapping decentralized fault detection and isolation. In *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pages 570–575. IEEE, 2010.

Srdjan S Stanković, Milorad J Stanojevic, and Dragoslav D Siljak. Decentralized overlapping control of a platoon of vehicles. *Control Systems Technology, IEEE Transactions on*, 8(5): 816–832, 2000.

DušAn M Stipanović, GöKhan Inalhan, Rodney Teo, and Claire J Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica*, 40(8):1285–1296, 2004.

Florin Stoican. *Fault tolerant control based on set-theoretic methods.* PhD thesis, Supelec, 2011.

Florin Stoican and Sorin Olaru. *Set-theoretic Fault-tolerant Control in Multisensor Systems.* John Wiley & Sons, 2013.

Florin Stoican, Sorin Olaru, and George Bitsoris. Brief paper-controlled invariance-based fault detection for multisensory control systems. *Control Theory & Applications, IET*, 7(4):606–611, 2013.

Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

Kokichi Sugihara. Voronoi diagrams in a river. *International Journal of Computational Geometry & Applications*, 2(01):29–48, 1992.

Herbert G Tanner. Flocking with obstacle avoidance in switching networks of interconnected vehicles. In *ICRA*, volume 3, pages 3006–3011. IEEE, 2004.

Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Flocking in teams of nonholonomic agents. In *Cooperative Control*, pages 229–239. Springer, 2005.

Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Flocking in fixed and switching networks. *Automatic Control, IEEE Transactions on*, 52(5):863–868, 2007.

Didier Theilliol, Hassan Noura, and Jean-Christophe Ponsart. Fault diagnosis and accommodation of a three-tank system based on analytical redundancy. *ISA transactions*, 41(3):365–382, 2002.

Claire Tomlin, George J Pappas, and Shankar Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *Automatic Control, IEEE Transactions on*, 43 (4):509–521, 1998.

PJ Van Overloop, RR Negenborn, B De Schutter, and NC Van De Giesen. Predictive control for national water flow optimization in the netherlands. In *Intelligent infrastructures*, pages 439–461. Springer, 2010.

Venkat Venkatasubramanian, Ravi Vaidyanathan, and Y Yamamoto. Process fault detection and diagnosis using neural networksi. steady-state processes. *Computers & Chemical Engineering*, 14(7):699–712, 1990.

Georges Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

Zhu Wang, Rui Yang, and Lingfeng Wang. Multi-agent control system with intelligent optimization for smart and energy-efficient buildings. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, pages 1144–1149. IEEE, 2010.

Alan S Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601–611, 1976.

Michael Wooldridge. *An introduction to multiagent systems.* John Wiley & Sons, 2009.

Michael Wooldridge, Nicholas R Jennings, et al. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152, 1995.

Qi-Di Wu, Dong Xue, and Jing Yao. Consensus analysis of networked multi-agent systems. *Physics Procedia*, 3(5):1921–1931, 2010.

Jin Xiao-Zheng and Yang Guang-Hong. Robust adaptive fault-tolerant compensation control with actuator failures and bounded disturbances. *Acta Automatica Sinica*, 35(3):305–309, 2009.

Ding-Li Yu, TK Chang, and Ding-Wen Yu. Fault tolerant control of multivariable processes using auto-tuning pid controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(1):32–43, 2005.

Ernst Zermelo. Über das navigationsproblem bei ruhender oder veränderlicher windverteilung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 11(2):114–124, 1931.

Peng Zhao, Siddharth Suryanarayanan, and Marcelo Godoy Simoes. An energy management system for building structures using a multi-agent decision-making control methodology. *Industry Applications, IEEE Transactions on*, 49(1):322–330, 2013.

Dong-Hua Zhou and PM Frank. Fault diagnostics and fault tolerant control. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(2):420–427, 1998.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies
de l'information
et de la communication (STIC)

Titre : Commande prédictive sous contraintes de sécurité pour des systèmes dynamiques Multi-Agents

Mots clefs : Systèmes dynamiques Multi-Agents, commande décentralisée, méthodes ensemblistes, partition de Voronoi, évitement de collision.

Résumé : Cette thèse porte sur des techniques de commande à base d'optimisation dans le cadre des systèmes dynamiques Multi-Agents sous contraintes, plus particulièrement liées à l'évitement des collisions. Dans un contexte ensembliste, l'évitement des collisions au sein de la formation se traduit par des conditions de non intersection des régions de sécurité caractéristiques à chaque agent/obstacle. Grace à sa capacité à gérer les contraintes, la commande prédictive a été choisie parmi les méthodes de synthèse fondées sur des techniques d'optimisation. Tout d'abord, une structure de type leader-suiveur est considérée comme une architecture décentralisée élémentaire. La zone de fonctionnement de chaque suiveur est décidée par le leader et puis une loi de commande locale est calculée afin de garantir que les suiveurs restent à l'intérieur de la zone autorisée, permettant d'éviter les collisions. Ensuite, un déploiement des agents fondé sur l'approche de commande prédictive décentralisée, utilisant des partitions dynamiques de Voronoi, est proposé, permettant de ramener chaque agent vers l'intérieur de sa cellule Voronoi. Une des contributions a été de considérer le centre de Chebyshev comme cible à l'intérieur de chaque cellule. D'autres solutions proposent l'utilisation du centre de masse ou du centre obtenu par l'interpolation des sommets. Finalement, des méthodes ensemblistes sont utilisées pour construire un niveau supplémentaire de détection de défauts dans le cadre du système Multi-Agents. Cela permet l'exclusion des agents défectueux ainsi que l'intégration des agents extérieurs certifiés sans défauts dans la formation en utilisant des techniques de commande prédictive centralisée.

Title : Safe predictive control for Multi-Agent dynamical systems

Keywords : Multi-Agents dynamical systems, decentralized control, set-theoretic methods, Voronoi partition, collision avoidance.

Abstract : This thesis presents optimization-based control techniques for dynamical Multi-Agent systems (MAS) subject to collision avoidance constraints. From the set-theoretic point of view, collision avoidance objective can be translated into non-overlapping conditions for the safety regions characterizing each agent/obstacle while maintaining the convergence towards a specified formation. Among the successful optimization-based control methods, Model Predictive Control (MPC) is used for constraints handling. First, a leader-follower structure is considered as a basic decentralized architecture. The followers functioning zone assignment is decided by the leader and then the local linear feedback control is computed such that the follower operates strictly inside its authorized zone, offering anti-collision guarantees. Second, a dynamic Voronoi partition based deployment of the agents using an inner target driver is developed. The main novelty is to consider the Chebyshev center as the inner target for each agent, leading to an optimization-based decentralized predictive control design. In the same topic, other inner targets are considered such as the center of mass or vertex interpolated center. Third, set-theoretic tools are used to design a centralized FDI layer for dynamical MAS, leading to the exclusion of a faulty agent from the MAS formation and the integration of an external healthy/recovered agent in the current formation. The set-based FDI allows detecting and isolating these faulty agents to protect the current formation using centralized predictive control techniques.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France