UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Automatique et Productique**

Arrêté Ministériel : 7 août 2006

Présentée par

Gabriel RODRIGUES DE CAMPOS

Thèse dirigée par **Carlos CANUDAS DE WIT**
et codirigée par **Alexandre SEURET**

préparée au sein du laboratoire **GIPSA-Lab, Département Automatique**, du centre de recherche **INRIA Grenoble Rhône-Alpes**
et de l'école doctorale **Électronique, Électrotechnique, Automatique et Traitement du Signal**

Agreement strategies for multi-robot systems

Stratégies de commande collaborative pour des systèmes multi-robots

Thèse soutenue publiquement le **23 novembre 2012**,
devant le jury composé de :

Mazen ALAMIR, Président
Directeur de Recherche CNRS, GIPSA-Lab (Grenoble, France)
António PASCOAL, Rapporteur
Professeur, Instituto Superior Técnico (Lisbonne, Portugal)
Christopher EDWARDS, Rapporteur
Professeur, University of Leicester (Leicester, Angleterre)
Sandra HIRCHE, Rapporteur
Professeur, Technische Universität München (Munich, Allemagne)
Dimos DIMAROGONAS, Examinateur
Assistant Professor, Royal Institute of Technology (Stockholm, Suède)
Alexandre SEURET, Codirecteur de thèse
Chargé de Recherche CNRS, GIPSA-Lab (Grenoble, France)

This thesis received the European Label

No one can whistle a symphony.
It takes a whole orchestra to play it.

— H. E. Luccock

To my parents

Acknowledgments

First of all, I would like to thank all the members of my committee for their involvement, questions and feedback: to Prof. Mazen Alamir for accepting to be the committee president; to Prof. Antonio Pascoal, Prof. Christopher Edwards and Prof. Sandra Hirche for kindly accepting to review this document; to Dr. Dimos V. Dimaragonas, not only for his guidance while staying at KTH but also for accepting to join us in Grenoble to participate at the discussion of this work. Their comments, observations, and suggestions greatly helped me to improve this thesis and their commitment was crucial for obtaining the European Label for this work.

I would like to thank my supervisor, Carlos Canudas de Wit, for the opportunity of joining the NeCS team. For his advice, support, and mostly for all the challenging possibilities he offered me during this period, especially throughout the FeedNetBack and Connect projects and the international mobility period I enjoyed. I am also thankful to my co-supervisor, Alexandre Seuret, for his trust and his encouraging guidance. Not only I found his support to all my projects and aspirations but I could develop with him an unforgettable friendship.

I was fortunate enough to undertake some of the work for this thesis in Sweden, where I spent an amazing six months at KTH. My gratitude goes to Prof. Karl H. Johansson and Dr. Dimos V. Dimarogonas, who provide me with their precious help and guidance and offered me the possibility of joining the Automatic Control Lab at KTH. This period was a particularly great experience for me, mainly because of all the wonderful people I had the chance to meet in Sweden. Finally, I also acknowledge Grenoble INP for the financial support that made this research visit possible.

I am very grateful to Janan Zaytoon and Noureddine Manamanni, for their helpful presence during my studies at Reims and for my first insight to academia and research, and to all the people that provide me with comments and suggestions regarding this document, especially to Prof. Linda Simões.

During these three years, I met great people and shared amazing experiences at the Gipsa- Lab. I would like to thank all its members, and in particular those from the Department of Automatic Control. Thank you to my office colleagues and friends Emilie, Valentina, Wenjuan and Dung, for all your daily smiles, jokes and support. Thank you to all PhD students and in particular to Nicolas, Lara B., and Augustin for their friendships, tips and all the cool moments.

Thank you to all the team members I had the pleasure to work with at the GipsADoc, especially to Antoine, Chloe, Emilie, Yo and Vincent. I also acknowledge Prof. Jean-Marc Thiriet and Dr. Nicolas Marchand for their support to student's initiatives in general.

I am especially grateful to the people that shared my life during these three years and to my friends with who I lived so many adventures and experiences: Marco, Lara J. and Elsa for their unconditional friendship, all my "amis grenoblois" and all the friends I had the chance to meet through conferences, meetings and projects.

I could not finish without mentioning my family, especially my brothers, who have always stood by me. Finally, I would like to express my gratitude to my parents for supporting me, my dreams and aspirations. This thesis is dedicated to them.

Lastly, I would like to thank everyone that, in a way or another, shared moments with me and contribute to this accomplishment.

Gabriel Campos
Grenoble, 2012

Abstract

The idea of deploying formations of relatively unsophisticated autonomous robots to accomplish complicated tasks has roots in the early works studying the flocking and foraging behaviors among birds. The main question was how one can mimic different behaviors witnessed in populations of birds, animals, insects, etc. among a population of artificial agents. The emerging use of large-scale multi-agent and multi-vehicle systems in various modern applications has recently raised the need for the design of control laws to perform challenging spatially-distributed tasks such as search and recovery operations, exploration, surveillance, environmental monitoring or pollution detection and estimation, among many others.

This dissertation focuses on distributed control strategies for a set of mobile robots, with a particular attention to agreement protocols. A significant part of the manuscript deals with consensus algorithms of arbitrary linear heterogeneous agents, representing, for example, different models or generations of robots. Motivated by the fact that only a few works consider heterogeneous cases of the synchronization problem, a control strategy is proposed based on a consensus algorithm which is decoupled from the original system. The new algorithm offers the major advantage to separate the stability analysis of each agent and the convergence analysis of the distributed consensus algorithm.

A second aspect of the work focusses on the consensus algorithm's convergence rate. Focusing in memory based approaches, the stabilizing delay principle is used. More precisely, a correctly weighted state sampled component is added to the control law allowing us to artificially manipulate the graph's algebraic connectivity.

Finally, algorithms for the compact deployment of agents are designed and analyzed. This manuscript proposes a completely distributed algorithm allowing swarm self-organization while improving the network's connectivity properties. For these protocols, the desired formation is entirely specified by the angles formed by agents within the formation.

Abrégé

L'idée de déployer des robots autonomes pour accomplir des tâches complexes puise ses racines dans des travaux qui étudient les comportements migratoires et l'organisation des populations d'animaux. La principale question à l'époque était de savoir comment imiter les différents comportements visibles dans les populations d'oiseaux, d'insectes, de poissons, etc., pour les appliquer à un groupe de robots. L'utilisation de plus en plus fréquente des réseaux multi-agents et des systèmes multi-véhicules dans diverses applications modernes révèle souvent la nécessité de lois de commande pour des scénarios compliqués tels que des opérations de sauvetage, d'exploration, de surveillance ou de détection de pollution, parmi beaucoup d'autres.

Cette thèse porte sur des stratégies de contrôle distribué pour un système multi-robots, avec une attention particulière aux protocoles de consensus. Une grande partie du document se focalise sur les algorithmes de consensus pour des agents hétérogènes qui peuvent représenter, par exemple, différents modèles ou générations de robots. Du fait que seuls quelques travaux abordent ce problème, on propose ici une stratégie de contrôle où l'algorithme de consensus est découplé du système original. Le nouvel algorithme offre l'avantage d'une analyse séparée de la stabilité de chaque agent et de celle de l'algorithme de consensus distribué. Un deuxième aspect de ce travail met l'accent sur le taux de convergence des algorithmes de consensus. On présente, en particulier, des protocoles avec mémoire, en utilisant le concept du délai stabilisant.

On s'occupe, finalement, de la définition et de l'analyse des algorithmes pour le déploiement compact d'agents. Des algorithmes totalement distribués sont proposés : ils permettent l'auto organisation d'un groupe de robots, tout en améliorant les propriétés de connectivité du réseau de communication, et ils ont la particularité de définir la formation souhaitée ayant pour base les angles inter-agents.

Contents

List of Figures

List of Acronyms

List of Notations and Definitions

- $\mathbb{R}^n$ is the real vector space of dimension n.
- $\mathbb{R}^{n\times p}$ is the set of real-valued matrices with dimension $n \times p$.
- N is the number of agents.
- q_i represents the state of agent i such that $q_i = [x_i, y_i]^T \in \mathbb{R}^2$.
- x_i represents the position of q_i on the x-axis.
- y_i represent the position of q_i on the y-axis.
- $\mathbf{0}$ denotes a vector or a matrix filled with 0's, of appropriate dimension.
- $\mathbf{1}$ denotes a vector or a matrix filled with 1's, of appropriate dimension.
- $\mathbf{L}$ denotes the Laplacian matrix for a given communication graph.
- I denotes the identity matrix of appropriate dimension.
- M^T designates the transpose of a matrix M.
- λ_i represents the i^{th} eigenvalue of a matrix M.
- $\min\{a; b\}$ gives the minimum between the scalar values a and b.
- $\text{He}\{A\} > 0$ denotes $A + A^T > 0$, for any matrix $A \in \mathbb{R}^{n\times n}$.

Preface

Contents

P.1 Problem statement and contributions

This preface aims to present the problem treated throughout this manuscript and to give a general overview of the content and contributions of this thesis.

Though organisms are inherently competitive, cooperation is widespread. Genes cooperate in genomes; cells cooperate in tissues; individuals cooperate in societies. Animal societies, in which collective action emerges from cooperation among individuals, represent extreme social complexity. Cooperative behavior in large groups of individuals, inherent to these societies, appear abundantly in nature. There exist well known examples of such behaviors such as schools of fish, flocks of birds or collective food-gathering in ant colonies, see Figure 1.2. These behaviors can be explained as swarm intelligence, or swarm theory, *i.e.*, the collective behavior of decentralized, self-organizing systems. The fundamental property of this cooperation is that the group behavior is not dictated by one of the individuals [207]. In swarm intelligence, simple creatures follow simple rules, each one acting on local information. No individual sees the big picture. No individual tells any other one what to do.

Curious and intrigued by how and why groups form and how individual behavioral roles are determined within groups, scientists have been trying to theorize such systems, symbols of a remarkable collective intelligence. The question of interest is how one can mimic different cooperation behaviors witnessed in populations of birds, insects, etc., among a population of artificially constructed stuctures/individuals. By leveraging these kinds of consensus-based systems, groups of independently-acting agents should solve problems more efficiently than they could if they were centrally controlled. Craig Reynolds was one of the first to be interested in this collective intelligence. In 1987, the behaviour of a flock of birds in motion was modeled and simulated in [227]. Reynolds' work, which was able to mimic swarm behavior, led to a frenetic study of self-organizing models, also called Multi-Agent Systems (MAS).

Self-organized swarming behaviors in biological groups with distributed agent-to-agent interactions have become the scientific motivation for studying coordination mechanisms of artificial mobile robots. See [225] for an overview of recent research. Nowadays, autonomous robots are recurrently used to help humans to perform certain tasks with improved performances and in better safety conditions. The deployment of large groups of autonomous vehicles is now possible because of technological advances in networking and in miniaturization of electromechanical systems. Indeed, groups of autonomous robots with computing, communication, and mobility capabilities have become economically feasible and can perform a variety of spatially distributed sensing tasks such as search and recovery operations, manipulation in hazardous environments, exploration, surveillance, environmental monitoring for pollution detection and estimation. etc. Employing teams of robots offers several advantages. For instance, certain tasks are difficult, if not impossible, when performed by a single vehicle. Furthermore, a group of vehicles inherently provides robustness to failures of single agents or communication links.

For several applications, teams of mobile autonomous agents need the ability to deploy over a region, assume a specified pattern, rendezvous at a common point, or move in a synchronized manner. Such abilities ask for motion coordination tasks, *i.e.*, collaborative behavior of a group of mobile agents in order to reach a common aim. Furthermore, coordination tasks must often be achieved with minimal communication between agents and, therefore, with limited information about the system. A recent survey on distributed coordination can be found in [165]. Currently engineering, and control engineers in particular, have to cope with many new problems arising from networked systems when designing complex systems. Indeed, many interesting questions still remain unanswered in the area of multi-agent systems. Something that was widely unclear a few years ago, but better understood today, is how to design local laws which yield a prescribed global behavior. In fact, this is clear in some specific and simple cases, such as, for instance, the distributed averaging problem. But even in this simple example there is still enormous space for improvement, specially in terms of robustness to failures, flexibility, reliability and adaptivity.

This thesis focuses on distributed agreement strategies to control a set of mobile robots. Several technical challenges are addressed in this dissertation such as agreement algorithms, communication constrained control design, connectivity maintenance and pattern control. It is also related to the European project FeedNetBack [1] supported by the European Commission, to the Connect [2] project supported by the Agence Nationale de la Recherche (ANR) and to the Groupement International de Recherche (GDRI) DelSys, supported by the Centre National de la Recherche Scientifique (CNRS).

In order to propose solutions to MAS control problems, the dissertation is partitioned into two main contributions:

Multi-agent systems rendezvous algorithms

A significant part of this manuscript deals with consensus algorithms of arbitrary linear heterogeneous agents, representing, for example, different models or generations of robots. Motivated by the fact that only a few works consider heterogeneous cases of the synchronization problem, we proposed a control strategy based on a consensus algorithm which is decoupled from the original system. In a second set of works, we focus on the consensus algorithm's convergence rate and more particularly, in accelerating it. Using the stabilizing delay principle, we add a state sampled component to the control law that can be seen as an artificial way to manipulate graph's algebraic connectivity.

1. www.feednetback.eu/
2. www.gipsa-lab.inpg.fr/projet/connect/

Multi-agent systems deployment algorithms

The main contribution in this topic is an effective algorithm for compact agent deployment. In our approach, the desired formation is specified entirely by angles formed by the agents within the formation. We proposed a completely distributed algorithm, based on only relative positions that allows swarm self-organization while improving the connectivity properties.

P.2 Dissertation outline

Chapter 1: Introduction

The purpose of this chapter is to contextualize the main topics related to this thesis and to give an exhaustive overview of the dissertation. At first, we present some of the cooperative behavior that motivated researchers to intensely study control strategies for MAS during the last couple of decades. The second part of the introduction is composed of a review of the basic tools and approaches to carry out cooperative tasks that must be achieved by a group of vehicles or sensors. This survey analyzes the applications of multi-agent systems and different collaborative control strategies present in the literature. Finally, we recall the structure and the main challenges and contributions proposed in this manuscript.

Chapter 2: Consensus strategies for heterogeneous multi-agent systems

The first objective of the thesis deals with consensus algorithms for heterogeneous agents, representing, for example, different models or generations of robots. Only a few works consider heterogeneous cases of the synchronization problem and, in particular, necessary and sufficient conditions for output synchronization were recently studied in [299]. In this chapter, we will propose a control strategy based on consensus algorithms which is decoupled from the original system. In other words, we attribute to each agent an additional control variable which achieves a consensus and thus the measurement variable of each agent should converge to this additional variable. The new algorithm offers the major advantage of separating the stability analysis of each agent and the convergence analysis of the distributed consensus algorithm. This conclusion inherently means that it is possible to extend the previous control law to more general situations, where for instance, the communication link induces transmission delays as in [179, 177, 251], or when one considers distributed filters as in [195]. Both cases will be studied in this dissertation.

Chapter 3: Improved stability of consensus algorithms

While in the previous chapter we focused on the design of effective consensus algorithms, in this chapter we will pay special attention to consensus algorithm's convergence rate. The speed of convergence of a consensus algorithm turns out to be equal to the second smallest eigenvalue of $\mathbf{L}$, also called algebraic connectivity. Accelerating the convergence of distributed synchronization algorithms have been studied in literature based on two main approaches: optimizing the topology-respecting weight matrix, summarizing the updates at each node [304], or incorporating memory into the distributed averaging algorithm. This approach will be studied throughout this manuscript. Although for most applications delays lead to a reduction of performances or can even lead to instability, there exist some cases where the introduction of a delay in the control loop can help to stabilize a system. This has been studied in [110] and [252]. For this second approach, adding a state sampled component to the control law can be seen as an artificial way to manipulate $\mathbf{L}$'s eigenvalues, by getting them further into the left part of the complex plan. This inherently means that the speed of convergence will change, and our objective is to maximize this value.

Chapter 4: Distributed control strategies for multi-agent systems compact formations

This third chapter addresses the design and analysis of an algorithm for compact agent deployment. In our approach, the desired formation is specified entirely by angles formed by the agents within the formation. We propose a completely leaderless and distributed algorithm that allows swarm self-organization. The first contribution corresponds to an extension of [76] for swarm dispersion, by adding a connectivity maintenance force. Each agent is equipped with potential functions that will, simultaneously, isolate it from any other agent and impose connectivity-maintenance, using only information of those located within each agent's sensing zone (a circular area around each agent and common for all nodes). We can find in the literature several applications of this type of algorithm including coverage control and optimal placement of a multi-robot team in small areas [63, 90]. On the other hand, and even though bearing-based algorithms were specifically applied to a triangular formation in [9], where the desired formation is specified entirely by the internal triangle angles, the approach that is going to be presented in the sequel consists of a completely distributed algorithm that allows large scale swarm self-organization. In fact, the second contribution consists of direct angle control using only relative positions. To the best of the author's knowledge, the design of a control law capable of establishing a specific formation acting on inter-agent angles has not been addressed so far. Two independent problems will be treated separately: dispersion and compactness. An individual stability analysis for these two strategies will be provided, but we will also propose a sequential control strategy gathering the two components. Theoretical arguments and calculations supporting the argument that such system corresponds to a hybrid system will also be discussed. We assume that no global positioning system such as the Global Position System (GPS) is available and that agents interact locally.

Chapter 5: Conclusion and future works

In the last chapter of the thesis, we make a general conclusion, which summarizes the dissertation contributions and describes ongoing and possible future extensions. Appendix A reviews the fundamentals of sampled systems needed for a complete understanding of the technical developments of this thesis.

P.3 List of publications

Journal articles under preparation

- Gabriel Rodrigues de Campos, Alexandre Seuret, ***Improved stability of consensus algorithms for MAS using appropriated sampling***.
- Gabriel Rodrigues de Campos, Dimos Dimarogonas, Alexandre Seuret, Karl Henrik Johansson ***Distributed control strategy for MAS compact Formations***.

Proceedings of peer-reviewed international conferences

- Gabriel Rodrigues de Campos, Alexandre Seuret, ***Improved consensus algorithms using memory effects***. In Proceedings of the 50^{th} IEEE Conference on Decision and Control and European Control Conference (IEEE CDC/ECC'11), Orlando, USA, 2011
- Gabriel Rodrigues de Campos, Alexandre Seuret, ***Continuous-time double integrator consensus algorithms improved by an appropriate sampling***. In Proceedings of the 2^{nd} IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'10), Annecy, France, 2010
- Gabriel Rodrigues de Campos, Lara Briñón-Arranz, Alexandre Seuret, and Siviu Niculescu, ***On the consensus of heterogeneous multi-agent systems: a decoupling approach***. In Proceedings of the 3^{rd} IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12), Santa Barbara, USA, 2012

Peer-reviewed national conference papers

- Gabriel Rodrigues de Campos, Alexandre Seuret, ***Algorithmes de consensus pour des systèmes double intégrateur continus améliorés par un échantillonnage approprié***. In 4^{mes} Journées Doctorales /Journées Nationales MACS, Marseille, France, 2011

Extended abstracts

- Gabriel Rodrigues de Campos, Iman Shames and Adrian Bishop, ***Distributed labeling in autonomous agent populations***. In 20^{th} International Symposium on Mathematical Theory of Networks and Systems, 2012.

Technical reports

- Alexandre Seuret, Daniel Simon, Emilie Roche, Lara Briñón-Arranz and Gabriel Rodrigues de Campos, ***Multi-agent systems architecture***. D01.03 Building blocks and architectures, Deliverable FeedNetBack project, 26 February 2010

Chapter 1

Introduction

Contents

1.1 Nature, source of inspiration

Many infrastructures and service systems nowadays can naturally be described as networks of a huge number of simple interacting units. Examples come from a large range of domains and include biological systems (genetic regulation, ecosystems), economic networks (production and distribution networks, financial networks), social networks (Facebook, Twitter or scientific networks) and, of course, technological networks (internet, sensor networks, robotics,...). For example, internet service relays on thousands of routers transmitting information all over the world [264], in power networks hundreds of power generators have to synchronize for correct performances [203], and inter-modal transportation systems consist of many trains, cars or airplanes [116]. All these cases ask for distributed decision making, where the process succeeds if all individuals eventually agree on some quantity of interest.

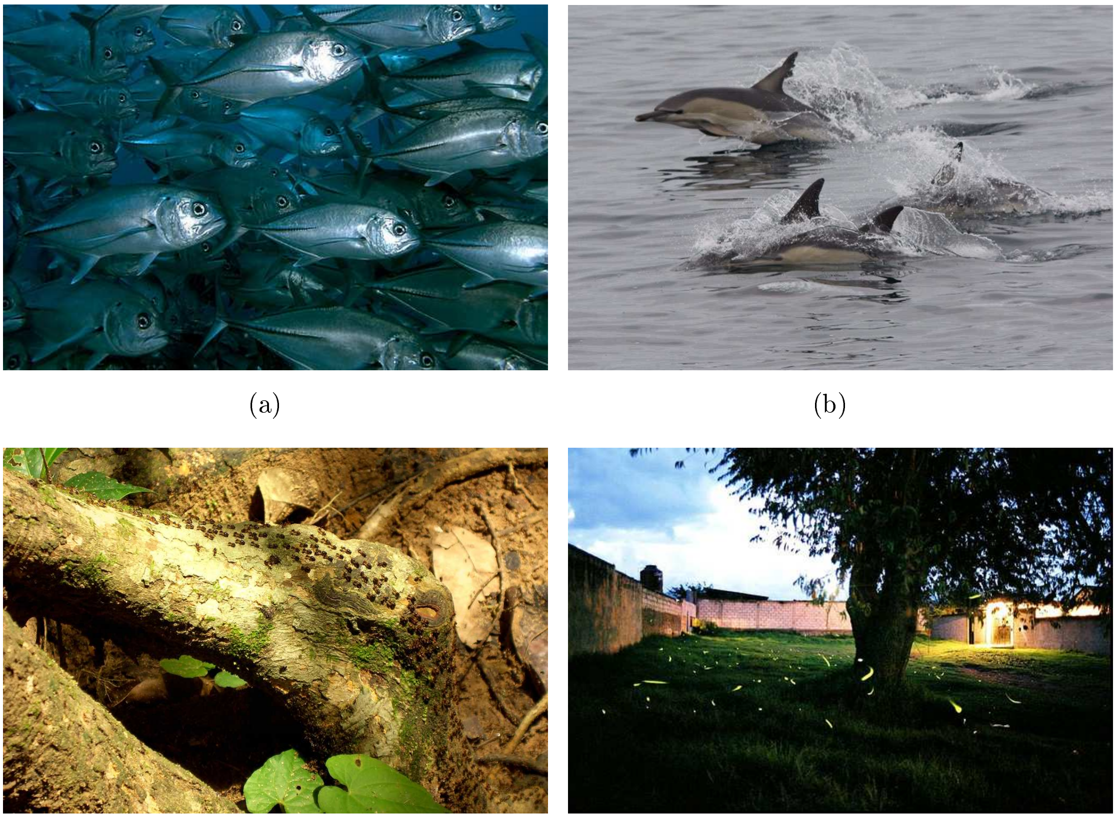
Though organisms are inherently competitive, cooperation is widespread. Genes cooperate in genomes; cells cooperate in tissues; individuals cooperate in societies, see Figure 1.1. Animal societies, in which collective action emerges from cooperation among individuals, represent extreme social complexity.

Figure 1.1: Flock of flamingos flying in formation. Self-organized behaviors emerge in biological groups, even though no individual has global knowledge of the group state. This image is property of António Luís Campos (www.antonioluiscampos.com). Photograph reproduced with the permission of the author.

Such societies are not only common in insects, mammals, and birds, but exist even in simple species like amoebas [268]. During the evolution process that has been taking place for the last thousands of years, individuals wondering whether to join or not a group necessarily weighed the cost-benefit ratio of living solitarily versus with others. When the benefits of living together outweigh the costs of living alone, animals will tend to form groups. Group-living typically provides benefits to individual group members that may include receiving assistance to deal with pathogens, easier mating opportunities, better conservation of heat, and reduced energetic costs of movements. However, living in groups may also confer costs to members such as increased predator attack rate, increased parasite burdens, misdirected parental care, greater reproductive competition or even increased competition for food. Furthermore, individuals may form short-term, unstable groups, *e.g.*, herds of wildebeest, colonies of gulls, or form long-term, stable social groups where interactions among members often appears to be altruistic. For example, when a meerkat or a squirrel sounds an alarm call to warn other group members of a nearby predator, it draws the predator's attention and increases the group's survival odds [55].

Cooperative behaviors in large groups of individuals, inherent to these societies, appear abundantly in nature. There exist well known examples of such behaviors such as schools of fish, flocks of birds or collective food-gathering in ant colonies, see Figure 1.2. These behaviors can be explained as swarm intelligence, or swarm theory, *i.e.*, the collective behavior of decentralized, self-organizing systems. Ants, for example, do not use any kind of centralized management in their colonies. Organization happens organically, through millions of interactions between individual ants who are following very simple behavior rules. In a colony, ants explore the environment randomly, looking for sources of food. While doing so, each ant produce pheromones on the ground allowing it to find its way back to the nest. Whenever an ant locates food, it will carry it back to the nest, following the original path, and inherently marking this route with more pheromones. As other ants are conditioned to most likely follow paths with the highest concentration of pheromones, more and more ants will eventually follow the shortest path between food and nest, as this is the path most ants have already taken, see Figure 1.2(c). Bees choose also their next hive location using a similar, self-organized system. They use scout bees flying out in all directions looking for new hive locations, sharing their interesting findings with others, and finally deciding together the new hive location. Also, fishes stay in large groups to scare predators and, in case of attack, to improve their chances of survival, see Figure 1.2(a).

The fundamental property of this cooperation is that the group behavior is not dictated by one of the individuals [207]. That is how swarm intelligence works: simple creatures following simple rules, each one acting on local information. No individual sees the big picture. No individual tells any other one what to do. The structural idea is that even complex behavior may be coordinated by relatively simple interactions [65, 207].



(a) (b) (c) (d)

Figure 1.2: From left to right, school of fish, dolphins, ants and fireflies. They have been posted to Flickr by Tom Weilenmann, Oldbilluk, Jonathan Pio and Lastbeats, respectively. These images are used under the CC-BY-2.0 licence (www.creativecommons.org/licenses/by/2.0/deed.fr).

Curious and intrigued by how and why groups form and how individual behavioral roles are determined within groups, scientists have been trying to theorize such systems, symbols of a remarkable collective intelligence. Even if scientists started looking at this kind of theory as early as the $40's$, the field exploded in the last twenty years with the rise of computer science, the internet and artificial intelligence. The question of interest is how one can mimic different cooperation behaviors witnessed in populations of birds, insects, etc. among a population of artificially constructed stuctures/individuals. By leveraging these kind of consensus-based systems, groups of independently-acting agents should solve problems more efficiently than they could if they were centrally controlled. The next section will show how these systems are perceived by engineers, and particularly by the control community.

1.2 Engineering perception: towards multi-robot systems

In Nature, the fundamental property of cooperation among several individuals is that the group behavior is not dictated by one of the individuals. On the contrary, the behavior results implicitly from the local interactions between the individuals and their neighbors. For instance, every fish in a school knows where the other fish in its neighborhood are heading, but it does not know the average heading of all fishes. Nonetheless, fishes stay together and move as a group in a certain direction, see Figure 1.2(a). Based on these natural, instinctive and very effective cooperative behaviors, control engineers have been paying a lot of attention to Networked Control Systems (NCS). Roughly speaking, NCS are spatially distributed systems with a communication network used between sensors, actuators, and controllers, which allows flexible architectures with reduced installation and maintenance costs [120, 311]. Indeed, networks provide a powerful metaphor for describing a system's behavior from disciplines as diverse as biology, computer science, physics, social science and, of course, engineering. Due to their numerous advantages, NCS's applications can be found in a wide range of areas such as mobile sensor networks [186], remote surgery, haptic collaboration over Internet, automated highway systems, averaging in communication networks [304], formation control [29, 30, 31, 32, 35, 75, 79, 131, 283] and Multi-Robot Systems (MRS) [29, 125, 167, 194, 286].

Craig Reynolds was one of the first to be interested in this collective intelligence. In 1987, the behaviour of a flock of birds in motion was modeled and simulated in [227]. In this work, generic birdlike objects, or boids, were each given three instructions: **(i)** avoid crowding nearby boids, **(ii)** fly in the average direction of nearby boids, and **(iii)** stay close to nearby boids. The model relied on assuming local motion strategies for each of the boids, *i.e.*, only local information from neighbouring boids was used by each individual. The result: a convincing simulation of flocking, including lifelike and unpredictable movements. Reynolds' work, which was able to mimic swarm behavior, led to a frenetic study of self-organizing models, also called Multi-Agent Systems (MAS), contributing to a better understanding of flocking like behaviors, see [65, 120, 191, 220] and the references therein.

Multi-agent systems were first considered in [288] by Tsitsiklis. This structural work, with a fair mixture of computer science and control/decision problems, considered distributed decision making and computation. On the other hand, it has also identified new challenging problems such as formation control, distributed sensing and optimization and even consensus algorithms. Due to several reasons, collecting measurements from distributed Wireless Sensor Network (WSN) at a single location for on-line data processing may not be feasible. Inherently, there is a growing need for new tools and algorithms that provide high performance in terms of control and estimation with constrained communication. Such algorithms should be robust to node failures and packet losses, reduce the communication load among all sensor nodes or be suitable for distributed control applications. Even though not discussed in this dissertation, several important contributions on MAS are distributed optimal control [214, 234], distributed estimation [1, 2, 194], distributed Kalman filtering [180, 190, 261], distributed computation [20, 27].

Self-organized swarming behaviors in biological groups with distributed agent-to-agent interactions [58, 65, 188] have become the scientific motivation for studying among all problems related with multi-agent system, coordination mechanisms of artificial mobile robots, see [225] for a overview of recent research. In particular, a lot of attention has been given to motion coordination, a remarkable phenomenon in biological systems. In nature, of course, animals travel in impressive large numbers. For these animals, coordinating their movements with one another can be a matter of life or death. Several examples can be identified, from basic molecules interactions to complex high order animal coordinated groups: we can mention how flocks of birds and schools of fish can travel in formation and act as one unit in order to scare predators and, in the case of attack, to improve their chances of survival; how fireflies synchronize their flashing cycles; how animals migrate, showing complex collective behaviors such as obstacle avoiding, leader election, and formation keeping; or even environment partitioning into nonoverlapping zones by individual animals. It is then clear that a deep understanding of such behavior is extremely useful tool for groups of vehicles, mobile sensors, and embedded robotic systems.

A common base to all this works relays on cooperative algorithms. The natural cooperative strategies observed in Nature (see Figure 1.2) might have different form, structure, or scale, but they do aim for the same thing: optimize a task by using all the tools available, *i.e.*, all the individuals. The criteria for the optimization might also be different from one example to another.

In order to explain the benefits of MAS to control engineering, some basic concepts and definitions need to be detailed. The first question that one might ask is "What is an Agent?". Within the scope of this thesis, an answer will be provided in the next section.

1.2.1 What is an agent?

The computer science community has produced various definitions for an agent, as for example in [115, 235, 300]. A comparison between these definitions and their relative merits and weaknesses, from a computer science point of view, can be found in [265]. Even though different, all the definitions referenced above share a basic set of concepts: the notion of an agent, its environment and autonomy. If we consider the environment as all external elements to the agent, that can be physical (*e.g.* the control system) or computing related (*e.g.* data sources, computing resources, and other agents), then, under Wooldridge's definition [300], we call it an agent if it can act autonomously in response to environmental changes. This definition is sufficiently close to those commonly used by the control community. In the scope of this dissertation, we consider an agent under the broad definition provided in [92]. An agent is a physical or virtual entity having several important characteristics:

- **Reactivity capabilities**: An agent is able to act and has a behaviour to satisfy its goals.
- **Autonomy**: An agent is at least partially autonomous.
- **Perception**: It is able to perceive its environment.
- **Local views**: No agent has a full global view of the system, or the system is too complex for an agent to make practical use of such knowledge.
- **Communication capabilities**: An agent is able to communicate with other agents.

Note that all these properties are closely related to those of many animals in Nature, where each one of them autonomously perceive and reacts to its environnement while locally communicating with those around it. Furthermore, note also that Wooldridge also defined an intelligent agent by extending the definition of autonomy to flexible autonomy, see [300] for more details.

But to what exactly corresponds an agent? How do we mathematically represent individuals? From engineers' point of view, an agent is nothing more than a dynamical equation. In fact, this equation states the relation between the input, or in other words the instructions, and the individual's behaviour, *i.e.*, the system's output. In the literature, different models to represent the dynamics of an agent have been used. Consider a multi-agent system formed by $i = 1, \dots, N$ agents. The state of agent i is represented by $q_i \in \mathbb{R}^m$ where m is the dimension of the state, and the control input of the system is denoted by u_i. Two major type of dynamics are available: linear and non linear models.

A) Linear models

i) *General linear models*

A general linear kinematic model for the agents is described by the following equations:

$$\dot{q}_i = Aq_i + Bu_i, \tag{1.2.1a}$$

$$y_i = Cq_i, \tag{1.2.1b}$$

where A, B, C are matrices of appropriated dimensions. This model is used in literature mainly dealt with formation control design such as in [88, 89, 105, 109, 154].

ii) *Single integrator model*

A particular case of linear models is the Simple Integrator (SI), expressed as:

$$\dot{q}_i = u_i. \tag{1.2.2}$$

Different applications for MAS contemplating this simple model include formation control [171], rendezvous [62], cycle pursuit [134, 161], coverage [63, 206, 243], coupled oscillators [52, 139, 140, 175, 267], among many others [111, 127].

iii) *Double integrator model*

Another particular case of linear systems is the Double Integrator (DI) model, often used in the literature of MAS. One of the reasons is that in many cases, several vehicles can be governed by controlling the acceleration of their actuators, for instance the time-derivative of the angular speed of motors. The consensus algorithm for double integrator dynamics is given by:

$$\ddot{q}_i = u_i. \tag{1.2.3}$$

As for before, this model is used in several approaches of cooperative control such as distributed formation control [193, 236], rendezvous [269] and flocking [191].

The previous models offer good properties for stability analysis, but in some situations they are somehow too simple to describe the dynamics of a real agent. In order to consider non-linearities, several authors have used nonlinear approaches for MAS coordination algorithms [51, 211, 238, 244, 245]. Two types of model have been considered: a general nonlinear modeling formulation, and unicycle kinematics as a particular case of the first one. They will be discussed in the sequel.

B) Nonlinear models

i) *General nonlinear models*

A general nonlinear model is described by:

$$\dot{q}_i = f(q_i, u_i), \tag{1.2.4a}$$

$$y_i = h(q_i, u_i), \tag{1.2.4b}$$

where $f(\cdot)$ and $h(\cdot)$ are functions that could satisfy some particular conditions according to the problem considered. Due to its potentialities, several applications assume this model. We can name formation control [15, 16, 85, 197, 213], motion planing [96], extremum-seeking problem [138] and plume tracking [237].

i) *Unicycle model*

A particular nonlinear dynamic model extensively considered in robotics and automatic control is the unicycle model. This non-holonomic model is widely used to represent dynamics of ground vehicles, Autonomous Underwater Vehicles (AUVs) and Unmanned Aerial Vehicles (UAVs). The state of the agent i is denoted by vector $[x_i, y_i\, \theta_i]^T$ where $\theta_i \in S^1$ is its heading angle and $q_i = [x_i, y_i]^T \in \mathbb{R}^2$ its position vector such that x_i and y_i represent the dynamics of q_i on the x-axis and y-axis, respectively. Considering that v_i, u_i are the control inputs, this model is therefore defined by:

$$\dot{x}_i = v_i \cos\theta_i, \quad (1.2.5a)$$

$$\dot{y}_i = v_i \sin\theta_i, \quad (1.2.5b)$$

$$\dot{\theta}_i = u_i, \quad (1.2.5c)$$

Among many others, this model has been considered for formation control [43, 59, 60, 72, 79, 118, 201, 247, 248], rendezvous [74], trajectory tracking [136, 137], motion planing [71], synchronization [200], coverage [141, 159], exploration task [145] and source-seeking problems [56, 57, 172, 186].

1.2.2 Networking

Multi-agent systems' structure relies particularly on the communication network connecting all agents. Indeed, since agents usually cooperate with other agents, they should have some social and communicative abilities. In order to communicate, agents must be able to:

- **Deliver and receive messages:** at this physical level, agents must communicate over agreed physical and network layers to be able to deliver and receive strings or objects that represent messages.
- **Parse the messages:** at the syntactic level, agents must be able to parse messages to correctly decode the message to its parts, such as message content, language, sender, and also must be able to parse the content of the message.
- **Understand the messages:** at the semantic level, the parsed symbols must be understood in the same way, *i.e.*, the ontology describing the symbols must be shared or explicitly expressed and accessible to be able to decode the information contained in the message.

Even if previous concepts define necessary abilities for inter-agent communication, they do not provide any specifications concerning the network. In fact, interactions between agents can be expressed in different ways. We can identify three types:

- **Mechanical networks:** In several structures, the coupling between the different players can be expressed through the interconnection of mechanical devices such as rigid bodies, springs, dampers or transmissions, satisfying both physical and mechanical properties. Moreover, the same concepts are also used in biology to define interactions between proteins or in neuro-science to express neurones' coupling laws.

- **Wired networks:** This type of network relies on a system of cables. There are many areas of application for wired networks. Electric, telephone, cable television or computational networks are some examples. In particular, the Ethernet is a widely used technology to establish a Local Area Network (LAN). More precisely, these structures are defined by a group of computers and associated devices that share a common communication line within a very small geographical area. Furthermore, it is also possible that several devices have to share the resources of a single processor or server.
- **Wireless networks:** Since having a wired connection is impossible in spatially distributed plants or in cars and airplanes, there is a lot of interest in wireless networks. The word wireless is defined in the dictionary as "having no wires". In networking terminology, wireless is the term used to describe any network where there is no physical wired connection between sender and receiver, but rather a communication network based on radio wave transmissions. A common application in every day life is the portable office. People on the road want to use their portable electronic equipment to send and receive telephone calls, faxes, and electronic mail, read remote files, login on remote machines and do this from anywhere on land, sea, or air. Wireless networks are of great value to fleets of trucks, taxis, buses, trains or planes. Another use is for rescue workers at disaster sites where the telephone system has been destroyed.

The introduction of the communication network in multi-agent systems offers some advantages, such as low cost, high reliability, less wiring, easy maintenance, but also various communication constraints including quantization effects [38, 44, 132], presence of noise [157, 318], delayed communication [42, 153, 157, 194, 274, 275, 284, 285, 301, 302, 310, 318], packet losses [86, 113, 209, 276, 306] and so on. More precisely, for first order multi-agent systems, [194] gives an initial study on the consensus problem for continuous-time systems in the frequency domain and provides a necessary and sufficient condition on the upper bound of time-delays under the assumption that all the delays are equal and time-invariant. For multi-agent systems with dynamically changing topologies and time-varying communication delays, [42, 301] prove that multi-agent systems can reach consensus under some connectivity condition, irrespective of time delays. When input delay also exists, a sufficient condition is provided in [284]. In [114, 209, 276], first-order consensus problem is formulated over networks with random packet losses, where it is shown that almost sure consensus is achieved if the expected interaction topology has spanning trees. Considering second-order consensus protocols, [153, 285] investigate the effect of constant input delays on the protocol's convergence properties and provides delay-dependent consensus conditions. Moreover, [318] investigates the consensus of second-order networked multi-agent systems with noise, packet losses and random communication delays. A queuing mechanism is introduced and the inherent packet loss process is assumed to obey the Bernoulli distribution.

In other fields, multi-agent systems are often addressed by computer science [92, 125, 290], distributed computation [20, 27], game theory [28] or social science [68]. In order to model multi-agent systems, control engineers use several tools and definitions from graph theory. The reader should refer to [23, 117, 238] and the references therein for detailed information. Next section presents the necessary tools used throughout the technical developments of this thesis.

1.2.3 Graph theory: concepts and tools

In a simplified way, MAS can be graphically represented by a network of nodes interconnected via communication links, usually modeled by arrows. In the context of this thesis, the existence of an edge necessarily means that the two concerned agents can exchange information. One should note that the graph connecting agents can be either directed or undirected. Therefore, the two following definitions hold.

Definition 1.1. *A* ***direct graph*** *or digraph is defined as a couple $\mathcal{G} = (V, E)$ consisting of a set of N elements called vertices, denoted by $V = \{1, 2, \dots, N\}$ and a set of ordered pair of vertices called edges, represented by $E \subseteq V \times V$. The pair (i, j) denotes an edge from the element i to j.*

Definition 1.2. *An* ***undirected graph*** *consists of a set of vertices V and a set of edges E such that, for all pair of elements $i, j \in V$, $(i, j) \in E$ and $(j, i) \in E$.*

As mentioned before, in MAS the group behavior is not dictated by one of the individuals. On the contrary, the behavior results implicitly from the local interactions between the individuals and their neighbors. Even if previously summarized, the following definition formalizes the neighborhood concept.

Definition 1.3. *The* ***neighborhood*** *of a vertex $i \in V$ is the set:*

$$\mathcal{N}_i = \{j \in V | (i, j) \in E\}.$$

Therefore, all the elements $j \in \mathcal{N}_i$ are called the neighbors of element i. This means that there is an edge from node i to each node j which belongs to the neighborhood. This inherently leads us to the following definition concerning node degree.

Definition 1.4. *The* ***degree*** *of a vertex $i \in V$ is the number of its neighbors, such that*

$$d_i = |\mathcal{N}_i|.$$

The previous definition allows us to define, in a matrix form, how many neighbors each agent has through what is called the degree matrix, defined as follows.

Definition 1.5. *The **degree matrix** of a digraph* $\mathcal{G} = (V, E)$ *is the* $N \times N$ *matrix* $\Delta = (d_{ij})$ *given for all* $i, j \in V$ *by:*

$$d_{ij} = \begin{cases} d_i, & \textit{if} \quad i = j \\ 0 & \textit{otherwise} \end{cases}$$

Until now, we have modeled and defined all concepts related to MAS except how and from who to who information is exchanged. This can be modeled through what is called adjacency matrix defined as follows.

Definition 1.6. *The **adjacency matrix** of a digraph* $\mathcal{G} = (V, E)$ *is the* $N \times N$ *matrix* $\mathcal{A} = (a_{ij})$ *given for all* $i, j \in V$ *by:*

$$a_{ij} = \begin{cases} 1, & \textit{if} \quad (i, j) \in E, \\ 0 & \textit{otherwise.} \end{cases}$$

Both adjacency and degree matrices have been previously defined. We are now ready to introduce what is called in literature the Laplacian matrix $\mathbf{L}$. This new matrix, which gathers both information of adjacency and neighborhood for each agent i, can be expressed as:

$$\mathbf{L} = \Delta - \mathcal{A},$$

and therefore

$$L_{ij} = \begin{cases} d_i, & \text{if} \quad i = j, \\ -1, & \text{if} \quad j \in \mathcal{N}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2.6}$$

where L_{ij} denotes the element on i^{th} line and j^{th} column of L, which is a $N \times N$ size matrix. One should note that Δ is a diagonal matrix, and by definition, each diagonal element of the degree matrix is equal to the sum of elements of its corresponding row in the adjacency matrix $(\Delta)_{ii} = \sum_{j=1}^{N} a_{ij}$. Moreover, all the eigenvalues of $\mathbf{L}$ have nonnegative real parts, such that:

$$0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1},$$

where $\lambda_k(\mathbf{L})$ represents the k^{th} eigenvalue of $\mathbf{L}$. It is worth mentioning that the second smallest eigenvalue of graph's Laplacian, called algebraic connectivity, quantifies the speed of convergence of consensus algorithms. The following definition holds.

Definition 1.7. *The second smallest eigenvalue λ_2 of the Laplacian matrix $\boldsymbol{L}$ is referred to as the* ***algebraic connectivity*** *of the undirected graph $\mathcal{G}$.*

Laplacians and their spectral properties [106, 168] play a crucial role in convergence analysis of consensus and alignment algorithms. In fact, several agreement algorithms can be formalized using the previously defined Laplacian matrix, and consequently, their convergence properties are dependent on the structure and derived properties of $\mathbf{L}$. One of the several interesting properties of $\mathbf{L}$ states that the vector of ones $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^N$ is always an eigenvector of the Laplacian matrix with eigenvalue zero, such that, $\mathbf{L1} = \mathbf{0}$, where $\mathbf{0} = (0, \ldots, 0)^T \in \mathbb{R}^N$ represents the vector of zeros.

Another concept that drastically affects convergence properties of consensus algorithms is graph's connectivity. The notion of connectivity is associated to the idea that the information transmitted by one node in the graph can be received for the rest of the nodes of the communication graph.

Consider a digraph $\mathcal{G} = (V, E)$ whose vertices are denoted by $V = \{i_1, i_2, \ldots, i_N\}$. A vertex of a digraph is globally reachable if it can be reached from any other vertex by traversing a direct path, which is defined as follows.

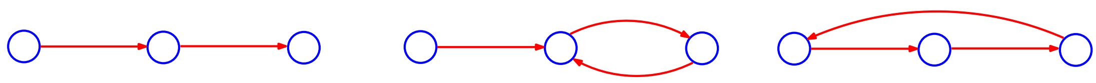
Definition 1.8. *A* ***direct path*** *in a digraph $\mathcal{G} = (V, E)$ is an ordered sequence of vertices $(i_1, i_2)(i_2, i_3) \ldots (i_{m-1}, i_m)$ such that any ordered pair of vertices appearing consecutively in the sequence is an edge of the digraph, i.e., $(i_{p-1}, i_p) \in E$ for all $p = 1, \ldots, m$, where $m \leq N$.*

In literature, there exist a huge number of contributions that rely on the properties of the Laplacian matrix. However, most solutions proposed in these works are graph constrained. In other words, strong assumption are made on the communication graph. In order to discuss, further in this thesis, assumption conservativeness, the following definitions hold.

Definition 1.9. *A digraph $\mathcal{G} = (V, E)$ is* ***strongly connected*** *if every vertex is globally reachable, such that, for all $k \in V$ there is a direct path starting from each other vertex $j \in V, j \neq k$ which finishes in k.*

Definition 1.10. *A digraph is* ***balanced*** *if each vertex $k \in V$ has the same number of incoming and outgoing edges.*

Requiring a strongly connected and balanced graph is, in fact, a very conservative assumption. However, and under fixed interaction topologies, it has been shown in [220] that agreement algorithms such as consensus asymptotically converge if and only if the directed interaction topology has a directed spanning tree. Therefore, the following three definitions are needed.



(a) (b) (c)

Figure 1.3: Three different communication topologies. From left to right, (a) and (b) are not strongly connected graphs, while (c) is (all its vertices are globally reachable). Also, (a) and (b) are not balanced, while (c) is (all vertices have the same number of incoming and outgoing edges).

Definition 1.11. *A graph $\mathcal{G}' = (V', E')$ is a* ***subgraph*** *of $\mathcal{G} = (V, E)$ if its set of vertices and its set of edges are subsets of the corresponding sets of graph $\mathcal{G}$ respectively, such that, $V' \subset V$ and $E' \subset E$. In addition, if $V' = V$ then $\mathcal{G}'$ is a spanning subgraph of $\mathcal{G}$.*

Definition 1.12. *A* ***spanning subgraph*** *is a subgraph in which $V' \equiv V$.*

Definition 1.13. *A* ***direct spanning tree*** *is a spanning subgraph in which there is a vertex, called root, such that any other vertex of the digraph can be reached by one and only one path starting at the root.*

One can easily conclude that requiring a directed spanning tree is less stringent than requiring a strongly connected and balanced graph. Note, though, that the consensus equilibrium is a function only of the initial information states of those vehicles that have a directed path to all other vehicles.

Considering that agents are connected through a network, a key feature of multi-vehicle groups is that communication between moving agents has several dynamic properties. In particular, not all agents may be able to communicate with all others, data rates may be low (either by environment or by design), dropouts may occur, etc. During a coordinated motion or a collaborative task, the interconnections between the agents can evolve such that new communication links are created and others are broken. Therefore, for the sake of both theoretical and practical interest, we will also consider in this thesis time-varying communication topologies. These topologies are represented by a time-varying graph, where the set of edges E and the adjacency matrix depend on time. Time-varying communication topologies are described by a time-varying ρ-digraph $\mathcal{G}(t) = (V, E(t))$, where the elements of its adjacency matrix $\mathcal{A}(t)$ are bounded and satisfy some threshold $\rho > 0$, that is, $a_{ij}(t) = 0$ in the absence of a communication link and $a_{ij}(t) \geq \rho$ in the presence of a communication link.

Definition 1.14. *Consider a time-varying graph $\mathcal{G}(t) = (V, E(t))$ with adjacency matrix $\mathcal{A}$, and let $\bar{\mathcal{G}}(t) = (\bar{V}, \bar{E}(t))$ be the graph in which $\bar{E}(t)$ contains all edges that appear in $\mathcal{G}(\tau)$ for $\tau \in [t, t+T]$ and its adjacency matrix is defined as $\bar{\mathcal{A}} = \int_t^{t+T} \mathcal{A}(\tau)d\tau$. A node i*

is said to be connected to node $j \neq i$ in the interval $[t, t+T]$ if there is a path from vertex i to j, which respects the orientation of the edges for the directed graph $\bar{\mathcal{G}}$. Moreover, $\mathcal{G}(t)$ is said to be ***uniformly connected*** *if there exists an index i and a time horizon $T > 0$ such that, for all t, node i is connected to all the other nodes across $[t, t+T]$.*

In this thesis, we consider both time-invariant and time-varying topologies, depending on the problem to be solved. These graphs are also assumed to be either directed or undirected. Note that in the case of undirected graphs, some of the Laplacian matrix's properties are stronger than for directed graphs. Furthermore, we have also considered cases with both time variant and invariant communication graphs, knowing that agents might be coupled by simple rules including nearest-neighbor or range-based neighborhood [79, 78, 127, 194].

The cooperative control of multi-agent systems poses significant theoretical and practical challenges. In fact, an important notion closely related to MAS is *decentralized control*. Generally, the question of whether to select centralized or decentralized control comes down to resources [325]. In fact, traditional large-scale systems have a centralized or at best a hierarchical architecture, which has the advantage to be relatively easy to be designed and has safety guarantees. However, these systems require very reliable sensors and actuators, which are generally very expensive and do not scale well. Furthermore, if one of the robots is defined as a central unit which is in charge of the data fusion and decision, the entire system can be brittle when the manager does not work. Additionally, communication overhead and response time are limiting factors for centralized control. This is a critical point for the control and analysis of MAS, especially if the multi-agent system is subject to communication limitations.

For successful control strategies, an important issue to be addressed includes the definition and management of shared information among a group of agents to facilitate the coordination of these agents. Regarding motion coordination, several works can be mentioned: pattern formation [3, 18, 29, 111, 131, 152, 171, 248, 260], motion planing [71], flocking [78, 127, 191, 281], self-assembly [135], swarm aggregation [102], gradient climbing [186], coverage and/or deployment [10, 11, 12, 61, 63, 101, 206, 243], rendezvous [3, 62, 150, 154], cyclic pursuit [36, 134, 161], vehicle routing [158, 254], trajectory tracking [136, 137], exploration task [145], source-seeking problems [56, 57, 172] and connectivity maintenance problems [240, 313].

1.2.4 Distributed control strategies

In a logical setup, a team of agents must be able to respond to unanticipated situations and/or to respond to changes in the environment. Therefore, it follows that agents must be in agreement as the environment changes. If we assume that shared information is a necessary condition for coordination, a direct consequence is that cooperation requires that the group of agents reach consensus on the coordination data. To converge to a common value is commonly called a *consensus* or *agreement* problem in the literature [87, 127, 129, 282].

Serving as an introduction to the technical achievements of this thesis, the next section will provide a precise definition of some of these types of algorithms.

Consensus algorithms

We consider a consensus algorithm (or protocol) as an interaction rule that specifies the information exchange between an agent and all of its neighbors over the network in order to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. For example, consensus is a useful algorithm to reach rendezvous, *i.e.*, all the agents converge to the same location [62, 74, 309, 308]. As a special case of MAS, consensus problems have a long history in the field of computer science. In fact, these algorithms have their roots in the analysis of Markov chains and have been deeply studied within the computer science community for load balancing [181, 289] and within the linear algebra community for the asynchronous solution of linear systems [99, 266]. However, they have been recently rediscovered by the control and robotics communities and applied to cooperative coordination of multi-agent systems. In fact, these distributed agreement problems are directly related to mobile multi-robot applications, since they represent an excellent tool to develop more complex cooperative control laws. The reader might refer to [37, 192, 222, 220] and the references therein for thorough information on this topic.

In the second chapter of this thesis, we pay special attention to consensus algorithms. Even though there are different models to represent the dynamics of multi-agent systems used in literature, in this thesis we mainly consider linear systems. Moreover, as particular cases of linear systems, we have intensively studied simple and double integrator dynamics, whose advantages and drawbacks are briefly discussed in the sequel.

Consensus algorithms for linear dynamics

Consider a multi-agent system formed by $i = 1, \ldots, N$ agents. As mentioned before, a general linear model can be described as follows:

$$\dot{q}_i = Aq_i + Bu_i,$$
$$y_i = Cq_i,$$

where q, y and A, B, C have been defined in (1.2.1). Our objective is to design a distributed control law which ensures that the output vectors of each agent reach an agreement. Therefore, the consensus algorithm can be written as:

$$u_i = -K \sum_{j \in \mathcal{N}_i} (y_i - y_j), \tag{1.2.8}$$

where $K > 0$ is a control matrix. Specially applied to formation control, this model was used in [88, 89, 105, 109, 154, 298, 299].

Consensus algorithms for simple integrator dynamics

Another model considered in this thesis is a special case of linear systems. A SI agent can be describe as:

$$\dot{q}_i = u_i.$$

As for before, in order to achieve state agreement between all agents, a commonly used SI consensus algorithm [13, 194, 251] is given by:

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij} \; (q_j - q_i).$$

This algorithm is distributed in the sense that each agent has only access to information from its neighbors, represented by the adjacency a_{ij} weights previously defined in Section 1.2.3. Introducing the vector $x(t) = [x_1, .., x_N]^T$ containing the state of all agents, we then derive:

$$\dot{q} = -\mathbf{L}q \; , \tag{1.2.9}$$

where $\mathbf{L}$ is the Laplacian matrix. This simple model is very popular within MAS analysis and control. The reader can find it applied to different applications: formation control [171], rendezvous [62], cycle pursuit [134, 161], coverage [63, 206, 243] among many others [111, 127].

For the sake of thoroughness of this survey, Kuramoto's oscillators will be briefly discussed here. Recently, a lot of research efforts focus on the mathematical analysis systems composed of interacting phase oscillators. A commonly used model relies on Kuramoto algorithms or Kuramoto oscillators, introduced in [139]. The proposed synchronization algorithm for coupled oscillators is given as:

$$u_i = -\sum_{j \in \mathcal{N}_i} \sin\left(q_j - q_i\right). \tag{1.2.10}$$

Based on this initial work, several extensions [52, 140, 175, 183, 267] led to a deep understanding of the synchronization problem, easily applicable to synchronize and stabilize different patterns in a MAS configuration, as for example in [199].

Consensus algorithms for double integrator dynamics

Finally, a third model considered in this thesis represents a different specific case of linear systems. Recalling equation (1.2.3), consider vehicles with DI dynamics given by:

$$\ddot{q}_i = u_i.$$

One should note that this model fits the behavior of real robotic agents more naturally, since such mechanical systems are controlled in most cases through their acceleration and not their velocity. Moreover, several robotic systems can be reduced to a double integrator via a transformation in their control law. But DI algorithms leads to several problems [217, 224]. For instance, if the graph is directed, the algorithm is not stable, and on the other hand, if the graph is undirected, this requires the knowledge of both position and velocity to converge to the same value. A consensus algorithm for (1.2.3) is studied in [216, 305] as:

$$u_i = -\sum_{j\in\mathcal{N}_i} a_{ij}[(q_j - q_i) + \varsigma\dot{q}_i], \tag{1.2.11}$$

where a_{ij} is the $(i,j)th$ entry of the adjacency matrix and ς a positive control gain representing the absolute damping. Following the same formalism as before, it follows:

$$\ddot{q}(t) = -\varsigma\dot{q} - \mathbf{L}q, \tag{1.2.12}$$

where $\mathbf{L}$ represents the Laplacian matrix. Consensus is reached for (1.2.3) if for all $q_i(0)$ and $\dot{q}_i(0)$, $q_i(t) \to q_j(t)$ and $\dot{q}_i(t) \to 0$ as $t \to \infty$. This model is used in several approaches of cooperative control such as distributed formation control [193, 236, 323], rendezvous [269] and flocking [191]. Note that this model allows the agents to reach an agreement in their velocities. This particular case of consensus problems is called flocking and, by definition, it is not possible to be applied to simple integrator dynamics of the agents.

On the other hand, a consensus algorithm for (1.2.3) is studied in [218] as given by:

$$u_i = -\sum_{j\in\mathcal{N}_i} a_{ij}[(q_j - q_i) + \varrho(\dot{q}_j - \dot{q}_i)], \tag{1.2.13}$$

where ϱ is a positive gain representing the relative damping. Following the same formalism as before, it follows:

$$\ddot{q}(t) = -\varrho \mathbf{L}\dot{q} - \mathbf{L}q \ . \tag{1.2.14}$$

Once again, consensus is reached for (1.2.3) if for all $q_i(0)$ and $\dot{q}_i(0)$, $q_i(t) \to q_j(t)$ and $\dot{q}_i(t) \to \dot{q}_j(t)$ as $t \to \infty$. Note that both communication topology and coupling gain ϱ affect consensus-seeking [220].

The previous discussion provided the essentials on consensus algorithms. Nevertheless, there exist a huge number of further contributions on consensus algorithms considering time-delays [26, 251], switching topologies [39, 111, 194, 219], consensus filters [192, 195], uncertainties [133], finite-time consensus [273], among many other cases [13, 14, 242, 322].

In this thesis we are particulary interested in collaborative behaviors ensuring good connectivity properties. Cohesiveness is characterized by a repulsion/attraction function which makes the agents in the network maintain desired relative distances between its neighbors or achieve collision avoidance [191, 279]. Due to its pertinence to the technical developments of the third chapter, a brief definition of these types of systems is presented in the following.

Inverse Agreement Algorithms

In literature, algorithms whose inverse would lead the multi-agent team to agreement are called dispersion algorithms. The goal is to design control laws that force the agents to converge to sufficiently large distances between them, *i.e.*, disperse in the workspace, or eventually to keep a certain distance between agents. In order to fulfil this purpose, several authors used artificial potential fields [73, 74, 76]. More precisely, each agent is equipped with a repulsive potential with respect to each other agent. Moreover, agents are considered to have limited communication range, a commonly setup widely considered in literature and specially suited to mobile robot applications. For any two agents (i, j), define the vector connecting agent i and j as:

$$q_{ij} = q_j - q_i,$$

and the squared distance between two agents as:

$$\beta_{ij} = \|q_j - q_i\|^2, \forall\ i, j \in \mathcal{N}.$$

The goal is for each agent to follow a flow, whose inverse leads the multi-agent team to agreement, just by taking in consideration the relative positions of agents located within each agent's sensing zone (a circular area of radius d around each agent and common for

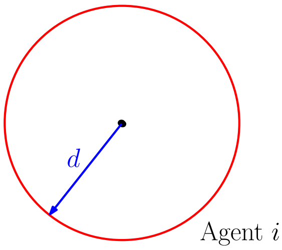


Figure 1.4: Illustration of agent i with range based sensing, where d represents the communication radius.

all nodes), see Figure 1.4. Consequently, the subset of $\mathcal{N}$ that includes the agents that agent i can sense at each time instant is denoted by $\mathcal{N}_i$ and defined by:

$$\mathcal{N}_i(t) = \left\{ j \in \mathcal{N} \backslash \{i\} | \ \ \beta_{ij} < d^2 \right\}.$$

The repulsive function, denoted γ_{ij}, is given by:

$$\gamma_{ij}(\beta_{ij}) = \begin{cases} \frac{1}{2}\beta_{ij}, 0 \leq \beta_{ij} \leq c^2 \\ \phi(\beta_{ij}), c^2 \leq \beta_{ij} \leq d^2 \\ h, d^2 \leq \beta_{ij} \end{cases} \tag{1.2.15}$$

where the positive scalars c, d, h and the function ϕ are chosen in such a way so that γ_{ij} is everywhere twice continuously differentiable. For instance, in [76] authors used a polynomial function of the form:

$$\phi(\beta_{ij}) = a_2 x^2 + a_1 x + a_0,$$

where

$$a_2 = \frac{1}{4(c^2 - d^2)}, \quad a_1 = \frac{d^2}{2(d^2 - c^2)}, \quad a_0 = \frac{c^4}{4(c^2 - d^2)}, \quad h = \frac{d^2 + c^2}{4}.$$

The reader can refer to Figure 1.5 for a graphical representation of γ_{ij}, as provided in [73].

Provided that the parameters of ϕ fulfil the previous relations, the differentiability requirement for γ_{ij} is satisfied. Based on the specific features of γ_{ij}, a controller for agent dispersion was proposed in, *e.g.*, [76]. The control law is defined by:

$$u_i = - \sum_{j \in \mathcal{N}_i} \frac{\partial (1/\gamma_{ij})}{\partial q_i}.$$

It is important to point out that a key feature of these algorithms is that there is no global knowledge imposed on any of the team members.

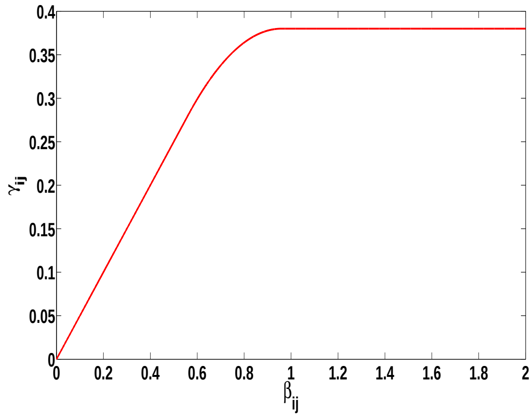


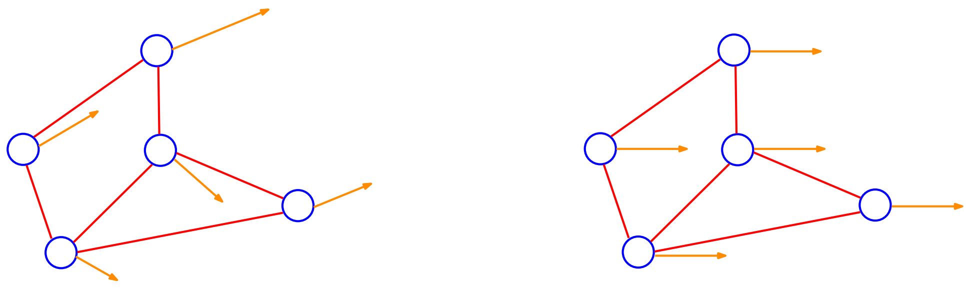
Figure 1.5: Graphical representation of the potential field γ_{ij} for $c^2 = 0.56$ and $d^2 = 0.96$, as presented in [73].

Due to the design characteristics of γ_{ij}, it has been shown in [76] that the closed loop system reaches a configuration in which the minimum distance between any pair of agents is larger than a specific lower bound, which coincides with the agents' sensing radius. It is worthy to mention that both the cases of an unbounded and a cyclic, bounded workspace have also been considered in this work. In the case of a bounded cyclic workspace, the control law was redefined in order to force the agents to remain within the workspace boundary throughout the closed loop system evolution, see [76] for details. Moreover, authors have also shown that the proposed control design guaranteed collision avoidance between the team members in both cases.

Similar approaches via potential fields can be found in literature: in [7, 43], robots move to their goal location while avoiding obstacles, collisions with other robots and remaining in formation; in [81, 215] authors combined the behavior-based approach with potential fields. Possible applications of dispersion algorithms include coverage control [63] and optimal placement of a multi-robot team in small areas [5, 8, 90, 226, 272].

Flocking

A popular definition of such behavior states that a group of mobile agents has to align their velocity vectors, and stabilize their inter-agent distances, using decentralized algorithms and taking into account the communication topology. An illustration is presented in Figure 1.6. As mentioned before, the study of animal behavior during the motion of a flock of birds, a herd of land animals, or a school of fish have been initially studied by [227]. Since then, a lot of effort has been made on this topic. For example, flocking for double integrator dynamics of the agents have been studied in [191, 279, 280].



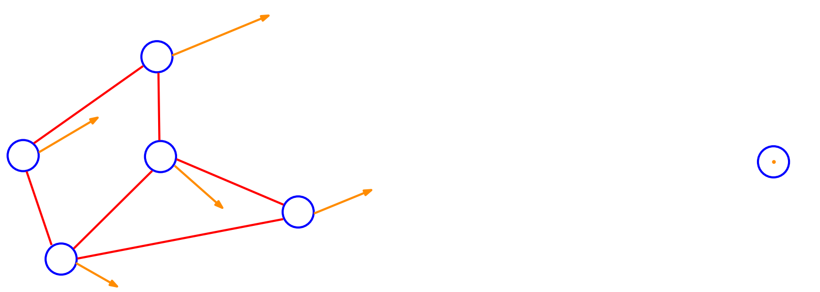
(a) (b)

Figure 1.6: Illustration of flocking behaviors: (a) initial configuration and (b) final configuration, where all speed vectors (yellow arrows) have the same length and direction.

Rendezvous

An important example of vehicle formation control is the rendezvous problem where all agents are required to meet at a common location using only relative position information for all initial conditions [17, 37, 74, 87, 148, 150, 221, 256, 259]. An illustration is presented in Figure 1.7. If one consider motion coordination on a x, y plan, this is in fact an intuitive application of consensus algorithms in two dimensions [127, 194]. The rendezvous problem was firstly introduced in [3]. Several extensions have been proposed in the last few years considering, for example, the synchronous case in [150], the asynchronous case in [149, 151] or *stop-and-go* maneuvers in [149, 148]. Furthermore, several applications typically consider limited communication range, see [62, 74, 87] and the references therein.

The literature on multi-agent systems is extensive and rapidly growing. The previous review can provide the reader a sufficiently thorough state of art on this topic, but it is important to state that several control and model approaches have been excluded from



(a) (b)

Figure 1.7: Illustration of rendezvous behaviors: (a) initial configuration and (b) final configuration, where all agents share a same location with zero velocity (yellow arrows).

this dissertation's discussion.

Because of technological advances in miniaturization of electromechanical systems and in networking, going from constrained wired networks to wireless configurations, the deployment of large groups of autonomous vehicles is now possible. Indeed, groups of autonomous robots with computing, communication, and mobility capabilities have become economically feasible and can perform a variety of spatially distributed sensing tasks such as search and recovery operations, manipulation in hazardous environments, exploration, surveillance, environmental monitoring for pollution detection and estimation, etc. Next section will discuss multi-robot systems, their applications and challenges.

1.2.5 Multi-robot systems, their applications and challenges

The use of groups of autonomous vehicles to perform coordinated and cooperative tasks has been receiving a growing amount of attention during the past decade. In fact, a group of robots can be treated as a multi-agent system in which each vehicle is considered as an agent with the previously mentioned capabilities. A recent review of the vast literature in the field can be found in [120, 192, 220, 221]. In this section, we will motivate our interest and provide applications for the novel strategies to be detailed throughout this manuscript.

Employing teams of robots offers several advantages. For instance, certain tasks are difficult, if not impossible, when performed by a single vehicle. Furthermore, a group of vehicles inherently provides robustness to failures of single agents or communication links. A key feature intensively used today, is the agents' mobility. Even though hundreds of applications rely on static computation chains, sensor or distribution networks, a whole new branch of tasks can be fulfilled assuming on agents' mobility. But an inherent characteristic of such mobile multi-vehicle groups is that the communication between moving vehicles has several dynamic properties. In particular, not all agents may be able to communicate with all others; data rates may be low (either by environment or by design); dropouts may occur, etc.

Since technological limits are being daily overtaken, new applications, technical challenges or control requirements rise on a proportional rhythm. For several applications, teams of mobile autonomous agents need the ability to deploy over a region, assume a specified pattern, rendezvous at a common point, or move in a synchronized manner. Such abilities ask for motion coordination tasks, *i.e.*, collaborative behavior of a group of mobile agents in order to reach a common aim. A particular class of motion coordination for multi-agent systems is studied in [145, 200, 246, 247, 248] under different constraints. These previous works study the problem of design feedback control laws that stabilize a collective motion. Moreover, a recent survey on distributed coordination can be found in [165]. In fact, achieving a coordination task corresponds to moving the agents and changing their state to maximize or minimize an objective function. Furthermore, coordination tasks must often be achieved with minimal communication between agents and, therefore, with limited information about the system.

Another important point concerns the interest/need to use a group of vehicles instead of one complex, expensive and high capability system. A major advantage is that a group of agents is able to realize tasks that can not easily be achieved by a single vehicle. As an example, search/recovey operations, specially in high seas, generally concern an enormous area of terrain to be thoroughly examined. It could take months, even years, to find what so ever if one single agent is deployed. Further, and related with the decentralized characteristics of the control strategy, a group of vehicles inherently provides robustness to failures of single agents or communication links. All previous topics motivate our interest in multi-robot systems, enhancing their numerous advantages. They also provide us insights on control and design of large scaled systems. We are now going to identify several challenging and useful applications for which the work presented in this manuscript is pertinent and can be easily applied to.

This thesis is related to the European project FeedNetBack [1] supported by the European Commission and the French project Connect [2] supported by the Agence Nationale de la Recherche (ANR). Both projects deal with networked control systems, with a special attention on multi-agent systems and their challenges. Moreover, some contributions of this manuscript also concern the GDRI DelSys, supported by the Centre National de la Recherche Scientifique (CNRS). In particular, the European FeedNetBack involves not only several academic partners, but also industrial participants in order to carry out the technological applications. One of the academic participants is the INRIA, and specifically the Networked Control System Team (NeCS) [3] with which this thesis was carried out. More precisely, NeCS is a joint team-projet between INRIA and Grenoble Images Parole Signal Automatique-Laboratoire (GIPSA-lab) whose goal is to develop a new control framework for assessing problems raised by the consideration of new technological low-cost and wireless components, the increase of systems complexity and the distributed and dynamic location of sensors and actuators. The main objectives of FeedNetBack are to design and analyse systems composed of several sub-systems, interconnected by a constrained communication network. More precisely, this project aims to preserve closed-loop system stability while taking into account shared computational resources, distributed sensing and/or constraints on the network topology. One of the study cases considered in this project involves formation control strategies of heterogeneous marine vehicles (surface and underwater vehicles such as autonomous crafts, AUVs or underwater gliders) in order to achieve a scientific mission, see [29, 196] for further details. The main objective of such formation is to carry out a gradient search and fol-

1. www.feednetback.eu
2. www.gipsa-lab.inpg.fr/projet/connect
3. http://necs.inrialpes.fr

lowing an underwater source, the nature of the source might be very different, see, *e.g.*, [33, 34]. To perform missions involving several vehicles, coordinated motion is required, especially when the goal of the mission is sensor driven. However, the challenge is even bigger when non-identical robots are taken into consideration and when simple modeling details can drastically influence the efficiency of the chosen control strategy.

Autonomous robots are being recurrently used to help humans to perform certain tasks with improved performances and in better safety conditions. Over the last decade, unfortunate aerial or even submarine accidents necessitated the use of high-teck technology either to search for wreckage, to recover survivors or to protect sensitive technology. This brings us to another important source of applications and inherent technological challenges: military operations.

Unmanned vehicles, whether air, land or sea, are one means to get airmen, soldiers, marines and sailors out of harm's way and are most likely a key driver to an upcoming revolution in military affairs. Indeed, new technology may be able to help answer the cries to reduce casualties resulting from friendly fire and collateral damage, as well as assist the military in performing urban operations. Military robots find major applications in surveillance, reconnaissance, location and destruction of mines, as well as for hostile military operations, see Figure 1.8. Moreover, recent military reports predict that robots might someday be sent into a hostile village to locate prisoners, into an earthquake-damaged building to find victims or onto chemical-spill sites to examine hazardous waste. Moreover, they might also be able to provide, in a near future, technology for protecting borders and safety in cities or and building smarter cities. In fact, some of the most advanced army robots carry dozens of sensors, including high-resolution night-vision cameras, 3-D imagers, and acoustic arrays. However, humans are still needed to operate these vehicles, interpret the data and coordinate tasks among multiple systems [297]. Because even if one day we are going to see fully autonomous robots capable of planning and carrying out missions and learning from their experiences, a basic question will remain: How can we equip these robots to make critical decisions on their own?

Without being able to answer all the previously mentioned problems, this thesis proposes solutions to some problems related to motion coordination among multiple individuals. We are motivated by recognition, recovery and search operations, such as the Feednetback study case or the civil/military missions described before. This manuscript is focused on distributed agreement strategies to cooperatively command mobile robots, and more precisely, to perform motion coordination based on local couplings [207]. The next section will present the objectives of this thesis.

(a) (b) (c)

Figure 1.8: Military applications. From left to right, RAF Reaper MQ-9 Remotely Piloted Air System, an aircraft fleet attack formation and Wheel Barrow Mk 8 Counter IED Robot. These images are used under the CC-BY-2.0 licence (http://creativecommons.org/licenses/by/2.0/deed.fr). The central image has been posted to Flickr by Archangel 12, while the remaining two were posted by the UK Ministry of Defence.

1.3 General objectives

Currently engineering, and control engineers in particular, has to cope with many new problems arising from networked systems when designing complex systems. Indeed, many interesting questions still remain unanswered in the area of multi-agent systems [312]. Something that was widely unclear a few years ago, but better understood today, is how to design local laws which yield a prescribed global behavior. In fact, this is solved in some specific and simple cases, such as, for instance, in the distributed averaging problem. But even in this simple example there is still enormous space for improvement, especially in terms of robustness to failure, flexibility, reliability and adaptivity. One of the objectives of this thesis pertains to identify optimal behavior criteria for multi-agent systems, which is strictly related with the possibilities and open

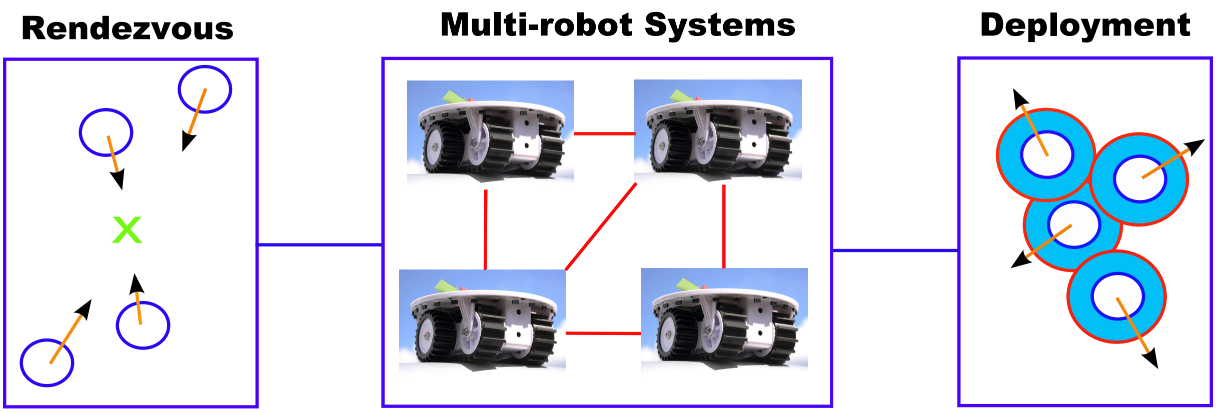


Figure 1.9: Contribution of the thesis

challenges of multi-agents systems that we have identified. Eventually, this study led to an adequate architecture to deal with the distributed characteristics of the system and its inherent communication constraints. A thorough state of art report, describing the existing approaches and most used tools is provided in this thesis. Finally, the ultimate objective of this thesis is to propose original/improved agreement algorithms for MAS. These algorithms should be decentralized, which offers several advantages in comparison to a centralized approach. They should also take into consideration communication limitations and model constraints. The proposed agreement strategies can be organized into two parts, illustrated in Figure 1.9 and explained as follows:

Chapter 2 and 3: MAS rendezvous

In this part, we want to study effective algorithms that are able to gather all agents at a same position, see Figure 1.10 for illustrations. In particular, we will consider consensus algorithms and two problems will be tackled in this dissertation:

i) How to design consensus algorithms for heterogeneous agents;

ii) How to improve traditional convergence properties.

Furthermore, one main constraint is considered: heterogeneous linear systems, which are particulary interesting for complex operations involving a large number of players, as the previously mentioned scientific search maneuvers or the complex military setups and fixed communication topologies.

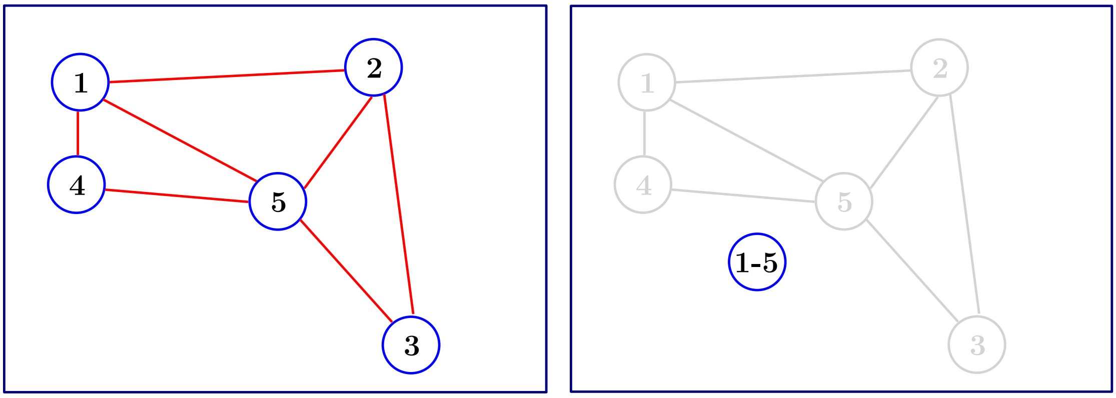
Chapter 4: MAS deployment

For several applications, teams of mobile autonomous agents need not only to agree on some quantity of interest, but also the ability to deploy over a region, assume a specified pattern, or move in a synchronized manner. This second part pays special attention to formation control, since to accomplish exploration, surveillance or rescue tasks agents need to coordinate in order to form a particular configuration that satisfies certain local and/or global constraints, *e.g.*, node degree and/or network connectivity. Three problems will be tackled:

i) How to improve the coverage rate for a given workspace;

ii) How to guarantee that two agents remain connected;

iii) How to improve connectivity properties for a given initial network.

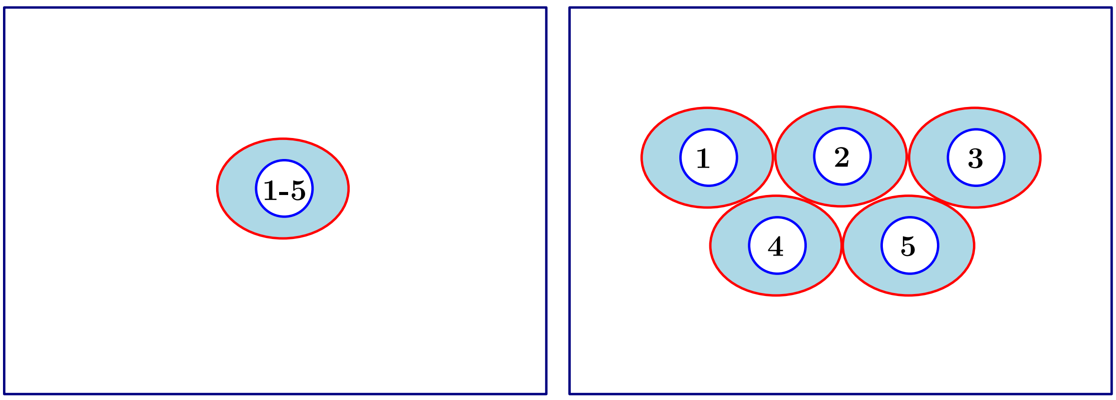


(a) (b)

Figure 1.10: Illustration of rendezvous algorithms for five agents (in blue) under a fixed communication graph (in red): 1.10(a) represents the initial configuration, while 1.10(b) shows the desired final configuration where all agents are at the same position.

In other words, we will try to provide a solution for agent deployment in order to achieve the best coverage rate possible, while keeping or improving connectivity properties, see Figure 1.11 for illustrations. This work takes two main constraints into consideration: homogeneous linear systems, for the sake of simplicity, and time-variant communication graph assuming that each agent has a limited sensing radius (and therefore the graph is dependent on the agents' motion).

In this section we present the objectives of this thesis and the different approaches therein. The main contributions of this manuscript are summarized in the sequel.



(a) (b)

Figure 1.11: Illustration of deployment algorithms for five agents (in blue) with range based communication (in red): 1.11(a) represents the initial configuration, while 1.11(b) shows the desired final configuration where all agents separated within the workspace.

1.4 Contributions of the thesis

The deployment of large groups of autonomous vehicles is now possible because of technological advances in networking and in miniaturization of electromechanical systems. Indeed, groups of autonomous robots with computing, communication, and mobility capabilities have become economically feasible and can perform a variety of spatially distributed sensing tasks such as search and recovery operations, manipulation in hazardous environments, exploration, surveillance, environmental monitoring for pollution detection and estimation. etc. This is one of the reasons why MAS are so popular within the automatic control community and therefore why this thesis focuses on MAS and their applications.

The main technical challenges addressed in this dissertation are:

- Agreement algorithms
- Communication constrained control design
- Connectivity maintenance
- Pattern control

Due to the distributed characteristics of multi-robots systems, the communication graph plays an important role in the efficiency of control strategies. In this thesis, we are particulary interested in studying this topic. More precisely, this manuscript presents pertinent results regarding the motion of a group of vehicles while keeping the best connectivity properties possible. We consider MAS with continuous-time linear dynamics and both direct and undirect graphs' topologies. Moreover, for specific cases, range based communication topologies are considered. The main scientific contributions presented in this manuscript can be organized as follows.

1.4.1 Chapter 2: MAS rendezvous

Consensus algorithms for heterogenous MAS A significant part of this dissertation is focused on consensus algorithms for heterogeneous agents, representing, for example, different models or generations of robots. Only a few works consider heterogeneous cases of the synchronization problem. We will propose here a control strategy based on consensus algorithms which is decoupled from the original system. In other words, we attribute to each agent an additional control variable which achieves a consensus and thus, the measurement variable of each agent should converge to this additional variable. The new algorithm offers the major advantage to separate the stability analysis of each agent and the convergence analysis of the distributed consensus algorithm.

Improved stability of consensus algorithms for MAS In a second part, we focus on consensus algorithm convergence rate. In several multi-agent control problems, the convergence properties and speed of the system depend on the algebraic connectivity of the graph. The algebraic connectivity is an important network property for all the previous systems to reach convergence and it characterizes the convergence rate. More precisely, the algebraic connectivity is equal to the second smallest eigenvalue of **L**. Accelerating the convergence of distributed agreement algorithms have been studied in the literature based on two main approaches: optimizing the weights of the topology-related matrix summarizing the updates at each node [304], or incorporating memory into the distributed averaging algorithm. This last approach will be studied throughout this manuscript. Even if for most applications delays lead to a reduction of performances or even to instability, there exist some cases where the introduction of a delay in the control loop can help to stabilize a system. This has been studied in [110, 252]. In [252], adding a state sampled component to the control law can be seen as an artificial way to manipulate the eigenvalues of **L**, by getting them further onto the left part of the complex plan and inherently changing the speed of convergence. In this thesis, the objective is to maximize the convergence rate value.

1.4.2 Chapter 3: MAS deployment

The third chapter addresses the design and analysis of an algorithm for compact agent deployment. In our approach, the desired formation is specified entirely by angles formed by agents within the formation. We propose a completely distributed and leaderless algorithm, only based on relative positions that allows swarm self-organization. The first contribution corresponds to an extension of [76] for swarms dispersion, by adding a connectivity maintenance force. The second contribution consists of direct angle control using only relative positions, which, at the best of our knowledge, has not been proposed in literature. This is motivated by the fact that for successful network operations, the deployment should result in configurations that not only provide good environment coverage but also satisfy certain local (*e.g.* node degree) and/or global (*e.g.* network connectivity) constraints. Thus, we intend to minimize inter-agent angles in order to achieve the most compact configuration possible. Note that the minimization of inter-agent angles can also be seen as a maximization or maintenance of node degree. Finally, we also propose a sequential control strategy gathering the two components, showing that such framework corresponds to a hybrid system.

The ultimate objective of this thesis is to provide pertinent solutions to some of the open problems of multi-robot systems. We are now ready to present our results.

Chapter 2

Consensus strategies for heterogeneous multi-agent systems

Contents

2.1 Context

In order to face the challenges mentioned in Chapter 1, this thesis is structured in two main parts, see Figure 2.1. A first part, composed of the next two chapters, focuses on consensus algorithms as one of the useful tools available for Multi-Agent Systems (MAS) agreement. In particular, Chapter 2 pertains to design consensus algorithms for heterogenous multi-agent systems. Since consensus is particulary adapted to rendezvous protocols, *i.e.*, to drive all agents to the same location [62, 74, 308], this specific application will be considered to validate our theoretical results.

We consider a consensus algorithm (or protocol) as an interaction rule that specifies the information exchange between an agent and all of its neighbors over the network in order to reach an agreement regarding a certain quantity of interest that depends on the state of all agents, see, *e.g.*, [220]. In a consensus algorithm, the communication topology can be expressed through a matrix called Laplacian, usually denoted $\mathbf{L}$, see [23, 117, 238] or previous chapter for further details. An important characteristic of such a matrix is that its eigenvalues define the system behavior [106, 168]. One of these structural characteristics relies on the fact that the column vector $\mathbf{1}$ of ones is an eigenvector associated with the zero eigenvalue, which implies that span $\{\mathbf{1}\}$ is contained in the kernel of $\mathbf{L}$. It follows that if zero is a simple eigenvalue of $\mathbf{L}$, then the state of all agents will converge to the same value, *i.e.* , achieve consensus. Therefore, convergence analysis have often focused on conditions to ensure that zero is a simple eigenvalue of $\mathbf{L}$, since otherwise the kernel of $\mathbf{L}$ includes elements that are not in span$\{\mathbf{1}\}$, in which case consensus is not guaranteed. It is well-known that zero is a simple eigenvalue of $\mathbf{L}$ if the

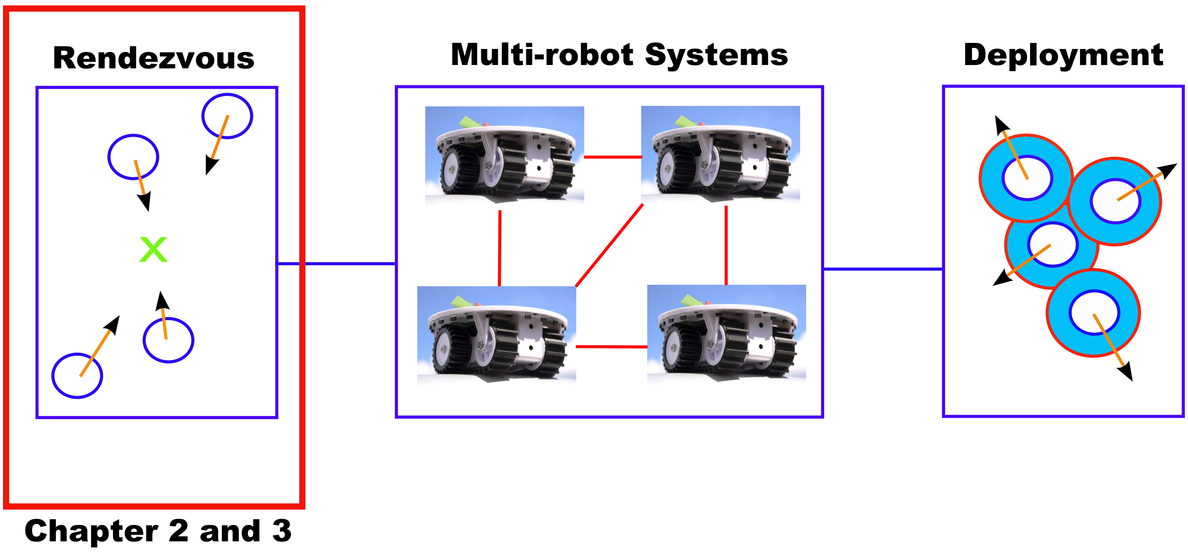


Figure 2.1: Context of Chapters 2 and 3

graph of $\mathbf{L}$ is strongly connected [54]. However, this is only a sufficient condition rather than a necessary one. Based on an induction approach, the formal statement that zero is a simple eigenvalue of the Laplacian matrix if and only if its digraph has a spanning tree has been shown in [223]. Furthermore, the same result is proven independently in [155] by a constructive approach. Consequently, it follows that under a time invariant information exchange topology, the continuous time protocol asymptotically achieves consensus if and only if the information exchange topology has a spanning tree. In a parallel way, but out of the scope of this manuscript, several works consider discrete time consensus algorithms [127, 174, 219, 324]. In fact, the well-known Perron-Frobenius theorem states that one is a simple eigenvalue of a stochastic matrix (used in an equivalent way as the Laplacian matrix) if the corresponding graph is strongly connected which, as for the continuous time case, is only a sufficient condition rather than a necessary one. Consequently, the authors of [219] prove that for a stochastic matrix, one is a unique eigenvalue of modulus one if and only if its digraph has a spanning tree.

There exist a huge number of contributions to this problem including: communication delays in [46, 69, 173, 194, 304], nonlinear consensus protocols in [14, 194, 258], stochastic algorithms in [113] or tracking of linear consensus on time-varying inputs in [262], among many others [40, 192, 195, 219, 251]. Furthermore, although consensus problems are significantly simplified by assuming a time-invariant information exchange topology, the information exchange topology between agents may change dynamically in reality. For instance, communication links between agents may be unreliable due to disturbances and/or subject to communication range limitations [31, 71, 78, 87, 240]. If information is being exchanged by direct sensing, the locally visible neighbors of an agent will likely change over time. Based on a particular model introduced in [294], which is a special case of the distributed behavioral model proposed by Reynold's in [227], many researchers worked on coordination of multiple autonomous agents under switching information exchange topologies, see [46, 128, 127, 154, 219, 239, 319] or [258] using nonlinear contraction theory.

Consensus algorithms are extensively studied in the literature for identical multi-agent systems [154, 173, 194, 241, 248] and in particular to simple and double integrator dynamics, see, for instance, [194, 220] and the references therein. However, increasing interest has turned to MAS with general linear time-invariant dynamical agents, see, *e.g.*, [89, 242, 298].

In this chapter, we are interested in consensus algorithms for heterogeneous multi-agent systems, *i.e.*, with non-identical dynamics representing, for example, different models or generations of robots. For a system consisting of heterogeneous dynamical agents, the first question to be answered is whether there exist a consensus solution to such a system. Only a few papers consider heterogeneous cases of the synchronization problem. In particular, [51, 211] solved the output synchronization problem under a non

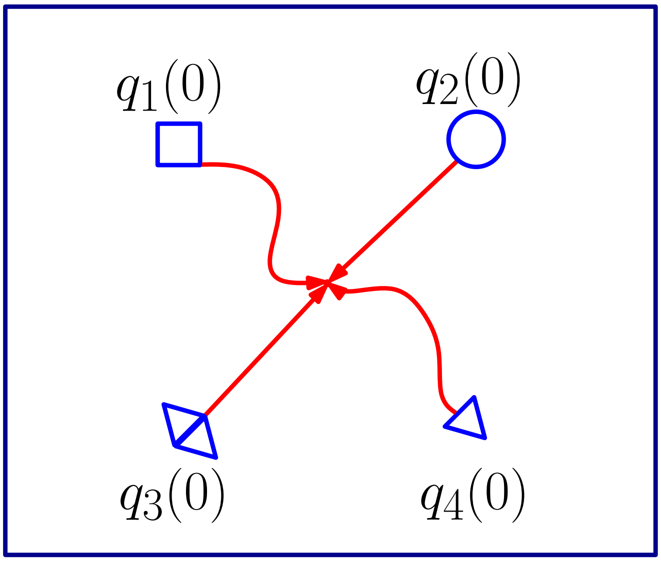


Figure 2.2: Illustration of heterogeneous agents rendezvous

linear approach. Recent results restrict their attention to heterogeneous linear dynamical systems [133, 299, 320, 322]. The consensus of heterogeneous linear agents applied to a formation control problem is presented in [130]. Consensus for heterogeneous multi-agent systems composed of simple and double integrators is presented in [320, 322]. In particular, the consensus problem of heterogeneous multi-agent system composed of second-order agents that cannot obtain the velocity measurements for feedback is studied in [320]. In this paper, authors get consensus criteria by using graph theory, the Lyapunov direct method and LaSalle's invariance principle. An extension to finite-time consensus algorithms with and without velocity measurements is proposed in [321]. Moreover, the authors of [299] focus on linear output synchronization of heterogeneous agents using an internal model approach. The same problem is analyzed in [133] taking into account uncertainties of the agents' models.

This chapter proposes a solution for heterogeneous MAS consensus. In particular, such approches are adapted to, for example, rendezvous applications, see Figure 2.2 for an illustration. In particular, we propose a control strategy based on a consensus algorithm which is decoupled from the original systems. In other words, we attribute to each agent an additional control variable which achieves a consensus and thus, the measurement variable of the each agent should converge with this additional variable. The new algorithm offers the major advantage to separate the stability analysis of each agent and the convergence analysis of the distributed consensus algorithm. To the best of the authors' knowledge, such an approach is new and has not been reported in literature.

2.2 Problem statement and preliminaries

Assuming systems composed of heterogenous agents representing, for example, different models or generations of robots, offer major advantages specially in terms of scalability and/or modularity. Moreover, such a setup naturally fits into several challenging applications relying, for instance, on surface and underwater marine vehicles or on ground and aerial robots. Consequently, this work is motivated by recognition, recovery and search operations on a civil or military framework.

Consider a graph $\mathcal{G}$ with N agents and an edge set given by $E = \{(i,j) : j \in \mathcal{N}_i\}$. Based on Chapter 1's concepts, the *adjacency matrix* $\mathcal{A} = \mathcal{A}(G) = (a_{ij})$ is a $N \times N$ matrix given by $a_{ij} = 1$, if $(i,j) \in E$ and $a_{ij} = 0$, otherwise. If there is an edge connecting two vertices i, j, *i.e.*,$(i,j) \in E$, then i, j are called *adjacent*. The *degree* d_i of vertex i is defined as the number of its neighboring vertices, *i.e.*, $d_i = \#j : (i,j) \in E$. Denote also $d_{max} = max\{d_i\}$. Let Δ be the $N \times N$ diagonal matrix of d_i's. Furthermore, the Laplacian of $\mathcal{G}$ is the matrix $L = \Delta - \mathcal{A}$.

In the scope of multi-robots systems, we are particulary motivated by motion control in a cartesian plane. Consequently, the position of each agent i is denoted:

$$q_i = [x_i, y_i]^T \in \mathbb{R}^2,$$

where x_i and y_i represent the dynamics of q_i on the x-axis and y-axis, respectively. However, for the sake of notation's clearness and without loss of generality, in this chapter we consider only the dynamics of x_i. Thus, consider the following multi-agent linear system:

$$\begin{cases} \dot{x}_i = \bar{A}_i x_i + B_i u_i \\ y_i = x_i \\ z_i = C_i x_i \end{cases} \quad \forall i \in \mathcal{N} = \{1, \ldots, N\}, \tag{2.2.1}$$

where $x_i \in \mathbb{R}^{n_i}$, $y_i \in \mathbb{R}^{n_i}$, $z_i \in \mathbb{R}^m$ and $u_i \in \mathbb{R}^m$ are the state, output, measurement and input vectors, respectively. Note that it is assumed that the state of the system is available for the design of the controller, *i.e.*, $y_i = x_i$ for all $i \in \mathcal{N}$. For all $i \in \mathcal{N}$, the matrices $\bar{A}_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m}$ and $C_i \in \mathbb{R}^{m \times n_i}$, with $m < min\{n_i\}$ and $n_i > m$ are assumed to be constant and known.

In this context, the objective is the design of a distributed control law ensuring that **(i)** each subsystem is stable and **(ii)** the measurement vectors of each agent reach an agreement. To deal with these problems, the following assumptions on the systems are considered:

Assumption 2.1. ***(Heterogeneity):*** *The N systems are assumed to be heterogeneous.*

In other words, this simply means that the matrices defining the systems may differ from one agent to another one and that the vectors x_i may have different dimensions.

Assumption 2.2. ***(Homogeneity of the measurement vector):*** *The measurement vectors z_i represents the same quantity of interests for all agents. As a consequence, the measurement vectors have the same dimension, or in other words, $z_i \in \mathbb{R}^m \; \forall i \in \mathcal{N}$, where $m < min\{n_i\}$.*

Assumption 2.3. ***(Structures of the systems):*** *For all $i \in \mathcal{N}$, the condition $rank(C_iB_i) = m$ holds.*

Technically speaking, it means that the input vectors directly affect the measurement vector.

Assumption 2.4. ***(Controllability):*** *For each agent, the pair (A_i, B_i) is controllable.*

In order to avoid a centralized solution of the problem, the agents are assumed to be connected through a communication network. To establish further stability results, the following assumption holds.

Assumption 2.5. ***(Graph's connectivity):*** *For any considered graph G, we assume that the communication graph has a directed spanning tree.*

This ensures that zero is a simple eigenvalue of $\mathbf{L}$ and the corresponding eigenvector is the vector of ones, $\overrightarrow{\mathbf{1}}$. Consequently, it implies that the algorithm will eventually reach consensus. Note that requiring a directed spanning tree is considerably less stringent than requiring a strongly connected and balanced graph [220].

The objective is to design an effective agreement strategy for heterogenous multi-agent systems. We propose here a control strategy based on trivial consensus algorithms which is decoupled from the original systems. In other words, we attribute to each agent an additional control variable which achieves a consensus such that the measurement variable of each agent should converge with this additional variable. The exact control objectives and needed assumptions have just been introduced. Next section presents the control design for arbitrary heterogeneous agents.

2.3 Controllers design

In order to achieve the goals mentioned above, a controller composed of two parts, one corresponding to the local controller and one representing the consensus algorithm,

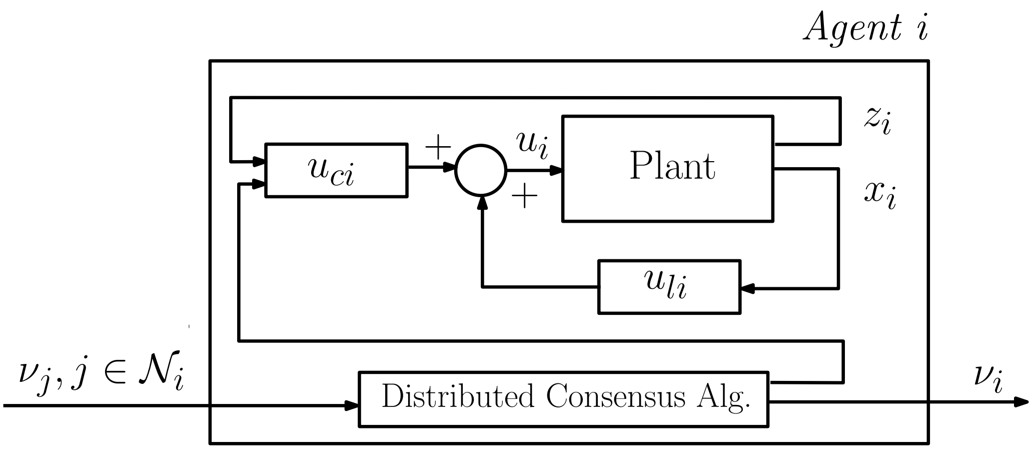


Figure 2.3: Control architecture for agent i

is proposed. Thus, the control law for each agent is represented by:

$$u_i(t) = u_{li}(t) + u_{ci}(t), \quad i \in \mathcal{N}. \tag{2.3.1}$$

where u_{li} and u_{ci} are the local and the agreement controllers respectively. The solution provided in this thesis to solve this problem is summarized in Figure 2.3.

In the sequel, a method is proposed for the design of the local and the agreement control laws.

2.3.1 Local control law

According to Assumption 2.4, there exist a local state feedback controller for each system given by:

$$u_{li} = -K_i x_i, \tag{2.3.2}$$

such that the matrix $A_i = \bar{A}_i - B_i K_i$ is *Hurwitz*. Thus, dynamics (2.2.1) can be written as:

$$\begin{cases} \dot{x}_i = A_i x_i + B_i u_{ci}, \\ z_i = C_i x_i, \end{cases} \qquad \forall i \in \mathcal{N} = \{1, \ldots, N\}. \tag{2.3.3}$$

The objective is to design a consensus algorithm which guarantees that the measurement vectors reach an agreement. In literature, classical distributed consensus algorithms have been intensively studied. Inherently, their stability and performance properties are well documented, see, for instance, [120, 192, 220] and the references therein.

2.3.2 Distributed consensus algorithm

In this section, we aim to design a consensus protocol which ensures that the measurement vectors reach an agreement. In literature, classical distributed consensus algorithms have been intensively studied. To achieve agents' agreement, consensus algorithms based on simple integrator dynamics [13, 194, 251] have been widely considered and applied to different applications for MAS such as formation control [171], rendezvous [62], cycle pursuit [134, 161], coverage [63, 206, 243] among many others [111, 127]. Inherently, the stability and performance properties of these simple controllers are well documented, for instance in [120, 192, 220] and the references therein.

One can easily realize that the control complexity for systems considering heterogeneous agents is greater than for simpler frameworks. Furthermore, even though some results for such a problem have been proposed in literature, most of them present major drawbacks such as calculation needs, complexity or accuracy of the solution. To overtake heterogeneous MAS agreement issues, we were able to propose an efficient way to simplify the control design. More precisely, the main idea is to add additional dynamics to the control law of each system which correspond to a trivial consensus algorithm. As a starting point, we will consider the simplest situation where the dynamics of these additional dynamics are driven by:

$$\dot{\nu}_i = -\sum_{j \in \mathcal{N}_i} (\nu_i - \nu_j), \quad \forall i \in \mathcal{N}, \tag{2.3.4}$$

where $\nu_i \in \mathbb{R}^m$. This allows defining the augmented vector $\nu = [\nu_1, \ldots, \nu_N]^T$ and thus the previous consensus algorithm can be rewritten in a matrix form such as $\dot{\nu} = -\mathbf{L} \otimes I_m \nu$, where $\mathbf{L}$ is the Laplacian matrix associated to the communication graph of the multi agent system and $\otimes$ represents the classical Kronecker product.

The stability of such a system has been widely studied in literature. See, for example, [192, 220] among many others. The well known convergence properties of (2.3.4) naturally motivate it as an appropriated "choice" for the additional dynamics. The rest of the contribution consists on using this well known consensus algorithm to reach an agreement on those additional dynamics, while applying a standard model tracking based controller to the remaining system. This ensures that the real system will have identical performances as the additional model. Due to the interactions between them, u_{ci} must be designed in a proper manner such that two correlated objectives are fulfilled:

$$\begin{cases} \lim\limits_{t \to \infty} (\nu_i - \nu_j) = 0, \\ \lim\limits_{t \to \infty} (z_i - \nu_i) = 0, \end{cases} \quad \forall i, j \in \mathcal{N}^2. \tag{2.3.5}$$

In other words, system (2.3.4) can then be seen as a reference model for system (2.2.1). Then, it is natural to introduce the error vector between the measurement vector z

from (2.2.1) and the additional dynamics ν given by:

$$\varepsilon_i = z_i - \nu_i. \tag{2.3.6}$$

We wish to ensure that each ε_i converges to zero as the time evolves. Therefore, the evolution of ε_i has the following dynamics:

$$\dot{\varepsilon}_i = -\beta\varepsilon_i,$$

where $\beta > 0$. Thus, it follows:

$$\begin{aligned} \dot{z}_i - \dot{\nu}_i &= -\beta(z_i - \nu_i), \\ C_i(A_i x_i + B_i u_{ci}) + (\mathbf{L})_i \otimes I_m \nu &= -\beta\varepsilon, \end{aligned} \tag{2.3.7}$$

where $(\mathbf{L})_i$ denotes the i^{th} line of the matrix $\mathbf{L}$. Due to the Assumption 2.3, C_iB_i is invertible for all agent i, and thus the proposed controller can be expressed as a standard asymptotic output tracking controller, see [126]. Therefore, it yields:

$$u_{ci} = (C_iB_i)^{-1}\left(\dot{\nu}_i - \beta(z_i - \nu_i) - C_iA_ix_i\right). \tag{2.3.8}$$

Remark 2.1. *One might see that in the previous calculus we inherently assume that, at each time instant, each agent i has access to its own full state. Despite the fact that, theoretical speaking, this does not represent a too conservative condition, the same comment does not hold in terms of practical application. However, this assumption might be relaxed by applying a observer and using the estimated state instead. For the sake of brevity, though, the synthesis of this observer will not be considered in this dissertation.*

Relying on a decoupling approach as well, a solution to the output synchronization problem for heterogeneous agents is presented in [299]. In particular, authors showed that an internal model requirement is necessary and sufficient for exponential synchronizability of a group of heterogeneous agents. Following [242], the authors add an internal model to the dynamics of each agent and synchronize these identical exosystems such that an output synchronization of the multi-agent systems is achieved. In order to compensate for heterogeneity in the individual system's dynamics, authors proposed in [299] specific coupling dynamics that are composed of **(i)** synchronized reference generators; **(ii)** Luenberger observers and **(iii)** static output regulation controllers. Therefore, each agent solves an output regulation problem with respect to its attached exosystem and the output synchronization problem for heterogeneous multi-agent systems is split into two parts: a homogeneous synchronization problem and local output regulation problem. Furthermore, these results can be easily extended to the case when reference generators contain exponentially unstable modes by imposing stronger connectedness assumptions onto the communication graph [242].

Our objective is to obtain a very simple control technique to reach consensus of heterogeneous multi-agent systems. In this chapter, the synchronization problem corresponds to single-integrator consensus and the output regulation problem is solved by an asymptotic output tracking controller due to the special structure of the agents under consideration, see Assumption 2.3. The main difference with respect to [299] is that the heterogeneous synchronization problem is solved using the properties of simple-integrator consensus algorithms. Consequently, this approach can easily be extended to more complex and realistic situations where, for instance, communication delays or external references are considered. Moreover, this structure seems to considerably reduce the control complexity and computational efforts with respect to [299].

The previous section presented and motivated the proposed control strategy. In the sequel we are going to provide stability conditions for this framework.

2.4 Stability analysis

In this section we are going to present stability conditions for the closed-loop multi-agent system (2.2.1) controlled by (2.3.1). This analysis offers the major advantage of separated stability analysis of each agent and the convergence analysis of the distributed consensus algorithm. The next theorem states our main result:

Theorem 2.1. *(Rodrigues de Campos et al. [228]) If Assumptions 2.1-5 are satisfied, then the control law* (2.3.1)*, given by:*

$$u_i = -K_i x_i + (C_i B_i)^{-1} \left(\dot{\nu}_i - \beta(z_i - \nu_i) - C_i A_i x_i\right) \tag{2.4.1}$$

where $\dot{\nu}_i$ is given by (2.3.4)*, guarantees that the multi-agent system* (2.2.1) *is stable and reaches an asymptotic measurement variable consensus, i.e., $z_i = z_j, \forall i, j \in \mathcal{N}$.*

Proof. In order to consider measurement variable consensus, an appropriate canonical representation of the locally controlled system is presented in the sequel following [6, 82, 83]. Consider that Assumption 2.3 is fulfilled and without loss of generality that $C = \begin{bmatrix} 0 & I_m \end{bmatrix}$. The input distribution matrix can be partitioned in such a way that:

$$B_i = \begin{bmatrix} \bar{B}_{1i} \\ \bar{B}_{2i} \end{bmatrix},$$

where $\bar{B}_{1i} \in \mathbb{R}^{(n_i - m)\times m}$ and $\bar{B}_{2i} \in \mathbb{R}^{m\times m}$. Then $CB = \bar{B}_{2i}$ and under Assumption 2.3 $rank(\bar{B}_{2i}) = m$. Hence, in particular, the left pseudo-inverse $\bar{B}_{2i}^{*} = (\bar{B}_{2i}^T \bar{B}_{2i})^{-1} \bar{B}_{2i}^T$ is well defined and there exist an orthogonal matrix $\tilde{T}_i \in \mathbb{R}^{m\times m}$ such that:

$$\tilde{T}_i \bar{B}_{2i} = \begin{bmatrix} 0 \\ B_{2i} \end{bmatrix},$$

where $B_{2i} \in \mathbb{R}^{m \times m}$ is non-singular. Therefore, with $\chi_i \in \mathbb{R}^{n_i - m}$ and $T_i \in \mathbb{R}^{m \times m}$, there exist a coordinate transformation of the form:

$$\begin{bmatrix} \chi_i \\ z_i \end{bmatrix} = T_i x_i, \tag{2.4.2}$$

where T_i is given by:

$$T_i = \begin{bmatrix} I_{n_i - m} & -\bar{B}_{1i}\bar{B}_{2i}^* \\ 0 & \tilde{T}_i^T \end{bmatrix}.$$

With respect to the new coordinated, the input distribution matrix is now given by:

$$B_i = \begin{bmatrix} 0 \\ B_{2i} \end{bmatrix},$$

and the output distribution matrix by:

$$C_i = \begin{bmatrix} 0 & \tilde{T}_i \end{bmatrix}.$$

Under the variable transformation (2.4.2), the new system matrix can be partitioned as follows:

$$T_i A_i T_i^{-1} = \begin{bmatrix} A_{11i} & A_{12i} \\ A_{21i} & A_{22i} \end{bmatrix},$$

and consequently, each system (2.3.3) can be rewritten as:

$$\begin{bmatrix} \dot{\chi}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} A_{11i} & A_{12i} \\ A_{21i} & A_{22i} \end{bmatrix} \begin{bmatrix} \chi_i \\ z_i \end{bmatrix} + \begin{bmatrix} 0 \\ B_{2i} \end{bmatrix} u_{ci}, \tag{2.4.3}$$

where $\chi_i \in \mathbb{R}^{n_i - m}$. Note that in this new representation, $A_{11i} \in \mathbb{R}^{(n_i - m)\times(n_i - m)}$ is *Hurwitz*, $A_{22i} \in \mathbb{R}^{m \times m}$ and $B_{2i} \in \mathbb{R}^{m \times m}$ is invertible. Further details and extensions to, for example, observability canonical forms can be found in [82] or [83]. The system is now rewritten in an appropriate canonical form particulary useful to consider the problem of measurement variable consensus. Due to previous transformations, the control law (2.4.1) can be rewritten as follows:

$$u_i = -K_i x_i + B_{2i}^{-1} \left[\dot{\nu}_i - \beta(z_i - \nu_i) - [0 \ I_m] T_i A_i x_i \right]. \tag{2.4.4}$$

Consider system (2.2.1) with the control law (2.4.4). For all $i \in \mathcal{N}$, each closed-loop system becomes:

$$\dot{x}_i = (\bar{A}_i - B_i K_i)x_i + B_i(B_{2i})^{-1}\left[\dot{\nu}_i - \beta(z_i - \nu_i) - [0\ I_m]T_i A_i x_i\right].$$

Following the line presented in the control design section, some computations show that the previous system is equivalent to:

$$\begin{bmatrix} \dot{\chi}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} A_{11i} & A_{12i} \\ 0 & -\beta I_m \end{bmatrix} \begin{bmatrix} \chi_i \\ z_i \end{bmatrix} + \begin{bmatrix} 0 \\ \beta I_m - (\mathbf{L}_i) \otimes I_m \nu \end{bmatrix}.$$

Recalling that $\varepsilon = z_i - \nu_i$, one has:

$$\begin{bmatrix} \dot{\chi}_i \\ \dot{\varepsilon}_i \end{bmatrix} = \begin{bmatrix} A_{11i} & A_{12i} \\ 0 & -\beta I_m \end{bmatrix} \begin{bmatrix} \chi_i \\ \varepsilon_i \end{bmatrix} + \begin{bmatrix} A_{12i}\nu_i \\ 0 \end{bmatrix}. \tag{2.4.5}$$

On the other hand, the variable ν_i is obtained by solving a consensus problem summarized as $\dot{\nu} = -\mathbf{L} \otimes I_m \nu$, which is known to be stable provided that the communication graph contains a directed spanning tree. Consequently, $\lim_{t\to+\infty}(\nu_i - \nu_j) = 0$, for all $(i,j) \in \mathcal{N}^2$. Finally, applying a separation principle allows the conclusion of the stability of the multi-agent system (2.4.5) for all $i \in \mathcal{N}$. Thus, for all $i \in \mathcal{N}$, $\lim_{t\to+\infty}(\nu_i - z_i) = 0$. This concludes the proof. □

Remark 2.2. *An other important issue concerns the agreement point. It has been shown (see the proof of Theorem 2.1) that (2.2.1) will achieve consensus where $z(\infty)$ depends on the initial conditions of the consensus variables ν_i. Thus, no matter which the initial state of the system is, (2.2.1) will always achieve consensus on the agreement value of ν (see for instance Figure 2.5(b)).*

2.5 Extensions to more complex situations

Heterogenous agents agreement strategies are known to be a top challenging problem for control theory. It was our intuition that it is possible to find a simple, non-complex and with low computational efforts controller for heterogenous agents agreement. As a result, the method proposed in this dissertation allow the analysis of each agent to be split of the analysis of the distributed consensus algorithm. Therefore, it is possible to extend the previous control law to more general situations, where for instance the communication link induces transmission delays as provided in [177, 179, 251], or when one considers distributed filters as in, *e.g.*, [195]. Both cases will be considered in the sequel.

2.5.1 Consensus algorithms with transmission delays

When agents are interconnected on a network, it is possible they are subject to communication delays. In literature, three main types of feedback controllers considering communication delays exist: **(i)** feedback without self-delay; **(ii)** feedback with identical self-delay and **(iii)** feedback with different self-delay. The feedback controller without self-delay only considers transmission delays for data sent from agent j to agent i over the communication network, and this was considered in [42, 50, 53, 144, 177, 178, 202, 301, 307]. Feedback with identical self-delay is usually proposed for MAS with delayed relative measurements or where agents have computation or reaction delays, and thus both the agent's own state and the neighbors' states are affected by the same delay. This model was studied, for example, in [26, 146, 194, 257, 301]. Finally, the feedback with different self-delay considers non-identical self-delay and neighboring delay. Such structure applies to situations where, for instance, there is a combination of computation delays and transmission delays. see [284, 302].

In this section, we only consider consensus algorithms without self-delay. In other words, it is assumed that agent i has access to its own variable without any delay, but receives the information from its neighbors after a time-delay caused by the communication network. Moreover, note that if we consider that all the communication delays are constant and equal to τ, then it can be assimilated as an average delay. The sequel is based on [176, 179] and the results within concerning delay robustness on consensus problems. Considering the controller (2a) of [179], the additional dynamics becomes:

$$\dot{\nu}_i(t) = K \textstyle\sum_{j=1}^{N} \frac{a_{ij}}{d_i}(\nu_j(t-\tau_{ij}) - \nu_i(t)) \quad i \in \{1, \dots, N\}, \tag{2.5.1}$$

The delay dependent adjacency matrix, containing the full information about the network topology and delay, is defined as follows:

$$\mathcal{A}_\tau(s) = \left[a_{ij}e^{-\tau_{ij}s}\right].$$

Then, the result from the Laplace transform of (2.5.1) is:

$$u(s) = -\mathbf{L}_n Y(s) = -(I - \Delta^{-1}\mathcal{A}_\tau(s))\nu(s),$$

where $u(s) = [u_1(s), \cdots, u_N(s)]^T$, $\nu(s) = [\nu_1(s), \cdots, \nu_N(s)]^T$. Therefore, the delay-interconnection matrix $\mathbf{L}_n$ can be defined as:

$$\mathbf{L}_n = I - \Delta^{-1}\mathcal{A}_\tau(s).$$

Since $\mathbf{L}_n$ is symmetric, the rightmost nonzero root of the corresponding characteristic equation is $-\lambda_2 K$, where $\lambda_2 > 0$ corresponds to the algebraic connectivity of the undirected graph [94]. An existent stability condition of consensus algorithms with communication delays, presented in [179], is described as follows.

Corollary 2.1. *(Münz et al. [179]) A single integrator (2.2.1) with feedback without self-delay (2.5.1), gain $K > 0$, arbitrary size of $N \in \mathbb{N}$, arbitrary delays $\tau_{ij} \leq \bar{\tau}$, and arbitrary topology with connected topology achieves consensus asymptotically for any $\bar{\tau}$.*

Theorem 2.2. *(Münz et al. [179]) A single integrator (2.2.1) with feedback without self-delay (2.5.1), gain $K \in (0, \frac{1}{\bar{\tau}})$, and arbitrary symmetric delays $\tau_{ij} = \tau_{ji} \leq \bar{\tau}$ achieves consensus exponentially with a convergence rate $\eta \in (0, \bar{\lambda}_2 K)$ that satisfies:*

$$e^{\eta\bar{\tau}} < min\left\{1 - \frac{\eta}{K} + \bar{\lambda}_2, \frac{1}{K\bar{\tau}}\right\}, \tag{2.5.2}$$

where $\bar{\lambda}_2 \in (0, 1)$ is a lower bound on the second smallest eigenvalue of λ_2 of $\boldsymbol{L}_n = I - \Delta^{-1}\mathcal{A}_\tau(s)$.

Consequently, the following result holds.

Corollary 2.2. *(Rodrigues de Campos et al. [228]) If Assumptions 2.1-5 and the conditions of both Corollary 2.1 and Theorem 2.2 are satisfied, the control law (2.4.1) where $\dot{\nu}$ is given in (2.5.1), ensures that the multi-agent system (2.2.1) is stable and reaches an asymptotic measurement variable consensus with a convergence rate expressed that satisfies (2.5.2).*

In [144, 202] it has been shown that system (2.2.1) controlled by (2.5.1) achieves consensus independent of delay using Gershgorin's circle theorem and Lyapunov-Razumikhin functions, respectively. However, in [179] authors used the generalized Nyquist criterion. Furthermore, the convergence rate has also been studied thoroughly for MAS without delays, see, *e.g.*, [189]. However, results for MAS with heterogeneous communication delays consider only discrete-time systems, see, *e.g.*, [184]. In [179], authors also present convergence rate conditions for continuous-time single integrator MAS with heterogeneous feedback delays. As for most of convergence rate conditions for MAS without delays [189] the derived conditions are not independent of the topology but depend on the connectivity of the underlying graph. These conditions are robust to unknown topologies as long as their algebraic connectivity is larger than a lower bound, *i.e.*, $\lambda_2 \geq \bar{\lambda}_2$.

2.5.2 Consensus algorithms with external reference

In this section we focus on reference-based consensus algorithms, see [195]. Our objective here is to develop a distributed algorithm that allows the agents to track an external signal while the consensus agreement is achieved. It is considered that each agent receives the same signal or different signals that can also be seen as the same signal corrupted by noise, for instance. In any case the objective is to achieve a measurement variable consensus, *i.e.*, $z_i = z_j, \forall i, j \in \mathcal{N}$, while tracking the average of all external signal references.

Let $r_i \in \mathbb{R}^m$ be the external signal reference received by agent $i \in \mathcal{N}$. In this situation, the new additional system can then be expressed as:

$$\dot{\nu}_i = -\alpha \sum_{j \in \mathcal{N}_i} (\nu_i - \nu_j) + \sum_{j \in \mathcal{J}_i} (r_j - \nu_i) \tag{2.5.3}$$

where $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$ and $r = [r_1, \dots, r_N]^T$ denotes the external signals. It follows from (2.5.3) that:

$$\dot{\nu} = -(\alpha \mathbf{L} + I_N + \Delta)\nu + (I_N + \mathcal{A})r \tag{2.5.4}$$

and finally:

$$\dot{\nu} = -A_\alpha \nu + B_\alpha r, \tag{2.5.5}$$

where $A_\alpha = \alpha \mathbf{L} + I_N + \Delta$ and $B_\alpha = I_N + \mathcal{A}$. As mentioned before, our objective is to reach an agreement on the average of measurements such that:

$$\nu_i \to r_c, \forall i \in \mathcal{N},$$

where $r_c = \frac{1}{N} \sum_{i=1}^{N} r_i$.

Theorem 2.3. *(Olfati-Saber* ***[195]****) Let r_i be a signal with uniformly bounded rate such that following inequality is satisfied $\|r^*\| \leq \gamma_1$, where $r^* = \vec{\mathit{1}} \otimes r_c$. Then $\nu = r^*$ is a globally asymptotically ϵ-stable equilibrium of the consensus algorithm given by* (2.5.3) *with:*

$$\epsilon = \frac{\gamma_1 \sqrt{N}(1 + d_{max}) + \gamma_2 \gamma_3) \lambda_{max}^{1/2}(A_\alpha)}{\lambda_{min}^{5/2}(A_\alpha)},$$

where constants γ_2 and γ_3 are defined as:

$$\|r - r^*\| \leq \gamma_2, \qquad \|B_\alpha^T A_\alpha\| \leq \gamma_3, \tag{2.5.6}$$

and where $A_\alpha = \alpha \boldsymbol{L} + I_N + \Delta$ and $B_\alpha = I_N + \mathcal{A}$.

In our case, the following result holds:

Corollary 2.3. *(Rodrigues de Campos et al.* ***[228]****) If Assumptions 2.1-5 and the conditions of Theorem 2.3 are satisfied, the control law* (2.4.1) *where $\dot{\nu}$ is given in* (2.5.3)*, ensures that the multi-agent system* (2.2.1) *is stable and reaches an measurement variable $\epsilon-$consensus.*

2.6 Simulation results

Autonomous robots are being recurrently used to help humans to perform certain task with improved performances and in better safety conditions. Indeed, we have mentioned in Chapter 1 that groups of autonomous robots with computing, communication, and mobility capabilities have become economically feasible and can perform a variety of spatially distributed sensing tasks such as search and recovery operations, manipulation in hazardous environments, exploration, surveillance, environmental monitoring for pollution detection and estimation.

In this section, some simulation results regarding the efficiency of the previously introduced approach will be presented. As mentioned before, consensus algorithms are a powerful tool for rendezvous applications as depicted in Figure 2.2. In the sequel, we are going to explore an realistic application setup based on flying robots, see Figure 2.4. In particular, such framework is adapted to several operations such as search, surveillance and recognition. Consider a set of $N = 4$ heterogeneous agents. In order to fit in with our problem, the matrices $\bar{A}_i, B_i$, and C_i for the different agents are defined by

$$
\begin{aligned}
\bar{A}_1 &= \begin{bmatrix} -1 & 0.5 \\ 0.05 & -1 \end{bmatrix}, & B_1 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, & C_1 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T, \\
\bar{A}_2 &= \begin{bmatrix} -2 & 1 \\ 0 & -0.9 \end{bmatrix}, & B_2 &= \begin{bmatrix} -1 \\ -1 \end{bmatrix}, & C_2 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T, \\
\bar{A}_3 &= \begin{bmatrix} 0 & 1 & 1 \\ -2 & 0.1 & 2 \\ 1 & 2 & 3 \end{bmatrix}, & B_3 &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, & C_3 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T, \\
\bar{A}_4 &= \begin{bmatrix} 0 & 1 & 1 & 0 \\ -2 & 0.1 & 2 & 2 \\ 0 & 1 & 2 & 3 \\ 0.2 & 0.4 & 0.1 & 0.3 \end{bmatrix}, & B_4 &= \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, & C_4 &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T.
\end{aligned}
$$

It is clear that the agents are characterized by different dimensions and stability properties. Moreover, note that Assumptions 2.1-4 are fulfilled. Thus, a simple pole placement allows us to find matrices K_i's such that the matrices $A_i = \bar{A}_i - B_i K_i$ are *Hurwitz*.

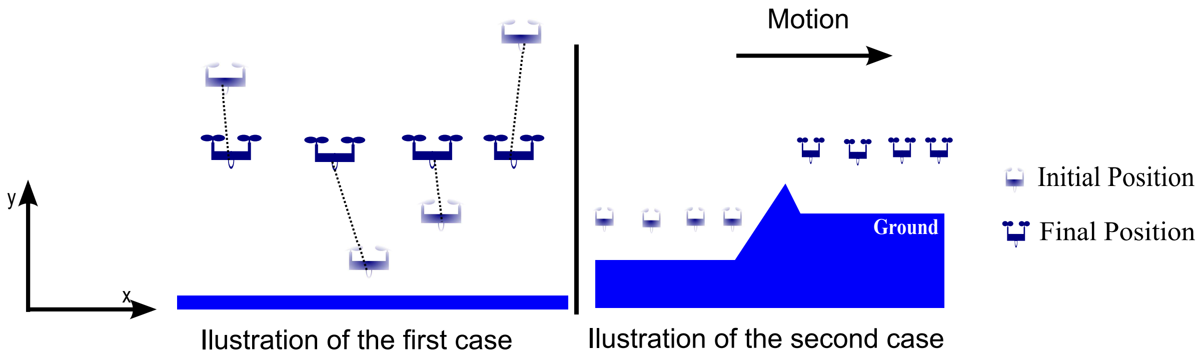


Figure 2.4: Application framework

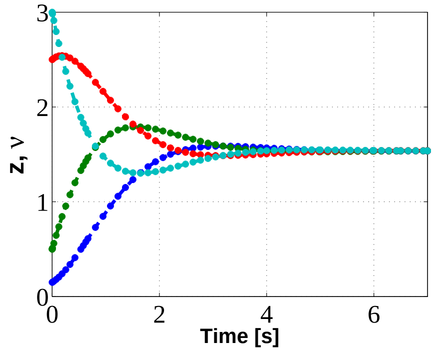
Consider that these four agents are connected through a graph expressed by the following Laplacian matrix:

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}.$$

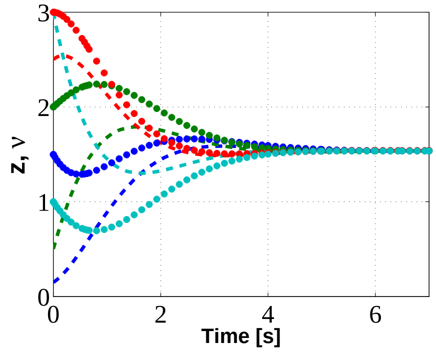
Note that this choice is not restrictive, as long as $\mathbf{L}$ represents a graph containing a spanning tree satisfying Assumption 2.5. For the different agents, the initial conditions are defined by:

$$x_1(0) = \begin{bmatrix} 1.5 & 0.15 \end{bmatrix}^T, \qquad x_2(0) = \begin{bmatrix} 5 & 0.5 \end{bmatrix}^T,$$
$$x_3(0) = \begin{bmatrix} 1.25 & 0.75 & 2.5 \end{bmatrix}^T, \quad x_4(0) = \begin{bmatrix} 1.35 & 0.45 & 3 & 1.5 \end{bmatrix}^T.$$

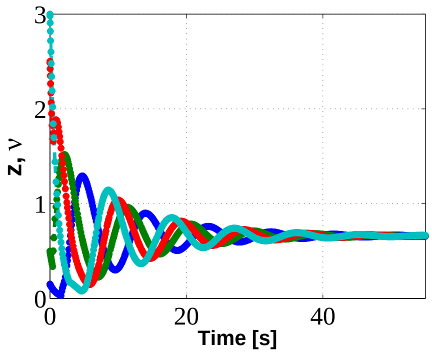
Let us comment Figure 2.4. In a first case, the heterogeneous agents must agree on the same height by applying the control strategy detailed in Section 2.3. Related to Remark 2.2, a special attention is paid to the influence of non-identical initial conditions (between the additional model and the multi-agent system) over convergence properties. In a second case, the robot fleet is moving through a changing environment. More precisely, the ground profile varies with the position of the fleet, while the objective aims to keep a constant height for the fleet with respect to the ground. This particular application has a pertinent practical meaning since, in several operations, such as search and recovery operations, robots usually move through a changing/hostile environment. Therefore, distributed collaboration, by exchanging individual measurement in order to achieve the control objectives, becomes a key issue.



(a) Evolution of the variables ν_i's and measurements z_i's when $\nu_i(0) = z_i(0)$



(b) Evolution of the variables ν_i's and measurements z_i's when $\nu_i(0) \neq z_i(0)$



(c) Evolution of the delayed variables ν_i's ($\tau = 0.6$) and measurements z_i's for random delays $\tau_{ij} \leq 3$, when $\nu_i(0) = z_i(0)$

Figure 2.5: Evolution of the additional dynamics ν (dashed line) and the measurement vector z (*star* line).

Consider for the moment Figure 2.5 [1]. Figure 2.5(a) shows simulation results for the closed-loop system (2.2.1) controlled by (2.3.1). We can clearly see that both systems reach a consensus, where the agreement value corresponds to $\nu(\infty)$. Note that the chosen **L** matrix is doubly stochastic, *i.e.*, all of its row-sums and column-sums are equal to 0. Therefore, $\nu(\infty)$ corresponds to the average of the initial conditions of the additional model, *i.e.*, $\nu(\infty) = ave\{\nu(0)\}$. Figure 2.5(b) shows the same set of heterogeneous agents initialized with different initial conditions from those of the additional variables. Once more, we can see that system (2.2.1) achieves consensus, converging to the agreement value of the additional model $\nu(\infty)$. These simulations enhance the conclusion mentioned in Remark 2.2, as well as the advantages of the referenced based algorithm introduced in Section 2.5.2.

1. For all figures, the dashed line corresponds to the additional dynamics, whereas the *star* line represents the measurement vector z evolution.

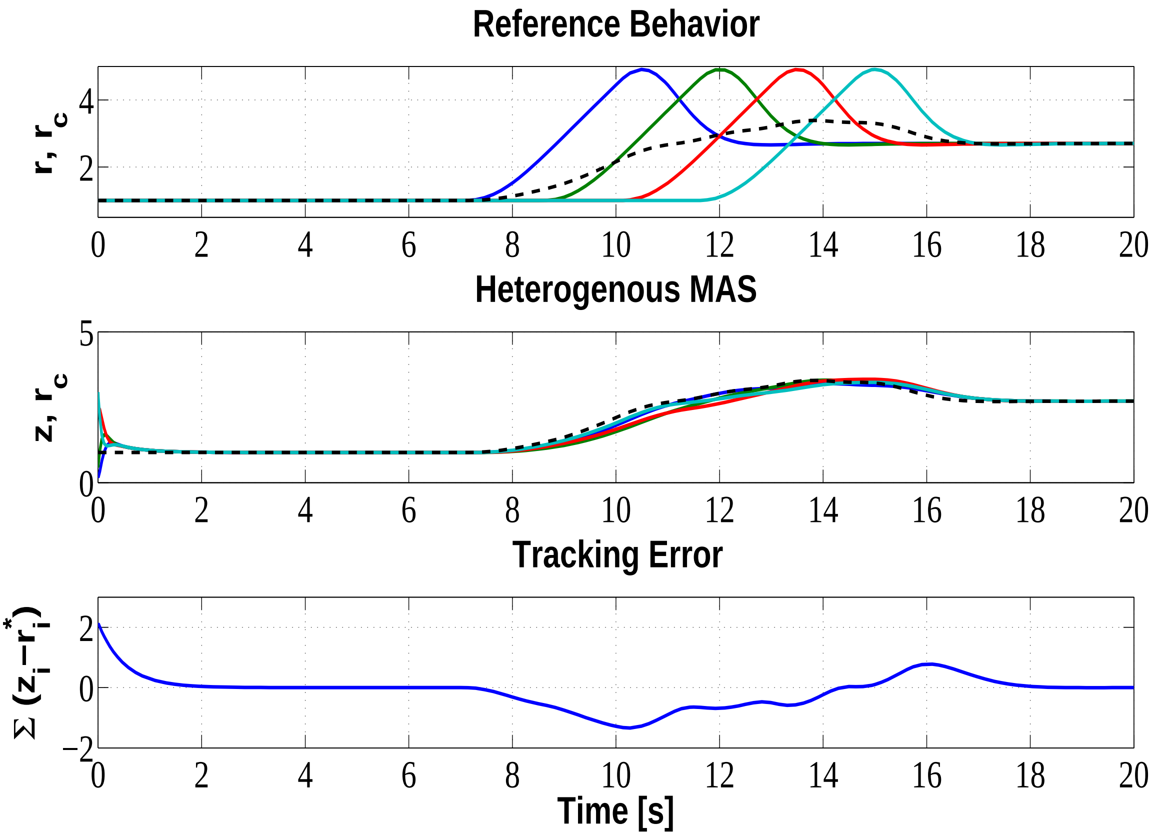


Figure 2.6: Non-identical reference signals and heterogeneous multi-agent systems response

In Figure 2.5(c), we can find simulation results for the closed-loop system (2.2.1) controlled by the delayed additional algorithm (2.5.1). For these simulations, we considered that random delays $\tau_{ij} \leq 3$, and we can conclude that the consensus protocol is robust to the network size and most important to bounded heterogenous delays. Moreover, similar performances as in [179] can be observed. Note that in Figures 2.5(a), 2.5(c) both the dashed line and the *star* line are completely overlapped due to equal initial conditions. Figure 2.5(c) shows that all elements of both the delayed additional dynamics ν_i and the measurement vector z_i, asymptotically converge to a common value ν_{eq} previously defined. These results enhance the efficiency of the proposed strategy when more complex situations are considered for the additional system.

Consider the application setup illustrated in Figure 2.4. While previous simulation results concern the first case, where flying robots must agree on a common height, consider now the second application case assuming that agents are moving on a changeable environment. Initially, agents are supposed to agree on a same height. At a certain time

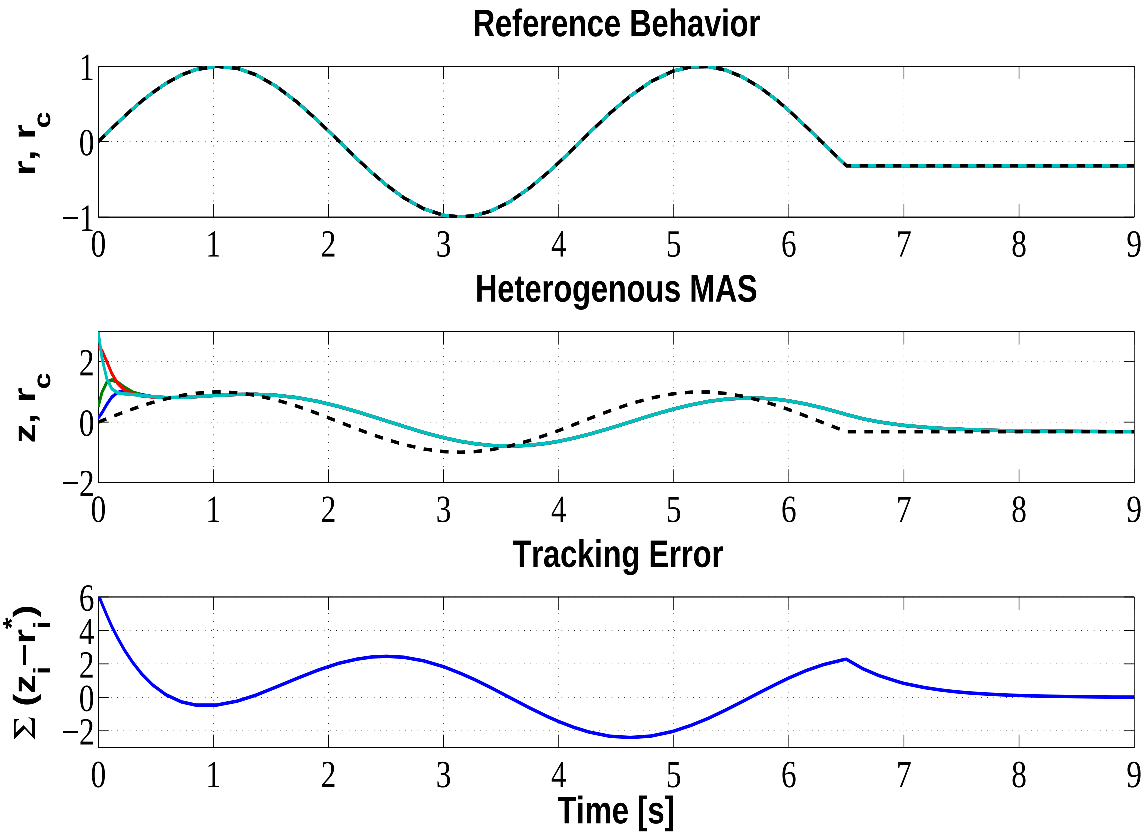


Figure 2.7: Identical reference signals and heterogeneous multi-agent systems response

instant the ground level changes due to a cliff or an obstacle. Therefore, such a case naturally asks for the external reference based control strategy mentioned in Section 2.5.2. Thus, agents are supposed to perform an efficient reference tracking and to finally agree on a common height. Figures 2.6 [2] and 2.7 [2] show the simulation of four flying vehicles controlled by the reference based control law (2.5.3). In Figure 2.6, the input signals r_i are different for every agent i, while in Figure 2.7 all elements r_i are equal. In both figures we can see the evolution of the measurement vector z of system (2.2.1), as well as the evolution of the error $z_i - r_c$, where r_c corresponds to the average of the reference signals. Therefore, we can see that all measurement vectors z_i of system (2.2.1) reach a common value r_c, satisfying our control objectives.

2. For all figures, r_c is displayed as a black dashed line.

2.7 Conclusions

In this chapter, a novel approach under which heterogeneous multi-agent systems reach measurement variable consensus has been presented, designed and analyzed. Moreover, Section 2.6 show the efficiency of our approach. We have shown that by using simple additional dynamics, the control of a high order heterogeneous multi-agent system not only becomes possible, but can be done with low constraints on system (2.2.1) and with low computational loads. The major advantage of the proposed algorithm remains in the separation of the stability analysis of each subsystem and the distributed control algorithm. As a result, we have also shown that different additional dynamics can be used, such as delayed Simple Integrator (SI) consensus and distributed consensus filter algorithms. This approach, when compared with the available results on literature, appears to have good performances, since other solutions offer some drawbacks as calculation complexity or restrictive assumptions.

The use of networks certainly offers several advantages, but not without drawbacks. In fact, the use of a shared network introduces new challenges, such as delays over communications, delays, data loss, or even communication blackout, see Chapter 1 and the references therein. Consequently, the Laplacian and its spectral properties [106, 168] play a crucial role in convergence analysis of consensus and alignment algorithms. In several multi-agent control problems, the convergence properties and speed of the system depend on the algebraic connectivity of the graph. Therefore, a special attention in paid in the sequel to the second smallest eigenvalue of graph's Laplacian, also called algebraic connectivity and that quantifies the speed of convergence of consensus algorithms. Some interesting and challenging questions recently raised by our community are still to be partially answered: **(i)** Is it possible to increase the convergence speed of a consensus algorithm? and **(ii)** If yes, how?

An approach able to answer these questions will be presented in the next chapter.

Chapter 3

Improved stability of consensus algorithms

Contents

3.1 Context

As for the previous chapter, the sequel focuses on consensus algorithms as one of the useful tools available for multi-agent systems agreement and in particular to rendezvous protocols, see Figure 2.1. While in the previous chapter we focus on the design of effective consensus algorithms for heterogeneous MAS, in this chapter a special attention is paid to consensus algorithm's convergence rate. More precisely, we will try to improve traditional convergence properties using memory based approaches.

In several multi-agent control problems, the convergence properties and speed of the system depend on the algebraic connectivity of the graph. The algebraic connectivity is an important network property for all the previous systems to reach convergence and it characterizes the convergence rate. More precisely, the algebraic connectivity is equal to the second smallest eigenvalue of $\mathbf{L}$. Connectivity control methods establish agent motions that preserve or maximize some network connectivity property. An optimization method of the topology-respecting weight matrix summarizing the updates at each node is presented in [112, 304] and the k-connectivity matrix of the graph is computed in a centralized fashion in [313]. Several distributed methods compute spanning subgraphs [314], specific Laplacian eigenvectors [212] or moments (mean, variance, skewness and kurtosis) of the Laplacian eigenvalue spectrum [210]. Some other works also maximize the algebraic connectivity through motion control without actually computing it, see, *e.g.*, [255]. Moreover, a distributed algebraic connectivity estimation method is proposed in [4]. More precisely, authors discuss a particular event-triggered consensus scenario and show that the availability of an estimate of the algebraic connectivity could be used for adapting the behavior of the average consensus algorithm.

Accelerating the convergence of distributed consensus algorithms have been also studied in literature by incorporating memory into the control laws. This chapter focuses in this specific approach. Even if for most applications delays lead to a reduction of performance and instability, in some cases the introduction of a delay in the control loop can help to stabilize a system. This has been studied in [110, 252]. More precisely, in [252] authors added a state sampled component to the control law, what can be seen as an artificial way to manipulate $\mathbf{L}$'s eigenvalues, by getting them further onto the left part of the complex plan. Considering Definition 1.7, this inherently means that the speed of convergence will change. An approach based on [252] will be proposed in the sequel aiming to maximize the speed of convergence. It is worth mentioning that these works are related with the International Research Group Groupement International de Recherche (GDRI) DelSys, supported by the Centre National de la Recherche Scientifique (CNRS). In the sequel, we consider both simple and double integrator dynamics and the communication graphs are supposed to be directed and undirected.

3.2 Simple integrator dynamics

3.2.1 Problem statement and preliminaries

In the scope of multi-robots systems, we are particulary motivated by motion control in a cartesian plane. Consequently, the position of each agent i is denoted

$$q_i = [x_i, y_i]^T \in \mathbb{R}^2,$$

where x_i and y_i represent the dynamics of q_i on the x-axis and y-axis, respectively. However, for the sake of notation's clearness and without loss of generality, in this chapter we consider only the dynamics of x_i.

Simple integrator dynamics are a particular case of the linear MAS widely used in literature [13, 176, 194, 251]. As mentioned before, we are interested in motion control of robot swarms, and in particular in rendezvous applications, see [62]. The desired behavior is depicted in Figure 3.1.

Consider the classical SI consensus algorithm given by:

$$\begin{cases} \dot{x}_i(t) = u_i(t) \\ u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \end{cases} \quad i \in \{1, \dots, N\}, \tag{3.2.1}$$

where x_i represents agent i variables. Introducing the vector $x(t) = [x_1(t), .., x_N(t)]^T$ containing the state of all agents, we then derive:

$$\dot{x}(t) = -\mathbf{L}x(t) \ , \tag{3.2.2}$$

where $\mathbf{L}$ is the Laplacian matrix. This algorithm is distributed in the sense that each agent has only access to information from its neighbors. Assume that there exist a constant and positive scalar μ such that:

$$\sum_{j \in \mathcal{N}_i} a_{ij} = \mu, \quad i \in \{1, \dots, N\}.$$

Since for achieving consensus some assumptions on the communication graph must be satisfied, the following assumptions hold.

Assumption 3.1. ***(Graph's Variance):*** *For any considered graph $\mathcal{G}$, we assume that the communication graph expressing neighborhood relations between agents is constant, and therefore the corresponding Laplacian matrix $\boldsymbol{L}$ is time-invariant.*

Other communication properties such as the presence of noise, packet loss and time delays will not be considered.

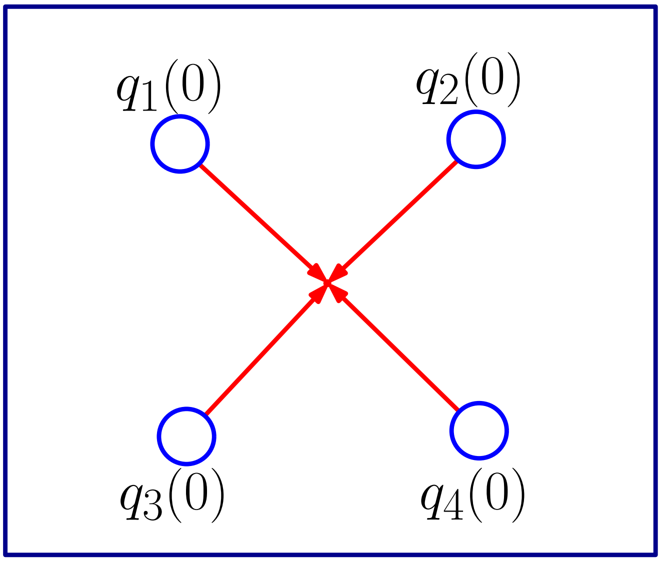


Figure 3.1: Illustration of simple integrator rendezvous

Assumption 3.2. ***(Graph's connectivity):*** *For any considered graph $\mathcal{G}$, we assume that the communication graph has a directed spanning tree.*

This ensures that zero is a simple eigenvalue of $\mathbf{L}$ and the corresponding eigenvector is the vector of ones, $\overrightarrow{\mathbf{1}}$. This implies that the algorithm will eventually reach consensus. Note that requiring a directed spanning tree is considerably less stringent than requiring a strongly connected and balanced graph [220].

The study of improved stability properties will be divided into two parts, one assuming *partial memory* and another considering *global memory*. This two concepts diverge on which sampled information is used for control matters. *Partial memory* uses neighbors's or agent's own past information while *global memory* considers all available information.

3.2.2 Partial memory

Controller design

For most applications, delays lead to a reduction in performances or can even lead to instability. However, there exist some cases where the introduction of a delay in the control loop can help to stabilize a system, see [110, 252]. In this thesis, we will prove that the simple integrator consensus algorithm belongs to this class of systems. To do so, algorithm (3.2.2) is modified into a new algorithm defined by:

$$\dot{x}(t) = (-\mathbf{L} - \delta\mathcal{A})x(t) + \delta\mathcal{A}x(t-\tau), \tag{3.2.3}$$

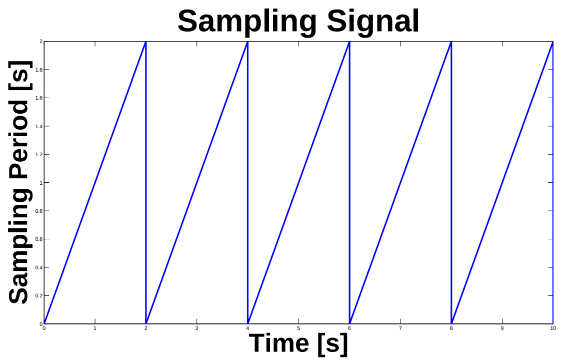


Figure 3.2: Graphical representation of the sampling signal.

where $\delta \in \mathbb{R}$ and $\tau \geq 0$ are additional parameters. Note that if δ and/or τ are taken as zeros, then the classical algorithm is retrieved. If δ and τ are not zero, then one can see that the proposed algorithm can be explained as follows. The non diagonal contribution of the Laplacian is split into two parts: one delayed and the other is kept at the current time. This allows conserving the averaging properties of the agreement algorithm. As the delay is now a control parameter, we can choose it from the most appropriate form. In this thesis, we will consider a sampling delay already used in [98] or in [250]. This sampling delay is given by:

$$\tau(t) = t - t_k, \; t_k \leq t < t_{k+1} \; ,$$

where the t_k's satisfies $0 = t_0 < t_1 < ... < t_k < ...$ corresponds to the sampling instants, see Figure 3.2. For the sake of simplicity, we will assume that the sampling process is periodic, *i.e.*, the difference between two successive sampling instants is constant and defined by:

$$t_{k+1} - t_k = T \; . \tag{3.2.4}$$

This makes sense in the situation of multi-agent systems. However, the latter analysis could be extended to asynchronous samplings.

From a computational point of view, to choose a sampling delay is relevant with respect to the introduction of a constant delay τ since in the sampling delay case, only one data is held in the algorithm whereas in the case of a constant delay, all values of x in the interval $[t - \tau \; t]$ should be kept in memory. However, a more dedicated stability analysis is required for such types of systems.

Finally, the algorithm (3.2.2) has been modified into a new algorithm shown in Fig.3.3. The improved algorithm is defined by:

$$\forall t \in [t_k \; t_{k+1}[, \quad \dot{x}(t) = (-\mathbf{L} - \delta\mathcal{A})x(t) + \delta\mathcal{A}x(t_k), \tag{3.2.5}$$

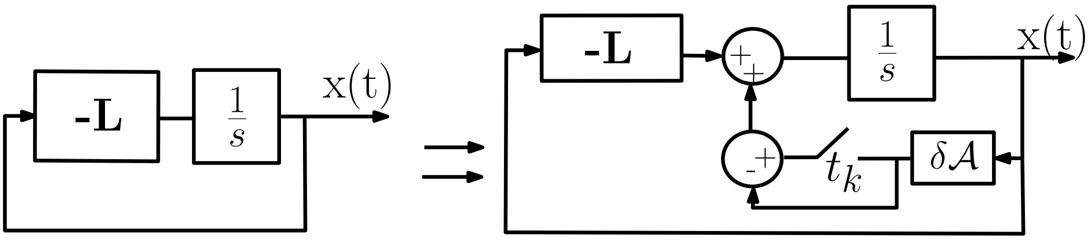


Figure 3.3: Bloc diagrams of the classical and the improved algorithms with neighboring partial memory.

where δ and T are additional parameters of the improved algorithm. From the point of view of agent i, the state x_i is available at every time t. However, both continuous and sampled data from the neighbor agents of agent i are used. Note that if δ and/or T are taken as zero, the classical algorithm is retrieved.

In the sequel, a stability analysis of the algorithm is proposed for any graph with a directed spanning tree, represented by the Laplacian $\mathbf{L}$. An inherent assumption is that all agents are synchronized and share the same clock. This analysis is composed of two parts, one dealing with the stability of the algorithm and another concerning the agreement of the agents. In particular, we will propose a method to choose appropriately the algorithm parameters δ and T for a given $\mathbf{L}$, considering a performance optimisation.

Definition of an appropriate model

This section focuses on the definition of a suitable modeling of the consensus algorithm (3.2.5) to analyze its convergence. Recall that the vector $\overrightarrow{\mathbf{1}}$ is an eigenvector of the Laplacian matrix associated to the eigenvalue 0. Thus, it is possible to find a change of coordinates $x = Wz$, as proposed in [251], such that:

$$U(-\mu I + \mathcal{A})W = \begin{bmatrix} \Omega & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}, \tag{3.2.6}$$

where $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = W^{-1}$ and $U_2 = (U)_N$. For graphs containing a directed spanning tree, the Laplacian eigenvalues are all positive and we denote them by $0 < \lambda_2 \leq \ldots \leq \lambda_N$. Let also $\Omega \in \mathbb{R}^{(N-1)\times(N-1)}$ be a diagonal matrix with $-\lambda_i$. The following lemma, based on the variable change proposed in [251], provides an appropriate way to rewrite (3.2.5) based on the properties of the matrix $\mathbf{L}$ for simple integrator agents.

Lemma 3.1. *(Rodrigues de Campos et al. [232]) The system (3.2.5) can be rewritten in the following way:*

$$\dot{z}_1(t) = -(\Omega + \delta(\Omega + \mu I))z_1(t) + \delta(\Omega + \mu I)z_1(t_k), \tag{3.2.7a}$$
$$\dot{z}_2(t) = -\delta\mu z_2(t) + \delta\mu z_2(t_k), \tag{3.2.7b}$$

where $z_1 \in R^{N-1}$, $z_2 \in R$ and the matrix Ω in given in (3.2.6).

Proof. By the Leibnitz formula, we have $x(t_k) = x(t) - \int_{t_k}^t \dot{x}(s)ds$, for all differentiable functions x. System (3.2.5) can be rewritten as:

$$\dot{x}(t) = -\mathbf{L}x(t) - \delta\mathcal{A}\int_{t_k}^t \dot{x}(s)ds. \tag{3.2.8}$$

This representation is a way to understand how memory components affect the algorithm. It is then possible to rewrite (3.2.5) into two equations defined by $z_1 = U_1x \ \in \mathbb{R}^{(N-1)}$ and $z_2 = U_2x \ \in \mathbb{R}^N$ representing, the $N-1$ first components and the last component of z, respectively. Consequently, we obtain:

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = -\begin{bmatrix} \Omega & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} - \delta\begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix}\int_{t-\tau}^t \dot{z}(s)ds,$$

where $\begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix} = U\mathcal{A}W$ and $A'_2 = (U\mathcal{A}W)_N$. From (3.2.6), simple matrix calculations lead us to:

$$\begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix} = UAW = \mu I + U(-\mu I + \mathcal{A})W = \begin{bmatrix} \Omega + \mu I & \vec{0} \\ \vec{0}^T & \mu \end{bmatrix}. \tag{3.2.9}$$

Using the Leibnitz formula, (3.2.5) can be rewritten as:

$$\begin{array}{l} \dot{z}_1(t) = -\Omega z_1(t) - \delta(\Omega + \mu I)\int_{t_k}^t \dot{z}_1(s)ds, \\ \dot{z}_2(t) = -\delta\mu\int_{t_k}^t \dot{z}_2(s)ds, \end{array} \tag{3.2.10}$$

which concludes the proof. □

The consensus problem is now expressed into an appropriate form to derive stability criteria. In the case of a symmetric network, the matrix W is an orthogonal matrix which means $U = W^T$. Then if the last column of W is $\beta\vec{\mathbf{1}}$, then $U_2 = 1/(\beta N)\vec{\mathbf{1}}$, which means that z_2 corresponds to the average of the position of all agents. This does not hold always for asymmetric communication network.

In the sequel, a stability analysis of the algorithm is proposed for any graph with a directed spanning tree, represented by the Laplacian $\mathbf{L}$. Note that requiring a directed spanning tree is less stringent than requiring a strongly connected and balanced graph [220]. This analysis is composed of two parts, one dealing with the stability of the algorithm and another concerning the agreement of the agents. More particularly, we will propose a method to choose appropriately the algorithm parameters δ and T for a given $\mathbf{L}$, considering a performance optimisation. Let us motivate this study by the following calculus. Take Ω as the diagonal matrix of the Laplacian eigenvalues such that:

$$\Omega = \begin{bmatrix} -\lambda_2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -\lambda_N \end{bmatrix}. \tag{3.2.11}$$

Thus, we establish for all $i = 1, \dots, N-1$

$$\dot{z}_{1i}(t) = (-\lambda_{i+1} + b)z_{1i}(t) - bz_{1i}(t_k), \tag{3.2.12}$$

with $b = \delta(\lambda_{i+1} + \mu)$. By integrating the previous equation, the following recurrence equation represents the discrete dynamics of the algorithm:

$$z_{1i}(t_{k+1}) = A(\lambda_{i+1}, \delta, T)z_{1i}(t_k), \tag{3.2.13}$$

with

$$A(\lambda_{i+1}, \delta, T) = \exp^{(-\lambda_{i+1}+b)T} \frac{-\lambda_{i+1}}{-\lambda_{i+1}+b} + \frac{b}{-\lambda_{i+1}+b}.$$

Note that system's (3.2.13) stability increases as $A(\lambda_{i+1}, \delta, T)$ decreases. We will prove that by varying δ and T values close to zero, we achieve a performance improvement for $\forall \lambda_{i+1}$, if:

$$\begin{aligned} \frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial T} &\leq 0, \text{ for some } \delta \text{ values}, \\ \frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial \delta} &\leq 0, \text{ for some T values}. \end{aligned}$$

From (3.2.13), by derivation of $A(\lambda_{i+1}, \delta, T)$, we have:

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial T} = -e^{(-\lambda_{i+1}+b)T}\lambda_{i+1}$$

and

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial \delta} = \frac{-\lambda_{i+1}e^{(-\lambda_{i+1}+b)T}}{(-\lambda_{i+1}+b)}\left[T(\lambda_{i+1}+\mu) - \frac{(\lambda_{i+1}+\mu)}{-\lambda_{i+1}+b}\right] + \frac{(\lambda_{i+1}+\mu)}{(-\lambda_{i+1}+b)^2}(\lambda_{i+1}+2b)$$

When we evaluate the previous equation for $T \simeq 0$ and for $\delta \simeq 0$, respectively, we obtain:

$$\begin{aligned} \frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial T} &= -\lambda_{i+1} \leq 0, \\ \frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial \delta} &= e^{-\lambda_{i+1}T}(\lambda_{i+1}+\mu)\left(T + \frac{1}{\lambda_{i+1}}\right) - \left(\frac{\lambda_{i+1}+\mu}{\lambda_{i+1}}\right) \leq 0. \end{aligned}$$

Since

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial T} = -\lambda_{i+1} < 0, \quad \forall\ \delta,$$

and

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial \delta} < 0, \quad \forall\, T,$$

we can then conclude that for small values of δ and T system (3.2.13) tends to converge more rapidly when compared with the trivial algorithm. The pertinent problem of how to choose these parameters values has been raised here, and will be treated in the next section.

Stability analysis

On a first step, we are going to deal with the stability analysis of (3.2.7b). The following lemma holds.

Lemma 3.2. *(Rodrigues de Campos et al. [232]) The system defined in* (3.2.7b) *is constant for any sampling period T and any δ such that:*

$$\forall t, \quad z_2(t) = z_2(0). \tag{3.2.14}$$

Proof. Consider $k \geq 0$, any $t \in [t_k\ t_{k+1}[$ and any parameters T, δ. The ordinary differential equation (3.2.7b) has known solutions of the form:

$$z_2(t) = e^{-\delta\mu(t-t_k)}C_0 - z_2(t_k) \tag{3.2.15}$$

where $C_0 \in \mathbb{R}$ represent the initial condition of the ordinary differential equation. The initial condition is determined at time $t = t_k$. We then obtain $C_0 = 0$ and thus:

$$\forall t \in [t_k\ t_{k+1}[, \quad z_2(t) = z_2(t_k) = z_2(0). \tag{3.2.16}$$

Then, we deduce that z_2 is constant. This concludes the proof. □

Now, consider the consensus algorithm (3.2.5) rewritten in the form of (3.2.7). We can establish:

$$\dot{z}_1(t) = M_{SIP}(\delta)z_1(t) + M^*_{SIP}(\delta)z_1(t_k), \tag{3.2.17}$$

with $M_{SIP}(\delta) = -[\Omega + \delta(\Omega + \mu I)]$ and $M^*_{SIP}(\delta) = \delta(\Omega + \mu I)$. For the sake of clearness, consider that for any matrix A in $\mathbb{R}^n$, the notation $He\{A\} > 0$ corresponds to the following sum $A + A^T > 0$. The following theorem holds.

Theorem 3.1. *(Rodrigues de Campos et al. [232]) Consider the proposed consensus algorithm* (3.2.5) *associated to a given Laplacian $\boldsymbol{L}$ representing a communication graph* *with a directed spanning tree, a given* $\alpha > 0$, $\delta > 0$, *and* $T > 0$. *Assume that there exist* $P > 0$, $R > 0$, S_1 *and* $X \in \mathbb{S}^n$, $S_2 \in \mathbb{R}^{n\times n}$ *and* $N \in \mathbb{R}^{2n\times n}$ *that satisfy*

$$\begin{array}{l} \Psi_1(T) = e_\alpha(\tau)\Pi_1 + f_\alpha(T,0)\Pi_2 + h_\alpha(T,0)\Pi_3 < 0, \\ \Psi_2(T) = \left[\begin{array}{cc} e_\alpha(T)\Pi_1 + h_\alpha(T,T)\Pi_3 & g_\alpha(T,T)N \\ * & -g_\alpha(T,T)R \end{array}\right] < 0, \end{array} \tag{3.2.18}$$

where

$$\begin{array}{l} \Pi_1 = He\{M_1^T P M_0 - M_{12}^T(\frac{1}{2}S_1 M_{12} + S_2 M_2 + N^T)\} + 2\alpha M_1^T P M_1 \\ \Pi_2 = M_0^T R M_0 + He\{M_0^T(S_1 M_{12} + S_2 M_2)\}, \\ \Pi_3 = M_2^T X M_2, \end{array}$$

and $M_0 = \left[\begin{array}{cc} M_{SIP}(\delta) & M^*_{SIP}(\delta) \end{array}\right]$, $M_1 = \left[\begin{array}{cc} I & 0 \end{array}\right]$, $M_2 = \left[\begin{array}{cc} 0 & I \end{array}\right]$, $M_{12} = M_1 - M_2$. *The functions* f_α, g_α *and* h_α *for all scalars* T *and* $\tau \in [0\ T]$ *are given by:*

$$\begin{array}{l} e_\alpha(\tau) = e^{2\alpha\tau}, \\ f_\alpha(T,\tau) = (e^{2\alpha T} - e^{2\alpha\tau})/2\alpha, \\ g_\alpha(T,\tau) = \left\{\begin{array}{ll} e^{2\alpha T}(e^{2\alpha\tau} - 1)/2\alpha, & \text{if } \alpha > 0 \\ (e^{2\alpha\tau} - 1)/2\alpha, & \text{if } \alpha < 0 \end{array}\right. \\ h_\alpha(T,\tau) = \frac{1}{\alpha}\left[\frac{e^{2\alpha T}-1}{2\alpha T} - e^{2\alpha\tau}\right]. \end{array}$$

Then, the consensus algorithm (3.2.5) *with the parameter* δ *and the sampling period* T *is thus* $\alpha-$*stable. Moreover, the consensus equilibrium is given by:*

$$x(\infty) = U_2 x(0).$$

Proof. Consider the consensus algorithm (3.2.5). Using Lemma 3.1, the algorithm is rewritten as (3.2.7). The stability of the second equation (3.2.7b) is ensured based on the discrete time Lyapunov theorem. Define $z_{1k} = z_1(t_k)$ and introduce a quadratic Lyapunov function defined by $V(z_1) = {z_1}^T P z_1$ for all $z_1 \in \mathbb{R}^n$. The objective is to ensure that the increment $\Delta_\alpha V(z_1)$ is negative definite [293]. A candidate for υ_α is defined for all $t \in [t_k\ t_{k+1}]$ by:

$$\begin{array}{r} \upsilon_\alpha(t, z_{1t}) = f_\alpha(T,\tau)\zeta^T(t)[S_1\zeta(t) + 2S_2 z_{1k}] \\ +f_\alpha(T,\tau)\int_{t_k}^t \dot{z}_{1k}^T(\theta) R z_{1k}^T(\theta)d\theta + h_\alpha z_{1k}^T X z_{1k}, \end{array} \tag{3.2.19}$$

where $\zeta(t) = z_{1t} - z_{1k}$, $f_\alpha(T,\tau) = (e^{2\alpha T} - e^{2\alpha\tau})/2\alpha$ and $h_\alpha(T,\tau) = \frac{1}{\alpha}\left[\frac{e^{2\alpha T}-1}{2\alpha T} - e^{2\alpha\tau}\right]$. Denote $W_\alpha(t, z_{1t}) = e^{2\alpha\tau}[V(z_1) + \upsilon_\alpha(t, z_{1t})]$. Consider a positive scalar $0 < \vartheta < T$ and the functional W_α at time $t_k - \vartheta$ and $t_k + \vartheta$. Since $\zeta(t_k+\vartheta)$ and $f_\alpha(T_{k-1}, T_{k-1}-\vartheta)$ tend to 0 as $\vartheta \to 0$ for all $\alpha > 0$, the functional $\upsilon_\alpha(t, z_{1t})$ satisfies Theorem A.2 of Appendix A, extracted from [250]. Denote $\xi(s) = z_1(s)^T$. By noting that $\frac{d}{dt}(e^{2\alpha\tau} f_\alpha) = -e^{2\alpha\tau}$ and $\frac{d}{dt}(e^{2\alpha\tau} h_\alpha) = h_\alpha$, it yields:

$$\begin{aligned}\dot{W}_\alpha(t, z_{1t}) &\leq \xi^T(t)[e_\alpha(\tau)[\Pi_1 + 2\alpha M_1^T P M_1] + f_\alpha(T,\tau)\Pi_2 \\ &+\tau e_\alpha(\tau) N R^{-1} N^T + h_\alpha(T,\tau)\Pi_3]\xi(t),\end{aligned} \tag{3.2.20}$$

The previous inequality does not depend linearly on τ but on both τ and a non linear function of τ,$e_\alpha(T,\tau)$ and $f_\alpha(\tau)$. The solution proposed here is to use the convexity property of the exponential function ensuring that $e^{2\alpha\tau} \geq 1 + 2\alpha\tau$ if $\alpha > 0$ and $e^{-2\alpha\tau} \geq 1 + 2\alpha\tau$ if $\alpha < 0$. Consequently, for all $\alpha \neq 0$, $\tau e_\alpha(\tau) \leq g_\alpha(T,\tau)$. Since R and R^{-1} are positive definite, we have:

$$\begin{aligned}\dot{W}_\alpha(t, z_{1t}) &\leq \xi^T(t)[e_\alpha(\tau)\Pi_1 + f_\alpha(T,\tau)\Pi_2 \\ &+g_\alpha(T,\tau) N R^{-1} N^T + h_\alpha(T,\tau)\Pi_3]\xi(t),\end{aligned} \tag{3.2.21}$$

The previous inequality is linear with respect to $e^{2\alpha\tau}$. To prove that $\dot{W}_\alpha(t, z_{1t})$ is negative definite for all τ we apply a lemma on positivity of matrix inequations taken from [182] with $\lambda(t) = e^{-2\alpha\tau}$, which leads to:

$$\begin{aligned}\Pi_1 + f_\alpha(T,0)\Pi_2 < 0, \\ \Pi_1 + g_\alpha(T,T) N R^{-1} N^T < 0.\end{aligned}$$

This leads to (3.2.18) using the Schur complement. Thus, the global algorithm (3.2.7) is then exponentially stable with the decay rate α. Moreover, according to Lemma 3.1, and since $z_1 \to 0$ and $z_2 \to z_2(\infty)$, we conclude that the consensus equilibrium is:

$$x(\infty) = U_2 x(0).$$

The proof is concluded by noting that $z_2 = U_2 x$. □

Through previous calculations we presented an improved consensus algorithm based on an appropriated sampling. We have analytically demonstrated the improved behaviors and we derived stability conditions for the proposed problem. Throughout the next section, we are going to propose a different controller based on what we called global memory, *i.e.*, where agents use all available information.

3.2.3 Global memory

Controller design

We aim now for each agent i to keep in memory all the available information. Algorithm (3.2.2) is modified into a new algorithm shown in Fig. 3.4. Considering notation and notions from the previous section, the proposed algorithm can be written as:

$$\forall t \in [t_k \; t_{k+1}[, \quad \dot{x}(t) = (-\mathbf{L} - \delta\mathcal{A} + \sigma I)x(t) + (\delta\mathcal{A} - \sigma I)x(t_k) \; . \tag{3.2.22}$$

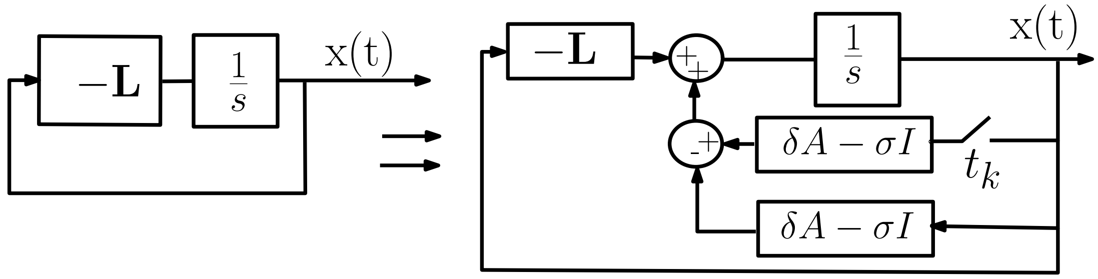


Figure 3.4: Bloc diagrams of the classical and the improved algorithms with global memory.

Here, we introduced a component proportional to σ and to the individual information of each agent. One can notice that the sampled information added to the classical algorithm corresponds to a weighted Laplacian matrix. Thus, a good weighting on the quantity of information to be used is essential. A procedure to find the optimal values as well as a critical analysis of new performances when compared with those of algorithm (3.2.5) will be presented next.

Definition of an appropriate model

The following lemma provides an appropriate modeling of (3.2.22).

Lemma 3.3. *(Rodrigues de Campos et al.* ***[230]****) The system* (3.2.22) *can be rewritten in the following way:*

$$\dot{z}_1(t) = -(\Omega + \delta(\Omega + \mu I) - \sigma I)z_1(t) + (\delta(\Omega + \mu I) - \sigma I)z_1(t_k), \quad (3.2.23a)$$

$$\dot{z}_2(t) = -(\delta\mu - \sigma)z_2(t) + (\delta\mu - \sigma)z_2(t_k), \quad (3.2.23b)$$

where $z_1 \in R^{N-1}$, $z_2 \in R$ and the matrix Ω in given in (3.2.6).

Proof. By the Leibnitz formula, we have $x(t_k) = x(t) - \int_{t_k}^{t} \dot{x}(s)ds$, for all differentiable functions x. We can then write:

$$\dot{x}(t) = -\mathbf{L}x(t) - (\delta\mathcal{A} - \sigma I)\int_{t_k}^{t} \dot{x}(s)ds. \quad (3.2.24)$$

This representation is a way to understand how memory components affect the algorithm. We then rewrite (3.2.22) as:

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = -\begin{bmatrix} \Omega & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} - \begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix}\int_{t-\tau}^{t} \dot{z}(s)ds,$$

where $\begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix} = U(\delta\mathcal{A} - \sigma I)W$ and $A'_2 = (U(\delta\mathcal{A} - \sigma I)W)_N$. Note that the previous equation is composed of two equations defined by $z_1 = U_1 x \in \mathbb{R}^{(N-1)}$ and $z_2 = U_2 x \in \mathbb{R}^N$ representing, respectively, the $N-1$ first components and the last component of z. From (3.2.6), simple matrix calculations lead us to:

$$\begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix} = \delta U\mathcal{A}W - \sigma UIW = \delta(\mu I + U(-\mu I + \mathcal{A})W) - \sigma UIW = \begin{bmatrix} \delta(\Omega + \mu I) - \sigma I & \vec{0} \\ \vec{0}^T & \delta\mu - \sigma \end{bmatrix}$$

To conclude the proof, we use the Leibnitz formula to rewrite (3.2.22) as:

$$\begin{array}{l} \dot{z}_1(t) = -\Omega z_1(t) - (\delta(\Omega + \mu I) - \sigma I) \int_{t_k}^{t} \dot{z}_1(s)ds, \\ \dot{z}_2(t) = -(\delta\mu - \sigma) \int_{t_k}^{t} \dot{z}_2(s)ds, \end{array} \tag{3.2.25}$$

□

The consensus problem (3.2.22) is now expressed into an appropriate form to establish stability criteria. Stability results are going to be derived in the next section.

Stability analysis

The stability of z_2 for a simple integrator consensus algorithms with partial memory has already been studied in Lemma 3.2. These results hold when global memory is considered. However, the stability of z_1 is still to be analyzed. In order to derive stability conditions ensuring exponential stability with a guaranteed decay rate, consider the consensus algorithm (3.2.23a) rewritten in the following form:

$$\dot{z}_1(t) = M_{SIG}(\delta,\sigma)z_1(t) + M^*_{SIG}(\delta,\sigma)z_1(t_k), \tag{3.2.26}$$

where

$$M_{SIG}(\delta,\sigma) = -[\Omega + \delta(\Omega + \mu I) - \sigma I], \quad M^*_{SIG}(\delta,\sigma) = [\delta(\Omega + \mu I) - \sigma I].$$

The following theorem holds.

Theorem 3.2. *(Rodrigues de Campos et al. [**230**]) Consider the proposed consensus algorithm (3.2.22) associated to a given Laplacian* $\boldsymbol{L}$ *representing a communication graph with a directed spanning tree, a given* $\alpha > 0$, $\delta > 0$, $\sigma > 0$ *and* $T > 0$. *Take:*

$$M_0 = \begin{bmatrix} M_{SIG}(\delta,\sigma) & M^*_{SIG}(\delta,\sigma) \end{bmatrix},$$

and M_1, M_2, M_{12} *as defined in Theorem 3.1. Assume that there exist* $P > 0$, $R > 0$, S_1 *and* $X \in \mathbb{S}^n$, $S_2 \in \mathbb{R}^{n\times n}$ *and* $N \in \mathbb{R}^{2n\times n}$ *that satisfy conditions (3.2.18) of Theorem 3.1.*

Then, the consensus algorithm (3.2.22) *with the parameters* δ, σ *and the sampling period* T *is thus* $\alpha-$*stable. Moreover, the consensus equilibrium is given by:*

$$x(\infty) = U_2 x(0).$$

Proof. The proof follows directly from Theorem 3.1, by taking into consideration now $M_0 = \left[\begin{array}{cc} M_{SIG}(\delta,\sigma) & M^*_{SIG}(\delta,\sigma) \end{array} \right]$, where $M_{SIG}(\delta,\sigma)$ and $M^*_{SIG}(\delta,\sigma)$ are defined in (3.2.26). □

3.3 Double integrator dynamics

3.3.1 Problem statement and preliminaries

In the scope of multi-robots systems, we are particulary motivated by motion control in a cartesian plane. Consequently, the position of each agent i is denoted:

$$q_i = [x_i, y_i]^T \in \mathbb{R}^2,$$

where x_i and y_i represent the dynamics of q_i on the x-axis and y-axis, respectively. However, for the sake of notation's clearness and without loss of generality, we consider only the dynamics of x_i in this chapter.

Double integrator dynamics are a particular case of the linear MAS, and extensively used in literature [216, 217, 269, 323]. As mentioned before, we are interested in motion control of robot swarms, and in particular in rendezvous protocols [269], as depicted in Figure 3.5.

Consider the classical Double Integrator (DI) consensus algorithm:

$$\begin{cases} \ddot{x}_i(t) = u_i(t), \\ u_i(t) = -\sum_{j\in\mathcal{N}_i}[\varsigma\dot{x}_i(t) + a_{ij}(x_j(t) - x_i(t))], \end{cases} \quad i \in \{1,\ldots,N\},$$

where x_i represents agent i variables. Introducing the vector $x(t) = [x_1(t),..,x_N(t)]^T$ containing the state of all agents, we obtain in a vector form:

$$\ddot{x}(t) = -\varsigma\dot{x}(t) - \mathbf{L}x(t)\ . \tag{3.3.1}$$

To achieve consensus, some assumptions on the communication graph must be satisfied. For instance, it is known that with $\varsigma > 0$, where ς denotes the absolute damping, algorithm (3.3.1) lead to an agreement if the communication graph is undirected but the algorithm is not necessarily stable if the graph associated to the Laplacian $\mathbf{L}$ is directed [217]. Therefore, the following assumptions hold.

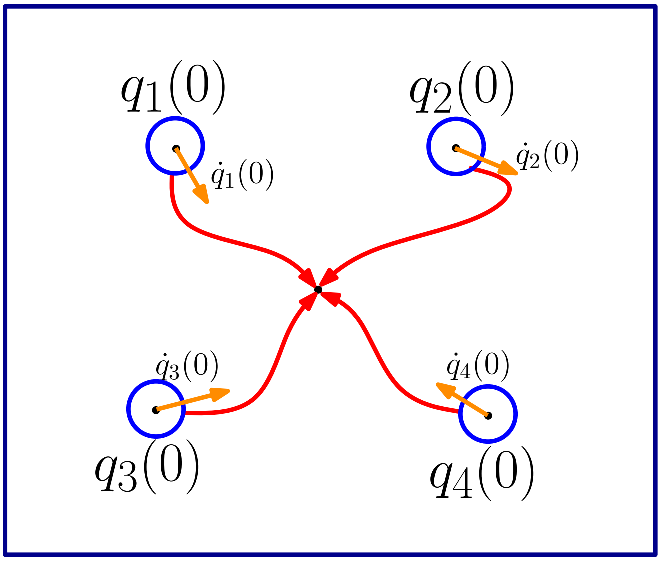


Figure 3.5: Illustration of double integrator rendezvous

Assumption 3.3. ***(Graph's Variance):*** *For any considered graph $\mathcal{G}$, we assume that the communication graph expressing neighborhood relations between agents is constant, and therefore the corresponding Laplacian matrix $\boldsymbol{L}$ is time-invariant.*

Other communication properties such as the presence of noise, packet loss and time delays will not considered.

Assumption 3.4. ***(Graph's connectivity):*** *For any considered graph $\mathcal{G}$, we assume that the communication graph has a directed spanning tree.*

This ensures that zero is a simple eigenvalue of $\mathbf{L}$ and the corresponding eigenvector is the vector of ones, $\overrightarrow{\mathbf{1}}$. This implies that the algorithm will eventually reach consensus. Note that requiring a directed spanning tree is considerably less stringent than requiring a strongly connected and balanced graph [220].

Assumption 3.5. ***(Absolute damping):*** *For every agent i, the absolute damping denoted by ς is supposed to be equal to zero, i.e., $\varsigma = 0$.*

Due to Assumption 3.5, additional difficulties are raised since we have:

$$\ddot{x}(t) = -\mathbf{L}x(t) \ , \tag{3.3.2}$$

and by introducing the augmented vector $\tilde{x}(t) = [x^T(t)\ \dot{x}^T(t)]^T$ we get:

$$\tilde{x}(t) = \begin{bmatrix} 0 & I \\ -\mathbf{L} & 0 \end{bmatrix} y(t) = \bar{L}\tilde{x}(t) \ . \tag{3.3.3}$$

It is important to point out that the trace of the matrix is zero whatever the communication graph and that the eigenvalues of $\mathbf{L}$ are either on the imaginary axis or there is at least one eigenvalue on the right side of the imaginary axis. Consequently, this leads to an oscillatory or unstable behavior of the algorithm. It is precisely this behavior that motivates our interest. In fact, since simple integrator consensus algorithm's performances can also be improved using an appropriated sampling, as shown in the previous section, it was our intuition that the benefits of such an approach should be even greater for double integrator dynamics. Indeed, it is important to bring forward the technical advantages of Assumption 3.5: by supposing $\sigma = 0$, we reduce drastically the information quantity needed for the control laws, and in a technical point of view, no velocity sensors are needed but only sensors to get the agent's position. This means economical, space and calculation savings.

The study of improved stability properties of double integrator consensus algorithms presented in the sequel will only consider partial memory, using for control agent's own information sampled information. However, these results can be easily extended in order to consider global memory.

3.3.2 Controller design

Using the concept of stabilizing delay, we will consider a sampling delay that was used in [98, 250] and defined in (3.2.4). It is important to recall that an inherent assumption is that all agents are synchronized and share the same clock. Using partial memory, algorithm (3.3.2) is modified into a new algorithm defined by:

$$\forall t \in [t_k \; t_{k+1}[, \quad \ddot{x}(t) = -(\mathbf{L} + \phi^2 I)x(t) + \phi^2 x(t_k) \; . \tag{3.3.4}$$

where $\phi \in \mathbb{R}$ and $T \geq 0$ are now additional control parameters. Moreover, as for the memory based controller (3.2.5), if ϕ and/or T are taken as zeros then the respective classical algorithm is retrieved. The structure of controller (3.3.4) is represented in Figure 3.6. It is important at this stage to enhance that (3.3.4) differs from (3.2.5) on the partial information used. More precisely, while in (3.2.5) the information of each agent's neighbors is incorporated in the controller weighted by the parameter δ, in (3.3.4) each agent i uses its own sampled information weighted by the parameter ϕ. Therefore, the diagonal contribution of the Laplacian is split in two parts: one delayed and the other is kept at the current time.

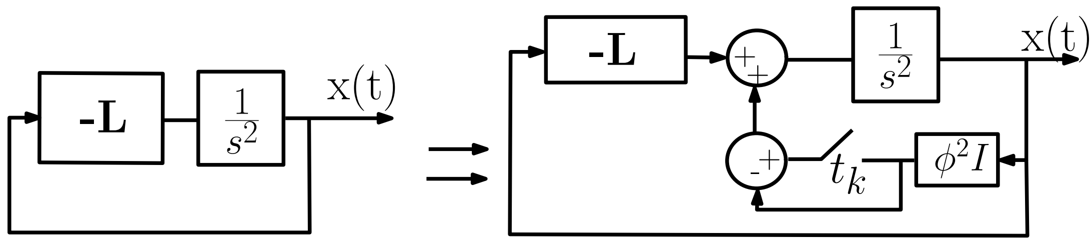


Figure 3.6: Bloc diagrams of the classical and the improved algorithms with neighboring partial memory.

3.3.3 Definition of an appropriate model

This section focuses on the definition of a suitable modeling of the consensus algorithm for convergence analysis. The next lemma shows an appropriate way to rewrite (3.3.4) based on the properties of **L**.

Lemma 3.4. *(Rodrigues de Campos et al. **[231]**) The consensus problem (3.3.4) can be rewritten using $z_1 \in \mathbb{R}^{N-1}$, $z_2 \in \mathbb{R}$ such that:*

$$\ddot{z}_1(t) = (\Omega - \phi^2 I)z_1(t) + \phi^2 z_1(t_k), \tag{3.3.5a}$$
$$\ddot{z}_2(t) = -\phi^2 z_2(t) + \phi^2 z_2(t_k), \tag{3.3.5b}$$

where the matrix Ω is given in (3.2.6).

Proof. Consider (3.3.4). It can be rewritten as follows:

$$\forall t \in [t_k\ t_{k+1}[, \quad \ddot{x}(t) = -\mathbf{L}x(t) - \phi^2 \int_{t_k}^{t} \dot{x}(s)ds.$$

Applying the change of coordinates $z = Ux$ it follows from (3.2.6) that (3.3.4) can be rewritten into two equations where $z_1 = U_1 x \ \in \mathbb{R}^{(N-1)}$ and $z_2 = U_2 x \ \in \mathbb{R}^N$ represent respectively the $N-1$ first components and the last component of z. Noting that $U(-\phi^2 I)W = -\phi^2 I$, (3.3.4) can be rewritten using the Leibnitz formula as:

$$\begin{array}{l} \ddot{z}_1(t) = \Omega z_1(t) - \phi^2 \int_{t_k}^{t} \dot{z}_1(s)ds, \\ \ddot{z}_2(t) = -\phi^2 \int_{t_k}^{t} \dot{z}_2(s)ds. \end{array} \tag{3.3.6}$$

The proof is concluded using $\int_{t_k}^{t} \dot{z}_i(s)ds = z_i(t) - z_i(t_k)$. □

Based on change of coordinates presented in (3.2.6), the sampled algorithm (3.3.4) is decomposed into two components: **(i)** a vectorial component z_1 associated with non-zero eigenvalues that converges to zero and **(ii)** a scalar component z_2 associated with the zero eigenvalue that converges to an agreement value dependent ont the system's initial positions. As mentioned before, in the case of a symmetric network, the matrix W is an orthogonal matrix which means $U = W^T$. Therefore, if the last column of W is $\beta\overrightarrow{\mathbf{1}}$,we obtain $U_2 = 1/(\beta N)\overrightarrow{\mathbf{1}}$, which means that z_2 corresponds to the average of the position of all agents. This does not always hold for asymmetric communication network.

The consensus problem is now expressed into an appropriate form to perform the stability analysis of (3.3.4). Due to the structure of this new model, the following analysis is composed of two parts. The first studies the agreement value and the second deals with the algorithm's stability. More precisely, we will propose a method to choose appropriately the algorithm parameters ϕ and T for a given $\mathbf{L}$, considering a performance optimisation.

3.3.4 Stability analysis

On a first step, we will deal with the stability analysis of (3.3.5b). The following lemma holds.

Lemma 3.5. *(Rodrigues de Campos et al.* ***[231]****) The system defined in* (3.3.5b) *is stable for any sampling period T and any ϕ such that* $\sin(\phi T) \neq 0\ [\pi]$, *i.e., $\phi T \neq k\pi$. The variable z_2 converges to:*

$$z_2(\infty) = z_2(0) + \gamma_{\phi T}\dot{z}_2(0), \tag{3.3.7}$$

where $\gamma_{\phi T} = \sin(\phi T)/(\phi(1-\cos(\phi T))) = \tan((\pi - \phi T)/2)/\phi$. *Moreover, the convergence rate of the solution to this equilibrium is* $-\log|\cos(\phi T)|$.

Proof. Consider $k \geq 0$ and any $t \in [t_k\ t_{k+1}[$ and any parameters T, ϕ such that $\phi T \neq 0\ [\pi]$. Define the augmented vector $\chi^* = [z_2^T(t)\ \dot{z}_2(t)]^T$. Equation (3.3.5) can be rewritten as follows:

$$\dot{\chi^*} = \bar{M}_{DIP}(\phi)\chi^*(t) + \bar{M}^*_{DIP}(\phi)\chi^*(t_k), \tag{3.3.8}$$

where

$$\bar{M}_{DIP}(\phi) = \begin{bmatrix} 0 & 1 \\ -\phi^2 & 0 \end{bmatrix},$$

and

$$\bar{M}^*_{DIP}(\phi) = \begin{bmatrix} 0 & 0 \\ \phi^2 & 0 \end{bmatrix}.$$

It is easy to see that $\bar{M}_{DIP}(\phi)$ is invertible and that:

$$\bar{M}_{DIP}(\phi)^{-1} = \begin{bmatrix} 0 & -1/\phi^2 \\ 1 & 0 \end{bmatrix}.$$

The previous ordinary differential equation has known solutions of the form:

$$\chi^*(t) = e^{M_{DIP}(\phi)(t-t_k)}[C_0\ C_1]^T - \bar{M}_{DIP}(\phi)^{-1}\bar{M}^*_{DIP}(\phi)\chi^*(t_k), \tag{3.3.9}$$

where C_0 and $C_1 \in \mathbb{R}$ represent the initial conditions of the ordinary differential equation. This leads to:

$$\chi^*(t) = \begin{bmatrix} \cos(w(t)) & \sin(w(t))/\phi \\ -\phi\sin(w(t)) & \cos(w(t)) \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \chi^*(t_k), \tag{3.3.10}$$

where $w(t) = \phi(t - t_k)$. The initial conditions are determined at time $t = t_k$. We then obtain $C_0 = 0$ and $C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix} \chi^*(t_k)$. Thus, we get the following recurrence equation:

$$\chi^*(t_{k+1}) = \begin{bmatrix} 1 & \sin(\phi T)/\phi \\ 0 & \cos(\phi T) \end{bmatrix} \chi^*(t_k) = \begin{bmatrix} 1 & \sin(\phi T)/\phi \\ 0 & \cos(\phi T) \end{bmatrix}^{k+1} \chi^*(0),$$

where $T = t_{k+1} - t_k$. Simple computations lead to:

$$\chi^*(t_{k+1}) = \begin{bmatrix} 1 & \sin(\phi T)\sum_{i=0}^{k+1}\cos(\phi T)^i/\phi \\ 0 & \cos(\phi T)^{k+1} \end{bmatrix} \chi^*(0)\ . \tag{3.3.11}$$

In the previous expression, we recognize a geometric sequence. Thus, we obtain the following expression:

$$\chi^*(t_{k+1}) - \begin{bmatrix} z_2(\infty) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -\gamma_{\phi T}\cos(\phi T)) \\ 0 & 1 \end{bmatrix} \cos(\phi T)^{k+1}\chi^*(0)\ . \tag{3.3.12}$$

The assumption on $\phi T \neq 0\ [\pi]$ implies $\cos(\phi T) < 1$. Then it implies that $\dot{z}_2(t_k)$ tends zero and z_2 to $z_2(\infty)$ defined in (3.3.7). This concludes the proof. □

Consider the consensus algorithm (3.3.5a) rewritten in the following form:

$$\dot{\chi}(t) = M_{DIP}(\phi)\chi(t) + M^*_{DIP}(\phi)\chi(t_k), \tag{3.3.13}$$

where $\chi(t) = [z_1^T(t)\ \dot{z}_1^T(t)]^T$ and

$$M_{DIP}(\phi) = \begin{bmatrix} 0 & I \\ (\Omega - \phi^2 I) & 0 \end{bmatrix}, \quad M^*_{DIP}(\phi) = \begin{bmatrix} 0 & 0 \\ \phi^2 I & 0 \end{bmatrix}.$$

We are now going to analyse the stability of (3.3.5a). The objective is to guarantee exponential stability with a guaranteed decay rate. The following theorem holds.

Theorem 3.3. *(Rodrigues de Campos et al. [231]) Consider the proposed consensus algorithm (3.3.4) associated to a given Laplacian $\boldsymbol{L}$ representing a communication graph with a directed spanning tree, a given $\alpha > 0$, $\phi > 0$ and $T > 0$. Take:*

$$M_0 = \left[\begin{array}{cc} M_{DIP}(\phi) & M_{DIP}^*(\phi) \end{array} \right],$$

and M_1, M_2, M_{12} as defined in Theorem 3.1. Assume that there exist $P > 0$, $R > 0$, S_1 and $X \in \mathbb{S}^n$, $S_2 \in \mathbb{R}^{n \times n}$ and $N \in \mathbb{R}^{2n \times n}$ that satisfy conditions (3.2.18) of Theorem 3.1. Then, the consensus algorithm (3.3.4) with the parameter ϕ and the sampling period T is thus α_g- stable, where $\alpha_g = \min\{\alpha, -\log(\cos(\delta T))\}$. Moreover, the consensus equilibrium is given by:

$$x(\infty) = U_2 \left(x(0) + \gamma_{\phi T} \dot{x}(0) \right). \tag{3.3.14}$$

Proof. The proof follows directly from Theorem 3.1, by taking into consideration now $M_0 = \left[\begin{array}{cc} M_{DIP}(\phi) & M_{DIP}^*(\phi) \end{array} \right]$, where $M_{DIP}(\phi)$ and $M_{DIP}^*(\phi)$ are defined in (3.3.13). Thus, (3.3.5a) and (3.3.5b) are exponentially stable with a respectively decay rate α and $-log|\cos(\delta T)|$, respectively. The global algorithm (3.3.5) is then exponentially stable with the decay rate α_g. Moreover, according to Lemma 3.4, and since $z_1 \to 0$ and $z_2 \to z_2(\infty)$, we conclude that the consensus algorithm is exponentially stable with the consensus equilibrium being:

$$x(\infty) = W z_2(\infty) = [W_1 \overrightarrow{1}] \left[\begin{array}{c} 0_{n-1\times 1} \\ z_2(\infty) \end{array} \right] = z_2(\infty) \overrightarrow{1}.$$

The proof is concluded by noting that $z_2 = U_2 x$ and that $z_2(\infty)$ is expressed in (3.3.7).

□

Note that previous calculations state stability conditions for the continuous system (3.3.4) in terms of Linear Matrix Inequality (LMI). However, Theorem (3.3) is based on the discrete-time Lyaponov theorem. Moreover, it is worth mention that the stability conditions proposed in this thesis are sufficient but not necessary conditions, for both simple and double integrator dynamics.

3.4 Simulation results

In the framework of the multi-robot swarms, and in particulary of the European Project FeedNetBack, cooperative control of a fleet of vehicles under varying-topology and communications constraints reveals challenging problems. For simulation purposes we have considered in this thesis three graphs depicted in Fig. 3.7.

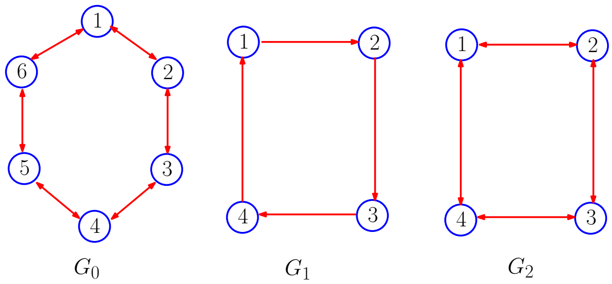


Figure 3.7: Corresponding graphs of the matrices L_0, L_1 and L_2.

To each graph is associated a Laplacian matrix given by:

$$L_0 = \begin{bmatrix} -1 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0.5 & -1 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & -1 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -1 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & -1 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0.5 & -1 \end{bmatrix}, L_1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \ L_2 = \begin{bmatrix} -1 & 0.5 & 0 & 0.5 \\ 0.5 & -1 & 0.5 & 0 \\ 0 & 0.5 & -1 & 0.5 \\ 0.5 & 0 & 0.5 & -1 \end{bmatrix}.$$

In the sequel, the initial positions over the x-axis for a four and a six agent network are denoted $x^{\ 4\ agents}$ and $x^{\ 6\ agents}$, respectively. For simulations purposes, they are defined as:

$$x^{\ 4\ agents}(0) = [30,\ 25,\ 15,\ 0]^T, \quad x^{\ 6\ agents}(0) = [30,\ 25,\ 15,\ 0,\ -10,\ -30]^T.$$

Moreover, to deal with consensus algorithms for double integrator dynamics, the initial velocity conditions of a set of four agents are defined by:

$$\dot{x}^{\ 4\ agents}(0) = [1,\ 2,\ 3,\ 2]^T.$$

Assume for the moment that agents obey simple integrator dynamics. Consider a set of four agents controlled by (3.2.5) connected through the undirected and directed graphs G_0 and G_1 respectively, see Figure 3.7. Those two graphs are balanced, which implies that consensus equilibrium value will be defined as the average of initial conditions presented just before.

Figure 3.8 shows the optimization results of the decay rate α satisfying Theorem 3.1 for graph G_0. Identical results considering the communication graph represented by L_1 are presented in Figure 3.9. Figures 3.8 and 3.9 show a 3-D representation of α stability results, enhancing the maximum convergence rate that guarantees the convergence of algorithm (3.2.5). We can identify a crest for specific values of (δ, T). These crests mean an improved behavior and the best positive value of α is obtained when $(\delta, T) = (1.96, 0.29)$ and $(\delta, T) = (1.96, 0.09)$, for graph G_0 and graph G_1 respectively. We have considered plausible sets of values for the different parameters such that $T \in [0, 1]s$ and $\delta \in [0, 2]$.

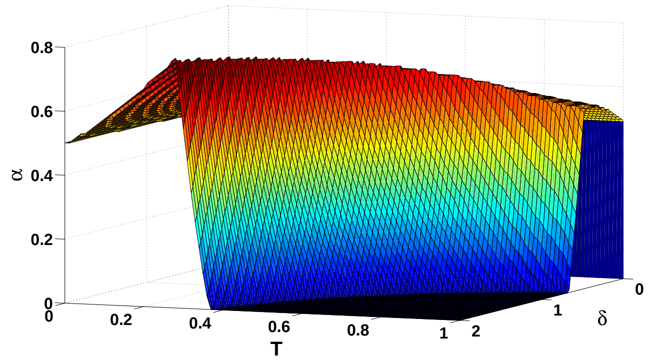


Figure 3.8: Convergence rate of the consensus algorithm (3.2.5) connected through graph G_0, for different values of (δ, T).

Figure 3.10 shows simulations from the classical algorithm (3.2.2) as well as the algorithm (3.2.5) considering G_0 and for several values of δ and T. Equivalent simulation results for G_1 are presented in Figure 3.11. The objective is to compare system performances with two different approaches and justify the interest of the proposed algorithm. Figures 3.10(a),3.11(a) show the evolution of the classical consensus algorithm. Figures 3.10(b),3.11(b) show simulation results using the optimal pair (δ, T) according to Theorem 3.1 and recovered on Figures 3.8, 3.9, respectively. We can see that they correspond to a faster algorithm when compared with the trivial algorithm. In Figures 3.10(c),3.11(c), we kept the optimal value of T and changed δ value. Finally, for Figures 3.10(d),3.11(d), we kept the optimal value of δ and changed T value. Since the balance properties of the graphs are not changed by the proposed approach, it is possible to observe that the modified algorithm keeps averaging properties.

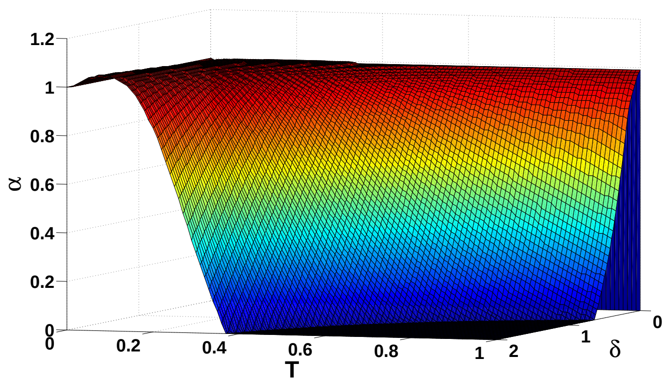


Figure 3.9: Convergence rate of the consensus algorithm (3.2.5) connected through graph G_1, for different values of (δ, T).

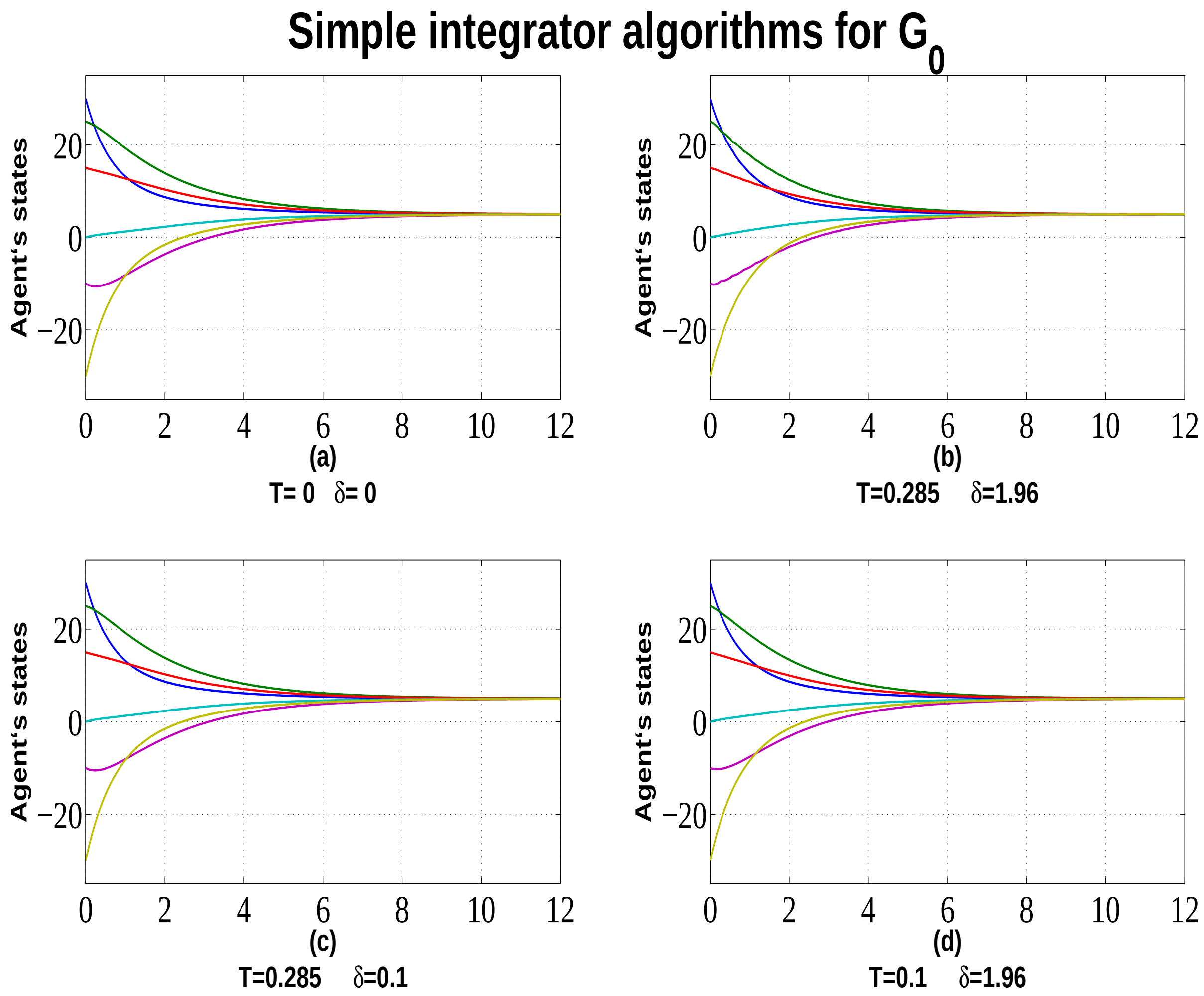


Figure 3.10: Simulation results for a set of six agents controlled by (3.2.5) and connected through graph G_0, for different values of (δ, T).

Take now a set of set of six agents controlled by (3.2.22) connected through the undirected graph G_0, shown in Figure 3.7. For the sake of brevity, the optimisation results of the controller's parameters for graph G_1 will be omitted in the sequel. However, the efficiency of (3.2.22) has also been confirmed for a set of four agents connected through G_1. Figure 3.12 show a 3-D representation of α stability results for graph G_0. In all figures, it's possible to identify a crest, which means an improved behavior, for specific values of (δ, σ, T), the control parameters. Therefore, taking all these results into consideration it is possible to compute the best positive value of α, obtained for $(\delta, \sigma, T) = (0.5, 6, 0.3)$.

Define $\varepsilon_{SI} = |x(t) - x_\infty|$ as the module of the error between agents states and the agreement value x_∞. Figure 3.13 shows the error ε_{SI} evolution for graph G_0 while

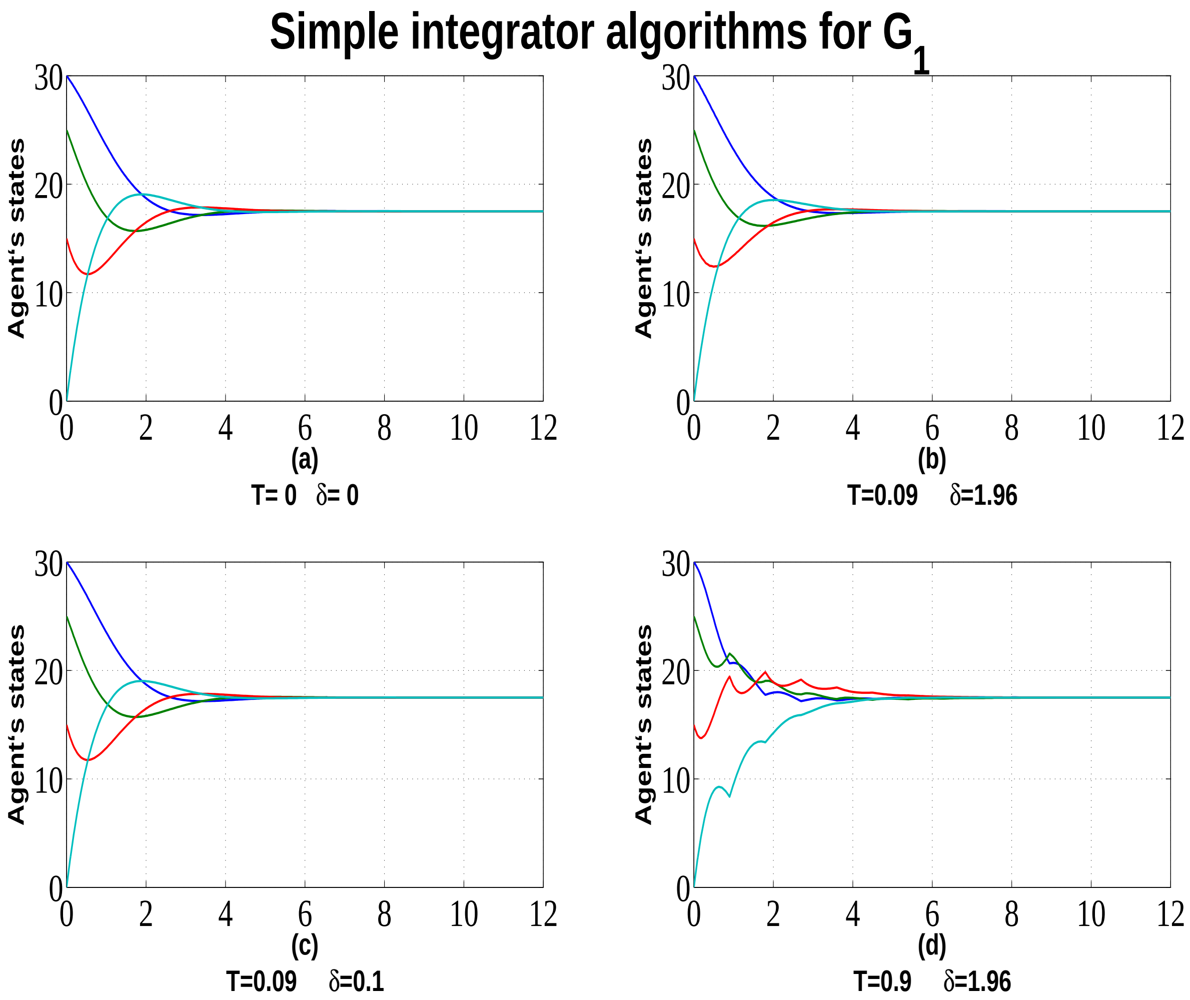
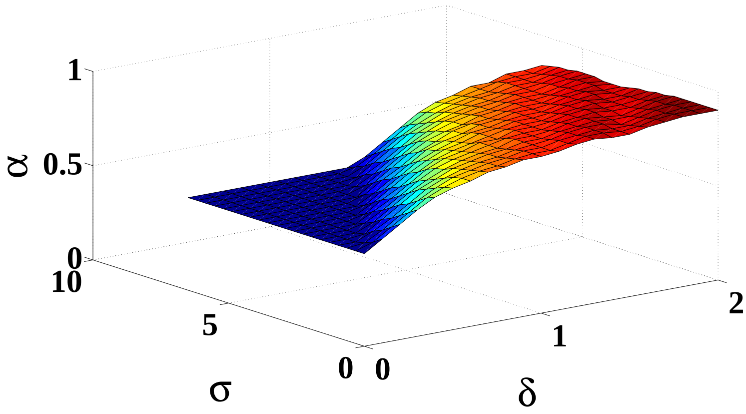
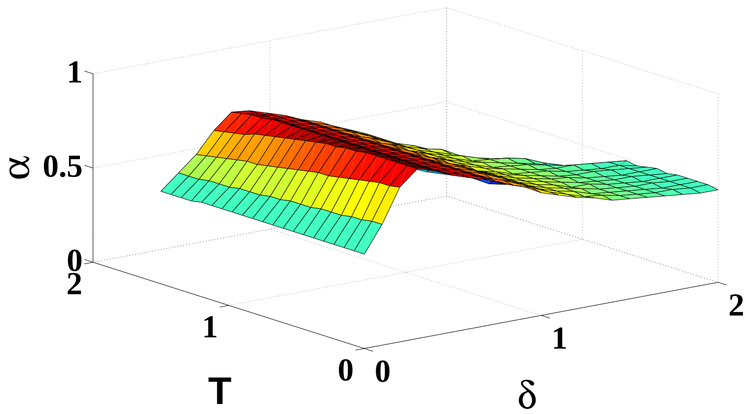


Figure 3.11: Simulation results for a set of four agents controlled by (3.2.5) and connected through graph G_1, for different values of (δ, T).

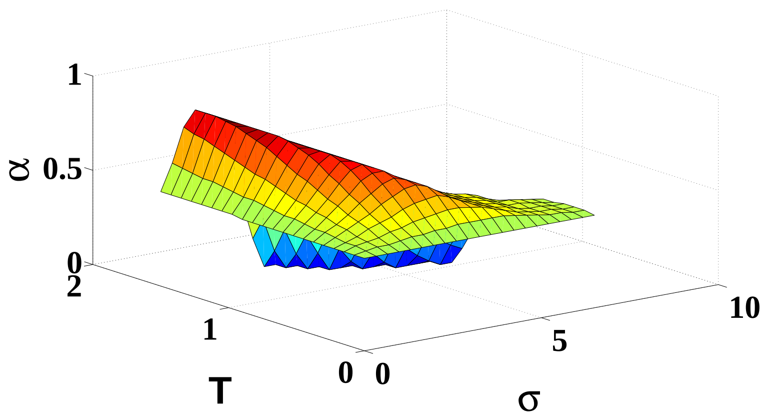
Figure 3.14 considers the evolution of the previously defined error ε_{SI} for graph G_1. Classical algorithm's performances correspond to the continuous line as the green dotted line shows the behavior of the improved algorithm (3.2.5). Note that the red dotted line corresponds to the evolution of algorithm (3.2.22). We can clearly observe that algorithm (3.2.5) converge more rapidly than the trivial SI consensus. Furthermore, we can also see that global memory improves both the trivial performances as well as those of protocol (3.2.5). Moreover, we can see that as $t \to \infty$, ε tends to zero for both setups. These results consider the best values of (δ, T) for algorithm (3.2.5) and the optimal values (δ, σ, T) for (3.2.22). They enhance the technical developments of this chapter concerning the improvement of consensus protocols convergence. Indeed, they seem to present better performances with respect to trivial consensus algorithms available in literature.



(a) for several values of (δ, σ)



(b) for several values of (δ, T)



(c) for several values of (σ, T)

Figure 3.12: Convergence rate of the consensus algorithm (3.2.22) for the communication graph G_0, for different values of (δ, σ, T).

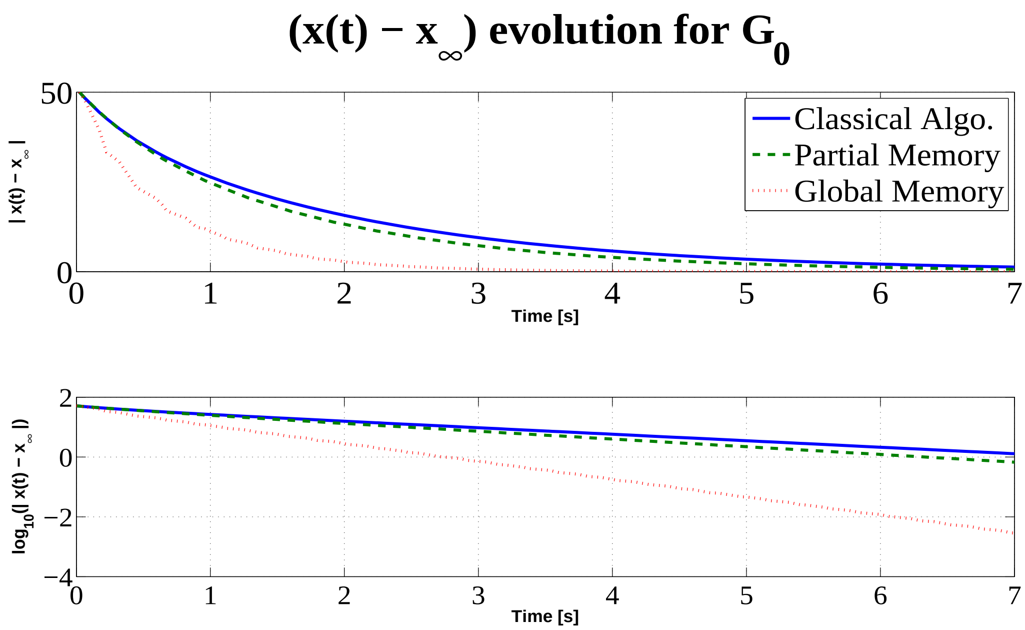


Figure 3.13: Time evolution of error ε_{SI} and $log_{10}(\varepsilon_{SI})$ for a set of six agents: performances comparison between (3.2.5) and (3.2.22) under a communication graph G_0.

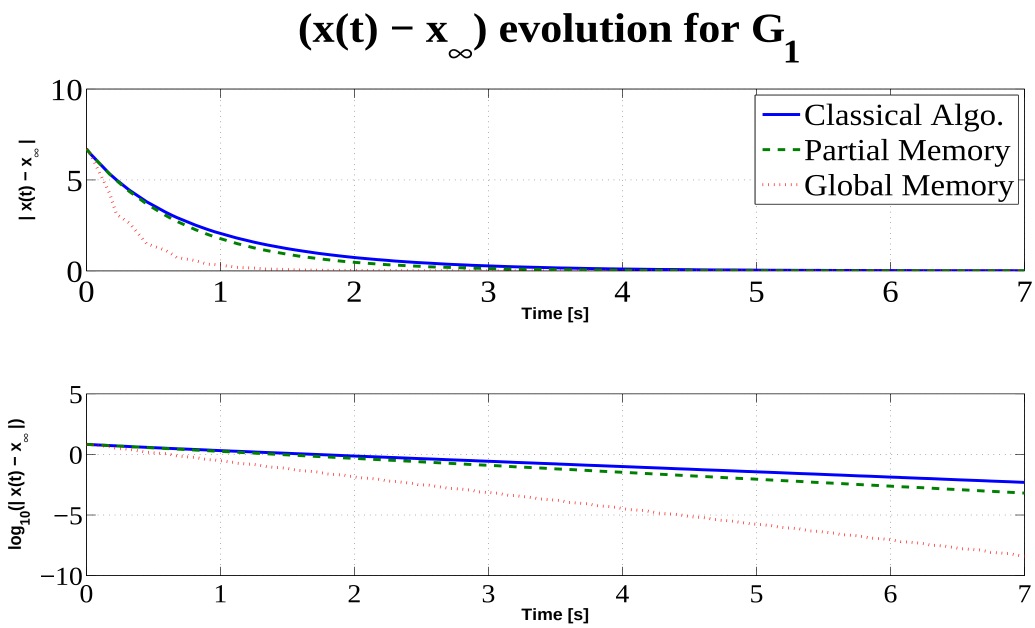
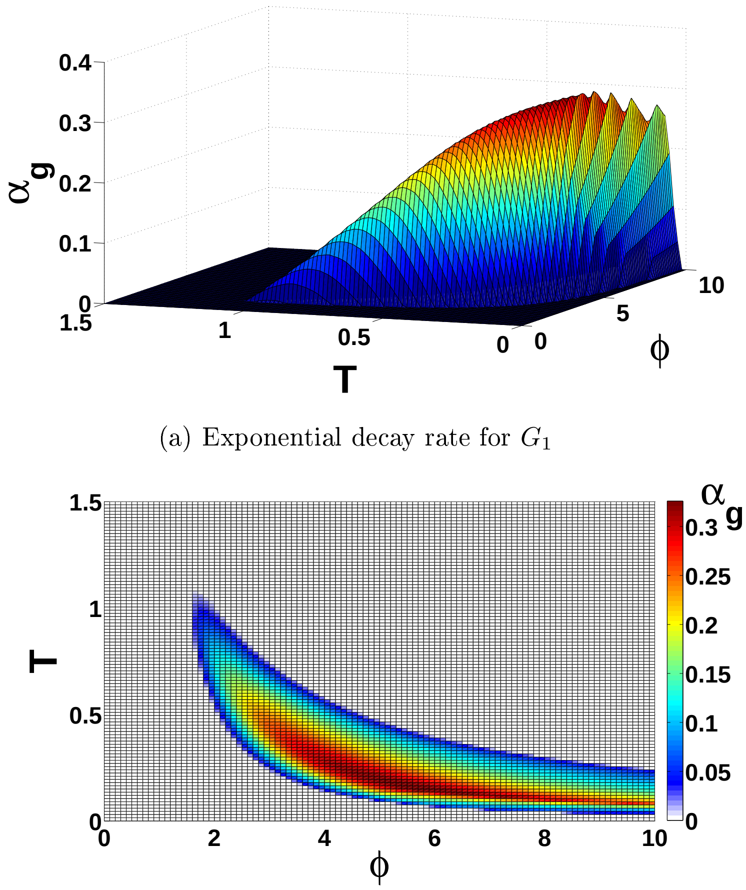


Figure 3.14: Time evolution of error ε_{SI} and $log_{10}(\varepsilon_{SI})$ for a set of four agents: performances comparison between (3.2.5) and (3.2.22) under a communication graph G_1.



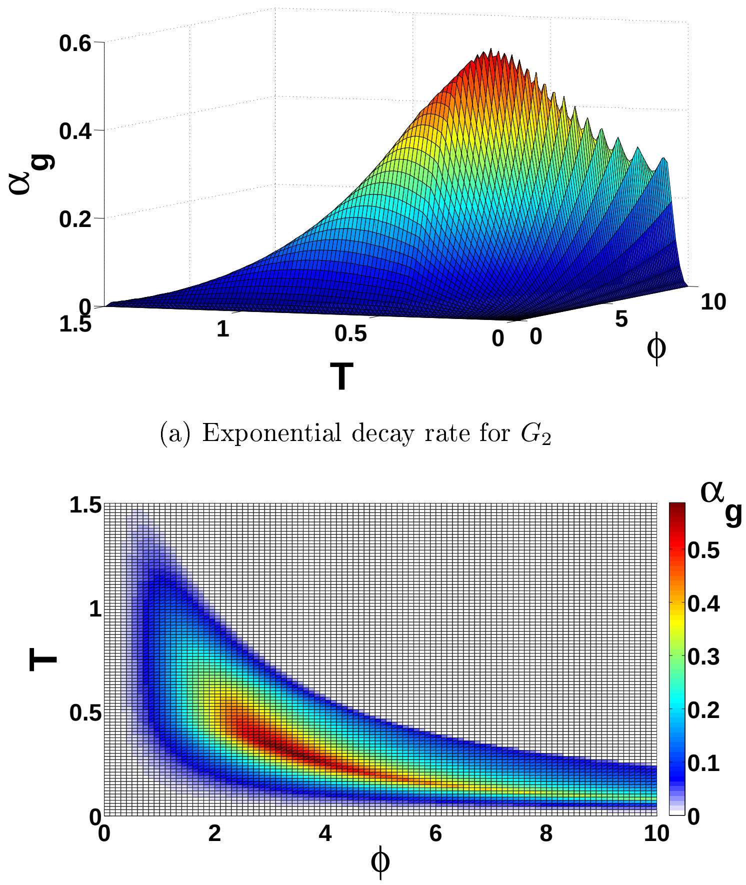
(a) Exponential decay rate for G_1

(b) Exponential decay rate for G_1 (Top view)

Figure 3.15: Convergence rate of the consensus algorithm (3.3.4) for several values of (ϕ, T) and for the communication graphs G_1.

Assume now that agents obey to a double integrator dynamics. Consider a set of four agents controlled by (3.3.4) connected through the undirected and directed graphs G_1 and G_2, shown in Figure 3.7. These two graphs are balanced, which implies that consensus equilibrium value will be dependent on the average of the initial conditions and on the values of ϕ and T.

Figures 3.15(a) and 3.16(a) are a 3-D representation of α_g stability results for graph G_1 and G_2, respectively. Figure 3.15(b),3.16(b) show a top-view from the previous figures allowing us to observe the distributions of α_g values. In these figures, we observe a region where $\alpha_g > 0$ that corresponds to the stability region of (3.3.4). The best positive value of α_g is obtained when (ϕ,T)=(6.4;0.15) and (ϕ,T)=(3;0.3), for graph G_1 and graph G_2, respectively. The fact thatthe systems' stability is not guaranteed by Theorem 3.3 does not necessary mean that the algorithm is unstable: in Figure 3.18(c), the algorithm is stable even though under our conditions we consider it unstable.



(a) Exponential decay rate for G_2

(b) Exponential decay rate for G_2 (Top view)

Figure 3.16: Convergence rate of the consensus algorithm (3.3.4) for several values of (ϕ, T) and for the communication graphs G_2.

Figure 3.17 shows simulations of algorithm (3.3.4) considering G_1 different values of ϕ and T. Simulation results of the same systems considering G_2 are presented in Figure 3.18. If $T = 0$, this algorithm is unstable for a directed and undirected graph Figures 3.17(a), 3.18(a). Moreover, an oscillating behavior is observed for the undirected graph, see Figure 3.18(a). Figures 3.17(b), 3.18(b) show simulation results using the optimal pair (ϕ,T) according to Theorem 3.3 and Figures 3.15 and 3.16, respectively. We can see that they correspond to the fastest algorithm presented.

Figure 3.18(c) shows a stable response but with a greater convergence rate with respect to Figures 3.18(b). In Figures 3.17(c),3.18(d) we can find the particular case where $T = \pi/\phi$, which does not fulfill assumptions of Theorem 3.3. Instability can be seen with oscillations around the consensus algorithm final value. This behavior can be explained as follows: z_1 dynamics, defining the global system behavior, eventually converge. However, z_2 do not converge to a stable agreement value, see Lemma 3.5.

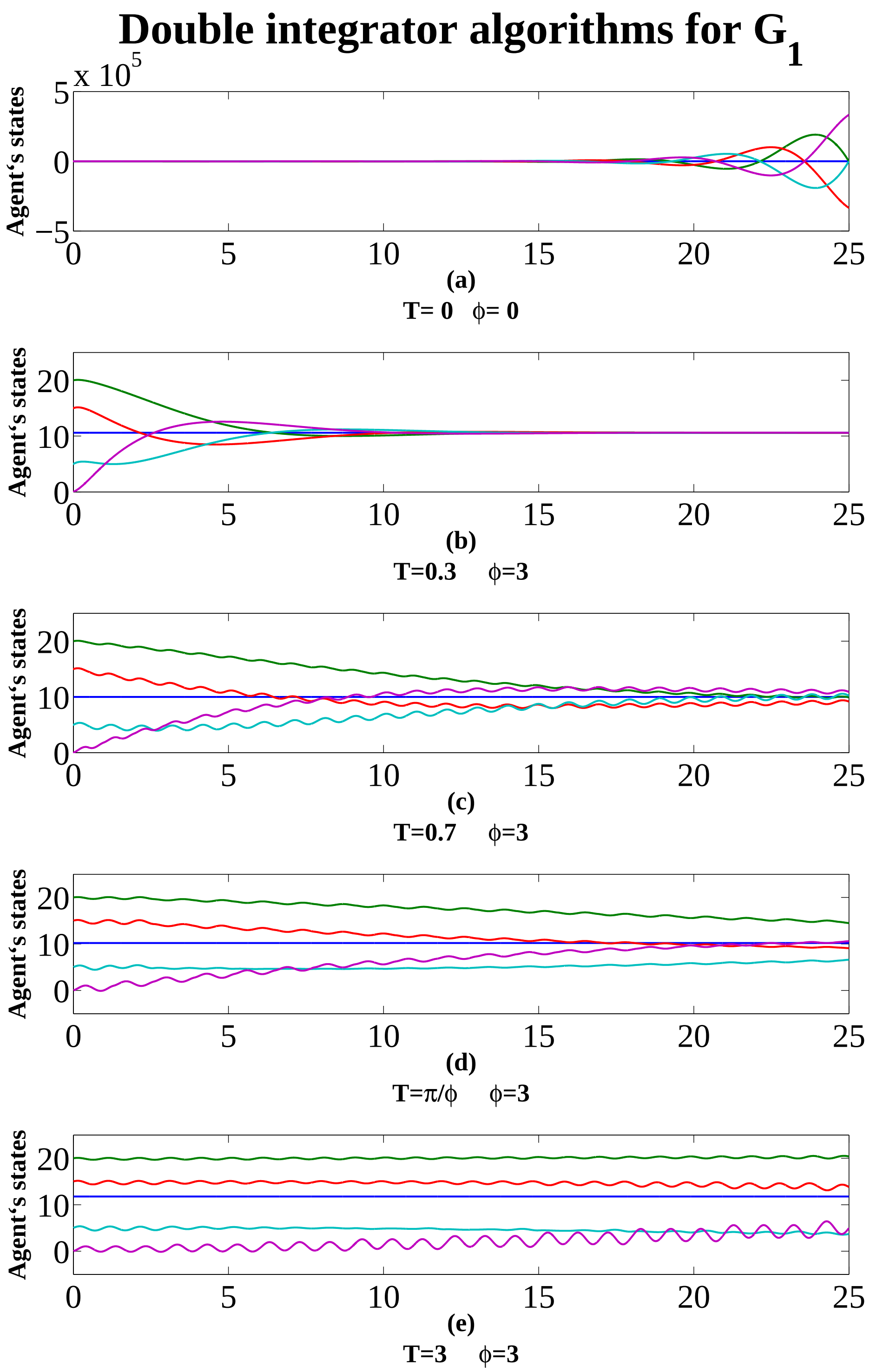


Figure 3.17: Simulation results for a set of four agents controlled by (3.3.4) and connected through graph G_1, for different values of (ϕ, T).

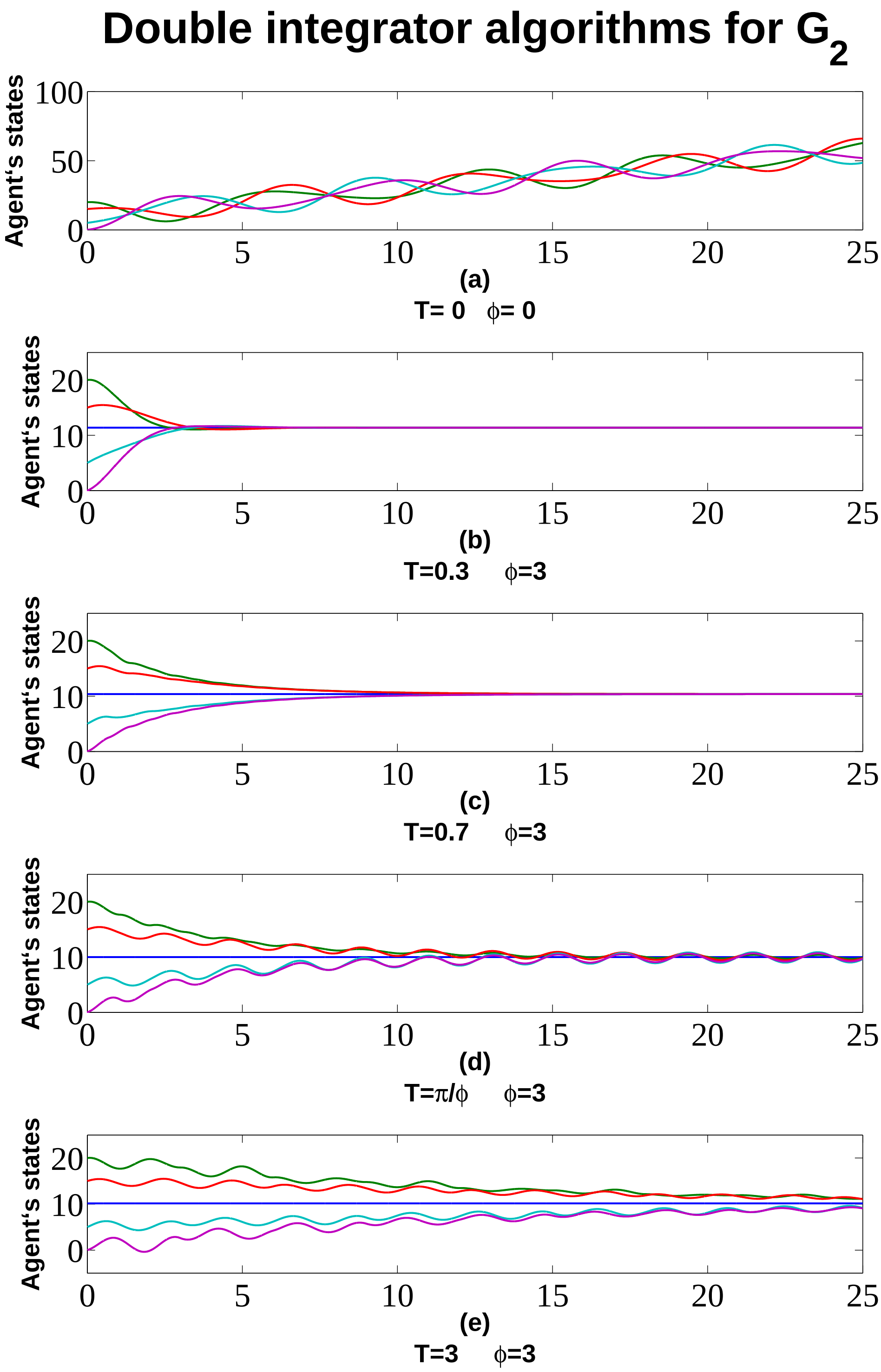


Figure 3.18: Simulation results for a set of four agents controlled by (3.3.4) and connected through graph G_2, for different values of (ϕ, T).

Finally, Figures 3.17(d-e),3.18(e) consider greater values of T. It is easy to conclude that algorithm's performances decrease and eventually become unstable. As for the SI case, optimality is obtained for a certain value of T. Consequently, small changes on the value of T lead to weaker performances.

In Table 3.1 we can find $\gamma_{\phi T}$ and $x(\infty)$ values for different ϕ and T. For the particular case where $T = \pi/\phi$ we can observe that consensus final value corresponds to the average of position's initial conditions ($\gamma_{\phi T}$=0), but no stability is achieved under the derived criteria. Note that the optimal behavior does not correspond to the point where we find the smallest erreur ε, but where decay rate α_g is the greatest.

	Graph 1					Graph 2				
	$\phi = 6,4$					$\phi = 3$				
T	0	0,15	π/ϕ	1,3	3	0	0,3	0,7	π/ϕ	3
$\gamma_{\phi T}$	48,82	0,30	0	0,09	0,88	222,22	0,69	0,19	0	0,07
$x(\infty)$	107,65	10,60	10	10,19	11,76	454,44	11,38	10,38	10	10,14

Table 3.1: Data overview of algorithm (3.3.4) for communication graphs G_1 and G_2.

The previously presented simulations complete the theoretical results on the convergence rate of the proposed control strategy. As mentioned before in this dissertation, consensus algorithms are powerful tools to deal with several motion coordination problems for robotic systems. Among many orders, a lot of attention has been paid to rendezvous applications, *i.e.*, where all the agents converge to the same location. In order to fit in this problem, in the sequel agents are supposed to move through a cartesian plane such that the system's configuration is represented by:

$$q = [q_i, q_j, \ldots, q_N]^T,$$

where $q_i = [x_i, y_i]^T \in \mathbb{R}^2, i \in \mathcal{N} = \{1, ..., N\}$. Throughout the rest of this section, the global initial configuration vectors for a four agent and a six agent network are denoted $q^{\ 4\ agents}$ and $q^{\ 6\ agents}$, respectively. For simulations purposes, they are defined as:

$$\begin{aligned} q^{\ 4\ agents}(0) &= [30,\ 30,\ 25,\ 10,\ 15,\ 9,\ 0,\ 20]^T, \\ \dot{q}^{\ 4\ agents}(0) &= [1,\ -1,\ 2,\ 5.5,\ 3,\ -3,\ 5,\ 3]^T, \\ q^{\ 6\ agents}(0) &= [30,\ 11,\ 25,\ 4,\ 15,\ -6,\ -10,\ 13,\ -30,\ -10,\ 0,\ 5]^T. \end{aligned}$$

In order to complete this section, we present through Figures 3.19 [1] and 3.20 [2] simulation results of rendezvous for simple and double integrator dynamics, respectively.

1. For all figures, the green cross corresponds to the average of the initial positions.
2. For all figures, the green cross corresponds to the average of the initial positions and the red cross stands for the agreement value.

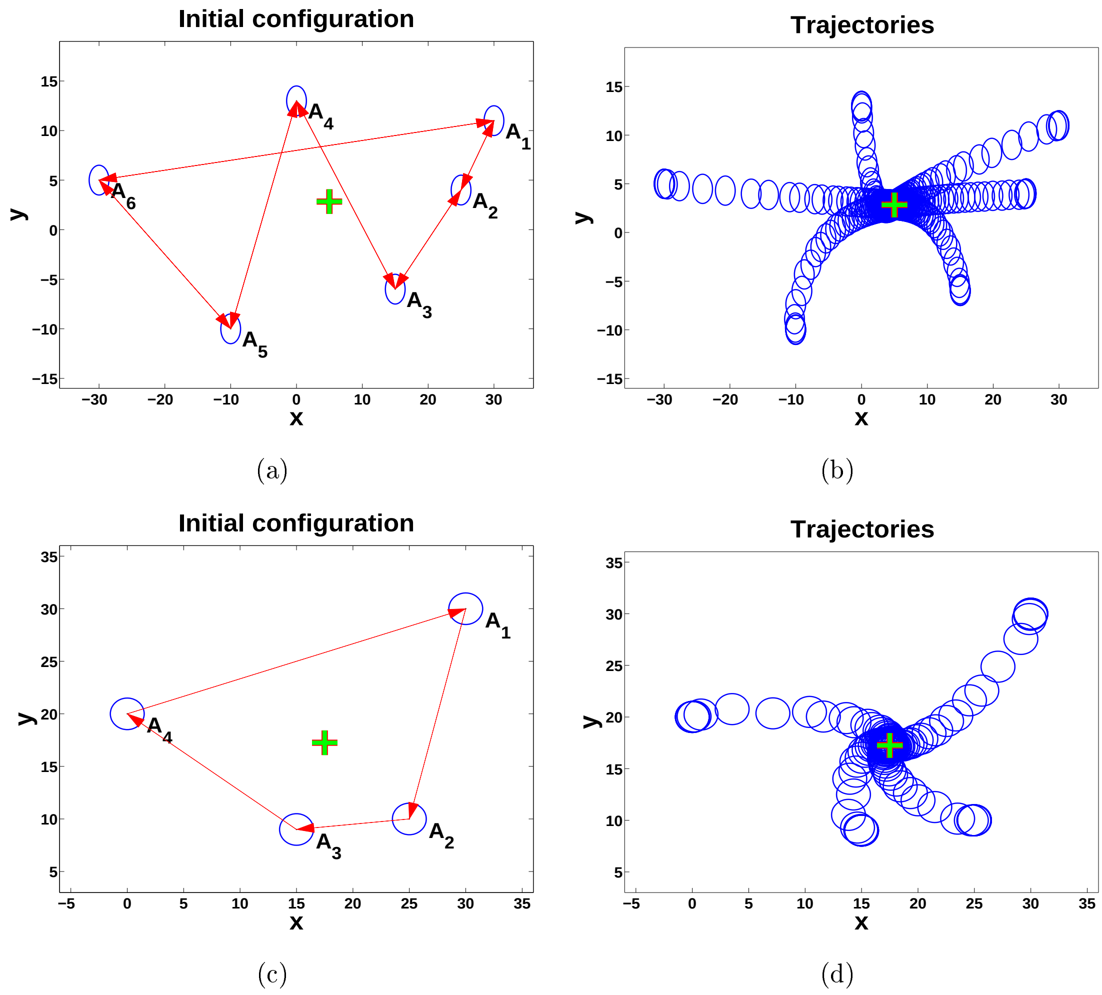


Figure 3.19: Simulation results of rendezvous algorithms for SI dynamics

Consider for the moment that agents obey to simple integrator dynamics and are controlled by (3.2.5). For a topology represented by the graph G_0, Figures 3.19(a) and 3.19(b) show the initial configuration and the trajectories of the swarm, respectively. On the other hand, Figures 3.19(c) and 3.19(d) show, respectively, the initial formation and the trajectories evolution for a set of agents connected through graph G_1. These simulations enhance the efficiency of rendezvous protocols using memory based control laws. Furthermore, they are complemented by the previous results on the convergence rate of such algorithms. It is also important to mention that these results consider the optimal values of δ and T for the respective graphs. We can observe that all agents meet at a same position which corresponds to the average of the swam initial positions. These conclusions join the theoretical results of this chapter regarding the averaging properties of the proposed strategy for simple integrator dynamics.

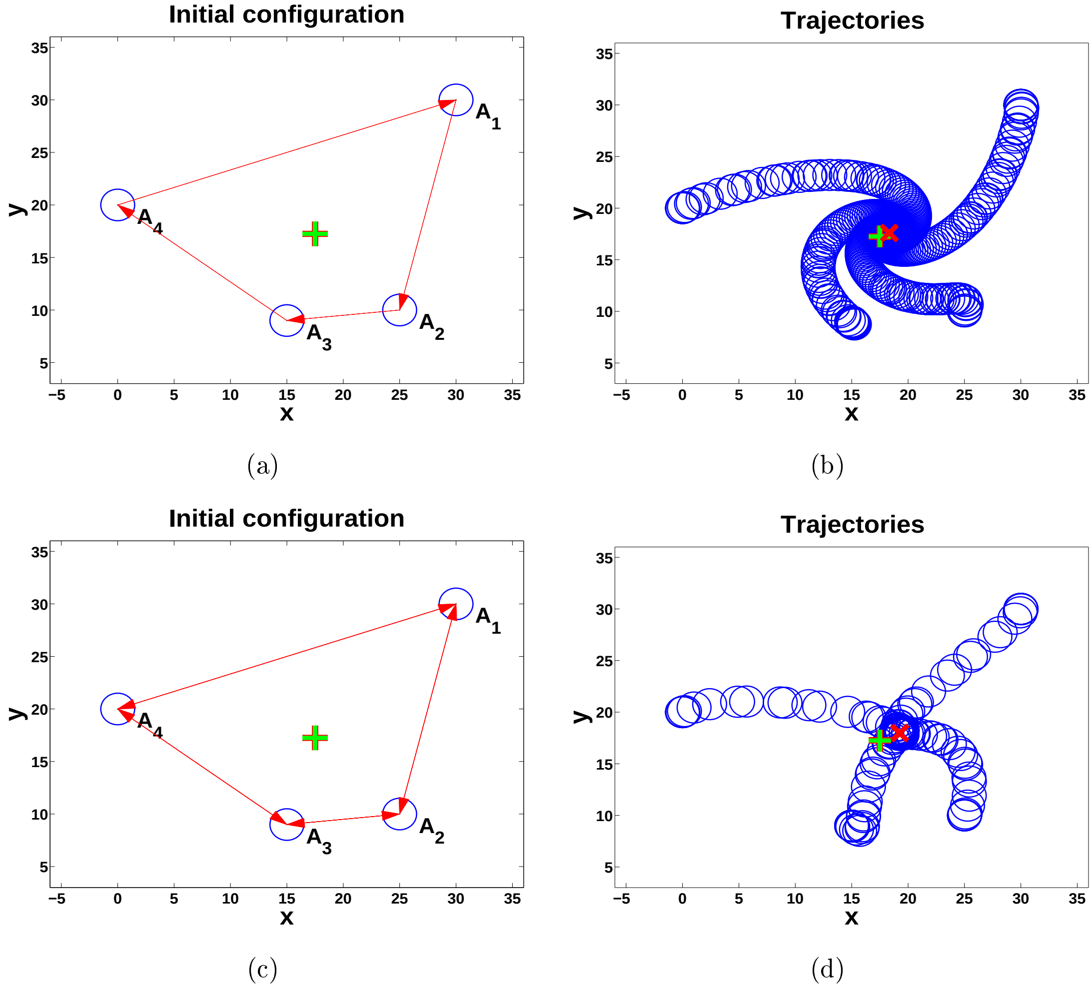


Figure 3.20: Simulation results of rendezvous algorithms for DI dynamics

Consider now a set of agents obeying to double integrator dynamics and controlled by (3.3.4). For a directed topology represented by G_1, the initial configuration and the system's trajectories are presented in Figures 3.20(a) and 3.20(b), respectively. Identically, Figures 3.20(c) and 3.20(d) show the initial formation and the resulting configuration, respectively, for a communication topology expressed by graph G_2. The same as before, these simulations enhance the efficiency of rendezvous protocols using memory. They are complemented by the previous results on the convergence rate the proposed controllers. Note that these results consider the optimal values of ϕ and T for the respective graphs. By analyzing Figures 3.20(b) and 3.20(d), one can see that agents eventually meet at a same position. Note that the agreement value is represented by the red cross. However, it is worth mentioning that the agreement value does not corresponds to the average of the initial positions. Analytically studied throughout Lemma 3.5, the equilibrium value of the proposed strategy is dependent on the average of the initial conditions and on the values of ϕ and T.

3.5 Conclusions

In this chapter, the incorporation of memory in consensus controllers for both simple and double integrator consensus dynamics has been studied. The objective was to propose and analyse an effective approach to accelerate the convergence of synchronous distributed averaging algorithms. In a first part, the influence of both local memory and global memory in consensus algorithms for simple integrator agents have been studied. We could conclude that using global memory drastically improves performances when compared to both trivial and partial memory based algorithms. An optimization method of controller parameters is proposed so that exponential stability of the solutions is achieved based on the discrete-time Lyaponov theorem and expressed in terms of LMI. Also, analytical conditions for improved performances based on Laplacian's eigenvalues have been provided. Simulation results show the efficiency of the proposed algorithm, as well as the conservation of averaging properties. In a second step, a new consensus algorithm (3.3.4) for double integrator agents was proposed. This chapter puts forward the technical advantages of such a protocol since it reduces information quantity needed for control: because there is no need for velocity sensors, there are economical, space and calculation savings. An optimisation of controller parameters is proposed so that exponential stability of the solutions is achieved. An expression of the consensus equilibrium is derived with respect to the initial position and velocity of the swarm.

The Laplacian and its spectral properties play a crucial role in convergence analysis of consensus and alignment algorithms. In order to answer the question if it is possible to increase the convergence speed of a consensus algorithms, this chapter presents our contribution to this topic by showing how memory can be incorporated in the controllers.

Chapter 4

Control strategies for multi-agent systems compact formations

Contents

4.1 Context

The deployment of large groups of autonomous vehicles is now possible because of technological advances in networking and in miniaturization of electromechanical systems. Indeed, groups of autonomous robots with computing, communication and mobility capabilities have become economically feasible and can perform a variety of spatially distributed sensing tasks such as search and recovery operations, manipulation in hazardous environments, exploration, surveillance, environmental monitoring for pollution detection and estimation. etc. The most important and commonly studied applications in the field of cooperative control were presented in Chapter 1. In the scope of agreement strategies for Multi-Robot Systems (MRS), we focused throughout the previous chapters on consensus algorithms of heterogeneous agents, representing, for example, different models or generations of robots, since consensus is a useful tool for several applications [62, 74, 308]. Motivated by the fact that only a few works consider heterogeneous cases of the synchronization problem, we proposed a control strategy based on consensus algorithms which is decoupled from the original system. We also designed efficient strategies that accelerate consensus algorithms' convergence rate, using the stabilizing delay concept and by incorporating memory into the distributed averaging algorithm. However, a common assumption to all previous developments is that agents are connected through a fixed communication topologie. Even if such a setup is realistic for some types of applications, a key feature of multi-vehicle groups is that communication between moving agents has several dynamic properties. In particular, data rates may be low (either by environment or by design), dropouts may occur or agents might not be able to communicate with

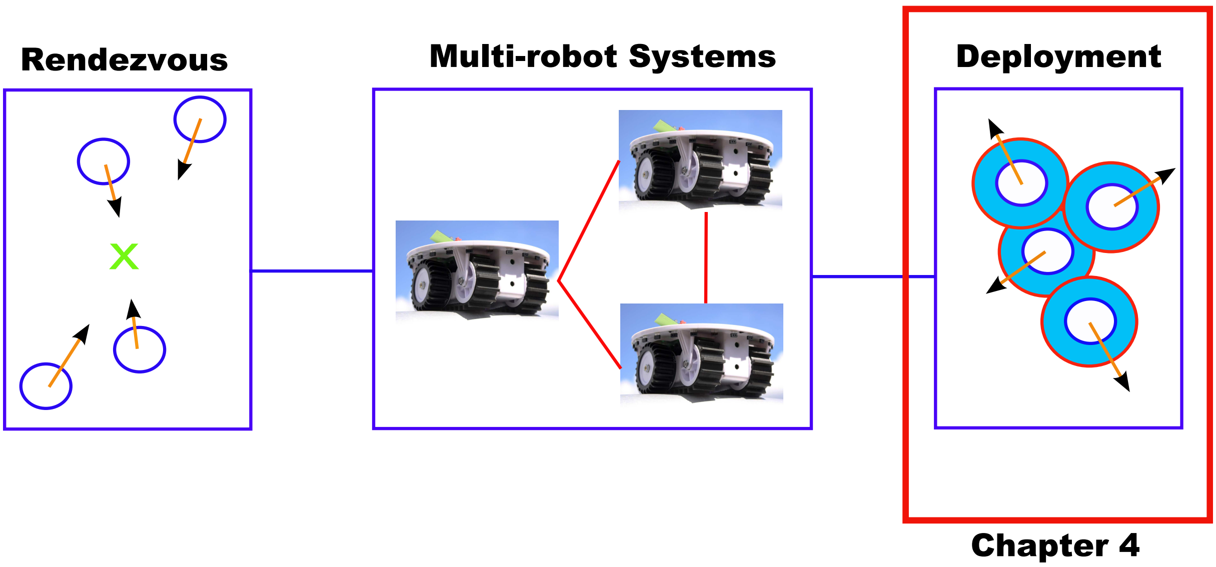


Figure 4.1: Context of Chapter 4

all nodes. In the sequel we consider this last setup. More precisely, we consider that during a coordinated motion or a collaborative task, the interconnections between the agents can evolve such that new communication links are created and others are broken. This setup has important repercussions from both a theoretical and practical point of view. In order to deal with such challenging cases, agents might be coupled by simple rules including nearest-neighbor or range-based neighborhoods [31, 66, 71, 79, 78, 194, 206, 240]. Flocking motions based on nearest-neighbor interaction rules have been developed for double-integrator dynamics of the agents [191, 279, 280] and rendezvous algorithms for Multi-Agent Systems (MAS) with limited sensing ranges studied in [87, 148, 150, 151]. Furthermore, and extending the results of [163], multi-agent flocking with random communication radius was recently studied in [162]. Considering that the communication network is determined by a metric rule based on a random interaction range, authors determine practical conditions (on the initial positions and velocities of agents) ensuring that the agents eventually agree with some probability on a common velocity, *i.e.*, flocking.

Recently achievements have shown that teams of mobile autonomous agents need not only agree on some quantity of interest, but also to have the ability to deploy over a region, assuming a specified pattern, or move in a synchronized manner. While previous chapters considered the first challenge, offering alternative and innovative agreement strategies to deal with commonly used MAS applications such as rendezvous, in this chapter we are mainly interested in motion and formation control, since to accomplish exploration, surveillance or rescue tasks, a recurrent choice is to coordinate the agents to form a particular configuration that satisfy certain local (*e.g.* node degree) and/or global (*e.g.* network connectivity) constraints, see Figure 4.1.

Widely spread in Nature, motion coordination is a useful tool for groups of vehicles, mobile sensors, and embedded robotic systems. Achieving a coordination task corresponds to moving the agents and changing their state to maximize or minimize a specific objective function. For instance, algorithms for placing individual network nodes with limited communication range into the environment, while guaranteeing certain communication properties of the resulting node ensemble have been proposed in [62, 63]. Under a bounded tracking error assumption, in [48, 85] the authors show that the formation error is stabilized when applied to rigid body constrained motions. In [208] control laws are based on probabilistic node degree constraints, whereas in [24, 25], the authors considered direction-based algorithms. There is a huge number of contributions to this topic. Even though several other approaches exist in literature, they are not going to be detailed in this dissertation. However, we present in the sequel some pertinent problems and related references. Illustrations of these behaviors are presented in Figure 4.2.

Leader-follower approaches

In this approach one agent is designated as being the leader and the mission of the rest of vehicles (followers) is to maintain a desired distance from the leader. Hence, the followers receive information from the leader in order to keep the desired formation. In [72], a controller is designed using input/output feedback linearization and an application of this strategy can be found in [277]. Another strategy to produce formation motions, *i.e.*, flight formations, is the virtual-leader approach [213] where a suitable inter-distance (and orientation) is set between agents. The motion of the formation results from the motion of the leader. Several extensions to multiple non-holonomic mobile robots are presented in [45, 59, 60, 70, 84].

Virtual structure based formation control

This method is developed to enforce a group of agents to stay in a rigid formation. The controller of each agent is designed to track the dynamics defined for the virtual structure. It means that, for a desired formation, the control laws designed minimize the error between the desired positions in the virtual structure and the real position of the agents [15, 16, 147, 278].

Coverage

The coverage problem [8, 10, 11, 12, 206, 226, 243], is the maximization of the total area covered by a robot's sensors. Static coverage is the problem of deploying robots in a static configuration, such that every point in the environment is under the robots' sensor shadow, *i.e.* covered, at every instant of time. Clearly, for complete static coverage of an environment the robot group should be larger than a critical size (depending on environment size, complexity, and robot sensor ranges), but determining the critical number is difficult or impossible if the environment is unknown a priori. On the other hand, the network deployment problem [62, 63] is the placement of individual network nodes, each with limited communication range, into the environment by a process that guarantees certain communication properties of the resulting node ensemble. Moreover, the deployed network can be used for tasks other than coverage.

Cyclic pursuit

Cyclic pursuit means that each agent i pursues agent $i+1$ modulo N, then each agent is required to sense information from only one other agent. Based on the notion of cyclic pursuit from mathematics, [161] proposes a collaborative strategy for MAS circular formations, and [134] presents a cooperative control of a multi-agent system to achieve a target-capturing task in 3-D space, among several other works [36, 256].

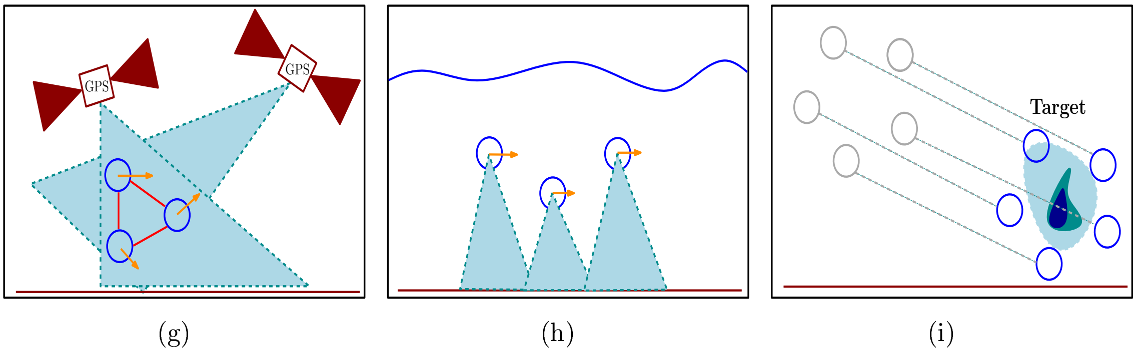


Figure 4.2: Illustration of several motion control approaches: (a) Leader-follower approach; (b) Virtual structure based formation control; (c) Coverage; (d) Cyclic pursuit; (e) Motion planning; (f) Coordinated trajectory tracking; (g) Generalized coordinates based formation control; (h) Exploration task; (i) Target tracking problem.

Motion planning

A typical motion planning system for coordinated motion of multiple robots must essentially consider various aspects ranging from modeling of robots and the environment to the generation of an optimal trajectory for achieving any specific task with multiple robots [71, 93, 143, 270]. The conventional motion planning systems consider all these aspects in a sequential fashion by devising strategies dependent on **(i)** the geometry used for robot and environment modeling, **(ii)** schemes used for collision avoidance [185, 49] and **(iii)** strategies adopted for coordination [19, 95, 292].

Coordinated trajectory tracking
: Coordinated path following is a control strategy where multiple vehicles are required to follow pre-specified spatial paths while keeping a desired inter-vehicle formation pattern in time [104, 136, 137, 164, 291].

Generalized coordinates based formation control
: In this strategy, the agent's position, its orientation and the shape with respect to a reference point in the formation are defined by the generalized coordinates. These coordinates can be used to specify the formation trajectories. See [108, 263].

Exploration task
: The purpose of exploration is to collect information by searching or traveling around an area of interest. Mobile sensor networks are often used in environmental applications such as ocean sampling, mapping and space exploration, see [67, 80, 145, 287, 315].

Target tracking problem
: The target tracking problem can be accomplished by a group of mobile vehicles or sensors. In this case, the objective for the agents is to locate and follow the trajectory of a moving target, as for source-seeking problems [56, 57, 41, 172]. There exist many different approaches to deal with this topic in the literature, the reader can refer to [164] and [291], and the references therein.

Recent robotics applications have also shown how interesting it is to impose a particular geometrical configuration. In fact, the geometry and symmetric proprieties of the desired configuration are directly related to control design for motion coordination [238]. Among many others, geometric formation control was tackled in [204, 200], and in particular circular formations were studied for instance in [31, 35, 136, 142, 145, 198, 247, 248]. The reader should refer to [29, 48] for a survey in different strategies dealing with formation control. Moreover, cohesiveness, which is characterized by a repulsion/attraction function which makes the agents in the network maintain desired relative distances between its neighbors was also studied in [191, 279]. In particular, formation control via behavior-based approach has been studied in [7, 43], and other works combined the behavior-based approach with potential fields, see, *e.g.*, [81, 103, 170, 215]. Furthermore, some techniques based on cooperative strategies [77, 166, 191] have already been used in order to guarantee that vehicles do not impact each other.

Sharing some of the motivations of previously mentioned works, this chapter addresses the design and analysis of an algorithm for *compact agent deployment*. Three problems will be considered:

i) How to improve the coverage rate for a given workspace;

ii) How to guarantee that two agents remain connected;

iii) How to improve connectivity properties for a given initial network.

The first contribution corresponds to an extension of [76] for swarms dispersion, by adding a connectivity maintenance force. Each agent is equipped with potential functions that will, simultaneously, isolate it from any other agent and impose connectivity-maintenance, using only information of those located within its sensing zone (a circular area around each agent and common for all nodes). We can find in literature several applications of this type of algorithms including coverage control and optimal placement of a multi-robot team in small areas [63, 90]. Though bearing-based algorithms were specifically applied to a triangular formation in [9], where the desired formation is specified entirely by the internal triangle angles, the second contribution of the approach presented in the sequel consists of a completely distributed algorithm that allows large scale swarm self-organization. In particular, this approach considers a direct angle control using only relative positions. To the best of our knowledge, the design of a control law capable of establishing a specific formation acting on inter-agent angles has not been addressed so far.

Motivated by the fact that for successful network operations, the deployment should result in configurations that not only provide good environment coverage but also satisfy certain local and/or global constraints, we intend to minimize inter-agent angles in order to achieve the most compact configuration possibilities, see Figure 4.3. Note that the minimization of inter-agent angles can also be seen as a maximization or maintenance of node degree (the number of neighbors of each node in the network). Our interest is strengthened by the fact that in several applications of sensor networks, the node degree plays an important role. For instance, several localization algorithms require a certain minimum degree for the nodes [208]. Moreover, in other applications, a high degree is required for the sake of redundancy.

Summarizing, we will try to provide a solution for the deployment of agents in order to achieve the best coverage rate possible, while keeping or improving connectivity properties. This work took two main constraints into consideration: homogeneous linear systems, for the sake of simplicity, and time-varying communication graph assuming

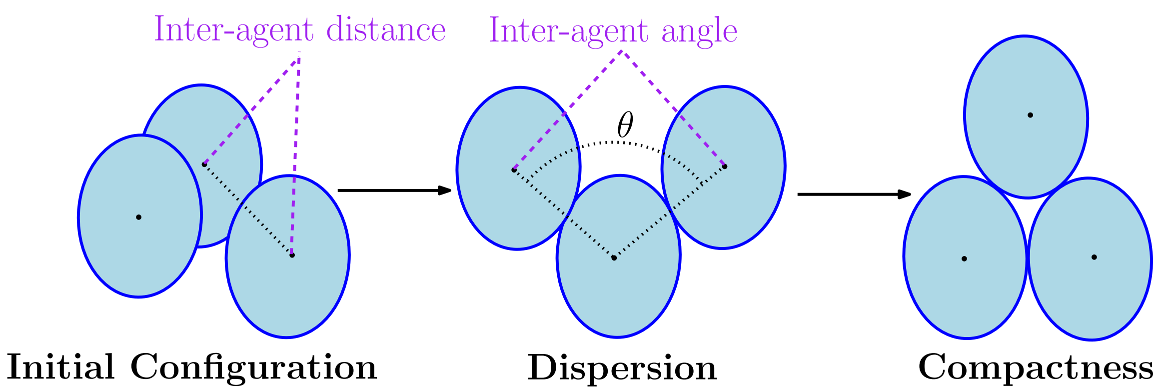


Figure 4.3: Objectives of Chapter 4

that each agent has a limited sensing range (and therefore the graph is dependent on the agents' motion). Two independent problems will be treated separately: dispersion and compactness. Individual stability analysis for these two strategies will be provided, but we will also propose a sequential control strategy gathering the two components. Theoretical arguments and calculations supporting the fact that such a system corresponds to a hybrid system will also be discussed. The control strategy proposed in this chapter is adaptive, since it adapts to changing environments, sensing task and network topology. It is also distributed, in the sense that the behavior of each vehicle depends only on the location of agents it can sense at each time, offering advantages as scalability and/or robustness. We consider applications of network deployment where a global map or knowledge of the environment is unnecessary or unavailable. We also assume that no global positioning system such as Global Position System (GPS) is available and that each calculus is based on local frameworks.

This work was carried out in collaboration with the Kungliga Tekniska Högskolan (KTH), the Royal Institute of Technology of Stockholm, Sweden. Furthermore, it was also supported by a International Mobility Grant of Institut National Polytechnique de Grenoble (Grenoble INP).

4.2 Problem statement and preliminaries

In this section we provide the definitions and tools necessaries for a deep understanding of the technical achievement of this thesis.

4.2.1 System description

Consider N agents operating in the workspace $W \subset \mathbb{R}^2$. The motion of each agent is described by the single integrator dynamics:

$$\dot{q}_i = u_i, \quad i \in \mathcal{N} = \{1, ..., N\}, \tag{4.2.1}$$

where q_i denote agent i position, $q = [q_1, ..., q_N]^T$ represents the agents configuration, and u_i denotes the control input for each agent. Since this chapter is motivated by motion control in a cartesian plane, we consider $q_i = [x_i, y_i]^T \in \mathbb{R}^2$ such that x_i and y_i represent the dynamics of q_i on the x-axis and y-axis, respectively.

Define now the vector connecting any two agents (i, j) as:

$$q_{ij} = q_j - q_i,$$

and β_{ij} denoting the squared distance between two agents as follows:

$$\beta_{ij} = \|q_j - q_i\|^2, \forall\ i, j \in \mathcal{N}.$$

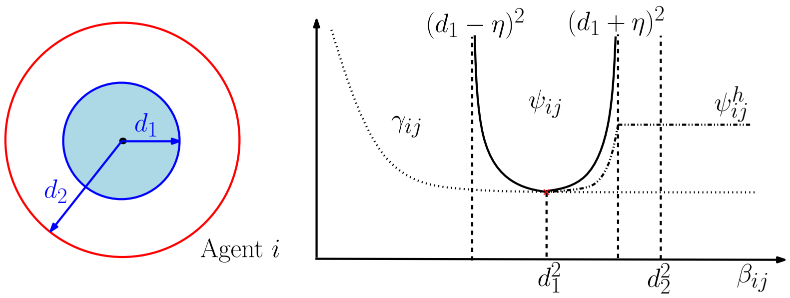


Figure 4.4: Communication setup and potential functions

Throughout this chapter, we denote for $i \in \mathcal{N}$ and $r > 0$:

$$\mathcal{N}_{i,r}(t) = \left\{ j \in \mathcal{N} \backslash \{i\} | \ \ \beta_{ij} < r^2 \right\},$$

as the subset of $\mathcal{N}$ including all neighbors of agent i, *i.e.*, all nodes that agent i can sense within a radius r at time t. Moreover, $|\mathcal{N}_{i,r}|$ denotes the number of neighbors of agent i. Define for each agent i a r-proximity graph such as:

$$\mathcal{P}_{i,r}(t) = (\mathcal{N}_{i,r}(t), \Upsilon_i(t)),$$

where $\Upsilon_i(t) \subseteq \mathcal{N}_{i,r}(t) \times \mathcal{N}_{i,r}(t)$ is the set of edges connecting agent i to all $j \in \mathcal{N}_{i,r}$ at time t, and $\mathcal{N}_{i,r}$ have been defined before.

In the sequel, each agent is supposed to have two overlapping sensing radii, as shown in Figure 4.4. Moreover, it is assumed that $d_1 < d_2$ satisfying:

$$d_1 = \xi d_2,$$

where $0 < \xi < 1$. Note that $1/\xi$ can be viewed as the safety factor, because the larger its value, the smaller the probability of losing an edge is. A pertinent question to be asked is why two communication radii are used. In fact, the use of two overlapping communication radii is strictly related to the problems treated in this thesis. As mentioned before, our objective is to propose control laws that **(i)** separate agents without losing communication and **(ii)** achieve a compact deployment by controlling inter-agent angles. In order to simplify the control strategy, we therefore consider two overlapping radii. The smaller one bounds the area wherein the inter-agent distances are controlled, while d_2-proximity graphs are used to establish a smaller domain wherein we control the inter-agent angles. For sake or clearness, an illustration of an inter-agent angle is provided in Figure 4.3.

Throughout this chapter, we particularly consider sub-graphs of three agents with specific characteristics. Thus, we introduce the following definitions.

Definition 4.1. *A triplet* $(i, j, k) \in \mathcal{N}^3$ *is a* Triangle *if* i, j, k *are all distinct and*

$$i, k \in \mathcal{N}_{j,d_2},\ i \notin \mathcal{N}_{k,d_2}.$$

Thus, a Triangle is a connected graph. Moreover, the central vertex can sense the other two agents, while the other two can only sense the central agent. Note that, in terms of notation, the order of agents matters. This means that, when Triangles are discussed, a triplet (i, j, k) is centered at j.

Definition 4.2. *A triplet* $(i, j, k) \in \mathcal{N}^3$ *is a* Compact Triangle *if* i, j, k *are all distinct and*

$$i, k \in \mathcal{N}_{j,d_2},\ i \in \mathcal{N}_{k,d_2}.$$

Then, it follows that a Compact Triangle is a three nodes complete graph. Also throughout this chapter, for each Triangle (i, j, k), we will often use the term "inter-agent angle", illustrated in Figure 4.3, to indicate the angle formed by these three agents such that:

$$\theta_{ijk} = \ \arccos\left(\frac{<q_{ji}, q_{jk}>}{\|q_{ij}\| \cdot \|q_{jk}\|}\right). \tag{4.2.2}$$

To minimize angles, we intend to apply to each triangle formed of three agents, a force proportional to the difference between the angle value and the reference. This force lies down on a perpendicular direction of q_{ij}, denoted $q_{ij}^{\perp}$. Since we intend to achieve compact formations composed of equilateral triangles, this inherently means we are aiming for each angle to be equal to $\frac{\pi}{3}$. Therefore, this value will be taken as reference.

In this chapter, we are going to analyze a dispersion and compactness control laws, individually, and finally a sequential controller. For the dispersion algorithm, we define the set of feasible initial conditions:

$$\mathcal{I}(d_1) = \left\{q \in W \mid \forall i \in \mathcal{N}, j \in \mathcal{N}_{i,d_2},\ \beta_{ij} \in \,]0, d_1^2]\right\},$$

and the set of desired final configurations as:

$$\mathcal{F}(d_1, \eta) = \left\{q \in W \mid \forall i \in \mathcal{N}, j \in \mathcal{N}_{i,d_2},\ (d_1 - \eta)^2 < \beta_{ij} < (d_1 + \eta)^2\right\}.$$

Define the set including all configurations where each agent has at most Δ neighbors in d_2-proximity graph as:

$$\mathcal{D}(d_2, \Delta) = \{q \in W \mid \forall i \in \mathcal{N}, |\mathcal{N}_{i,d_2}| \leq \Delta\}.$$

Note that Δ is a positive integer value to be defined. Define now:

$$\mathcal{E}(d_1) = \left\{q \in W \mid \text{ for all } \textit{Triangles } (i, j, k),\ \beta_{ij} = \beta_{jk} = d_1^2\right\},$$

as the set including the configurations where all Triangles have two equal edges of length d_1. Thus, for the compactness control strategy, we define the set of feasible initial conditions as:

$$\mathcal{F}'(d_1, d_2) = \mathcal{D}(d_2, 2) \cap \mathcal{E}(d_1).$$

This means that we assume formations where each agent has at most two neighbors and where the length of the existing edges is constant and equal to d_1. It is important to point out that this assumption includes, among many others, the ring formation. More precisely, this particular configuration can be represented by a graph where each agent has exactly two neighbors. Consequently, it follows from a geometrical point of view that the control strategy presented in this article is not suited for such a problem. In particular, due to the perfect balanced forces applied to each agent, an inter-agent angle based controller is unable to solve the compact formation problem. Therefore, this special formation is excluded from the following discussion.

Since the objective is to achieve the most compact configuration possible, we can define a set of desired configurations such that:

$$\mathcal{G}(d_1, d_2) = \left\{ q \in W \; | \forall j \in \mathcal{N}, \exists (i,k) \in \mathcal{N}^2_{j,d_2} \quad \text{st.} \quad \beta_{ij} = \beta_{jk} = \beta_{ki} = d_1^2 \right\}.$$

As a consequence, this triplet (i, j, k) is a *Compact Triangle*.

4.2.2 Definition of the potential functions

From the control theory point of view, cohesiveness is characterized by a repulsion/attraction function which makes the agents in the network maintain desired relative distances between its neighbors, see, *e.g.*, [191, 279]. In particular, formation control via behavior-based approach have been studied in [7, 43], and others works combined the behavior-based approach with potential fields as in, for example, [81, 103, 170, 215]. More precisely, in [215] group formation behavior is based on social potential fields and artificial potential trenches are used to represent the formation trajectory of the group in [103]. Moreover, some techniques based on cooperative strategies have already been used in order to guarantee that vehicles do not impact each other [77, 166, 191]. For instance, the authors of [170] apply this method to a non-linear dynamic system for obstacle avoidance and trajectory generation. The reader should refer to [48] for a survey in different strategies dealing with formation control.

Our deployment algorithm is based on artificial potential fields. In the sequel, a special attention is paid to the inverse agreement protocols presented in [76], among many other works in this field. In [76], authors showed that the closed loop system reaches a configuration in which the minimum distance between any pair of agents is larger than a specific lower bound. Moreover, it was proven [76] that this lower bound coincides with the agents' sensing radius. Based on this, our contribution consists on an extension of this work by adding a connectivity maintenance component. In Figure 4.4, we can see the two potential functions, γ_{ij} and $\psi_{ij} \in C([0, +\infty))$, whose argument is β_{ij} (defined before).

These two functions are designed with the key features described for all $i, j \in \mathcal{N}$ and $i \neq j$ as follows:

Definition of the potential function γ_{ij} for a couple of distinct agents $(i, j) \in \mathcal{N}^2$:

- γ_{ij} is a decreasing function, twice continuously differentiable;
- γ_{ij} tends to $+\infty$ when β_{ij} tends to zero;
- $\frac{\partial \gamma_{ij}}{\partial \beta_{ij}} = 0$ if $\beta_{ij} \geq d_1^2$.

Definition of the potential function ψ_{ij} for a couple of distinct agents $(i, j) \in \mathcal{N}^2$:

- ψ_{ij} tends to ∞ when β_{ij} tends to $(d_1 \pm \eta)^2$;
- $\psi_{ij} = \gamma_{ij}$ and $\frac{\partial \gamma_{ij}}{\partial \beta_{ij}} = 0$ if $\beta_{ij} = d_1^2$.

  where η is a positive scalar defining the size of the desired neighborhood for connectivity maintenance.

Using this framework, one has not taken into account dynamical graphs, where new edges are added, *i.e.*, when two agents come close enough to sense each other but did not form an edge before. This scenario should also be considered here. Indeed, it is possible that, as a consequence of the second stage, agents will get new neighbors as time evolves. Define:

$$\mathcal{N}^h_{i,d_1+\eta} = \left\{ j \in \mathcal{N} \backslash \{i\},\ \beta_{ij} < (d_1 + \eta)^2,\ \dot{\beta}_{ij} < 0 \right\},$$

as the subset of the sensing zone of agent i including all agents that form a new edge with agent i. In order to ensure a smooth transition to ψ_{ij}, we define ψ^h_{ij} such that:

Additional features of the potential function ψ^h_{ij}:

- ψ^h_{ij} is differentiable everywhere;
- $\psi^h_{ij} = \psi_{ij}$ and $\frac{\partial \psi^h_{ij}}{\partial \beta_{ij}} = \frac{\partial \psi_{ij}}{\partial \beta_{ij}}$ when $\beta_{ij} \leq d_1^2$;
- ψ^h_{ij} is strictly increasing when $d_1^2 < \beta_{ij} < (d_1 + \eta)^2$;
- ψ^h_{ij} is constant when $\beta_{ij} > (d_1 + \eta)^2$.

We have introduced here the considered model as well as the potential fields needed for the design of the dispersion controller. This algorithm will now be treated in the following section.

4.3 Dispersion algorithm

In this section, we present a controller for agents' dispersion with connectivity maintenance based on the potential fields described in the previous section. For a set of two agents (i, j), the dispersion controller's principle is illustrated in Figure 4.5. The forces applied on the direction of the edge connecting both agents are represented by the orange arrows. Moreover, the blue and red lines represent the d_1 and d_2, respectively.

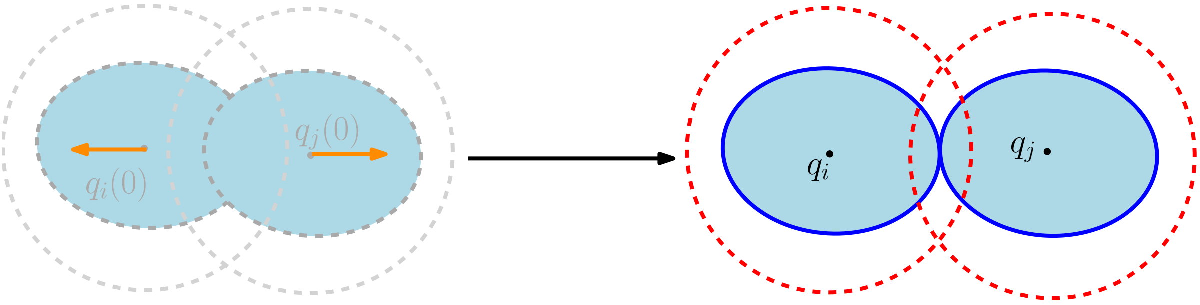


Figure 4.5: Dispersion controller's principle

4.3.1 Controller design

Consider a set of N agents (4.2.1). In this section we present a controller designed for *Dispersion*, denoted u_1. For each pair of agents (i, j) initially in communication, γ_{ij} is active, implying that agents aim at dispersing from each other. Designed to keep connectivity, ψ_{ij} becomes active at t^*_{ij} (*Confinement*), with:

$$t^*_{ij} = \min_{i \in \mathcal{N},\ j \in \mathcal{N}_{i,d_1}} \{t |\ \beta_{ij}(0) < d_1^2, \beta_{ij}(t^*_{ij}) = d_1^2\}.$$

This means that t^*_{ij} represents the instant for which the distance between two agents i and j, initially closer than d_1 and such that $\dot{\beta}_{ij} > 0$, reaches the threshold distance d_1 for the first time. Consequently, the instant for which the length of an edge within the over all network reaches the threshold distance d_1, for the first time, is defined as:

$$t^*_a = \min\{t |\ \beta_{ij}(0) < d_1^2, \beta_{ij}(t^*_{ij}) = d_1^2, \forall i \in \mathcal{N},\ j \in \mathcal{N}_{i,d_1}\}.$$

In other words, such instant corresponds to the smallest value of all t^*_{ij} among all $i, j \in \mathcal{N}$ for which $\beta_{ij}(0) < d_1^2$. The scheduling of our approach is presented in Figure 4.6. It presents the controller sequence regarding the evolution of both time and β_{ij}. For any pair of agents (i, j) for which $\beta_{ij}(0) < d_1^2$, the dispersion controller is applied while $t < t^*_{ij}$. Then, the connectivity maintenance controller takes over for all $t > t^*_{ij}$, whenever $\beta_{ij} < (d_1 + \eta)^2$.

The partial derivative of the potential functions can be computed as:

$$\begin{cases} \frac{\partial \gamma_{ij}}{\partial q_i} = 2\frac{\partial \gamma_{ij}}{\partial \beta_{ij}}(D_{ij})_i q, \\ \\ \frac{\partial \psi_{ij}}{\partial q_i} = 2\frac{\partial \psi_{ij}}{\partial \beta_{ij}}(D_{ij})_i q, \end{cases}$$

and their gradient as:

$$\begin{cases} \nabla \gamma_{ij} = 2\frac{\partial \gamma_{ij}}{\partial \beta_{ij}} D_{ij} q, \\ \\ \nabla \psi_{ij} = 2\frac{\partial \psi_{ij}}{\partial \beta_{ij}} D_{ij} q. \end{cases}$$

Matrices D_{ij}, $(D_{ij})_i$, for any $i < j$, are given by:

$$D_{ij} = \begin{bmatrix} & & O_{(i-1)\times N} & & \\ O_{1\times(i-1)} & 1 & O_{1\times(j-i-1)} & -1 & O_{1\times(N-j)} \\ & & O_{(j-i-1)\times N} & & \\ O_{1\times(i-1)} & -1 & O_{1\times(j-i-1)} & 1 & O_{1\times(N-j)} \\ & & O_{(N-j)\times N} & & \end{bmatrix} \otimes I_2.$$

Thus, the proposed controller can be expressed as:

$$u_{i1} = -\sum_{j\in\mathcal{N}_{i,d_1}} \frac{\partial \gamma_{ij}}{\partial q_i} - \sum_{j\in\mathcal{N}^h_{i,d_1+\eta}} \frac{\partial \bar{\psi}_{ij}}{\partial q_i} - \sum_{j\in\mathcal{N}_{i,d_2}} \frac{\partial \psi^h_{ij}}{\partial q_i}, \tag{4.3.1}$$

with

$$\bar{\psi}_{ij}(t) = \begin{cases} 0, & \text{if } t < t^*_{ij}, \\ \psi_{ij}, & \text{otherwise.} \end{cases}$$

Equation (4.3.1) corresponds to the sum of the negative gradients of the potential functions. This means that each agent is then equipped with a repulsive and attractive

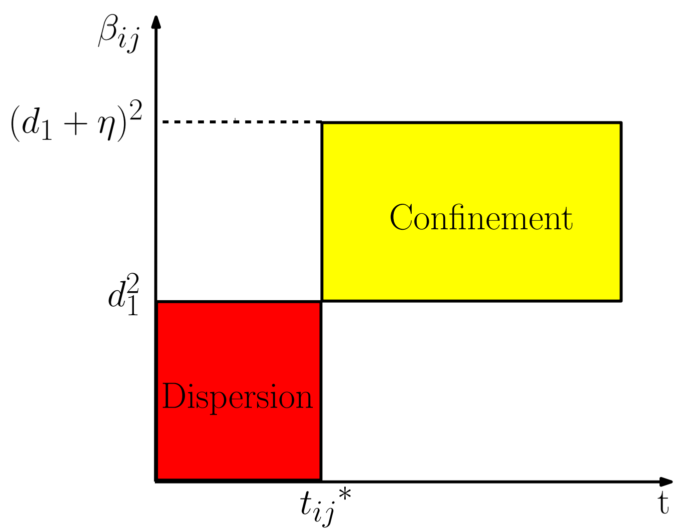


Figure 4.6: Dispersion controller's scheduling

field with respect to another agent within its sensing zone. As mentioned before, in order to take into account newly formed edges, consider ψ_{ij}^H as a modified potential function based on ψ_{ij} and ψ_{ij}^h, such that:

$$\psi_{ij}^H = \begin{cases} \psi_{ij}^h, & \text{if } j \in \mathcal{N}_{i,d_1+\eta}^h, \\ \bar{\psi}_{ij}, & \text{otherwise .} \end{cases}$$

Whenever an agent j forms a new edge with agent i, the function ψ_{ij}^H switches from ψ_{ij}^h to $\bar{\psi}_{ij}$. Define also:

$$\sigma_{ij}^H = \begin{cases} \frac{\partial \psi_{ij}^h}{\partial \beta_{ij}}, & \text{if } j \in \mathcal{N}_{i,d_1+\eta}^h, \\ \\ \frac{\partial \bar{\psi}_{ij}}{\partial \beta_{ij}}, & \text{otherwise .} \end{cases}$$

Thus, it yields:

$$u_1 = -2 \left[R_1 \otimes I_2 + R_2 \otimes I_2 \right] q, \tag{4.3.2}$$

where R_1 and R_2 are $N \times N$ matrices defined as follows:

$$(R_1)_{pq} = \begin{cases} \displaystyle\sum_{j \in \mathcal{N} \setminus \{p\}} \frac{\partial \gamma_{pj}}{\partial \beta_{pj}}, & \text{if } p = q, \\ -\dfrac{\partial \gamma_{pq}}{\partial \beta_{pq}}, & \text{otherwise.} \end{cases} \qquad (R_2)_{pq} = \begin{cases} \displaystyle\sum_{j \in \mathcal{N} \setminus \{p\}} \sigma_{pj}^H, & \text{if } p = q, \\ -\sigma_{pq}^H, & \text{otherwise.} \end{cases}$$

The proposed controller for agent's dispersion is expressed in (4.3.2). The following section will provide stability conditions and performance analysis.

4.3.2 Stability analysis

This section provides a theoretical stability analysis for the dispersion algorithm. The respective controller ensures dispersion of all agents in the workspace, while maintaining connectivity and avoiding loosing edges. The next theorem states our results.

Theorem 4.1. *(Rodrigues de Campos et al. [229]) Consider N agents (4.2.1) driven by the control law (4.3.1). Assume a set of feasible initial conditions $\mathcal{I}(d_1)$. Then the system reaches a final static configuration belonging to $\mathcal{F}(d_1, \eta)$.*

Proof. Consider V_d as a Lyapunov function such that:

$$V_d(q) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} \left[\gamma_{ij}(\beta_{ij}) + \psi_{ij}^H(\beta_{ij}) \right]. \tag{4.3.3}$$

For the sake of readability, the argument of the Lyapunov function will be omitted. We can compute:

$$\begin{cases} \nabla\gamma_{ij} = 2\frac{\partial\gamma_{ij}}{\partial\beta_{ij}}(q_i - q_j), \\ \\ \nabla\psi^h_{ij} = 2\frac{\partial\psi^h_{ij}}{\partial\beta_{ij}}(q_i - q_j), \end{cases}$$

and

$$\frac{\partial\gamma_{ij}}{\partial\beta_{ij}} = \frac{\partial\gamma_{ji}}{\partial\beta_{ji}}, \quad \frac{\partial\psi_{ij}}{\partial\beta_{ij}} = \frac{\partial\psi_{ji}}{\partial\beta_{ji}}, \quad \sigma^H_{ij} = \sigma^H_{ji},$$

due to symmetry. Thus, it follows that:

$$\nabla V_d = \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}\setminus\{i\}}\left[\nabla\gamma_{ij} + \nabla\psi^H_{ij}\right] = 4(R_1\otimes I_2)q + 4(R_2\otimes I_2)q = 4\left[(R_1+R_2)\otimes I_2\right]q.$$

Finally, the derivative of the Lyapunov function can be expressed as:

$$\dot{V}_d = (\nabla V_d)^T\,\dot{q} = -8\|[(R_1\otimes I_2) + (R_2\otimes I_2)]q\|^2. \tag{4.3.4}$$

Therefore, $\dot{V}_d$ is strictly negative for all $t > 0$, guaranteeing the system's convergence. Let us discuss this conclusion. Consider any $t < t^*_a$. For any two agents initially close such that $\beta_{ij}(0) < d_1^2, \forall j \in \mathcal{N}_{i,d_1}$, we have $\frac{\partial\gamma_{ij}}{\partial\beta_{ij}} < 0$ if $\beta_{ij} < d_1^2$. Moreover, $\frac{\partial\gamma_{ij}}{\partial\beta_{ij}} = 0$ if $\beta_{ij} \geq d_1^2$, and therefore $\beta_{ij} = d_1^2$ corresponds to the equilibrium point of γ_{ij}. Since the contribution of the remaining potential fields is null for $\beta_{ij} < d_1^2$, it follows that for any two agents $\dot{\beta}_{ij} > 0$ such that β_{ij} will eventually reach a close neighborhood of d_1^2. At $t = t^*_a$, the potential function ψ_{ij} is activated for at least one pair of agents. For this pair of agents, two evolution cases are possible: either $\dot{\beta}_{ij} < 0$ such that $(d_1-\eta)^2 < \beta_{ij} < d_1^2$ (situation **(i)**), or $\dot{\beta}_{ij} > 0$ such that $d_1^2 < \beta_{ij} < (d_1+\eta)^2$, (situation **(ii)**):

(i) By definition, if $(d_1-\eta)^2 < \beta_{ij} < d_1^2$, the quantities $\frac{\partial\gamma_{ij}}{\partial\beta_{ij}}$ and $\frac{\partial\psi_{ij}}{\partial\beta_{ij}}$ are strictly negative. This means that throughout (4.3.1) a repulsive force is applied to agents i and j, and consequently β_{ij} will increase such that $\dot{\beta}_{ij} > 0$. Moreover, since V_d tends to $+\infty$ whenever β_{ij} tends to $(d_1-\eta)^2$, we can conclude that the distance separating agents will never reach $(d_1-\eta)^2$.

(ii) By definition, if $d_1^2 < \beta_{ij} < (d_1+\eta)^2$, the quantities $\frac{\partial\gamma_{ij}}{\partial\beta_{ij}}$ and $\frac{\partial\psi_{ij}}{\partial\beta_{ij}}$ are equal to zero and strictly positive, respectively. This means that throughout (4.3.1) a attractive force is applied to agents i and j and therefore β_{ij} will decrease such that $\dot{\beta}_{ij} < 0$. Since V_d tends to ∞ when β_{ij} tend to $(d_1+\eta)^2$, we can conclude that the distance separating agents will never reach $(d_1+\eta)^2$.

Considering the specific design characteristics of ψ^H_{ij}, it follows that if a new edge is created, *i.e.*, if the distance separating agents i and j is lower than a certain threshold $d_1+\eta$, the transition between ψ^h_{ij} and ψ_{ij} is held in a sufficiently smooth manner. This means that once an edge is added it is never deleted. Thus, this yields that $V_d(q(t)) \leq V_d(q(0))$ for all $t \geq 0$ and $V_d \to \infty$ when $\beta_{ij} \to d_1^2 \pm \eta$ for at least one par of agents (i, j). Then, we can conclude that $q(t) \in \mathcal{F}(d_1, \eta)$ for all $t \geq 0$ where all agent pairs that come into distance less or equal to $d_1 + \eta$ for the first time, remain within distance $d \in [d_1 - \eta, d_1 + \eta]$ for all future times. In the sequel, we use LaSalle's Principle for hybrid systems [160]. A collection of other results that can be viewed as extensions of LaSalle's Invariance Principle to certain classes of switched linear systems is also presented in [119], and a recent review on hybrid systems in [107]. Considering LaSalle's Principle for hybrid systems, the trajectories of the closed loop system converge to the largest invariant subset of the set:

$$S = \left\{ q | \dot{V}_d = 0 \right\} = \{ q | ([R_1 + R_2] \otimes I_2) q = 0 \} .$$

Since $\dot{q} = u_1 = -2 \left[R_1 \otimes I_2 + R_2 \otimes I_2 \right] q$, we have $u_1 = 0$, *i.e.*, when all edges have equal length d_1 and all potential field's derivatives are equal to zero. Therefore, the system will eventually reach a static configuration, *i.e.*, all agents eventually stop such that $u_{i1} = 0$ for all $i \in \mathcal{N}$. □

In this section we presented and analyzed a controller for agents' deployment using potential fields. We have proven its efficiency for both dispersion and connectivity maintenance. Moreover, it is worth mentioning that the proposed analysis is valid for all N, η and ξ. Simulation results validating these theoretical developments will be presented further in this chapter.

4.4 Compactness controller

In this section, we design a controller for formation compactness, denoted u_2. We intend to preform direct angle control using only relative positions, by minimizing inter-agent angles in order to achieve the most compact configuration possible. We make the following assumption.

Assumption 4.1. *We assume the length of existing edges is equal to d_1, and that ξ is sufficiently close to 1. This inherently means that we assume that the communication radii are completely overlapping, such that $d_1 = d_2$, and that initially formed edges are taken to be equal to d_1 as a result of a completely independent control action.*

4.4.1 Controller design

In order for the system to reach the most compact formation possible, we intend to minimize inter-agent angles among the overall formation. It is worth mentioning

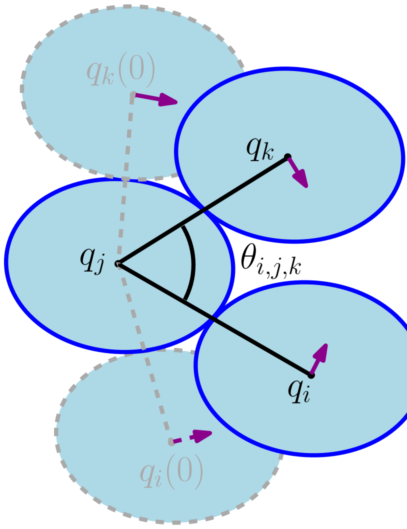


Figure 4.7: Compactness controller's principle

that minimizing inter-agent angles can be seen as an artificial way to manipulate the graph connectivity. Indeed, through the proposed strategy, we inherently control the node degree, *i.e.*, the number of neighbors of each node, and therefore the global graph connectivity properties.

Thus, the new controller for each agent i can then be presented as:

$$u_{i2} = -\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \backslash \{i\}} \sum_{k \in \mathcal{N} \backslash \{i,j\}} \chi_{ijk} R_{ot} q_{ji}, \tag{4.4.1}$$

with

$$\chi_{ijk} = \begin{cases} K[\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}], & \text{if } (i,k) \in \mathcal{N}^2_{j,d_2} \backslash \mathcal{N}^2_{j,d_1}, \\ 0, & \text{otherwise,} \end{cases}$$

where K is a positive real gain. Each angle is calculated by the central vertex, transmitting afterwards, in a distributed way, this information to its neighbors. By acting directly over angles, controller (4.4.1) allows us to improve the formation compactness. The underlying principle is based on angular springs. Using the physical and mechanical concepts relating the applied force and the error regarding the equilibrium position of the system, we apply, on $q_{ij}^{\perp}$ direction, a force proportional to difference between the actual angle value and the desired one. This principle is illustrated in Figure 4.7, where the previously mentioned forces are represented by the purple arrows. Consequently, when applied to a set of three agents, we obtain a Compact Triangle where all internal angles are equal to $\pi/3$, see Figure 4.3. Note that due to this geometric argument, one might conclude that agent i can have at the most six neighbors.

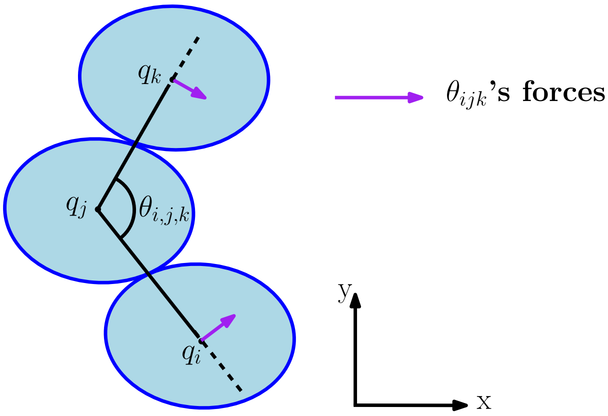


Figure 4.8: Configuration of three agents

Rewritten in vector form, equation (4.4.1) yields:

$$u_2 = -\left[(R_3 \otimes I_2)R_{ot}\right] q, \tag{4.4.2}$$

where

$$(R_3)_{pq} = \begin{cases} \sum_{j\in\mathcal{N}\setminus\{p\}} \sum_{k\in\mathcal{N}\setminus\{p,j\}} \chi_{pjk}, & \text{if } p = q, \\ \sum_{k\in\mathcal{N}\setminus\{p,q\}} \chi_{pqk}, & \text{otherwise.} \end{cases}$$

By applying controller (4.4.2), we can transform the existing formation in the most compact structure possible. Moreover, it has been mentioned that this approach can also be seen as a method to increase the network connectivity rate. In the next section we provide analytical stability conditions for the proposed control strategy.

4.4.2 Stability analysis

The following two Lemmas state our first results concerning the efficiency of the compactness controller.

Lemma 4.1. *(Rodrigues de Campos et al. [229]) Consider $N = 3$ agents described by (4.2.1), denoted (i, j, k), driven by the control law (4.4.1). Note that j is the central vertex and assume the set of initial conditions $\mathcal{F}'(d_1, d_2)$, as depicted in Figure 4.8. Then, the system reaches a final configuration belonging to $\mathcal{G}(d_1, d_2)$.*

Proof. Based on the definition of the set $\mathcal{F}'(d_1, d_2)$, we can consider that the distance separating two connected agents is constant and equal to d_1. From equation (4.2.2), we can compute:

$$\frac{d}{dt}\cos(\theta_{ijk}) = \frac{[< \dot{q_{ji}}, q_{jk} > + < q_{ji}, \dot{q_{jk}} >]}{\|q_{ji}\| \ \|q_{jk}\|}. \tag{4.4.3}$$

We can easily obtain:

$$< q_{ji}, {q_{jk}}^{\perp} >= -\|q_{ji}\|\|q_{jk}\| \sin(\theta_{ijk}), \quad < {q_{ji}}^{\perp}, q_{jk} >= \|q_{ji}\|\|q_{jk}\| \sin(\theta_{ijk}).$$

Considering controller (4.4.1), (4.4.3) can be written as:

$$\frac{d}{dt}\cos(\theta_{ijk}) = 2K\left(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}\right)\sin(\theta_{ijk}).$$

Finally, for all $sin(\theta_{ijk}) \neq 0$ we have:

$$\dot{\theta}_{ijk} = -2K\left(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}\right).$$

Note that previous manipulations exclude cases where $|\theta_{ijk}| = k\pi$, since if $sin(\theta_{ijk}) = 0$ then:

$$\dot{\theta}_{ijk} = 2K\left(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}\right)\sin(\theta_{ijk}) = 0,$$

which corresponds to a particular equilibrium offering two possible trajectories. In this work, and for the sake of simplicity, we consider that $\theta_{ijk} \neq 0 \ [\pi]$, *i.e.*, $\theta_i jk \neq k\pi$. Consider now the following Lyapunov function candidate:

$$V_c(q) = \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}_{j,d_2}} \sum_{k\in\mathcal{N}_{j,d_2}\backslash\{i\}} \left(|\theta_{ijk}| - \frac{\pi}{3}\right)^2.$$

Note that for each Triangle, there is only one inter-agent angle to be controlled. Due to the geometric constraints that $\mathcal{F}'(d_1, d_2)$ implies, $|\theta_{ijk}| \geq \pi/3, \forall j \in \mathcal{N}, (i,k) \in \mathcal{N}^2_{j,d_2}, i \neq k$ (since in a isosceles Triangle edges are at least of length d_1). Thus, either $\pi/3 \leq \theta_{ijk} < \pi$ or $-\pi < \theta_{ijk} \leq -\pi/3$, which leads us to a continuous function $V_c(q)$. Its derivative can then be written as:

$$\dot{V}_c(q) = -4K\left(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}\right)^2. \tag{4.4.4}$$

Thus, $\dot{V}_c(q)$ remains non-positive for all $t \geq 0$, so that $|\theta_{ijk}|$ tends to $\pi/3$. Based on simple geometric arguments, this necessarily means that the third edge's length tends to d_1, and consequently that a Compact Triangle is eventually formed. This concludes the proof. □

A first result on formation's compactness control for a three agents network was presented in Lemma 4.1. Since the evolution of each angle is dependent on the evolution of three different agents, it has been our intuition that the complexity of our approach will tend to increase as the size of the network increases. The next Lemma states a new result for a four agent network.

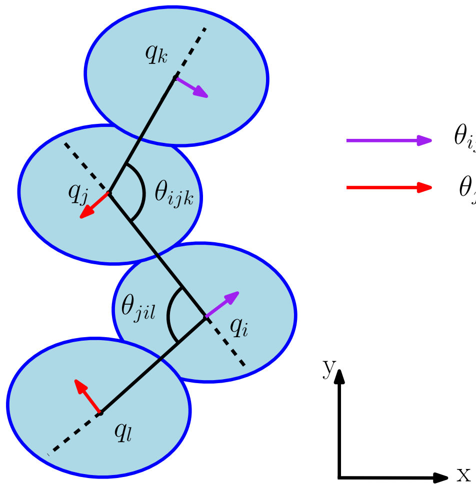


Figure 4.9: Configuration of four agents

Lemma 4.2. *(Rodrigues de Campos et al. [229]) Consider four agents described by (4.2.1), denoted (i, j, k, l), driven by the control law (4.4.1). Note that i and j are the central vertices, and assume that the set of initial conditions is $\mathcal{F}'(d_1, d_2)$, as depicted in Figure 4.9. Then, the system reaches a final configuration belonging to $\mathcal{G}(d_1, d_2)$.*

Proof. From (4.2.2), we have:

$$\frac{\mathrm{d}}{\mathrm{d}\theta_{ijk}} \cos(\theta_{ijk}) = K \left[2(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) + (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) \right] sin(\theta_{ijk}), \tag{4.4.5a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta_{jil}} \cos(\theta_{jil}) = K \left[2(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) + (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) \right] sin(\theta_{jil}). \tag{4.4.5b}$$

Note that we can see in (4.4.5) the relation between adjoining angles sharing one edge. Consider V_c as a Lyapunov function candidate defined by:

$$V_c(q) = \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{N}_{j,d_2}} \sum_{k \in \mathcal{N}_{j,d_2} \setminus \{i\}} \left(|\theta_{ijk}| - \frac{\pi}{3} \right)^2 .$$

Since for a set of four agents there are two controllable angles, and considering the same geometric concepts as before, then $V_c(q) = (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})^2 + (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})^2$. Retrieving $\dot{\theta}_{ijk}$ and $\dot{\theta}_{jil}$ from (4.4.5), we can write:

$$\dot{V}_c(q) = -4K \Big[(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})^2 + (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})^2 + \\ + (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) \Big] .$$

Since the right hand side of the previous equation is strictly negative, it follows that:

$$V_c(q(t)) < V_c(q(0)) < \infty, \ \forall\, t \geq 0,$$

and that $|\theta_{ijk}| \to |\theta_{jil}| \to \pi/3$, and inherently, all the triangles' edges are equal to d_1. □

Previous calculations show that control complexity and inherent analysis tend to increase as the network's size grows. Furthermore, it has been shown that the evolution of any angle is closely dependent on the evolution of its adjoining angles. This point will be enhanced by the following discussion.

Consider a set of five agents (4.2.1) driven by controller (4.4.1), in a formation as in Figure 4.10. It follows that, for a five agents network, there are three controllable angles. Following the same reasoning as before, calculations lead us to:

$$\frac{\mathrm{d}}{\mathrm{d}\theta_{ijk}}\cos(\theta_{ijk}) = K\left[2(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) + (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) + (\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3})\right] sin(\theta_{ijk}),$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta_{jil}}\cos(\theta_{jil}) = K\Big[2(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) + (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) +$$
$$(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})(\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3})\frac{\sin\left((\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) - (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})\right)}{\sin\left(\theta_{jil}\right)}\Big] sin(\theta_{jil}),$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta_{mkj}}\cos(\theta_{mkj}) = K\Big[2(\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3}) + (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) +$$
$$(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})(\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3})\frac{\sin((\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) - (\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3}))}{\sin(\theta_{mkj})}\Big] sin(\theta_{mkj}).$$

Consider:

$$V_c(q) = \sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{N}_{j,d_2}}\sum_{k\in\mathcal{N}_{j,d_2}\setminus\{i\}}\left(|\theta_{ijk}| - \frac{\pi}{3}\right)^2 \tag{4.4.6}$$

as a candidate Lyapunov function candidate, with a minimum equal to zero for $\theta_{ijk} = \theta_{jil} = \theta_{mkj} = \frac{\pi}{3}$. We can then derive:

$$\dot{V}_c(q) = -K\begin{bmatrix}(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})\\(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})\\(\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3})\end{bmatrix}^T \mathcal{M}(\varepsilon)\begin{bmatrix}(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})\\(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})\\(\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3})\end{bmatrix}, \tag{4.4.7}$$

with

$$\mathcal{M}(\varepsilon) = \begin{bmatrix}4 & 2 & \varepsilon\\2 & 4 & 2\\\varepsilon & 2 & 4\end{bmatrix}.$$

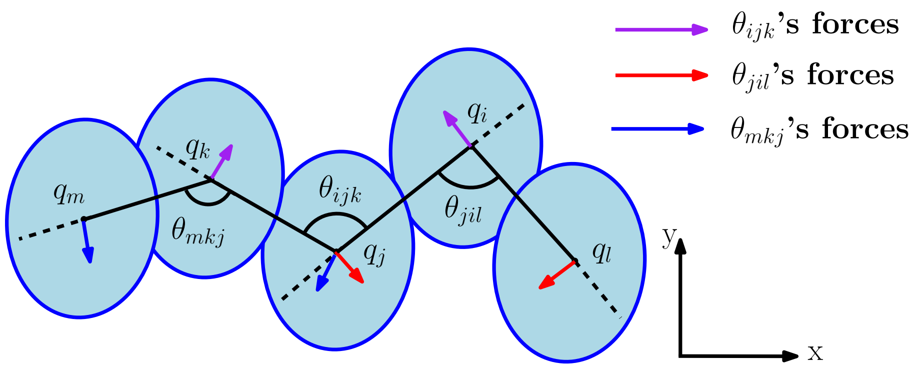


Figure 4.10: Configuration of five agents

Knowing that:

$$\varepsilon = \varphi \sin(\theta_{ijk}), \tag{4.4.8}$$

with

$$\varphi = \left[\frac{\sin\left((\theta_{mkj} - sign(\theta_{mkj})\frac{\pi}{3}) - (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) \right)}{\sin(\theta_{jil})\sin(\theta_{mkj})} \right],$$

it follows that $-\varphi \leq \varepsilon \leq \varphi$.

In the previous analysis, we identified a term relating angles sharing a same edge (as θ_{ijk} and θ_{jil} or θ_{ijk} and θ_{mkj}). However, the evolution of the system is also dependent on the evolution of angles sharing a same vertex (as θ_{jil} and θ_{mkj}), as we can see in (4.4.8). Thus, we can conclude from (4.4.7) that the system's stability is conditioned by the value of ε, which establishes relations between the three angles of the system, $\theta_{ijk}, \theta_{jil}, \theta_{mkj}$. As $\mathcal{M}(\varepsilon)$ is a 3×3 matrix, the calculus of its eigenvalues is straightforward and can be expressed as:

$$\lambda_1 = -\varepsilon + 4 \text{ and } \lambda_{2,3} = \frac{\varepsilon + 8 \pm \sqrt{\varepsilon^2 + 32}}{2}.$$

Thus, in order to have positive eigenvalues, ε has to be chosen in $[-2, 4]$. Thus, if the set of initial conditions $\mathcal{F}'(d_1, \eta)$ satisfies $\varepsilon \in [-2, 4]$, then $\dot{V}_c < 0$ and the system remains stable, eventually reaching a formation belonging to $\mathcal{G}(d_1, d_2)$. Figure 4.11 shows the validity zones for such conditions. More precisely, red zones represent the set of values of $(\theta_{ijk}, \theta_{jil}, \theta_{mkj})$ for which the desired condition is not fulfilled.

The previous discussion is an endeavour to analyze networks of five or more agents, but formal stability analysis is still to be provided. This study also revealed that for particular initial formations the system reaches singular formations that do not satisfy our control objectives. Since these singular formations can be explained by a stable equilibrium between inter-agent angles, the next section will propose a solution to overtake these local mimina and to avoid singular configurations.

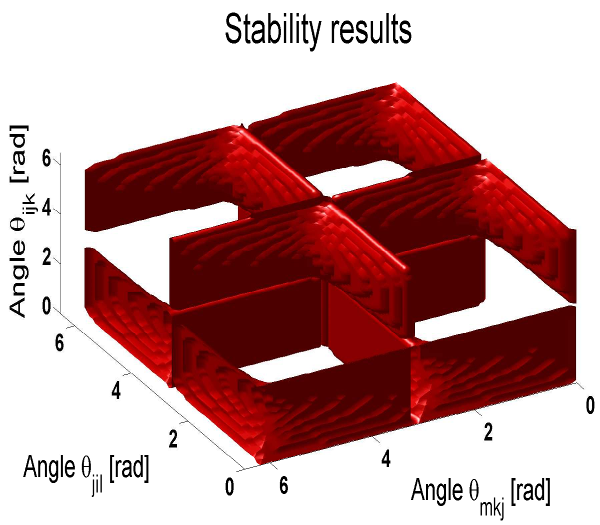


Figure 4.11: Stability zones for a configuration of five agents

4.4.3 Improved controller with variable gain

Consider now a set of $N = 4$ agents (4.2.1), denoted (i, j, k, l). Under Assumption 4.1, define:

$$\mathcal{I}'(d_1, d_2) = \mathcal{D}(d_2, 3) \cap \mathcal{E}(d_1).$$

as the considered set of feasible formations. The following result can be derived.

Lemma 4.3. *(Rodrigues de Campos et al. [229]) Consider four agents (4.2.1), denoted (i, j, k, l), driven by the control law (4.4.1). Note that j is the central vertex and assume the set of initial conditions belongs to $\mathcal{I}'(d_1, d_2)$, as shown in Figure 4.12. Thus, the system reaches a final configuration where all inter-agent angles $\theta_{ijk}, \theta_{lji}, \theta_{kjl}$ are equal to $2\pi/3$.*

Proof. Following the same reasoning as before, calculations lead us to:

$$\frac{\mathrm{d}}{\mathrm{d}\theta_{ijk}} \cos(\theta_{ijk}) = K\left[2(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) - (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) - (\theta_{jkm} - sign(\theta_{jkm})\frac{\pi}{3})\right] sin(\theta_{ijk}),$$
$$\frac{\mathrm{d}}{\mathrm{d}\theta_{jil}} \cos(\theta_{jil}) = K\left[2(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) - (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) - (\theta_{jkm} - sign(\theta_{jkm})\frac{\pi}{3})\right] sin(\theta_{jil}),$$
$$\frac{\mathrm{d}}{\mathrm{d}\theta_{lji}} \cos(\theta_{lji}) = K\left[2(\theta_{lji} - sign(\theta_{lji})\frac{\pi}{3}) - (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}) - (\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})\right] sin(\theta_{lji}).$$

It is obvious that, if all agents are sharing the same vertex j, the evolution on an angle is closely dependent on the others, such that:

$$2\pi = \theta_{ijk} + \theta_{kjl} + \theta_{lji}. \tag{4.4.9}$$

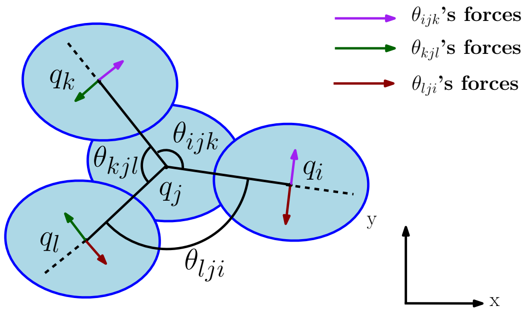


Figure 4.12: Singular formation of four agents

Furthermore, if we consider $V_c(q)$ as a Lyapunov function candidate defined by:

$$V_c(q) = \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}_{j,d_2}} \sum_{k\in\mathcal{N}_{j,d_2}\setminus\{i\}} \left(|\theta_{ijk}| - \frac{\pi}{3}\right)^2,$$

its derivative can be expressed as:

$$\begin{aligned}\dot{V}_c(q) = -4K\Big[&-(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3}) - (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})(\theta_{lji} - sign(\theta_{lji})\frac{\pi}{3}) - \\ &-(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})(\theta_{lji} - sign(\theta_{lji})\frac{\pi}{3}) + (\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3})^2 + \\ &+(\theta_{jil} - sign(\theta_{jil})\frac{\pi}{3})^2 + (\theta_{lji} - sign(\theta_{lji})\frac{\pi}{3})^2\Big].\end{aligned} \tag{4.4.10}$$

By analyzing (4.4.10), we can see that $\dot{V}_c(q)$ is strictly negative, and has a global maximum equal to zero when $\|\theta_{ijk}\| = \|\theta_{kjl}\| = \|\theta_{lji}\|$. Thus, from (4.4.9), it follows that this system will reach a stable configuration where:

$$\|\theta_{ijk}\| = \|\theta_{kjl}\| = \|\theta_{lji}\| = \frac{2\pi}{3}.$$

This concludes the proof. □

Previous calculations show that a set of four agents (i, j, k, l) on an initial configuration as presented in Figure 4.12 will eventually reach a final configuration where all inter-agent angles $\theta_{ijk}, \theta_{lji}, \theta_{kjl}$ are equal to $2\pi/3$. Despite the fact that the system achieves a stable configuration, our control requirements are not satisfied. The inter-agent angles reach an undesired equilibrium that can be explained by a perfect balance of the forces applied to each agent. Therefore, in order to avoid local minima and inherent singular configurations we present in the sequel an improved controller equipped with a variable gain. This gain, denoted $K(\theta_{ijk})$, can be expressed as:

$$K(\theta_{ijk}) = \frac{1}{k(|\theta_{ijk}|)^2}\frac{\partial k(|\theta_{ijk}|)}{\partial \theta_{ijk}}, \tag{4.4.11}$$

where $k(|\theta_{ijk}|)$ is depicted in Figure 4.13. Finally, our improved controller can be expressed as:

$$u_{i3} = -\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}\backslash\{i\}}\sum_{k\in\mathcal{N}\backslash\{i,j\}}\chi'_{ijk}R_{ot}(D_{ij})_i q, \tag{4.4.12}$$

with

$$\chi'_{ijk}(t) = \begin{cases} K(\theta_{ijk})(\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}), & \text{if } (i,k) \in \mathcal{N}^2_{j,d_2}\backslash\mathcal{N}^2_{j,d_1}, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the following result holds.

Lemma 4.4. *(Rodrigues de Campos et al. [229]) Consider four agents (4.2.1), denoted (i,j,k,l), driven by the control law (4.4.12). Note that j is assumed to be the central vertex. Assume the set of initial conditions $\mathcal{I}'(d_1,d_2)$, provided that the initial inter-agent angles do not have the absolute value. Thus, the systems reach a final configuration belonging to $\mathcal{G}(d_1,d_2)$.*

Proof. From its definition, we have $k(|\theta_{ijk}|) > 0$ and $\frac{\partial k(|\theta_{ijk}|)}{\partial|\theta_{ijk}|} \geq 0$. Furthermore, it follows from (4.4.11) that $K(\theta_{ijk})$ value increases as $|\theta_{ijk}| \to \pi/3$. Therefore, for each Triangle (i,j,k), the smaller θ_{ijk}'s value is the stronger the contribution of (4.4.1) will be. Based on geometric arguments, it yields that for a set of four agents in similar formation as in Figure 4.12, there are three controllable inter-agent angles and that their sum is equal to 2π. Under the assumption that the initial inter-agent angles do not have the same absolute value, it follows that the absolute value of an inter-agent angle will be strictly inferior to all the others' such that, for example, $\theta_{ijk} < \theta_{ljk} < \theta_{kjl}$. Since $K(\theta_{ijk})$ value increases as $|\theta_{ijk}| \to \pi/3$, the reader can see in this approach a priority based strategy. In other words, the force applied to small angles has a higher priority (higher $K(\theta_{ijk})$'s value), while large angles have lower priority (smaller $K(\theta_{ijk})$'s value). For the sake of brevity, the rest of the proof can been deduced from the proof of Lemma 4.3 and is omitted. □

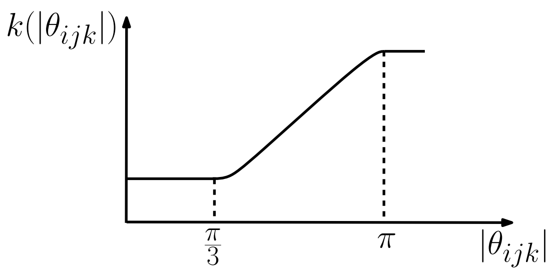


Figure 4.13: Evolution of the variable gain $k(|\theta_{ijk}|)$

Particular formations as the one presented in Figure 4.12 lead our original approach expressed in (4.4.1) to singular results where all angles are equal. However, an alternative solution using variable gains was identified and proposed previously in this section. In the sequel, we are going to discuss and analyze a sequential approach.

4.5 Sequential controller

In this section we present what we call a sequential controller, composed of the two algorithms presented in previous sections. One phase will ensure dispersion of all agents in the workspace, while the second one is going to minimize inter-agent angles in order to achieve a compact formation. Without lost of generality for the dispersion analysis, we make the following assumption in order to ensure the efficiency of our approach

Assumption 4.2. *We assume that η and ξ are, respectively, sufficiently close to* 0 *and* 1*. This inherently means that, after the dispersion is achieved, the distance separating two neighbors is taken to be constant and equal to d_1. Furthermore, it is assumed that initially each agent should have, at most, two neighbors.*

The strategy structure is represented in Fig. 4.14. Consider now a Triangle, with agents initially closer than d_1. The smallest time value for which the distance separating two neighbors reaches the threshold distance d_1 for the first time and for two edges is defined as:

$$t^*_{ijk} = \min\{t|\ \beta_{ij}(0), \beta_{jk}(0) < d_1^2, \beta_{ij}(t^*_{ijk}) = \beta_{jk}(t^*_{ijk}) = d_1^2,\ j \in \mathcal{N},\ i \in \mathcal{N}_{j,d_1},\ k \in \mathcal{N}_{j,d_1} \backslash \{i\}\}.$$

The smallest time value satisfying for the first time the previous condition is defined as:

$$t^*_b = \min_{i,j,k}\{t|\ \beta_{ij}(0) < d_1^2, \beta_{jk}(0) < d_1^2, \beta_{ij}(t^*_{ijk}) = \beta_{jk}(t^*_{ijk}) = d_1^2\}.$$

It follows that $t^*_b \geq t^*_a$. Keeping in mind previous definitions, the force minimizing inter-agent angles becomes active for each Triangle (i, j, k) at $t = t^*_{ijk}$ (Fig. 4.14). Note that dispersion algorithm's final configuration corresponds to the initial set of the compactness algorithm.

Define now:

$$\mathcal{I}''(d_1, d_2) = \mathcal{D}(d_2, 2) \cap \mathcal{I}(d_1).$$

as the feasible set of initial configurations for the proposed sequential controller. We are now ready to introduce our approach.

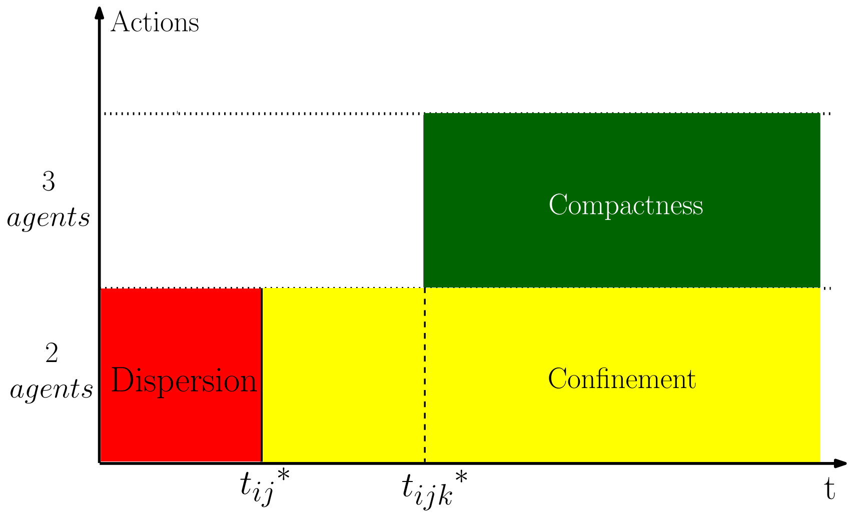


Figure 4.14: Sequential controller's scheduling

4.5.1 Controller design

The overall framework can be seen as a hybrid system, with transitions at instants whenever ψ_{ij} is activated and when new edges are added to the graph, or when compactness forces are being applied. Agent i's control input will be denoted by:

$$u_i = u_{i1} + u_{i2},$$

and $u = [u_1, \ldots, u_N]^T$. Define:

$$(R_1)_{pq} = \begin{cases} \sum_{j \in \mathcal{N} \setminus \{p\}} \frac{\partial \gamma_{pj}}{\partial \beta_{pj}}, & \text{if } p = q, \\ -\frac{\partial \gamma_{pq}}{\partial \beta_{pq}}, & \text{otherwise.} \end{cases} \qquad (R_2)_{pq} = \begin{cases} \sum_{j \in \mathcal{N} \setminus \{p\}} \sigma_{pj}^H, & \text{if } p = q, \\ -\sigma_{pq}^H, & \text{otherwise.} \end{cases}$$

and

$$(R_3)_{pq} = \begin{cases} \sum_{j \in \mathcal{N} \setminus \{p\}} \sum_{k \in \mathcal{N} \setminus \{p,j\}} \chi_{pjk}, & \text{if } p = q, \\ \sum_{k \in \mathcal{N} \setminus \{p,q\}} \chi_{pqk}, & \text{otherwise.} \end{cases}$$

In a similar formalization as in [107], we can then express our system as follows:

$$\begin{cases} \begin{cases} \dot{q} & = -\left[2(R_1 \otimes I_2) + 2(R_2 \otimes I_2)\right] q, \\ \bar{\theta}_{ijk} & = \pi, \end{cases} & \text{if } t < t^*_{ijk}, \\ \begin{cases} \dot{q} & = -\left[2(R_1 \otimes I_2) + 2(R_2 \otimes I_2) + (R_3 \otimes I_2) R_{ot}\right] q, \\ \bar{\theta}_{ijk} & = \theta_{ijk}, \end{cases} & \text{otherwise.} \end{cases} \tag{4.5.1}$$

It is easy to see that the proposed controller expressed in the previous equation presents a hybrid structure. More precisely, it exhibits both continuous and discrete dynamic behavior, with jumps described by the evolution of $\bar{\theta}_{ijk}$. Therefore, it follows that the system's state changes either continuously or discretely. We are now going to analyze its stability and performances.

4.5.2 Stability analysis

Based on our results presented in Sections 4.3 and 4.4, we are going to analyze the proposed sequential controller's performances. From the definition of the controller presented before, it is clear that the resulting system has a hybrid behavior as described in [107]. Therefore, in this section we are going to present a stability analysis based on this framework and stated as follows.

Theorem 4.2. *(Rodrigues de Campos et al. [229]) Consider $N \in \{2,3,4\}$ agents (4.2.1) driven by the control law (4.5.1). Assume the set of initial conditions $\mathcal{I}''(d_1,d_2)$. Thus, the system reaches a final configuration belonging to $\mathcal{G}(d_1,d_2)$.*

Proof. Since the two stages are sequentially related, we consider a global Lyapunov function candidate $V_g(q)$ such that:

$$V_g(q) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}\setminus\{i\}} \left[\gamma_{ij} + \psi_{ij}^H\right] + \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}_{j,d_2}} \sum_{k\in\mathcal{N}_{j,d_2}\setminus\{i\}} \left(|\bar{\theta}_{ijk}| - \frac{\pi}{3}\right)^2. \tag{4.5.2}$$

For all $t < t_b^*$, $V_g(q)$ corresponds to:

$$V_g(q) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}\setminus\{i\}} \left[\gamma_{ij} + \psi_{ij}^H\right] + C_1, \text{ for } t < t_b^*$$

where $C_1 = \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}_{j,d_2}} \sum_{k\in\mathcal{N}_{j,d_2}\setminus\{i\}} \left(|\bar{\theta}_{ijk}| - \frac{\pi}{3}\right)^2$ is constant (recall the definition of $\bar{\theta}_{ijk}$). Under Assumption 4.2, it follows from Theorem 4.1 that $\dot{V}_g(q)$ is strictly negative such that the system will eventually reach a configuration where agents are separated from any other agent by a distance exactly equal to d_1 and where each agent has at most two neighbors.

At $t = t_b^*$, the compactness controller becomes active for at least one triplet (i,j,k) while the dispersion controller remains active for all agents (i,j) satisfying $\beta_{ij} < d_1^2$.

Therefore, we consider the function:

$$V_g(q) = \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}_{j,d_2}} \sum_{k\in\mathcal{N}_{j,d_2}\setminus\{i\}} \left(|\bar{\theta}_{ijk}| - \frac{\pi}{3}\right)^2 + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}\setminus\{i\}} \left[\gamma_{ij} + \psi_{ij}^H\right], \text{ for } t > t_b^*.$$

The second right hand term is related with the dispersion control strategy. Under Assumption 4.2, it follows from Theorem 4.1 that the closed-loop system controlled by (4.5.1) will converge to a configuration where all agents are separated from any other agent by a distance exactly equal to d_1. On the other side, the first term of the right hand side corresponds to the compactness evolution, which can be analyzed using Lemma 4.1 and Lemma 4.2. Under Assumption 4.2, and using Theorem 4.1 and Lemma 4.1 and 4.2, it yields:

$$\dot{V}_g(q) < 0, \text{ for } t \in]t_b^*, \infty[.$$

As mentioned before, the proposed sequential controller can be seen as a hybrid system, where transitions are generated by triplets (i, i, k) that become Triangles, see Definition 4.1. Recall the definition of $\bar{\theta}_{ijk}$ presented in (4.5.1). At each instant $t = t_{ijk}^*$, the value of $\bar{\theta}_{ijk}$ switches from π to the real value of θ_{ijk} for all triplet (i, j, k). The increment of $|\theta_{ijk}|$ will be denoted by $\Delta\bar{\theta}_{ijk}$. From (4.5.1), it yields:

$$\Delta\bar{\theta}_{ijk} < 0, \forall \text{ Triangle } (i, j, k) \ .$$

Denote the sum of increments of $V_g(q)$ by $\sum \Delta V_g(q)$. Thus, it yields from the previous equation:

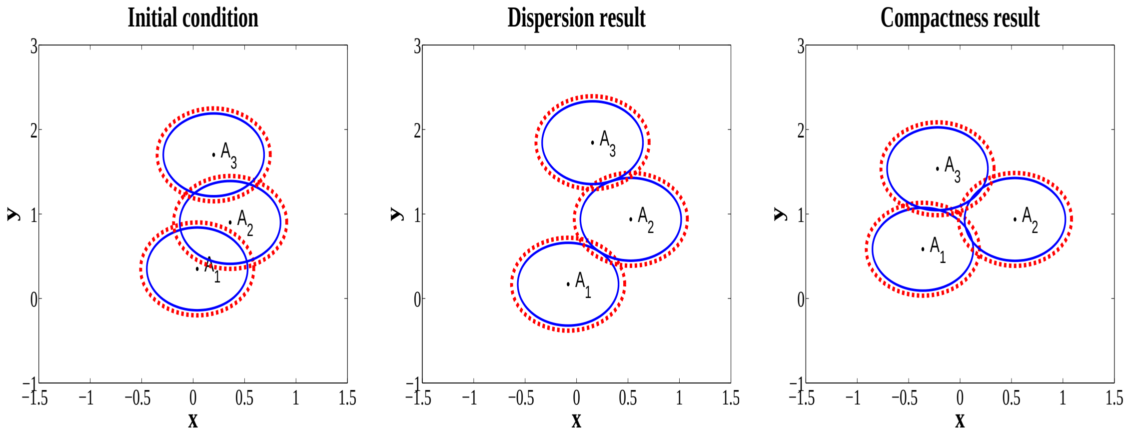
$$\sum \Delta V_g(q) < 0.$$

Finally, under Theorem 20 of [107] it holds that, since $\dot{V}_g(q) < 0$ for $t < t_b^*$, $\dot{V}_g(q) < 0$ for $t > t_b^*$, and $\Delta V_g(q) < 0$, the system is asymptotically stable. Agents will eventually reach a configuration belonging to $\mathcal{G}(d_1, d_2)$, where equilateral triangles are formed and all internal angles are equal to $\pi/3$. □

Throughout previous sections, we designed and analyzed controllers for dispersion and compactness. Improved strategies to avoid singular configurations were also derived. Moreover, we also proposed a sequential controller composed of both dispersion and compactness controllers. We are now going to present some simulation results supporting the theoretical contributions of this chapter.

4.6 Simulation results

In this section, we present some results that validate the technical developments of this chapter. We are particularly interested in the sequential controller, since the validation of its performances inherently leads us to the validation of the dispersion and compactness controllers.



(a) (b) (c)

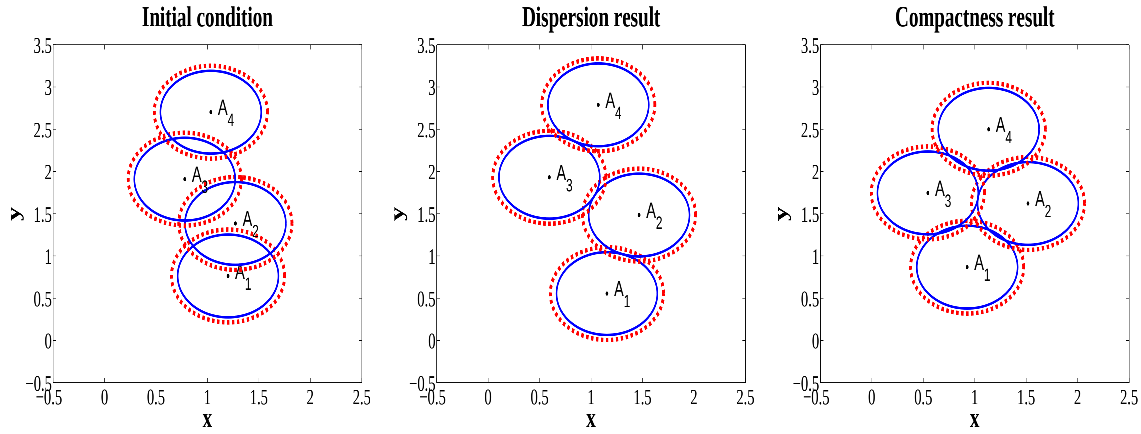
Figure 4.15: Simulation results for a configuration of three agents

The position of each agent i is defined by $q_i = [x_i, y_i]^T \in \mathbb{R}^2, i \in \mathcal{N} = \{1, ..., N\}$ such that $q = [q_i, q_j, \ldots, q_N]^T$. In the sequel, the initial global configuration vectors for a four, six and eight agent network are denoted $q^{4\ agents}$, $q^{6\ agents}$, and $q^{8\ agents}$, respectively. For simulations purposes, they are defined as:

$$\begin{aligned} q^{3\ agents}(0) &= \begin{bmatrix} 0.04, & 0.35, & 0.36, & 0.90, & 0.20, & 1.70 \end{bmatrix}^T, \\ q^{4\ agents}(0) &= \begin{bmatrix} 1.20, & 0.76, & 1.27, & 1.39, & 0.78, & 1.91, & 1.03, & 2.70 \end{bmatrix}^T, \\ q^{8\ agents}(0) &= [\ 1.20, \ 0.48, \ 0.76, \ 1.08, \ 0.81, \ 2.00, \ 0.60, \ 2.40 \\ &\quad 1.60, \ 0.08, \ 1.60, \ 0.64, \ 2.40, \ -1.00, \ 2.00, \ -1.72\ ]^T. \end{aligned}$$

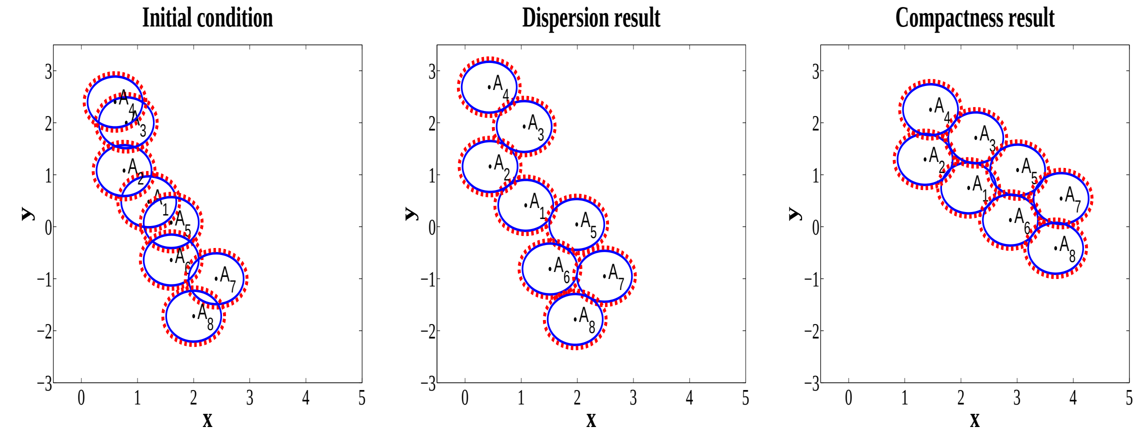
Figures 4.15, 4.16, 4.17 show the configuration evolution for several networks. More precisely, Figures 4.15(a), 4.16(a) show the initial configuration for a group of three and four agents, respectively. These figures confirm the theoretical results proving the stability of systems containing three and four agents. However, simulation results for bigger networks are also available, as for example in Figure 4.17 for a network of eight agents. Note, though, that this subject still demands further research and complete theoretical proofs are yet to be provided. In all figures, the red radius (dotted line) represents $d_2/2$, while the blue line (complete line) stands for radius $d_1/2$.

Figures 4.15(b), 4.16(b), 4.17(b) show the resulting formation of the dispersion algorithm. We observe that, for any pair of agents, they are located at a distance d_1 from each other, which means that no other agent is within a radius of $d_1/2$. Since the potential forces oblige agents to get apart and there is no pair (i, j) satisfying $q_i(0) = q_j(0)$, it is worth mentioning that coalisions are avoided by the proposed control strategy. Figures 4.15(c), 4.16(c), 4.17(c) show the ultimate resulting configuration. We can see that equilateral triangles have been formed, and that the internal angles were minimized to



(a) (b) (c)

Figure 4.16: Simulation results for a configuration of four agents



(a) (b) (c)

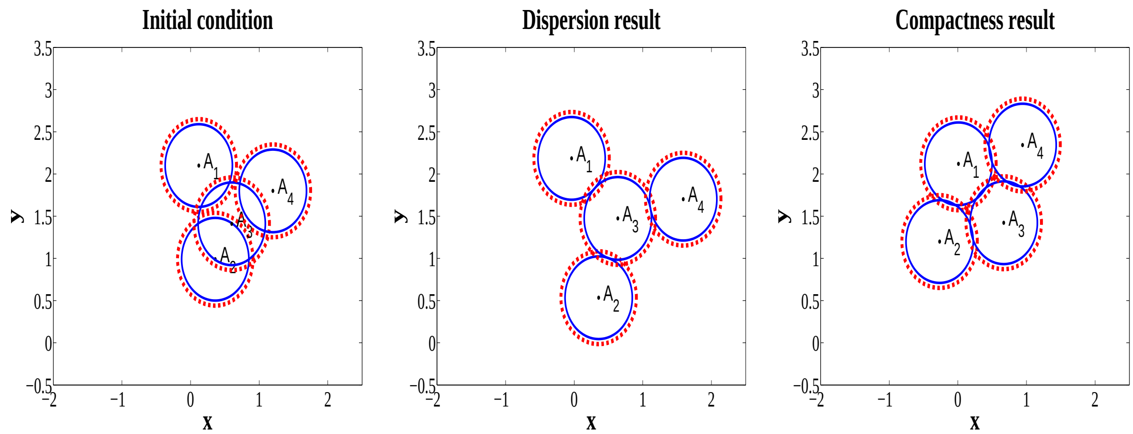
Figure 4.17: Simulation results for a configuration of eight agents

$\pi/3$ so that formation is the most compact possible. We also observe that agents continue to keep a d_1 distance from any other agent, which shows the compatibility of the different components of (4.5.1) and the efficiency of the proposed strategy.

Finally, Figure 4.18 show some results for a particular initial configuration depicted in Figure 4.18(a). The initial conditions for such formation are defined as:

$$q^{4\ agents}(0) = \left[\begin{array}{cccccccc} 0.12 & 0.36 & 0.6 & 1.2 & 2.1 & 0.99 & 1.41 & 1.8 \end{array}\right]^T.$$

In Figure 4.18(b) we can observe the resulting configuration when controller (4.4.1) is applied, leading us to a singular equilibrium where all angles are equal to $2\pi/3$. Then, Figure 4.18(c) shows the resulting formation when the controller (4.4.12), using a variable gain, is implemented. We can easily conclude the improved controller (4.4.12) satisfies our objective, avoiding the singular equilibrium visible in Figure 4.18(b).



(a) (b) (c)

Figure 4.18: Simulation results for a singular configuration of four agents

4.7 Conclusions

In this chapter, we consider the compact deployment of agents problem. This problem asks for effective motion coordination strategies and, in particular, for formation control approaches. This work is motivated by the fact that for recent robotic applications is important to impose a particular configuration. In this context, distributed control strategies that guarantee specific connectivity properties of the overall graph were derived.

The first development corresponds to an extension of existing potential based approaches for swarm dispersion. In this context, the main contribution presented in this chapter is a Theorem presented in Section 4.3 ensuring agents' dispersion while keeping graphs connectivity, *i.e.*, while avoiding edges to break due to agents' motion. Moreover, the proposed strategy also allows collision avoidance, an important feature to several mobile robot applications. The second contribution of this chapter consists of direct angle control using only relative positions. In fact, and to the best of the author's knowledge, the design of a control law capable of establishing a specific formation acting on inter-agent angles has not been addressed so far. In this chapter, four Lemmas and two Theorems for compact deployment are presented. These results show that the system will eventually reach a configuration where each three agents form a Compact Triangle (see definition in Section 4.2). A sequential controller composed of both dispersion and compactness control laws has also been formulated. The proof was derived using hybrid systems theory for switched systems. Furthermore, particular formations that led the initial system to local minima asked for an improvement of the initial strategy by considering variable gains. To enhance the technical developments of this chapter, simulation results for several initial configurations and variable number of agents were presented. The simulation setup considers 2-D motion in a cartesian plan. The resulting configurations are indeed as compact as possible such that the connectivity of the communication graph is optimized. Such feature is extremely important to several cooperative strategies for multi-robot systems. In particular, connectivity plays a crucial role on the efficiency and rapidity of distributed protocols. It also allows redundancy, useful for critical applications such as search/recovery and operations in hazardous environnements. Moreover, the resulting geometrically constrained formation may offer a interesting trade-off in some sensor driven applications as, *e.g.*, in gradient search maneuvers.

Chapter 5

Conclusions and future works

Contents

5.1 Review of the contributions and conclusions

Cooperative control is an important issue due to its large number of applications. Collaborative behavior of a group of agents means that there exist several interconnections between them in order to reach a common objective. This thesis pertains to distributed control strategies for multi-robot systems and its applications. Due to the distributed characteristics of multi-vehicle systems, the communication graph plays an important role in the efficiency of control strategies. This manuscript is particulary interested in studying this topic. More precisely, this dissertation presents pertinent results regarding the motion of a group of vehicles while keeping the best connectivity properties possible. The main contributions of this thesis were developed considering several assumptions on the agents' model, the communication graph and on the environment. An overview is presented in the sequel.

5.1.1 Consensus algorithms

Throughout a significant part of this dissertation a special attention is paid to consensus algorithms for heterogeneous agents, representing, for example, different models or generations of robots. One can easily state that the control complexity of systems considering heterogeneous agents is greater than for simpler frameworks. These works were motivated by some extremely useful applications demanding for cooperation among heterogenous agents such as search, recovery or surveillance operations in civil or military setups. An overview of challenging applications relaying in multi-robots systems is presented in Chapter 1. It is also worth it mentioning that despite the fact several works consider consensus algorithms for homogenous sets of agents, only a few works consider heterogeneous cases of the synchronization problem. Furthermore, most of them present major drawbacks such as calculation needs, complexity or accuracy of the solution. Chapter 2 proposes an efficient way to simplify the control design for this problem. In this context, the objective is the design of a distributed control law which ensures that:

- each subsystem is stable;
- the measurement vectors of each agent reaches an agreement.

More precisely, a control strategy based on consensus algorithms, which is decoupled from the original system, was designed in Chapter 2. In other words, we attributed to each agent an additional control variable which achieves a consensus and thus, the measurement variable of the each agent should converge with this additional variable.

In these works it is assumed that:

- the N systems are heterogeneous;
- each agent $i \in \mathcal{N}$ is controllable;
- for all $i \in \mathcal{N}$, the input vectors directly affects the measurement vector;
- for all agent $i \in \mathcal{N}$ the measurement vectors z_i represents the same quantity of interests for all agents;
- the communication graph has a directed spanning tree.

Classical distributed consensus algorithms have been intensively studied in literature. Inherently, their stability and performance properties are well documented. The well known convergence properties of simple integrator algorithms naturally motivate it as an appropriated "choice" for the additional dynamics. The rest of the contribution consists of using this trivial consensus algorithm to reach an agreement on those additional dynamics, while applying a model tracking based controller to the remaining system. This ensures that the real system will have identical performances as the additional model. The new algorithm offers the major advantage to separate the stability analysis of each agent and the convergence analysis of the distributed consensus algorithm. Therefore, it is possible to extend the previous control law to more general situations, where for instance the communication link induces transmission delays or when one considers distributed filters. Both cases were studied in Chapter 2. However, several other problems of multi-robot systems can be treated with the proposed approach, and specially simplified. The initial assumptions ask only for a fairly enough connected graph and that the input vectors directly affects the measurement vector. Moreover, the presented approach relays in a simpler and lighter structure with respect to [299], a recent result on heterogeneous Multi-Agent Systems (MAS) consensus.

Following the previous study, we were interested in knowing if it is possible to improve traditional convergence properties. Consequently, Chapter 3 focuses on consensus algorithm convergence rate. The speed of convergence of a consensus algorithm, also called algebraic connectivity, is equal to the second smallest eigenvalue of $\mathbf{L}$. Accelerating the convergence of synchronous distributed averaging algorithms have been studied in literature based on two main approaches: optimizing the topology-respecting weight matrix summarizing the updates at each node or incorporating memory into the distributed averaging algorithm. Chapter 3 pertains to dealing with this second approach. Even if for most applications delays lead to a reduction of performances or can even lead to instability, there exist some cases where the introduction of a delay in the control loop can help to stabilize a system. In Chapter 3, a state sampled component is added to the control law, which can be seen as an artificial way to manipulate $\mathbf{L}$'s eigenvalues by getting them further on the left part of the complex plan. The incorporation of memory in consensus controllers for both simple and double integrator consensus dynamics was studied in this manuscript, which proposes and analyzes an effective approach to accelerate the convergence of synchronous distributed agreement algorithms. In these works it is assumed that:

- the N systems are homogeneous;
- the communication graph has a directed spanning tree.

In a first part, the influence of both partial and global memory in consensus algorithms for simple integrator agents was studied. An important conclusion states that global memory drastically improves performances when compared to both trivial and partial memory based algorithms. An optimization method of the controller parameters is proposed so that exponential stability of the solutions is achieved. These results are based on discrete-time Lyaponov Theorem and expressed in terms of Linear Matrix Inequality (LMI). Also, analytical conditions for improved performances based on Laplacian's eigenvalues were provided. Simulation results show the efficiency of the proposed algorithm, as well as the conservation of averaging properties. On a second step, a new consensus algorithm for double integrator agents is proposed. For this specific case, Chapter 3 brings forward an improved algorithm that reduces the information quantity needed for control since velocity sensors' data is not considered. This inherently means economical, space and calculation savings. This work particulary shows the influence of artificial delays over the system's convergence, since the introduction of a delay in the control loop helped to stabilize an originally non-stable system. An optimisation method of controller parameters is proposed so that exponential stability of the solutions is achieved and an expression of the consensus equilibrium is derived with respect to the initial position and velocity of the swarm.

5.1.2 Compact formations

Chapter 4 addresses the design and analysis of an algorithm for compact agent deployment. This problem asks for effective motion coordination control strategies and in particular for formation control approaches. Some of the different approaches proposed in recent literature on formation control is provided at the beginning of the chapter. This work is motivated by the fact that for recent robotic applications it is important to impose a particular configuration. In this context, a special attention was paid to distributed control strategies that guarantee specific connectivity properties of the overall graph. In the proposed approach, the desired formation is entirely specified by the angles that agents within the formation form among themselves. Consequently, a completely distributed and leaderless algorithm that allows swarm's self-organization is proposed in Chapter 4. In these works it is assumed that

- the N systems are homogeneous;
- each agent has range based sensing habilites;
- the initial configuration satisfies particular conditions.

The first contribution corresponds to an extension of previous works on swarms dispersion. By adding a connectivity maintenance force, agents' dispersion is ensured while keeping graphs connectivity, *i.e.*, while avoiding edges to break due to agents' motion. Moreover, the proposed strategy also allows collision avoidance, an important feature to several mobile robot applications. The second contribution on this topic consists of direct angle control since to the best of our knowledge, the design of a control law capable of establishing a specific formation acting on inter-agent angles has not been addressed so far. To solve the compact agent deployment problem inter-agent angles are minimized in order to achieve the most compact configuration possible, motivated by the fact that for successful network operations the deployment should result in configurations that not only provide good environment coverage but also satisfy certain local and/or global constraints such as the node degree or the network connectivity. Finally, a sequential problem composed of both dispersion and compactness control laws is formulated. The proof has been derived using hybrid systems theory for switched systems. Moreover, since for some particular initial formations the original approach leads to particular and undesired configurations that can be explained by a balance between the inter-agent angles, an improved controller that considers variable gains was also derived.

5.2 Ongoing and future works

This dissertation proposes control strategies to carry out several applications of multi-robot systems. Some interesting areas of future work are outlined in the sequel. Firstly, some nearby extensions of the presented results are described before indicating more distant research directions.

5.2.1 Perspectives in consensus algorithms

This thesis considers linear MAS that offer good properties for stability analysis. But in some situations such models are too simple to describe the dynamics of a real agent. Even if an effective solution for linear heterogenous MAS is presented in Chapter 2, an adequate solution to deal with nonlinear heterogenous MAS is still to be provided. This is a logical extension of Chapter 2's results.

It is also known that asymptotically stabilizing control design is generally more accessible than finite-time control design, especially for the lack of effective analysis tools. Various finite-time stabilizing control laws have been proposed in literature as for instance in [21, 22, 91, 123, 295, 303, 320]. In particular, finite-time control design has been extended to nth order systems with both parametric and dynamic uncertainties in [124] and since non-smooth finite-time control synthesis can improve the system behaviors in some aspects like high-speed, control accuracy, and disturbance- rejection, finite-time control applied to first order agent dynamics using gradient flow and Lyapunov theory was proposed in [64]. Even if finite-time distributed control for homogeneous multi-agent systems was already tackled in literature as for example in [296], there is still free space for improvements concerning finite-time consensus algorithms for heterogenous MAS. Considering the decoupled algorithm presented in Chapter 2, those results can likely be extended to fit this problem and/or eventually simplify it.

Consider now the results on consensus convergence rate presented in Chapter 3. As mentioned before, an optimization method of controller parameters and inherent stability conditions were expressed in terms of LMI. The proposed solutions consist of off-line calculations that guarantee exponential stability for a given graph. Even though LMI remains a solid and a powerful tool for the analysis of delayed and sampled systems, they also present some drawbacks. In fact, their complexity will drastically increase for large networks which can compromise the feasibility of the proposed algorithms. Therefore, in order to overtake these drawbacks for large sized networks, an interesting research direction copes with developing other tools for the analysis of memory based consensus algorithms. This would considerably reduce the calculation needs and also the conservativeness of the initial LMI based conditions. Moreover, this could eventually lead us to on-line solutions that might be able to calculate the optimal controller setting in real time. On the other hand, an inherent assumption of these works is that all agents are synchronized and share the same clock to ensure that the agents also share the same sampling. It is also assumed that the sampling process is periodic. Even if this setup makes sense in the situation of multi-agents systems, these results might likely be extended to asynchronous samplings. Consequently, it is also logical to consider event based control strategies, a useful tool for modern control applications relying on heavy communication frameworks.

5.2.2 Perspectives in compact formations control

Chapter 4 considers the compact agent deployment problem for linear agents and in particular for simple integrator dynamics. One of the major challenges of this work are the non linearities raised by the angle based control approach. Despite the fact that agents' dynamics correspond to simple integrators, in Section 4.4 the trajectory of each agent is dependent on the non linear evolution of one or several inter-agent angles. Therefore, the simple initial system becomes through the chosen control strategy a highly non linear and constrained system. The presented works revealed that for particular initial formations the system reaches singular formations that do not satisfy the control objectives. Despite the fact that the system achieves a stable configuration, the control requirements are not fulfilled: the inter-agent angles reach an singular equilibrium, leading the system to a local minima. But this drawback was overtaken by considering variable gains. On the other hand, though stability results from dispersion algorithms for large groups of agents were provided in this manuscript, the compactness controller's analysis is for the moment only valid for three and four agents networks. However, Chapter 4 provides an endeavour to analyze networks of five or more agents, but formal stability analysis is still to be provided. Therefore, it is pertinent to analyse the efficiency of the proposed strategy for high order systems. This analysis should be done at two levels: one dealing with agent-to-agent relationships, and another, more abstract, dealing with edge-to-edge and edge-to-local communication graph relationships. Such an approach would give us insights on how to efficiently describe and formalize a multi-agent network based on "edges' neighborhood" instead of the commonly used "agents' neighborhood". Furthermore, one might consider to extend the previous result to non equal communication radii. This corresponds to a more realistic setup for several applications such as, for instance, ship-based radar systems recently studied in [253].

5.2.3 Perspectives in distributed labeling in artificial populations

Something that has not gained much attention until now is that in an ever changing dynamical environment the agents need to be able to communicate with each other to improve their adaptability and the overall performance of the group. In the animal kingdom this may be done via the secretion of pheromones or, in the particular example of humans, it is done by communicating via Language. This is the main motivation to study the possibility of artificial agents developing their own Language in the context of exploring their environment. It is possible to assume that there is no global frame of reference and there are certain prominent environmental features with unknown global positions, beacons or landmarks, such as a communication antenna or an urban landmark, that all the agents can recognize and can assign a label to. From a control point of view, it is interesting to investigate the possibility of agents determining each other's positions via communicating their own position relative to these locally chosen labeled beacons. To achieve this, it is of course necessary for the agents to agree on the same name for the landmarks, and in a sense develop a Language of their own. A fist work in this work is presented in [233].

Generally speaking, a logical extension of the results presented in this dissertation should focus on relaxed assumptions in order to consider more realistic situations. Here, we assume perfect communication between two connected vehicles. With a view to analyze the performance of the control algorithms previously presented in the presence of more realistic communication constraints, cooperative approaches dealing with packet loss, noise and time delays can be considered in further research. Furthermore, throughout this thesis we assume a two-dimensional kinematic model of the vehicles. In consequence, the motions and formations obtained are planar, *i.e.*, the vehicles are moving in a 2-D framework. It would be also pertinent to consider the possibility of developing three-dimensional motion controllers.

Appendix A

Fundamentals on stability of sampled-data systems

Contents

A.1 Context

This appendix looks forward to providing a sufficiently detailed description of sampled systems. The sequel completes the results on exponential stability analysis of memory based consensus algorithms presented in Chapter 3.

The following is based on [250], a relevant work on linear sampled systems. This article proposes a novel approach to assess stability of continuous linear systems with sampled-data inputs. The method, which is based on the discrete-time Lyapunov theorem, provides easy tractable stability conditions for the continuous-time model. Indeed, sufficient conditions for asymptotic and exponential stability are provided dealing with synchronous samplings. This appendix focuses on these results. However, [250] also copes with asynchrony, multiple samplings, packet losses and uncertain systems. But due to the lack of pertinence within the scope of this thesis, they will be excluded from the following discussion.

Networked control systems are generally composed of several distributed plants which are connected through a communication network. As mentioned before, the existence of such networks raises several challenges. In the context of this appendix, a heavy temporary load of computation in a processor can corrupt the sampling period of a given controller. In such situations, the variations of the sampling period will affect the stability properties. Thus an important issue is the development of robust stability conditions with respect to the variations of the sampling period.

Sampled-data systems have been extensively studied in literature, see [47, 98, 100, 316, 317] and the references therein. Before presenting important results for systems under periodic samplings, let us discuss some works on asynchronous samplings, a more realistic setup that sill lead to several open problems. Several articles drive the problem of time-varying periods based on a discrete-time approach [122, 187, 271]. An input delay approach using the Lyapunov-Krasovskii (LK) theorem is provided in [98]. Improvements are provided in [100, 169], using the small gain theorem and in [182] based on the analysis of impulsive systems. Recently [97, 156, 249] refine those approaches and obtain tighter conditions.

Considering the previously mentioned works, among many others, it is now reasonable to design controllers which guarantee the robustness of the solutions of the closed-loop system under periodic samplings. Based on [250], the sequel proposes a novel framework for the stability analysis of linear sampled-data systems using the discrete-time Lyapunov theorem and the continuous-time model of sampled-data systems. In particular, asymptotic and exponential stability criteria are derived from this method to support the results presented in Chapter 3. For sake of clearness, the sets $\mathbb{N}$, $\mathbb{R}^+$, $\mathbb{R}^n$, $\mathbb{R}^{n\times n}$ and $\mathbb{S}^n$ denote the sets of nonnegative integers, nonnegative scalars, n-dimensional vectors, $n \times n$ matrices and symmetric matrices of $\mathbb{R}^{n\times n}$, respectively. Define $\mathbb{K}$, as the set of differentiable functions from an interval of the form $[0,\ T]$ to $\mathbb{R}^n$, where $T \in \mathbb{R}^+$. The notation $|\cdot|$ and the superscript 'T' stand for the Euclidean norm and for matrix transposition, respectively. The notation $P > 0$ for $P \in \mathbb{S}^n$ means that P is positive definite. For any matrix $A \in \mathbb{R}^{n\times n}$, the notation $\mathrm{He}\{A\} > 0$ refers to $A + A^T > 0$. The symbols I and 0 represent the identity and the zero matrices of appropriate dimensions.

A.2 Problem statement

Let $\{t_k\}_{k\in\mathbb{N}}$ be an increasing sequence of positive scalars such that:

$$\bigcup_{k\in\mathbb{N}} [t_k,\ t_{k+1}) = [0,\ +\infty),$$

for which there exist two positive scalars $\mathcal{T}_1 \leq \mathcal{T}_2$ such that:

$$\forall k \in \mathbb{N}, \quad T_k = t_{k+1} - t_k \in [\mathcal{T}_1,\ \mathcal{T}_2]. \tag{A.2.1}$$

Consider the following sampled-data system:

$$\forall t \in [t_k,\ t_{k+1}), \quad \dot{x}(t) = Ax(t) + Bu(t_k), \tag{A.2.2}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ represent the state and the input vectors. The sequence $\{t_k\}_{k\in\mathbb{N}}$ represents the sampling instants of the controller. The matrices A and B are constant, known and of appropriate dimension. The control law is a linear state feedback, $u = Kx$ with a given gain $K \in \mathbb{R}^{m\times n}$. The system is governed by:

$$\forall t \in [t_k,\ t_{k+1}), \quad \dot{x}(t) = Ax(t) + BKx(t_k). \tag{A.2.3}$$

Integrating the previous differential equation, the dynamics of the system satisfy:

$$\begin{array}{ll} \forall t \in [t_k,\ t_{k+1}], & x(t) = \Gamma(t - t_k)x(t_k), \\ \forall \tau \in [0,\ T_k], & \Gamma(\tau) = \left[e^{A\tau} + \int_0^\tau e^{A(\tau-\theta)}\mathrm{d}\theta BK\right]. \end{array} \tag{A.2.4}$$

This equality leads naturally to the introduction of the following notation. For any integer $k \in \mathbb{N}$, define the function $\chi_k : \mathbb{K}$, such that, for all $\tau \in [0,\ T_k]$:

$$\left\{ \begin{array}{lll} \chi_k(\tau) = x(t_k + \tau) & = \Gamma(\tau)\chi_k(0), \\ \dot{\chi}_k(\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\chi_k(\tau) & = A\chi_k(\tau) + BK\chi_k(0). \end{array} \right. \tag{A.2.5}$$

The definition of χ_k yields $x(t_{k+1}) = \chi_k(T_k) = \chi_{k+1}(0)$.

If A, BK are constant and known and $T_k = \mathcal{T}$, the dynamics become:

$$x(t_{k+1}) = \Gamma(\mathcal{T})x(t_k).$$

The system is thus asymptotically stable if and only if $\Gamma(\mathcal{T})$ has all eigenvalues inside the unit circle. If T_k is time-varying, this does not hold anymore. Relevant stability analysis based on uncertain representations of $\Gamma(T_k)$ have been already investigated for instance in [121, 187, 271]. Note, though, that extensions to uncertain systems lead to additional difficulties induced by the definition of Γ. Concerning the analysis using continuous-time models, the input delay approach provided in [98] and refined in [97, 169, 182, 249] leads to relevant stability criteria since it is able to take into account uncertain systems. However, this approach is still conservative in comparison to discrete-time ones. The results of [250] establish a novel framework for the stability analysis of sampled-data systems.

A.3 Asymptotic and exponential stability analysis

In this section, we present asymptotic and exponential stability conditions for synchronous sampled-data systems.

A.3.1 Asymptotic stability criteria

The following theorem shows an equivalence between the discrete-time and the continuous-time approaches.

Theorem A.1. *Let $0 < \mathcal{T}_1 \leq \mathcal{T}_2$ be two positive scalars and $V : \mathbb{R}^n \to \mathbb{R}^+$ be a differentiable function for which there exist positive scalars $\mu_1 < \mu_2$ and p such that:*

$$\forall x \in \mathbb{R}^n, \quad \mu_1 |x|^p \leq V(x) \leq \mu_2 |x|^p. \tag{A.3.1}$$

Then the two following statements are equivalent.

(i) *The increment of the Lyapunov function is strictly negative for all $k \in \mathbb{N}$ and $T_k \in [\mathcal{T}_1,\ \mathcal{T}_2]$, i.e.,*

$$\Delta_0 V(k) = V(\chi_k(T_k)) - V(\chi_k(0)) < 0;$$

(ii) *There exists a continuous and differentiable functional $\mathcal{V}_0 : [0,\ \mathcal{T}_2] \times \mathbb{K} \to \mathbb{R}$ which satisfies for all $z \in \mathbb{K}$:*

$$\forall T \in [\mathcal{T}_1,\ \mathcal{T}_2] \quad \mathcal{V}_0(T, z(\cdot)) = \mathcal{V}_0(0, z(\cdot)), \tag{A.3.2}$$

and such that, for all $(k, T_k, \tau) \in \mathbb{N} \times [\mathcal{T}_1,\ \mathcal{T}_2] \times [0\ T_k]$,

$$\dot{\mathcal{W}}_0(\tau, \chi_k) = \frac{\mathrm{d}}{\mathrm{d}\tau}[V(\chi_k(\tau)) + \mathcal{V}_0(\tau, \chi_k)] < 0. \tag{A.3.3}$$

Moreover, if one of these two statements is satisfied, then the solutions of the system (A.2.3) are asymptotically stable.

Proof. Let $k \in \mathbb{N}$, $T_k \in [\mathcal{T}_1,\ \mathcal{T}_2]$ and $\tau \in [0,\ T_k]$. Assume that (ii) is satisfied. Integrating $\dot{\mathcal{W}}_0$ with respect to τ over $[0,\ T_k]$ and assuming that (A.3.2) holds, this leads to:

$$\int_0^{T_k} \dot{\mathcal{W}}_0(\tau, \chi_k)\mathrm{d}\tau = \Delta_0 V(k).$$

Then $\Delta_0 V(k)$ is strictly negative since $\dot{\mathcal{W}}_0$ is negative over $[0,\ T_k]$.
Assume that (i) is satisfied. Introduce the functional:

$$\mathcal{V}_0(\tau, \chi_k) = -V(\chi_k(\tau)) + \tau/T_k \Delta_0 V(k),$$

as in Lemma 2 in [205]. By simple computations, it is easy to see that it satisfies (A.3.2) and $\dot{\mathcal{W}}_0(\tau, \chi_k) = \Delta_0 V(k)/T_k$. This proves the equivalence between (i) and (ii). The function $\Gamma(\cdot)$ is continuous and consequently bounded over $[0,\ \mathcal{T}_2]$. Then Equation (A.2.4) proves that $x(t)$ and the continuous Lyapunov function uniformly and asymptotically tend to zero. □

The main contribution of the Theorem A.1 is the introduction of a new kind of Lyapunov functionals for sampled-data systems. Even though similar functionals were derived from the Lyapunov-Krasovskii (LK) theorem by several other authors [97, 182, 249], the relation between the discrete-time Lyapunov theorem and the LK theorem was not provided so far. Theorem A.1 not only proves that they are equivalent but also allows relaxing the constraint on the positivity of the functional. For a graphical illustration of the proof of Theorem A.1, please see Figure A.1.

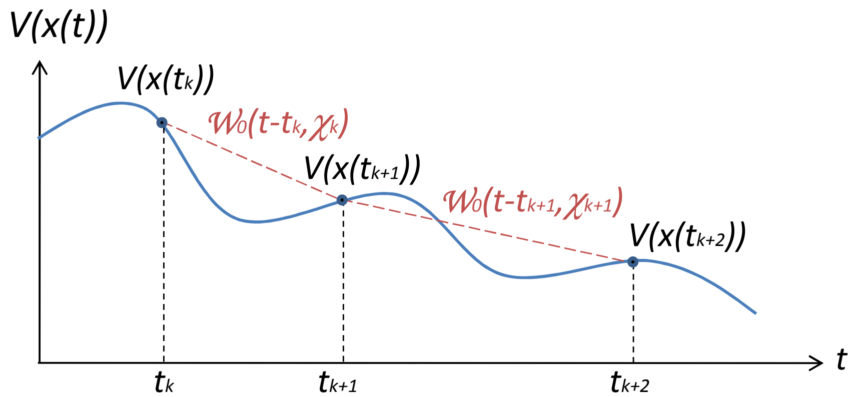


Figure A.1: Illustration of Theorem A.1 with $\mathcal{V}_0(T_k, \chi_k) = \mathcal{V}_0(0, \chi_k) = 0$.

A.3.2 Exponential stability criteria

In the scope of this thesis, and in particular in Chapter 3, we are interested in guaranteeing exponential stability with a guaranteed decay rate. In the sequel an extension of Theorem A.1 is presented.

Theorem A.2. *Consider positive scalars* α, $0 < \mathcal{T}_1 \leq \mathcal{T}_2$ *and a function* $V : \mathbb{R}^n \to \mathbb{R}^+$, *which satisfies* (A.3.1). *The two following statements are equivalent.*

(i) *The function* V *satisfies for all* $(k, T_k) \in \mathbb{N} \times [\mathcal{T}_1,\ \mathcal{T}_2]$

$$\Delta_\alpha V(k) = e^{2\alpha T_k} V(\chi_k(T_k)) - V(\chi_k(0)) < 0.$$

(ii) *There exists* $\mathcal{V}_\alpha : [0,\ \mathcal{T}_2] \times \mathbb{K} \to \mathbb{R}$ *satisfying*

$$\forall (T, z) \in [\mathcal{T}_1,\ \mathcal{T}_2] \times \mathbb{K}, \quad e^{2\alpha T} \mathcal{V}_\alpha(T, z(\cdot)) = \mathcal{V}_\alpha(0, z(\cdot)), \tag{A.3.4}$$

such that the functional $\mathcal{W}_\alpha(\tau, \chi_k) = e^{2\alpha\tau}[V(\chi_k(\tau)) + \mathcal{V}_\alpha(\tau, \chi_k)]$ *satisfies*

$$\forall (k, T_k, \tau) \in \mathbb{N} \times [\mathcal{T}_1,\ \mathcal{T}_2] \times [0,\ T_k], \quad \dot{\mathcal{W}}_\alpha(\tau, \chi_k) < 0.$$

Moreover, if one of these statements holds, the solutions of the system (A.2.3) *are exponentially stable with the rate* α.

Proof. Consider a given $\alpha > 0$ and a positive integer $k \in \mathbb{N}$. Following Theorem A.1, (ii) implies (i). Assume now that (i) holds. Consider the functional:

$$\mathcal{V}_\alpha(\tau, \chi_k) = -V(\chi_k(\tau)) + \Delta_\alpha V(k)/(e^{2\alpha T_k} - 1).$$

This functional satisfies (A.3.4) and

$$\dot{\mathcal{W}}_\alpha(\tau, \chi_k) = \frac{2\alpha T_k}{e^{2\alpha T_k} - 1} \frac{e^{2\alpha\tau}}{T_k} \Delta_\alpha V(k).$$

Since $(e^{2\alpha T_k} - 1)/2\alpha T_k$ is positive, for all α, $\dot{\mathcal{W}}_\alpha$ has the same sign as $\Delta_\alpha V(k)$. This proves the equivalence between (i) and (ii). The proof is concluded as in Theorem A.1. □

It is worth mentioning that even if $\alpha < 0$, Theorem A.2 still holds. In other words, it means that the solutions of the system can be unstable but the divergence rate of the solutions is not greater than α.

This appendix presents some results extracted from [250] that are necessary for a complete understanding of this thesis, and in particular of Chapter 3. More precisely, these results were used to study the $\alpha-$stability of the solutions of memory based consensus algorithms under a constant sampling period.

Appendix B

Résumé en Français

Sommaire

B.1 Préface

B.1.1 Introduction

De nos jours, beaucoup d'infrastructures et de services peuvent facilement être décrits par des réseaux d'unités qui interagissent entre eux. On peut trouver des exemples dans différents domaines tels que la biologie, les réseaux économiques et, bien sûr, les domaines technologiques. Par exemple : Internet, qui est basé sur des milliers de routeurs transmettant des informations partout dans le monde [264] ; les réseaux de distribution d'électricité, composés de centaines de groupes électrogènes qui doivent se synchroniser [203] ; certains systèmes de transport, composés de nombreux éléments [116]. Tous ces exemples exigent des systèmes de contrôles décentralisés où le processus ne réussit qu'au moment où tous les individus se mettent d'accord sur un certain nombre d'intérêts. Ce résumé en français n'a pas la prétention d'être un document complet sur les travaux de cette thèse. En effet, il n'en présentera que le contexte et les objectifs, ainsi que la description des parties les plus pertinentes. Pour l'intégralité des travaux, les preuves mathématiques, les résultats élargis ou pour toute information complémentaire, le lecteur est prié de se référer à la version anglaise de la thèse, présentée auparavant.

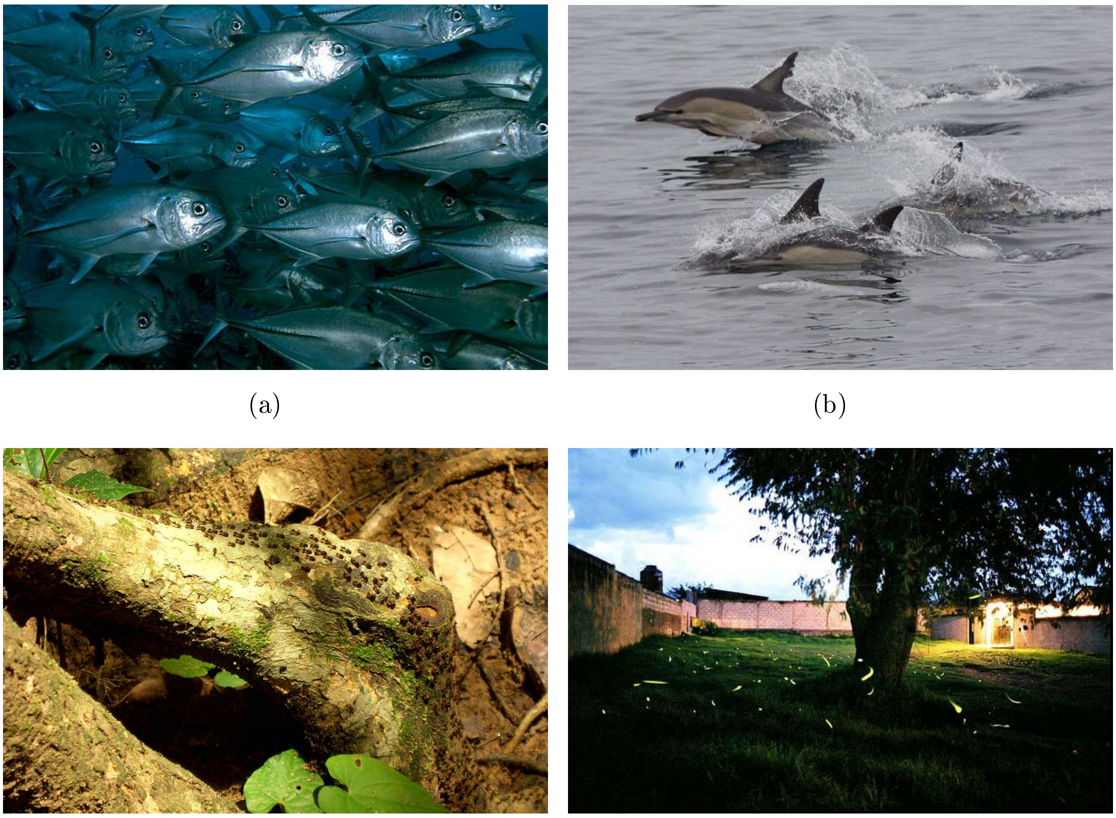
Bien que les organismes soient, par nature, compétitifs, la coopération entre eux est très répandue. Les gènes coopèrent dans le génome, les cellules coopèrent dans les tissus, les individus coopèrent dans les sociétés. Les sociétés animales, dans lesquelles des actions collectives émergent de la coopération entre individus, ont parfois une grande complexité sociale, voir Figure B.1. Des comportements coopératifs dans de grands groupes d'individus apparaissent en abondance dans la nature. Il existe des exemples bien connus de ces comportements : les bancs de poissons ou les nuées d'oiseaux, voir Figure B.2. La propriété fondamentale de cette coopération est que le comportement du groupe n'est pas dicté par l'un des individus [207] : chaque membre suit des règles très simples, chacun agissant suivant des informations locales. Aucun individu ne voit l'image complète. Aucun individu ne dit à l'autre quoi faire. Curieux et intrigués, quant au mode et à la raison de la formation de ces groupes et quant à la méthode selon laquelle les rôles joués par chacun y sont déterminés, les scientifiques ont tenté de modéliser ces systèmes, symboles d'une intelligence collective remarquable. La principale question est de savoir comment on peut simuler les différents comportements de coopération dont témoignent les populations d'oiseaux, d'insectes, etc., sur une population de structures/individus artificiels. En utilisant des algorithmes de consensus, un groupe d'agents indépendants doit être capable de résoudre des problèmes d'une manière plus efficace que ce qu'ils pourraient faire dans le cas où ils seraient contrôlés de manière centralisée. Craig Reynolds a été l'un des premiers à s'intéresser à cette intelligence collective. En 1987, le comportement d'un groupe d'oiseaux en mouvement a été modélisé et simulé dans [227]. Reynolds, qui a réussi à imiter le comportement de groupe, a mené une étude minutieuse sur les

FIGURE B.1 – Nuée de flamants roses volant en formation : des comportements auto organisés sont aperçus dans plusieurs systèmes biologiques, même si aucun individu n'a une connaissance globale du groupe. Cette image est propriété de António Luís Campos (www.antonioluiscampos.com). Photographie utilisée sous permission de l'auteur.

systèmes multi-agents. Ces comportements auto organisés, visibles dans les systèmes biologiques qui présentent des interactions distribuées entre individus, ont rapidement inspiré l'étude des mécanismes de coordination pour des robots mobiles. Le lecteur peut consulter [225] pour un aperçu récent des recherches dans ce domaine. Aujourd'hui, des robots autonomes sont utilisés de façon récurrente pour aider les humains à accomplir certaines tâches avec des performances améliorées et dans de meilleures conditions de sécurité. En effet, le déploiement de grands groupes de véhicules autonomes est désormais possible, grâce aux progrès des technologies réseau et à la miniaturisation des systèmes électromécaniques, ce qui permet d'accomplir une variété de tâches telles que des opérations de sauvetage et des manœuvres ou activités d'exploration dans des environnements dangereux. Cette thèse se focalise sur les algorithmes de contrôles nécessaires à ces systèmes robotisés et aux multiples problèmes relevés par ces applications complexes. Pour diverses applications, un groupe de robots pourra avoir besoin de se déployer sur une région, assumer une configuration spécifique, se réunir à une localisation déterminée ou se déplacer de manière synchronisée. Cette capacité exige, évidemment, des méthodes de coordination en vue d'un objectif commun. En plus, ces tâches doivent souvent être réalisées avec un minimum de communication et, par conséquent, avec des informations limitées sur le système. Une analyse récente sur ces systèmes de coordination distribuée peut être trouvée dans [165].

Cette thèse porte sur des stratégies de contrôles coopératifs. Ses principales contributions sont présentées ci-dessous :

Algorithmes de rendez-vous pour des systèmes multi-agents Cette thèse porte une attention particulière aux protocoles de consensus. Une grande partie du document se focalise sur des algorithmes de consensus pour des agents hétérogènes qui peuvent représenter, par exemple, différents modèles ou générations de robots. Considérant le fait que seuls quelques travaux étudient ce problème, une stratégie de contrôle est proposée, où l'algorithme de consensus est découplé du système original. Le nouvel algorithme offre l'avantage d'une analyse séparée de la stabilité de chaque agent et de celle de l'algorithme de consensus. Finalement, un deuxième aspect de ce travail met l'accent sur le taux de convergence des algorithmes de consensus. En particulier, des protocoles avec mémoire sont proposés, en utilisant le concept du délai stabilisant.



(a) (b) (c) (d)

FIGURE B.2 – De gauche à droite, un groupe de poissons, de dauphins, de fourmis et de lucioles. Ces images ont été postées sur Flickr par Tom Weilenmann, Oldbilluk, Jonathan Pio and Lastbeats, respectivement. Elles ont été utilisées dans le cadre de la licence CC-BY-2.0. (www.creativecommons.org/licenses/by/2.0/deed.fr).

Algorithmes de déploiement pour des systèmes multi-robots La principale contribution à ce sujet est un algorithme pour le déploiement compact d'agents. Dans notre approche, la configuration souhaitée est entièrement spécifiée par les angles formés entre agents. Nous avons proposé une solution totalement distribuée qui améliore les propriétés de connectivité du graphe de communication.

B.1.2 Contexte de la thèse

Cette thèse a été élaborée au sein deNetworked Control System Team, une équipe appartenant au Grenoble Images Parole Signal Automatique-Laboratoire et à l'INRIA. Elle fait partie de FeedNetBack [1], financé par la Commission européenne, et du projet Connect [2], financé par l'Agence Nationale de la Recherche. Ces deux projets étudient les systèmes commandés en réseau, Networked Control Systems (NCS en anglais), avec une attention particulière portée aux problèmes du contrôle des systèmes multi-agents, c'est-à-dire, des systèmes composés de plusieurs sous-systèmes interconnectés par un réseau de communication hétérogène. Le défi principal de ces projets est d'apprendre à concevoir des contrôleurs, prenant en compte les contraintes sur la topologie du réseau et sur la possibilité de partager des ressources informatiques, tout en préservant la stabilité du système. Un cas d'étude commun à ces deux projets se concentre sur la commande coopérative d'un groupe de véhicules mobiles. Une partie de ces travaux a aussi été développée dans le cadre du Groupement International de Recherche DelSys, du Centre National de la Recherche Scientifique, qui regroupe plusieurs acteurs scientifiques majeurs dans ce domaine.

B.1.3 Structure du document

Chapitre 1 : Introduction

Le but de ce chapitre est de détailler les principaux sujets liés à cette thèse et de donner un aperçu du travail réalisé. Dans un premier temps, nous présentons quelques comportements coopératifs pour des systèmes multi-agents. La deuxième partie de l'introduction porte sur les outils et les approches nécessaires pour ces algorithmes collaboratifs. Les différentes applications pour des systèmes multi-agents et les diverses approches présentées dans la littérature sont aussi décrites. Finalement, nous y présentons les principaux défis techniques et les contributions de ce document à ce sujet.

1. www.feednetback.eu/
2. www.gipsa-lab.inpg.fr/projet/connect/

Chapitre 2 : Stratégies de consensus pour des systèmes multi-agents hétérogènes

Le premier objectif de la thèse est le design d'algorithmes de consensus pour des agents hétérogènes qui représentent, par exemple, différents modèles ou générations de robots. Peu de travaux s'intéressent à ce problème. Les conditions nécessaires et suffisantes pour la synchronisation ont été récemment étudiées dans [299]. Dans ce chapitre, nous proposons une stratégie basée sur un algorithme de consensus découplé du système d'origine. En d'autres termes, nous attribuons à chaque agent une variable de contrôle supplémentaire qui atteint un consensus, tout en garantissant une convergence de la variable d'intérêt de chaque agent vers la variable supplémentaire correspondante. Le nouvel algorithme offre le grand avantage d'une analyse séparée de la stabilité de chaque agent et de celle de l'algorithme de consensus. Cette conclusion signifie fondamentalement qu'il est possible d'utiliser d'autres lois de commande plus générales comme, par exemple, des protocoles de consensus avec retards [179, 177, 251] ou des filtres distribués [195]. Ces deux cas sont étudiés dans cette thèse.

Chapitre 3 : Algorithmes de consensus améliorés par un échantillonnage approprié

Alors que dans le chapitre précédent nous nous sommes concentrés sur la conception d'algorithmes de consensus, dans celui-ci nous porterons une attention particulière à leur vitesse de convergence. La vitesse de convergence d'un algorithme de consensus s'avère être égale à la deuxième plus petite valeur propre de $\mathbf{L}$, aussi appelée connectivité algébrique. Les moyens pour accélérer la convergence des algorithmes de synchronisation ont déjà été étudiés dans la littérature d'après deux approches principales : l'optimisation de la matrice correspondante à la topologie de communication [304] ou l'ajout de mémoire dans l'algorithme. Cette approche sera étudiée dans ce document. Bien que la présence de retards puisse conduire, normalement, à une réduction des performances ou même à l'instabilité, il y a des cas où l'introduction d'un retard dans la boucle de commande peut aider à stabiliser le système. Cela a été étudié dans [110] et [252]. Pour la deuxième approche, le fait d'ajouter une composante échantillonnée à la loi de commande peut être considéré comme un moyen artificiel pour manipuler les valeurs propres de la matrice Laplacienne et cela signifie, implicitement, que la vitesse de convergence changera. Notre objectif est donc de maximiser cette valeur.

Chapitre 4 : Stratégies pour le déploiement compact d'un système multi-robots

Ce troisième chapitre porte sur la conception et l'analyse d'un algorithme de déploiement d'un ensemble de robots mobiles. Dans notre approche, la configuration souhaitée est entièrement spécifiée par les angles formés entre les agents. Nous proposons un algorithme complètement hiérarchisé et distribué qui permet l'auto organisation du système. La première contribution correspond à une extension de [76] dans des protocoles de dispersion, où une fonction de maintien de la connectivité a été ajoutée. Chaque agent est équipé d'une fonction potentielle qui, simultanément, permet de l'isoler de tout autre agent et de maintenir la connectivité parmi les agents. La deuxième contribution consiste en un protocole de contrôle des angles inter-agents. Ceci semble être la grande contribution de ce chapitre, étant donné que la synthèse d'une loi de commande, capable d'établir une formation spécifique en agissant directement sur les angles inter-agents, n'a pas été abordée jusqu'à présent. Deux problèmes indépendants ont été traités séparément : la dispersion et la compacité. L'analyse de ces deux stratégies a été faite de façon indépendante, mais une structure séquentielle regroupant les deux composantes a aussi été étudiée. Quelques arguments et calculs, confirmant qu'un tel système correspond à un système hybride, sont aussi présentés.

Chapitre 5 : Conclusions et perspectives

Dans ce dernier chapitre de la thèse, nous résumons les contributions de ce travail et nous en décrivons les extensions et les évolutions possibles. De plus, l'Annexe A reprend les principes fondamentaux des systèmes échantillonnés nécessaires à certains développements théoriques présentés dans cette thèse.

B.2 Stratégies de consensus pour des systèmes multi-agents hétérogènes

B.2.1 Contexte

Cette thèse a pour but de répondre aux défis soulevés par les systèmes multi-agents et leurs applications ou, tout au moins, d'apporter des solutions partielles à quelques-uns des problèmes qui en découlent. Ce document est structuré en deux parties, voir Figure B.3. Dans la première partie, composée des deux prochaines sections, on se concentre sur les algorithmes de consensus. En particulier, le présent chapitre s'occupera des algorithmes de consensus pour des systèmes multi-agents hétérogènes. Ces algorithmes seront validés dans des applications de rendez-vous [62, 74, 308].

Un algorithme (ou protocole) de consensus est une loi de contrôle coopératif qui a pour objectif de parvenir à un accord au sujet d'une certaine quantité d'intérêt, ne se basant que sur une interaction entre un certain agent et tous ses voisins [220]. Dans un algorithme de consensus, la topologie de communication peut être représentée par le biais d'une matrice, appelée Laplacienne et normalement notée $\mathbf{L}$, voir [23, 117, 238]. Une caractéristique importante de cette matrice est que ses valeurs propres définissent le comportement du système [106, 168]. On trouve plusieurs contributions dans ce domaine : des protocoles considérant des retards de communication [46, 69, 173, 194, 304], des protocoles non linéaires,[14, 194, 258], des algorithmes stochastiques [113], parmi beaucoup d'autres [40, 192, 195, 219, 251, 262].

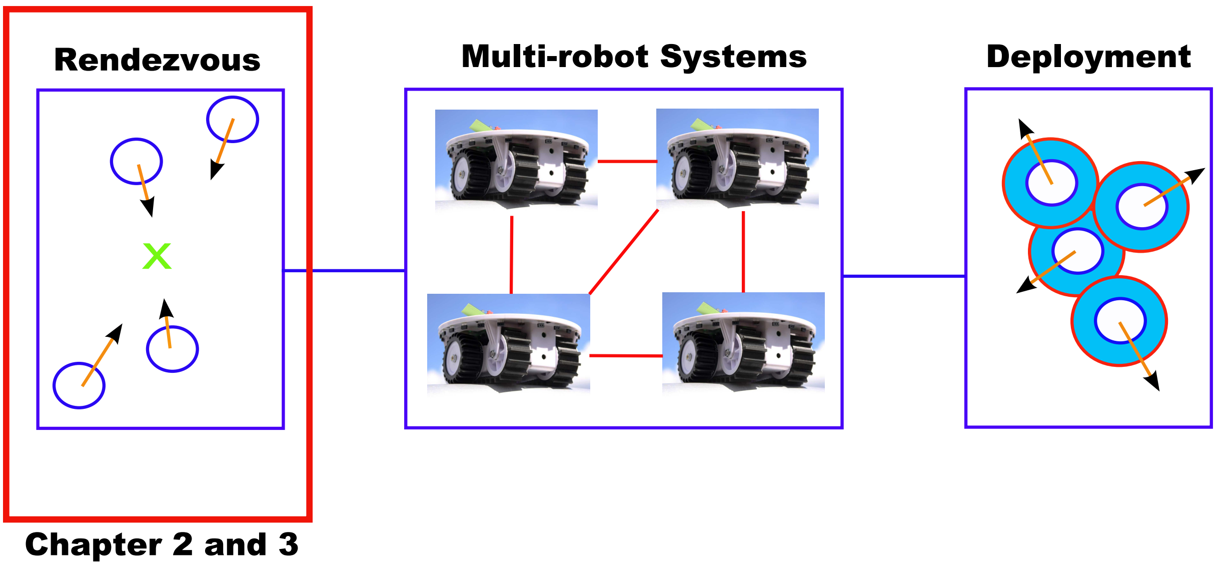


FIGURE B.3 – Contexte des chapitres 2 et 3

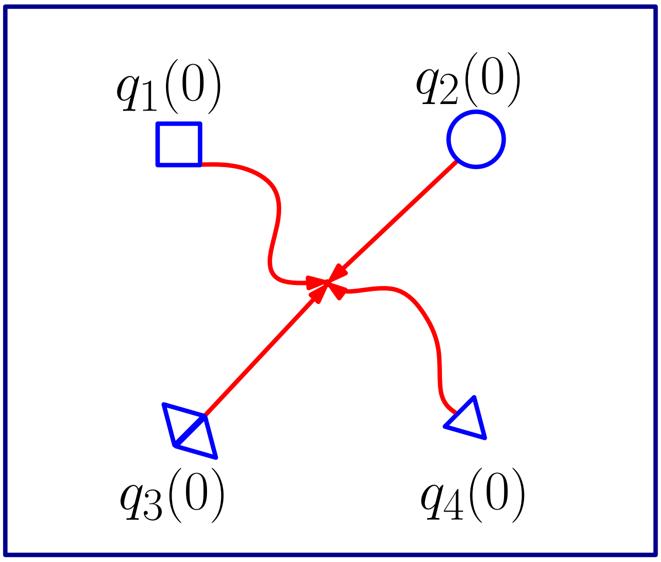


FIGURE B.4 – Illustration d'un protocole de rendezvous pour des systèmes hétérogènes.

Dans ce chapitre, nous nous intéressons à des algorithmes de consensus pour des systèmes multi-agents hétérogènes, c'est-à-dire, avec des dynamiques non identiques représentant, par exemple, différents modèles ou générations de robots. Si on considère un système constitué d'agents dynamiques hétérogènes, la première question à se poser est de savoir s'il existe une solution de consensus pour ce système. Seuls quelques articles abordent le problème de synchronisation dans le cas hétérogène. En particulier, le problème de synchronisation en sortie avec une approche non linéaire est considérée dans [51, 211]. Des résultats récents se limitent aux systèmes dynamiques linéaires hétérogènes [133, 299, 320, 322]. Des protocoles de consensus pour agents hétérogènes linéaires appliqués à un problème de contrôle de formation sont présentés dans [130], et des résultats sur des systèmes multi-agents hétérogènes composés d'intégrateurs simples et doubles sont présentés dans [320, 322, 320, 321]. En plus, les auteurs de [299] étudient la synchronisation selon une approche modèle interne. Le même problème est analysé dans [133], en tenant compte des incertitudes sur les modèles des agents.

Ce chapitre présente une solution pour le problème de consensus quand on considère un groupe de robots hétérogènes, portant une attention particulière aux protocoles de rendez-vous, voir Figure B.4.

En particulier, on propose une stratégie où le protocole de consensus est découplé du système original. Dans d'autres termes, on attribue à chaque agent une variable additionnelle de contrôle suivant une dynamique de consensus triviale, tandis qu'une loi de suivi de trajectoire est définie entre la variable de mesure (variable d'intérêt) et cette même variable additionnelle. Le nouvel algorithme offre l'avantage d'une analyse séparée de la stabilité de chaque agent et de celle de l'algorithme de consensus.

B.2.2 Définition du problème

Considérons un graphe $\mathcal{G}$ avec N agents et un ensemble d'arêtes défini par $E = \{(i,j) : j \in \mathcal{N}_i\}$. La matrice d'adjacence peut être définie par $\mathcal{A} = \mathcal{A}(G) = (a_{ij})$, une matrice de dimension $N \times N$ avec $a_{ij} = 1$, si $(i,j) \in E$ et $a_{ij} = 0$ sinon.

S'il y a une arête connectant deux nœuds i, j, $i.e.$,$(i,j) \in E$, alors i, j sont désignés voisins. Le *degrée* d_i est défini comme le nombre de voisins d'un nœud i, c'est-à-dire, $d_i = \#j : (i,j) \in E$, tel que $d_{max} = max\{d_i\}$. Soit Δ une matrice diagonale de dimension $N \times N$, où les élements de la diagonale sont donnés par d_i's. Alors la matrice Laplacienne correspondant à $\mathcal{G}$ est définie par $L = \Delta - \mathcal{A}$.

Dans le vaste domaine des systèmes multi-agents, nous sommes particulièrement intéressés par les applications de contrôle de robots mobiles dans un plan cartésien. La position d'un robot i est définie par :

$$q_i = [x_i, y_i]^T \in \mathbb{R}^2,$$

où x_i et y_i représentent les dynamiques de q_i dans les axes x et y, respectivement. Cependant, pour des questions de notation et de clarté, nous n'avons pris en compte dans ce chapitre que les dynamiques de chaque agent sur l'axe x.

Soit le système multi-agents suivant :

$$\begin{cases} \dot{x}_i = \bar{A}_i x_i + B_i u_i \\ y_i = x_i \\ z_i = C_i x_i \end{cases} \qquad \forall i \in \mathcal{N} = \{1, \ldots, N\}, \tag{B.2.1}$$

où $x_i \in \mathbb{R}^{n_i}$, $y_i \in \mathbb{R}^{n_i}$, $z_i \in \mathbb{R}^m$ et $u_i \in \mathbb{R}^m$ sont les vecteurs des variables d'état, de sortie, des variables de mesure et d'entrée, respectivement. On remarque qu'on suppose être capable d'accéder à l'état du système, c'est-à-dire, $y_i = x_i$ pour tout $i \in \mathcal{N}$. De plus, et pour tout $i \in \mathcal{N}$, on suppose que les matrices $\bar{A}_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m}$ et $C_i \in \mathbb{R}^{m \times n_i}$, avec $m < min\{n_i\}$ et $n_i > m$ sont constantes et connues.

Dans ce contexte, on a pour objectif la synthèse d'une loi de contrôle distribué garantissant que **(i)** chaque système est stable et **(ii)** que les variables de mesures des systèmes atteignent un consensus. Les hypothèses suivantes sont considérées :

Hypothèse B.1. ***(Hétérogénéité) :*** *On suppose que les N agents sont hétérogènes.*

Dans d'autres termes, ceci implique que les dynamiques peuvent varier d'un agent à l'autre et que les vecteurs x_i peuvent avoir des dimensions différentes.

Hypothèse B.2. ***(Homogénéité du vecteur de mesure) :*** *Les vecteurs de mesure z_i sont censé représenter la même quantité d'intérêt pour chaque agent. Donc, les vecteurs de mesure ont la même dimension, c'est-à-dire, $z_i \in \mathbb{R}^m$ $\forall i \in \mathcal{N}$, où $m < min\{n_i\}$.*

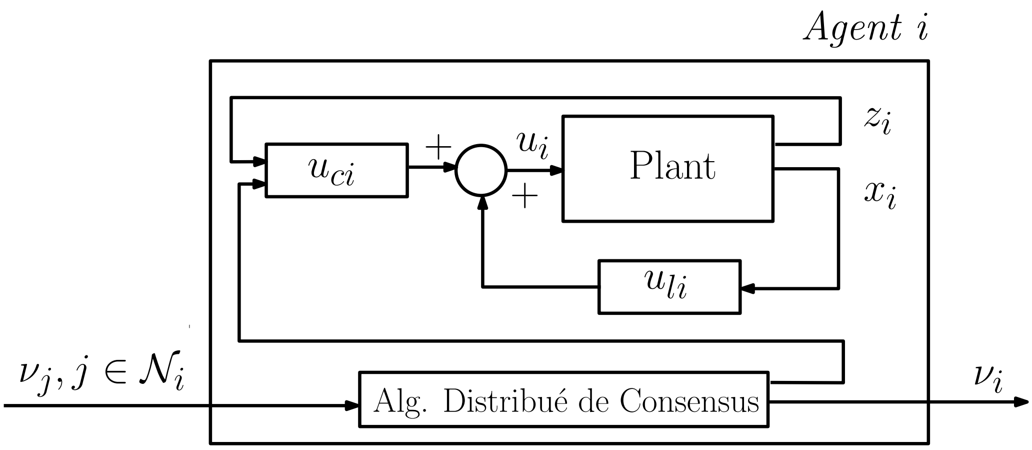


FIGURE B.5 – Structure de contrôle pour tout agent i

Hypothèse B.3. ***(Structure du système) :*** *Pour tout $i \in \mathcal{N}$, la contrainte $rang(C_iB_i) = m$ est satisfaite.*

Ceci signifie que les vecteurs d'entrée affectent directement les vecteurs de mesure.

Hypothèse B.4. ***(Contrôlabilité) :*** *Pour chaque agent, la paire (A_i, B_i) est contrôlable.*

On suppose aussi qu'il existe un graphe de communication avec une structure qui garantit qu'une seule valeur propre est égale à zéro et que le vecteur propre correspondant est un vecteur unitaire. En d'autres termes, ceci signifie qu'un consensus pourra être atteint asymptotiquement [220].

L'objectif est de développer des lois de commande efficaces pour des systèmes multi-agents hétérogènes. On propose ici une stratégie de commande, découplée du système d'origine, et qui est basée sur des algorithmes de consensus communs. En d'autres termes, nous attribuons à chaque agent une variable de contrôle supplémentaire qui permet d'atteindre un consensus, tout en garantissant que la grandeur d'intérêt de chaque agent converge vers cette variable supplémentaire. La méthode de synthèse de ces nouveaux contrôleurs est présentée dans la section suivante.

B.2.3 Synthèse des lois de commande

Afin d'atteindre les objectifs mentionnés auparavant, on propose dans la suite un contrôleur composé de deux parties, l'une correspondant à un contrôleur local et une autre représentant l'algorithme de consensus.

La loi de commande pour chaque agent est décrite par :

$$u_i(t) = u_{li}(t) + u_{ci}(t), \quad i \in \mathcal{N}, \tag{B.2.2}$$

où u_{li} et u_{ci} sont respectivement le contrôleur local et l'algorithme de consensus. Une illustration de la stratégie de contrôle proposée est présentée dans la Figure B.5.

B.2.4 Loi de commande locale

D'après hypothèse B.4, il existe, pour chaque système, une loi de retour d'état donnée par :

$$u_{li} = -K_i x_i, \tag{B.2.3}$$

tel que $A_i = \bar{A}_i - B_i K_i$ est *Hurwitz*. Donc, il suit que :

$$\begin{cases} \dot{x}_i = A_i x_i + B_i u_{ci}, \\ z_i = C_i x_i, \end{cases} \qquad \forall i \in \mathcal{N} = \{1, \ldots, N\}. \tag{B.2.4}$$

L'objectif est de proposer un algorithme de consensus qui garantisse la convergence des vecteurs de mesure vers une valeur de consensus. Dans différents travaux de la littérature, des algorithmes de consensus distribués considérés classiques ont été intensivement étudiés, et tant leur stabilité que leurs propriétés sont donc bien documentées, voir, par exemple, [120, 192, 220] et les références à l'intérieur.

B.2.5 Loi de commande distribuée

Dans cette section, nous proposons un protocole de consensus qui garantit que les vecteurs de mesure parviennent à un accord.

On comprend facilement que la complexité des systèmes composés d'agents hétérogènes est plus élevée que celle d'un système homogène. En outre, même si certaines solutions pour un tel problème ont été proposées dans la littérature, la plupart d'entre elles présente des inconvénients tels que les besoins de calcul ou la complexité de la solution. Dans ce document, on simplifie la synthèse des contrôleurs pour ces systèmes. Plus précisément, l'idée principale est l'ajout de dynamiques supplémentaires correspondant à un algorithme de consensus dit simple. Comme point de départ, assumons que ces dynamiques sont décrites par :

$$\dot{\nu}_i = -\sum_{j \in \mathcal{N}_i} (\nu_i - \nu_j), \quad \forall i \in \mathcal{N}, \tag{B.2.5}$$

où $\nu_i \in \mathbb{R}^m$. Considérez $\nu = [\nu_1, \ldots, \nu_N]^T$. Le précédent système peut donc être réécrit suivant $\dot{\nu} = -\mathbf{L} \otimes I_m \nu$, où $\mathbf{L}$ est la matrice Laplacienne associée au graphe de communication et $\otimes$ représente le produit de Kronecker. La stabilité d'un tel système a été largement étudiée dans la littérature comme par exemple dans [192, 220], parmi beaucoup d'autres. Les bonnes propriétés de convergence de l'équation (B.2.5) nous conforte dans notre choix d'utiliser ce système comme dynamique supplémentaire. Le reste de l'approche consiste à utiliser cet algorithme de consensus bien connu pour parvenir à un accord sur ces dynamiques supplémentaires, tout en appliquant une loi de suivi entre le système original et celui-ci. Cette approche garantit que le système réel aura des performances identiques à celles du modèle supplémentaire. La synthèse de u_{ci} doit considérer les deux objectifs suivants :

$$\begin{cases} \lim_{t\to\infty} (\nu_i - \nu_j) = 0, \\ \lim_{t\to\infty} (z_i - \nu_i) = 0, \end{cases} \quad \forall i, j \in \mathcal{N}^2. \tag{B.2.6}$$

Dans cette approche, le système (B.2.5) sera considéré comme le modèle de référence pour (B.2.1). Définissons le vecteur d'erreur entre le vecteur de mesure z du système (B.2.1) et le vecteur de variables additionnelles ν comme suit :

$$\varepsilon_i = z_i - \nu_i. \tag{B.2.7}$$

Le but est donc d'assurer que ε_i convergera vers zéro. Donc, l'évolution de ε_i doit satisfaire l'expression suivante :

$$\dot{\varepsilon}_i = -\beta\varepsilon_i,$$

où $\beta > 0$. Ceci signifie que :

$$\begin{aligned} \dot{z}_i - \dot{\nu}_i &= -\beta(z_i - \nu_i), \\ C_i(A_i x_i + B_i u_{ci}) + (\mathbf{L})_i \otimes I_m \nu &= -\beta\varepsilon, \end{aligned} \tag{B.2.8}$$

où $(\mathbf{L})_i$ représente la i^{me} ligne de la matrice $\mathbf{L}$. D'après hypothèse B.3, C_iB_i est inversible pour tout agent i, et donc le contrôleur proposé peut être décrit comme une loi de suivi de trajectoire classique, voir [126]. Donc, on peut écrire que :

$$u_{ci} = (C_iB_i)^{-1} \left(\dot{\nu}_i - \beta(z_i - \nu_i) - C_iA_ix_i\right). \tag{B.2.9}$$

B.2.6 Extensions à des situations plus complexes

C'était notre intuition qu'il était possible de trouver une stratégie peu complexe et avec peu de contraintes sur les moyens de calcul dans le cadre de la synthèse de protocoles de consensus pour des systèmes hétérogènes. Comme mentionné auparavant, le grand avantage de la méthode proposée réside dans une analyse découplée entre les différents composants du système. Par conséquent, il est possible d'étendre la structure précédente à des situations plus complexes, en considérant par exemple des retards de communication [177, 179, 251] ou des filtres distribués [195]. Dans ce document, on a considéré, en tant que dynamiques supplémentaires, les deux cas suivants :

- **Protocoles de consensus considérant des retards de communication**
  D'après les résultats de [176, 179], la loi de commande correspondante est donnée par :

$$\dot{\nu}_i(t) = K \sum_{j=1}^{N} \frac{a_{ij}}{d_i} (\nu_j(t - \tau_{ij}) - \nu_i(t)) \quad i \in \{1, \dots, N\}. \tag{B.2.10}$$

- **Protocoles de consensus considérant des références externes**
  D'après les résultats de [195], la loi de commande correspondante est donnée par :

$$\dot{\nu}_i = -\alpha \sum_{j \in \mathcal{N}_i} (\nu_i - \nu_j) + \sum_{j \in \mathcal{J}_i} (r_j - \nu_i), \tag{B.2.11}$$

  où $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$ et $r = [r_1, \dots, r_N]^T$ répresente les signaux/références externes.

B.2.7 Résultats théoriques

L'étude des protocoles de consensus distribués appliqués à des systèmes où les agents sont hétérogènes nous a conduits à plusieurs résultats théoriques de cette thèse. Ces résultats ont été aussi soutenus par des résultats de simulation qui seront présentés plus tard dans ce document. Considérant la structure de contrôle présentée dans les sections précédentes, voici notre résultat principal :

Théorème B.1. *(Rodrigues de Campos et al. [228]) Si les hypothèses B.1-B.4 sont satisfaites, alors la loi de commande* (B.2.2)*, donnée par :*

$$u_i = -K_i x_i + (C_i B_i)^{-1} (\dot{\nu}_i - \beta(z_i - \nu_i) - C_i A_i x_i) \tag{B.2.12}$$

où $\dot{\nu}_i$ est défini dans (B.2.5)*, et garantit que le système multi-agents* (B.2.1) *est stable et atteint un consensus sur les variables de mesure, c'est-à-dire , $z_i = z_j, \forall i, j \in \mathcal{N}$.*

En plus, ce résultat a été étendu à des situations et à des systèmes plus complexes, en considérant différentes formulations pour les dynamiques supplémentaires présentées dans (B.2.10) and (B.2.11). L'intégralité de ces résultats est fournie dans le Chapitre 2.

B.2.8 Résultats de simulation

Dans cette section, on affiche quelques résultats de simulation concernant l'efficacité de l'approche présentée précédemment. Dans la suite, on explore un scénario réaliste considérant un groupe de robots aériens, voir Figure B.6. On prend un groupe de $N = 4$ agents hétérogènes, définis par des matrices de différentes structures et dimensions.

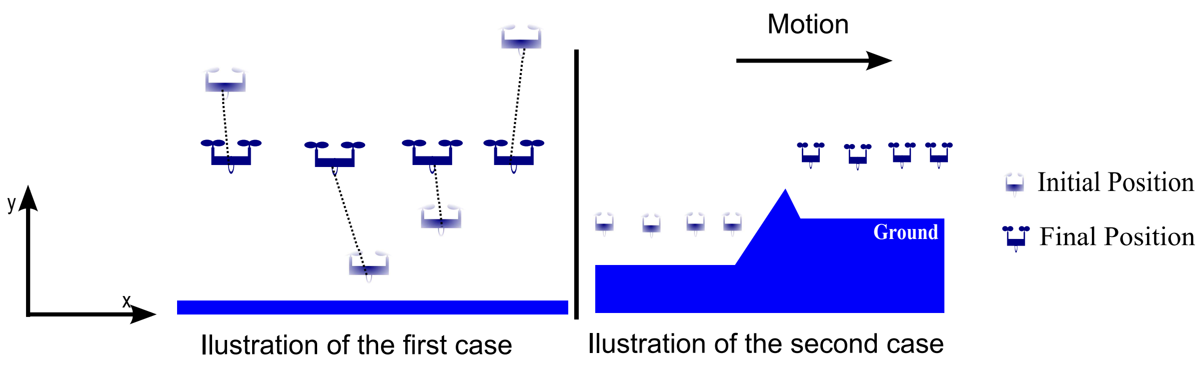


FIGURE B.6 – De gauche à droite, illustration du premier et deuxième cas d'étude. "Motion" et "ground" signifient en français, respectivement, "mouvement" et "sol", tandis que "initial position" et "final position" signifient "position initiale" et "position finale".

De plus, la matrice Laplacienne correspondant au graphe de communication du système est donnée par :

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}.$$

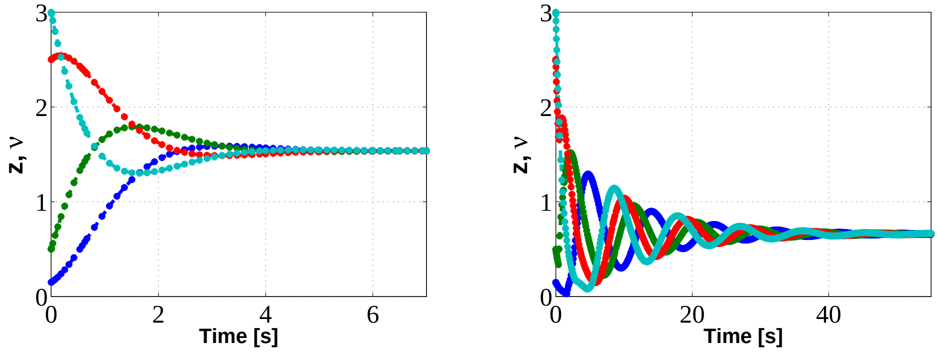
Les conditions initiales sont données par :

$$x_1(0) = \begin{bmatrix} 1.5 & 0.15 \end{bmatrix}^T, \qquad x_2(0) = \begin{bmatrix} 5 & 0.5 \end{bmatrix}^T,$$
$$x_3(0) = \begin{bmatrix} 1.25 & 0.75 & 2.5 \end{bmatrix}^T, \quad x_4(0) = \begin{bmatrix} 1.35 & 0.45 & 3 & 1.5 \end{bmatrix}^T.$$

Quelques commentaires à propos de la Figure B.6 s'imposent. Dans un premier cas, les agents hétérogènes doivent se mettre d'accord sur la même hauteur. Dans un deuxième cas, le groupe de robots est en mouvement dans un environnement dynamique. Plus précisément, le profil du sol varie dans le temps, alors que l'objectif reste le même : maintenir une hauteur constante par rapport au sol. Cette application a un sens pratique particulier puisque, dans plusieurs opérations comme, par exemple, celles de recherche ou de sauvetage, les robots se déplacent normalement dans un environnement hostile et dynamique.

La Figure B.7(a)[1] présente les résultats de simulation pour le système (B.2.1) contrôlé par (B.2.2). On peut facilement remarquer que les deux systèmes atteignent un consensus où la valeur d'accord, $\nu(\infty)$, correspond à la moyenne des conditions initiales du système, c'est-à-dire, $\nu(\infty) = moy\{\nu(0)\}$.

1. Pour toutes les figures, la ligne pointillée correspond aux dynamiques supplémentaires, tandis que la ligne *étoilée* représente l'évolution du vecteur de mesure z.



(a) Evolution des variables ν_i's et z_i's quand $\nu_i(0) = z_i(0)$

(b) Evolution des variables ν_i's ($\tau = 0.6$) et z_i's pour des retards aléatoires $\tau_{ij} \leq 3$, quand $\nu_i(0) = z_i(0)$

FIGURE B.7 – Évolution des variables supplémentaires ν (ligne pointillée) et du vecteur de mesure z (ligne *étoilée*).

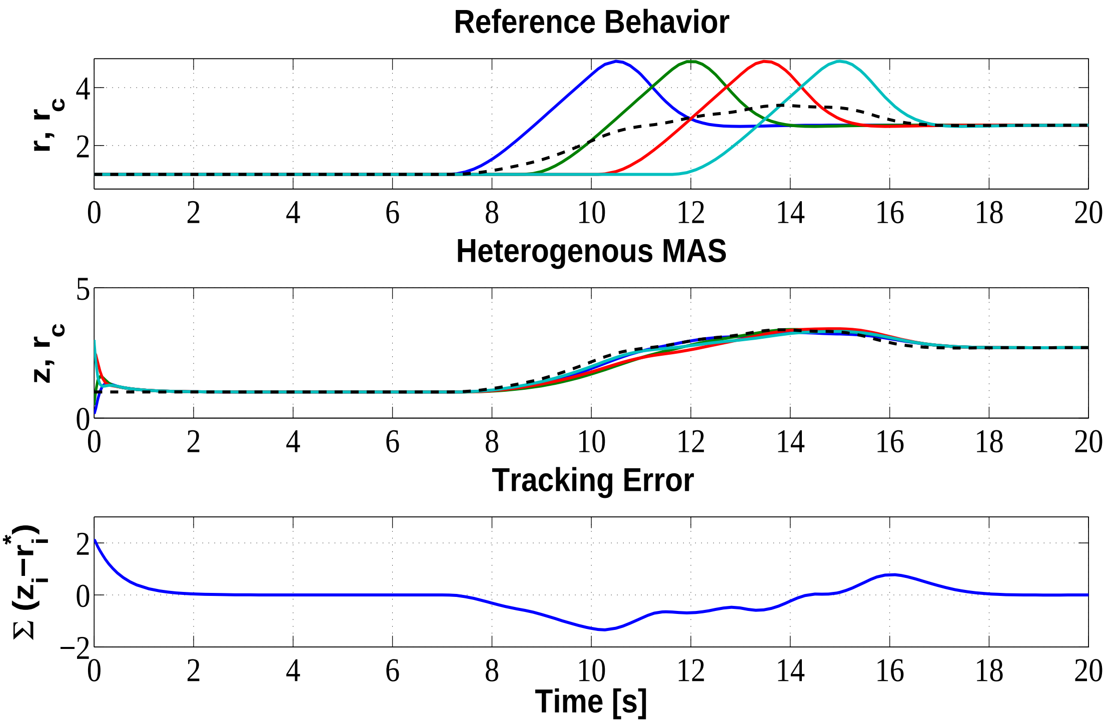


FIGURE B.8 – Résultats de simulation basés sur des signaux de référence non identiques. Les expressions anglaises “reference behavior”, “heterogeneous MAS” et ‘tracking error” signifient en français, respectivement, “comportement des signaux de référence”, “systèmes multi-agents hétérogènes” et “erreur de suivi”.

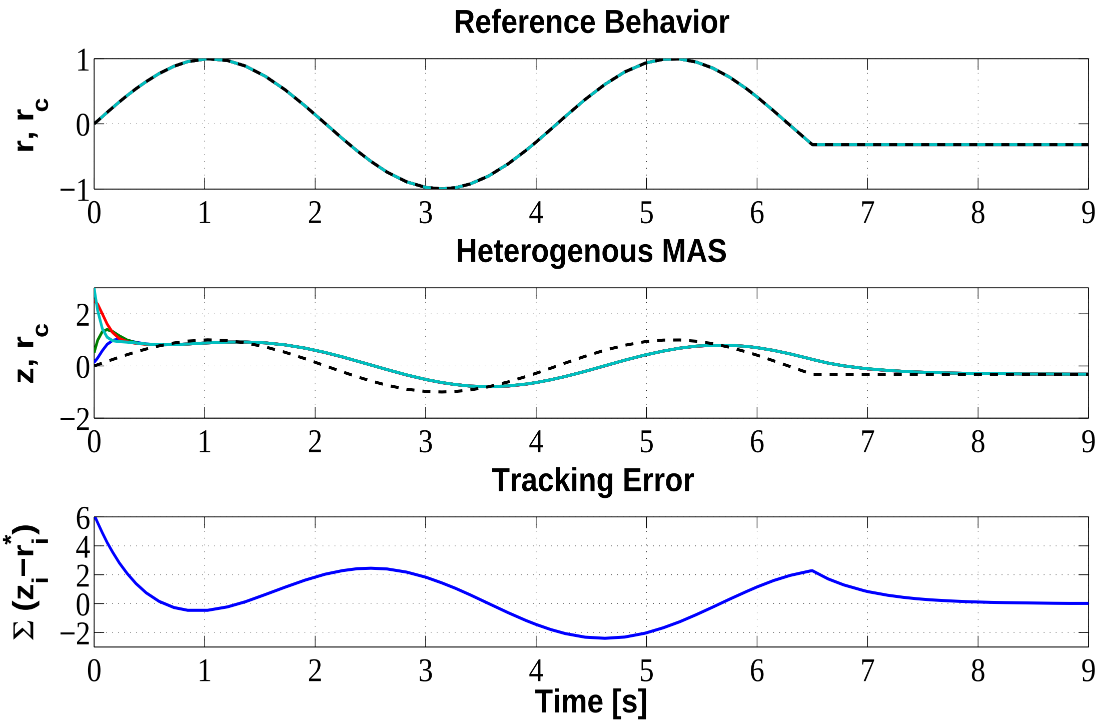


FIGURE B.9 – Résultats de simulation basés sur des signaux de référence identiques. Les expressions anglaises "reference behavior", "heterogeneous MAS"' et 'tracking error" signifient en français, respectivement, "comportement des signaux de référence", "systèmes multi-agents hétérogènes" et "erreur de suivi".

La Figure B.7(b) présente les résultats de simulation pour le système (B.2.1) contrôlé par (B.2.10) avec des retards aléatoires $\tau_{ij} \leq 3$. On peut facilement conclure que le protocole de consensus est robuste par rapport à des retards hétérogènes bornés. De plus, ces résultats sont concordants avec ceux présentés dans [179].

Considérons maintenant le deuxième cas d'étude présenté dans la figure B.6. Dans ce cas, les agents sont supposés s'accorder sur une hauteur commune et maintenir une distance constante par rapport au sol. Les figures B.8 [2] et B.9 [2] présentent des résultats correspondant à ce cas, où les différents systèmes sont contrôlés par (B.2.11). Dans la Figure B.8, les signaux r_i sont différents pout tout agent i, tandis que dans la Figure B.9 tous les éléments de r_i sont égaux. On peut observer que tous les vecteurs de mesure z_i du système atteignent une valeur commune r_c (égale à la moyenne des références externes), satisfaisant donc les objectifs initiaux.

2. Pour toutes les figures, r_c correspond à la moyenne des références externes et est représenté par une ligne noire pointillée.

B.2.9 Conclusions

Dans ce chapitre, une nouvelle approche pour la synthèse de lois de commande du type consensus a été présentée. D'ailleurs, divers résultats de simulation montrent l'efficacité de notre approche. Nous avons démontré que, en utilisant des dynamiques supplémentaires simples, la commande d'un système multi-agents hétérogène devient possible, en même temps qu'elle limite les contraintes sur le système. Le grand avantage de la méthode proposée est de permettre une analyse découplée entre les différents composants du système. Par conséquent, il a été possible d'étendre nos résultats à des situations plus complexes, en considérant par exemple des retards de communication ou des filtres distribués avec des références externes.

B.3 Algorithmes de consensus améliorés par un échantillonnage approprié

B.3.1 Contexte

Cette section porte sur des algorithmes de consensus avec une attention particulière sur leur vitesse de convergence. L'objectif est, plus précisément, l'amélioration des propriétés de convergence, en utilisant de la mémoire dans la synthèse des lois de commande.

Pour plusieurs problèmes de contrôle dans le cadre des systèmes multi-agents, les propriétés de convergence et la vitesse du système dépendent de ce qu'on nomme la connectivité algébrique du graphe de communication. Plus concrètement, la connectivité algébrique est égale à la deuxième plus petite valeur propre de la matrice Laplacienne **L**. Des méthodes permettant le contrôle de la connectivité du graphe peuvent être trouvées dans [112, 210, 212, 304, 313, 314], parmi d'autres. En particulier, les travaux de [255] maximisent la connectivité algébrique, sans forcément la calculer, et une approche distribuée pour l'estimation de la connectivité est présentée dans [4], où les auteurs détaillent un scénario du type "event-triggered" ou, en français, un système déclenché par des événements.

L'accélération de la vitesse de convergence des algorithmes de consensus a aussi été étudiée dans la littérature en intégrant de la mémoire dans les lois de commande. C'est précisément sur cette approche que cette section se focalise. Même si, dans la plupart des cas, la présence de retards conduit à une réduction des performances ou même à l'instabilité, il y a d'autres scénarios où l'introduction d'un retard dans la boucle de commande peut améliorer les propriétés de convergence du système. Ceci a été étudié dans [110, 252], par exemple.

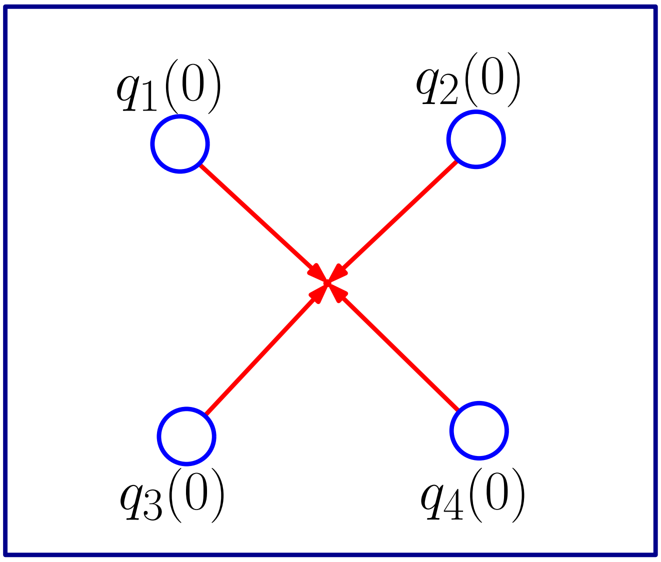


FIGURE B.10 – Illustration des protocoles de rendez-vous pour des agents du type simple intégrateur.

Plus précisément, dans [252] les auteurs ont ajouté aux lois de commande une composante échantillonnée de l'état, ce qui peut être considéré comme une façon de manipuler artificiellement les vecteurs propres de **L**. Une approche basée sur ces idées est proposée par la suite, en considérant deux systèmes différents : simple et double intégrateurs. Cependant, dans ce travail on ne considérera que le cas simple intégrateur (SI). L'intégralité des travaux sur ce domaine est présentée dans Chapitre 3. Il est aussi important de mentionner que ce travail a été effectué dans le cadre du Groupement International de Recherche DelSys, supporté par le Centre National de la Recherche Scientifique.

Définition du problème et préliminaires

Dans le vaste domaine des systèmes multi-agents, on s'intéresse particulièrement aux applications de contrôle de robots mobiles dans un plan cartésien. La position d'un robot i est définie par :

$$q_i = [x_i, y_i]^T \in \mathbb{R}^2,$$

où x_i et y_i représentent les dynamiques de q_i suivant les axes x et y, respectivement. Cependant, pour des questions de notation et de clarté, on ne considère, dans cette section, que les dynamiques de chaque agent sur l'axe x.

Les systèmes simple intégrateur sont un cas particulier des systèmes multi-agents linéaires, largement utilisés dans la littérature [13, 176, 194, 251]. Dans ce cadre, nous nous intéressons en particulier aux applications du type rendez-vous [62]. Le comportement souhaité est illustré dans Figure B.10.

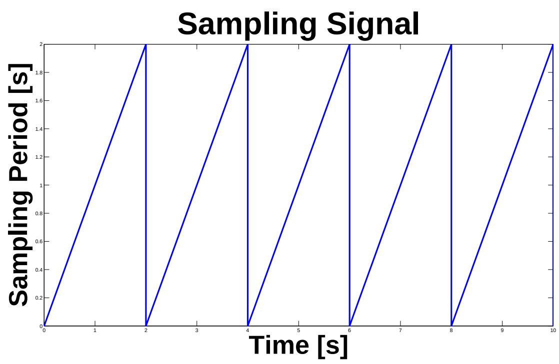


FIGURE B.11 – Représentation graphique du signal d'échantillonnage. Les expressions anglaises "sampling signal" et "sampling period" signifient en français, respectivement, "signal d'échantillonnage" et "période d'échantillonnage".

Considérons le système suivant :

$$\begin{cases} \dot{x}_i(t) = u_i(t) \\ u_i(t) = \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \end{cases} \qquad i \in \{1, \ldots, N\}, \tag{B.3.1}$$

où x_i représente les variables de l'agent i. Si on définit $x(t) = [x_1(t), .., x_N(t)]^T$, contenant les variables de tous les agents, on obtient :

$$\dot{x}(t) = -\mathbf{L}x(t) \ , \tag{B.3.2}$$

où $\mathbf{L}$ est la matrice Laplacienne. Considérons qu'il existe une constante positive μ telle que :

$$\sum_{j\in\mathcal{N}_i} a_{ij} = \mu, \quad i \in \{1, \ldots, N\}.$$

On suppose aussi qu'il existe un graphe de communication invariant dans le temps avec une structure qui garantisse une seule valeur propre égale à zéro, le vecteur propre correspondant étant un vecteur unitaire. En d'autres termes, ceci signifie qu'un consensus pourra être atteint asymptotiquement [220].

Dans la suite, on considère l'incorporation de *mémoire partielle* et de *mémoire globale*. Ces deux concepts divergent en ce qui concerne l'information du système qui est échantillonnée et appliquée dans la synthèse. En particulier, la *mémoire partielle* utilise de l'information sur le propre agent ou sur ses voisins, tandis que la *mémoire globale* considère toute l'information du système. Dans ce résumé, on ne présentera que des travaux sur la *mémoire partielle*.

B.3.2 Synthèse des contrôleurs

Considérons le système (B.3.2). Il peut être modifié de façon à incorporer de la mémoire, tel que :

$$\dot{x}(t) = (-\mathbf{L} - \delta\mathcal{A})x(t) + \delta\mathcal{A}x(t-\tau), \tag{B.3.3}$$

où $\delta \in \mathbb{R}$ et $\tau \geq 0$ sont des paramètres supplémentaires. Remarquez que, si δ et/ou τ sont égaux à zéro, alors l'algorithme classique est retrouvé.

Sachant que le retard est maintenant un paramètre de contrôle, on considère dans nos travaux un retard échantillonné [98, 250], ce qui offre des avantages en termes de mémoire par rapport à des retards continus. Ce retard est défini par :

$$\tau(t) = t - t_k,\ t_k \leq t < t_{k+1}\ ,$$

où $0 = t_0 < t_1 < ... < t_k < ...$ correspond aux instants d'échantillonnage, voir Figure B.11.

On suppose par la suite que le processus d'échantillonnage est périodique tel que :

$$t_{k+1} - t_k = T\ . \tag{B.3.4}$$

Toutefois, l'approche suivante peut être étendue à des systèmes non asynchrones. Finalement, l'algorithme initial (B.3.2) a fait place à un nouvel algorithme illustré dans la Figure B.12. Cet algorithme est défini par :

$$\forall t \in [t_k\ t_{k+1}[, \quad \dot{x}(t) = (-\mathbf{L} - \delta\mathcal{A})x(t) + \delta\mathcal{A}x(t_k), \tag{B.3.5}$$

où δ et T sont de nouveaux paramètres de contrôle.

B.3.3 Définition d'un modèle approprié et résultats théoriques

Cette section se concentre sur la définition d'un modèle approprié à l'analyse de convergence pour l'algorithme de consensus (B.3.5) En supposant que le vecteur propre unitaire $\overrightarrow{\mathbf{1}}$ est associé à la valeur propre 0 de la matrice Laplacienne, il est possible de trouver un changement de coordonnées du type $x = Wz$, comme proposé dans [251], tel que :

$$U(-\mu I + \mathcal{A})W = \begin{bmatrix} \Omega & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}, \tag{B.3.6}$$

où $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = W^{-1}$ et $U_2 = (U)_N$.

Le Lemme suivant permet de présenter une nouvelle formulation de l'algorithme (B.3.5).

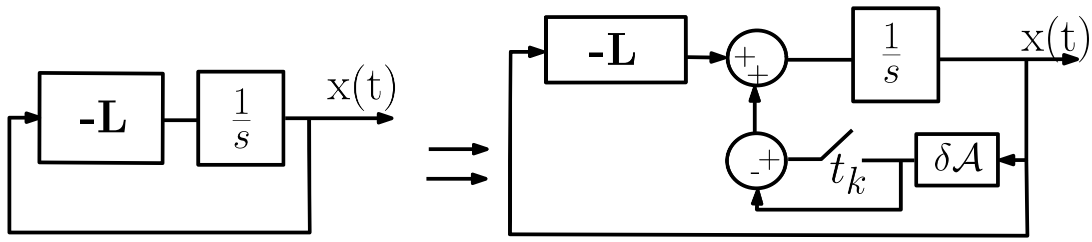


FIGURE B.12 – Structure de contrôle classique et avec mémoire partielle pour des systèmes du type SI.

Lemme B.1. *(Rodrigues de Campos et al. [232]) Le système (B.3.5) peut être réécrit comme :*

$$\dot{z}_1(t) = -(\Omega + \delta(\Omega + \mu I))z_1(t) + \delta(\Omega + \mu I)z_1(t_k), \tag{B.3.7a}$$
$$\dot{z}_2(t) = -\delta\mu z_2(t) + \delta\mu z_2(t_k), \tag{B.3.7b}$$

où $z_1 \in R^{N-1}$, $z_2 \in R$ et Ω sont donnés en (B.3.6).

Considérons le système (B.3.5) réécrit comme dans (B.3.7). On peut obtenir :

$$\dot{z}_1(t) = M_{SIP}(\delta)z_1(t) + M^*_{SIP}(\delta)z_1(t_k), \tag{B.3.8}$$

où $M_{SIP}(\delta) = -[\Omega + \delta(\Omega + \mu I)]$ et $M^*_{SIP}(\delta) = \delta(\Omega + \mu I)$. En plus, pour toute matrice $A \in \mathbb{R}^n$, la notation $He\{A\} > 0$ correspond à la somme suivante $A + A^T > 0$. Notre résultat principal sur les algorithmes de consensus incorporant de la mémoire est donné dans la suite.

Théorème B.2. *(Rodrigues de Campos et al. [232]) Soit l'algorithme (B.3.5) associé à une matrice Laplacienne $\boldsymbol{L}$, et $\alpha > 0$, $\delta > 0$, et $T > 0$. Supposons qu'il existe $P > 0$, $R > 0$, S_1 et $X \in \mathbb{S}^n$, $S_2 \in \mathbb{R}^{n\times n}$ et $N \in \mathbb{R}^{2n\times n}$ tel que l'expression suivante est satisfaite*

$$\begin{aligned} \Psi_1(T) &= e_\alpha(\tau)\Pi_1 + f_\alpha(T,0)\Pi_2 + h_\alpha(T,0)\Pi_3 < 0, \\ \Psi_2(T) &= \begin{bmatrix} e_\alpha(T)\Pi_1 + h_\alpha(T,T)\Pi_3 & g_\alpha(T,T)N \\ * & -g_\alpha(T,T)R \end{bmatrix} < 0, \end{aligned} \tag{B.3.9}$$

où

$$\begin{aligned} \Pi_1 &= He\{M_1^T P M_0 - M_{12}^T(\tfrac{1}{2}S_1 M_{12} + S_2 M_2 + N^T)\} + 2\alpha M_1^T P M_1 \\ \Pi_2 &= M_0^T R M_0 + He\{M_0^T(S_1 M_{12} + S_2 M_2)\}, \\ \Pi_3 &= M_2^T X M_2, \end{aligned}$$

et $M_0 = \begin{bmatrix} M_{SIP}(\delta) & M^*_{SIP}(\delta) \end{bmatrix}$, $M_1 = \begin{bmatrix} I & 0 \end{bmatrix}$, $M_2 = \begin{bmatrix} 0 & I \end{bmatrix}$, $M_{12} = M_1 - M_2$.

Les fonctions f_α, g_α et h_α pour tout scalaire T et $\tau \in [0\ T]$ sont données par :

$$
\begin{aligned}
&e_\alpha(\tau) = e^{2\alpha\tau},\\
&f_\alpha(T,\tau) = (e^{2\alpha T} - e^{2\alpha\tau})/2\alpha,\\
&g_\alpha(T,\tau) = \begin{cases} e^{2\alpha T}(e^{2\alpha\tau} - 1)/2\alpha, & \textit{if } \alpha > 0\\ (e^{2\alpha\tau} - 1)/2\alpha, & \textit{if } \alpha < 0\end{cases}\\
&h_\alpha(T,\tau) = \frac{1}{\alpha}\left[\frac{e^{2\alpha T}-1}{2\alpha T} - e^{2\alpha\tau}\right].
\end{aligned}
$$

Alors, l'algorithme (B.3.5) avec des paramètres δ et la période d'échantillonnage T est $\alpha-$stable. De plus, la valeur d'équilibre/d'accord est donnée par :

$$x(\infty) = U_2 x(0).$$

Il est important de rappeler que ce résumé ne contient qu'une petite partie des résultats de cette thèse. En complément des éléments présentés précédemment, le Chapitre 3 contient d'autres résultats portant sur des protocoles de consensus avec *mémoire globale* pour des systèmes du type simple intégrateur, ainsi que pour des systèmes du type double intégrateur avec *mémoire partielle* et *globale*.

B.3.4 Résultats de simulation

Pour illustrer l'efficacité de ces nouveaux protocoles, nous avons considéré plusieurs types de graphes de communication, dirigés ou non dirigés, voir Figure B.13.

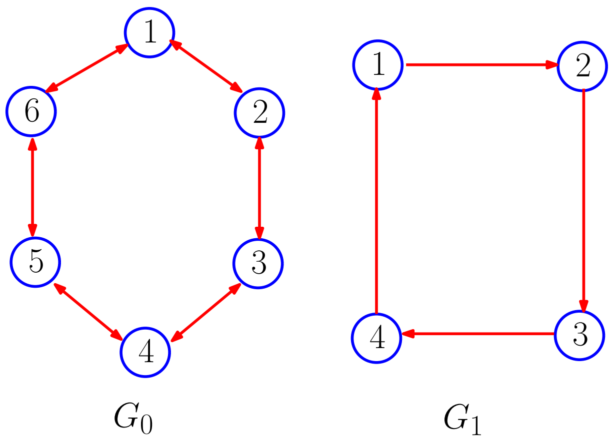


FIGURE B.13 – Graphes correspondant aux matrices L_0 et L_1.

A chaque graphe, la matrice Laplacienne associée est donnée par :

$$L_0 = \begin{bmatrix} -1 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0.5 & -1 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & -1 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -1 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & -1 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0.5 & -1 \end{bmatrix}, L_1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix}.$$

Les positions initiales dans l'axe x pour un réseau de quatre et de six agents sont définies comme $x^{\ 4\ agents}$ et $x^{\ 6\ agents}$, respectivement, et données par :

$$x^{\ 4\ agents}(0) = [30,\ 25,\ 15,\ 0]^T, \quad x^{\ 6\ agents}(0) = [30,\ 25,\ 15,\ 0,\ -10,\ -30]^T.$$

Considérons un ensemble de quatre agents contrôlés par (B.3.5) et connectés par les graphes G_0 et G_1 respectivement, voir Figure B.13. Remarquez que ces deux graphes sont équilibrés, ce qui implique que la valeur de consensus correspond à la moyenne des conditions initiales.

La figure B.14 présente les résultats de simulation du taux de convergence α d'après le Théorème B.2 pour G_0. Des résultats équivalents pour le graphe G_1 sont présentés dans la Figure B.15. Les figures B.14 et B.15 montrent une représentation 3-D des résultats précédents, mettant en évidence le taux maximal de convergence du protocole (B.3.5). En particulier, on peut identifier une crête pour des valeurs spécifiques de (δ, T). En effet, la meilleure valeur de α est observée quand $(\delta, T) = (1.96, 0.29)$ pour le graphe G_0 et $(\delta, T) = (1.96, 0.09)$ pour le graphe G_1.

La Figure B.16 présente des simulations du protocole dit classique (B.3.2) ainsi que du protocole (B.3.5), avec G_0 et différentes valeurs de δ et T. Des résultats équivalents, pour le graphe G_1, sont présentés dans la Figure B.17. Ces courbes mettent en évidence la différence entre la performance des deux systèmes. D'après leur analyse, on

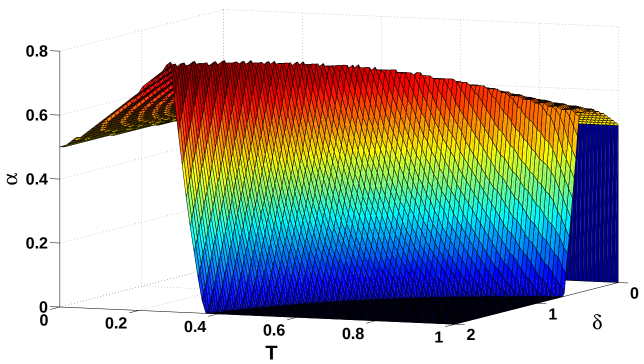


FIGURE B.14 – Taux de convergence du protocole de consensus (B.3.5) connecté avec G_0, pour des valeurs différentes de (δ, T).

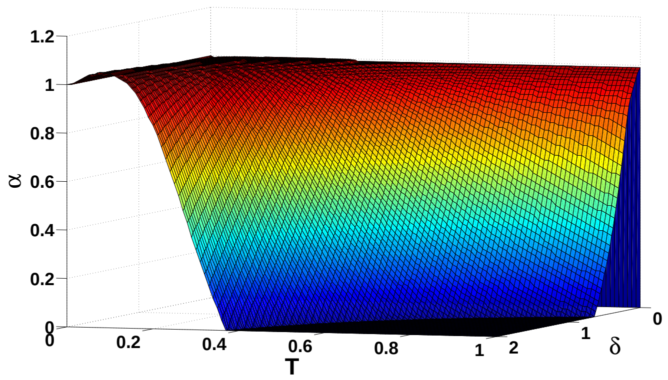


FIGURE B.15 – Taux de convergence du protocole de consensus (B.3.5) connecté avec G_1, pour des valeurs différentes de (δ, T).

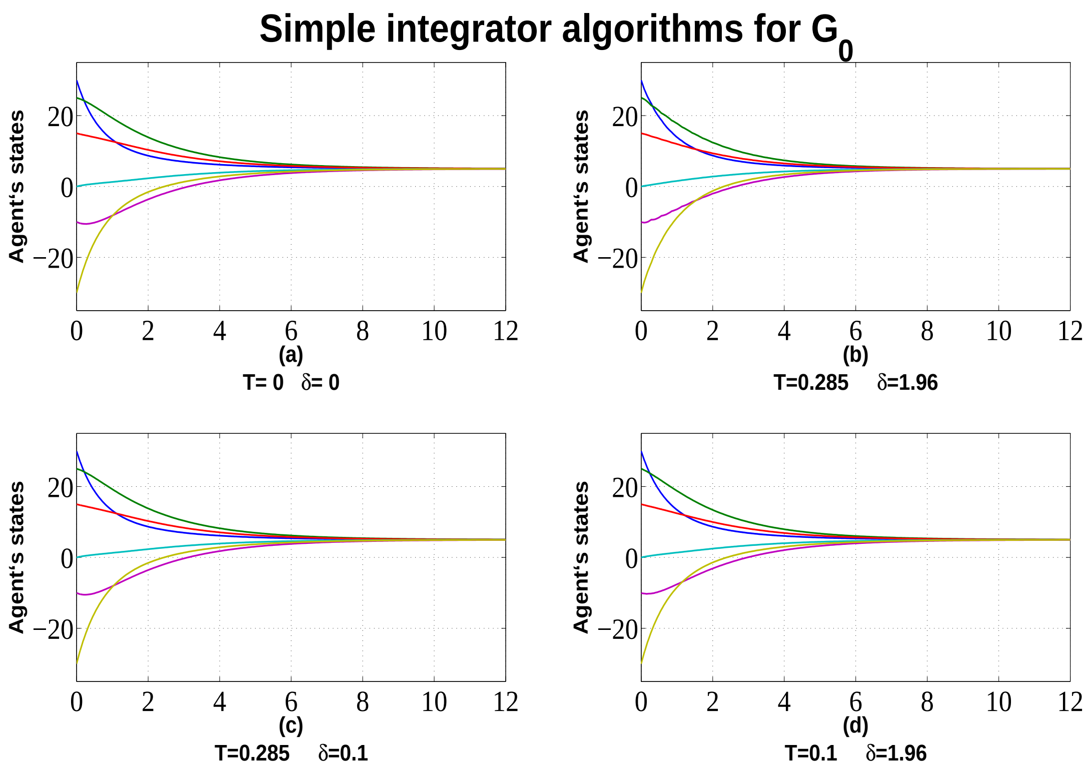


FIGURE B.16 – Résultats de simulation du système (B.3.5) connecté avec G_0, pour des valeurs différentes de (δ, T). Les expressions anglaises “simple integrator algorithms” et “agent’s states ” signifient en français, respectivement, “algorithmes pour des systèmes du type simple intégrateur” et “états des agents”.

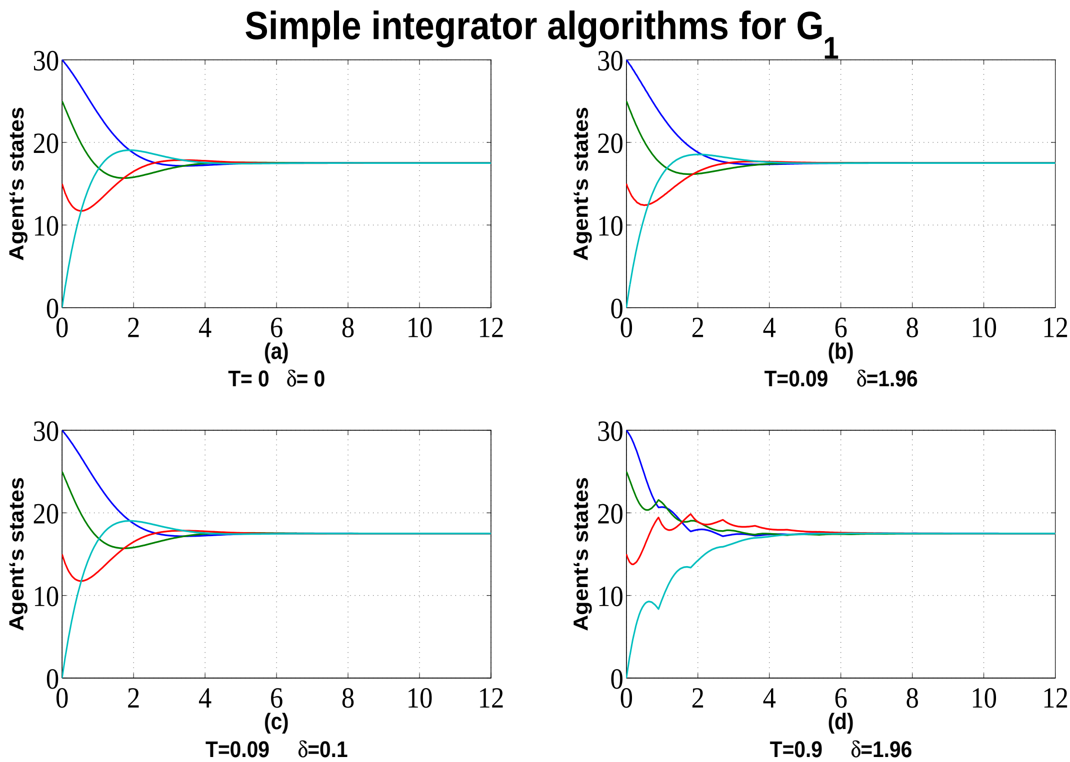


FIGURE B.17 – Résultats de simulation du système (B.3.5) connecté avec G_0, pour des valeurs différentes de (δ, T). Les expressions anglaises “simple integrator algorithms” et “agent's states ” signifient en français, respectivement, “algorithmes pour des systèmes du type simple intégrateur” et “états des agents”.

peut facilement observer que le taux de convergence de l'algorithme (B.3.5) est supérieur à celui du protocole original, confirmant donc que l'incorporation de mémoire peut améliorer les propriétés de certains systèmes. Les figures B.16(a),B.17(a) montrent l'évolution du système dit classique. Les figures B.16(b),B.17(b) correspondent aux résultats qui prennent en compte les valeurs optimales de (δ, T) d'après le Theorème B.2 déduites des figures B.14 et B.15, respectivement. On peut remarquer que ces deux images correspondent au protocole le plus rapide, par rapport à l'algorithme classique. Dans les figures B.16(c),B.17(c), la valeur optimale de T est maintenue et la valeur de δ est changée. Finalement les figures B.16(d),B.17(d), considèrent la valeur optimale de δ et des valeurs différentes pour T. On remarque aussi que la réponse la plus rapide est corrélée avec les valeurs optimales de (δ, T). En revanche, pour d'autres valeurs, on peut constater une détérioration des performances, voire même l'instabilité du système dans le cas de systèmes double intégrateur. Pour le cas particulier où $T = \pi/\delta$, l'instabilité peut être observée, avec des oscillations autour de la valeur de consensus finale.

B.3.5 Conclusions

Dans cette section, l'incorporation de la mémoire dans des lois de commande du type consensus a été présentée, dans le but d'améliorer les performances de convergence de ces systèmes. Seules des dynamiques du type simple intégrateur avec *mémoire partielle* ont été considérées dans ce résumé ; cependant, des résultats plus généraux compris dans le Chapitre 3 considèrent aussi des systèmes du type double intégrateur, pour lesquels l'efficacité et les avantages de cette approche sont plus visibles, et la *mémoire globale*.

Nous avons pu conclure que l'incorporation de *mémoire globale* améliore considérablement les performances par rapport aux algorithmes classiques et également à ceux où la *mémoire partielle* est utilisée. Nous avons également proposé une méthode d'optimisation des paramètres des contrôleurs, basée sur les principes de la théorie de Lyapunov et exprimée sous la forme de LMI (Linear Martix Inequality). Pour mettre en valeur les résultats théoriques de ces études, plusieurs résultats de simulations montrent clairement l'efficacité de l'approche illustrée dans ce document. Cette section met aussi en évidence les avantages techniques de cette approche, car elle conduit à une réduction de la quantité d'information requise : l'absence de capteurs de vitesse, une fois qu'ils ne sont plus nécessaires, entraîne des avantages en termes d'économie, d'espace et de calcul.

B.4 Commande distribuée pour le déploiement compact d'agents

B.4.1 Contexte

Le déploiement d'ensembles de véhicules autonomes est désormais possible grâce aux progrès technologiques de ces dernières décennies. En effet, de nos jours, des robots munis de capacité de calcul, de communication et de mobilité permettent d'accomplir une variété de tâches telles que des opérations de surveillance, de recherche et de sauvetage, des manœuvres dans des environnements dangereux, etc.

Récemment, le vaste nombre d'applications possibles a démontré que des agents mobiles ont besoin non seulement de s'accorder quant à une certaine variable d'intérêt, mais aussi d'avoir la possibilité de se déployer sur une région, d'assumer une configuration prédéfinie ou de se déplacer de manière synchronisée.

Alors que dans les sections précédentes on a proposé des solutions pour des application du type rendez-vous, on s'intéresse ici au contrôle de formation, voir Figure B.18. Des applications robotiques récentes ont également montré comment il est intéressant d'imposer une configuration géométrique particulière à l'ensemble des robots. En fait, la géométrie et la symétrie de la configuration finale sont directement liées au problème de contrôle et à la synthèse des lois de commande [238]. Parmi tant d'autres, le contrôle de la géométrie des formations a été traité dans [200, 204]. En particulier, le contrôle des formations circulaires a été proposé dans [31, 35, 136, 142, 145, 198, 247, 248]. Une étude approfondie portant sur les différentes stratégies peut être trouvée dans [29, 48].

Cette section porte sur la synthèse et l'analyse d'un algorithme pour le *déploiement compact des agents*. Trois problèmes seront pris en considération :

i) Comment améliorer le taux de couverture de l'environnement ;

ii) Comment garantir que deux agents restent connectés ;

iii) Comment améliorer la connectivité d'un graphe de communication.

La première contribution de cette thèse correspond à une extension de [76] pour la dispersion des agents, basée sur des forces potentielles. En effet, chaque agent est équipé de ces fonctions qui, simultanément, lui permettent de s'isoler de tout autre agent et d'imposer, aussi, le maintien des liens de voisinage. Finalement, la deuxième contribution dans ce domaine correspond au contrôle des angles inter-agents. Plus précisément, le but est d'obtenir la formation la plus compacte possible, à travers la minimisation des angles inter-agents, voir Figure B.19.

En résumé, nous proposons ici une stratégie de contrôle pour le déploiement compact d'agents afin d'obtenir le meilleur taux de couverture possible, tout en conservant ou en améliorant les propriétés de connectivité. Deux problèmes indépendants sont traités dans cette thèse : le contrôle de la dispersion et celui de la compacité de la formation. On propose finalement une approche séquentielle regroupant ces deux composantes.

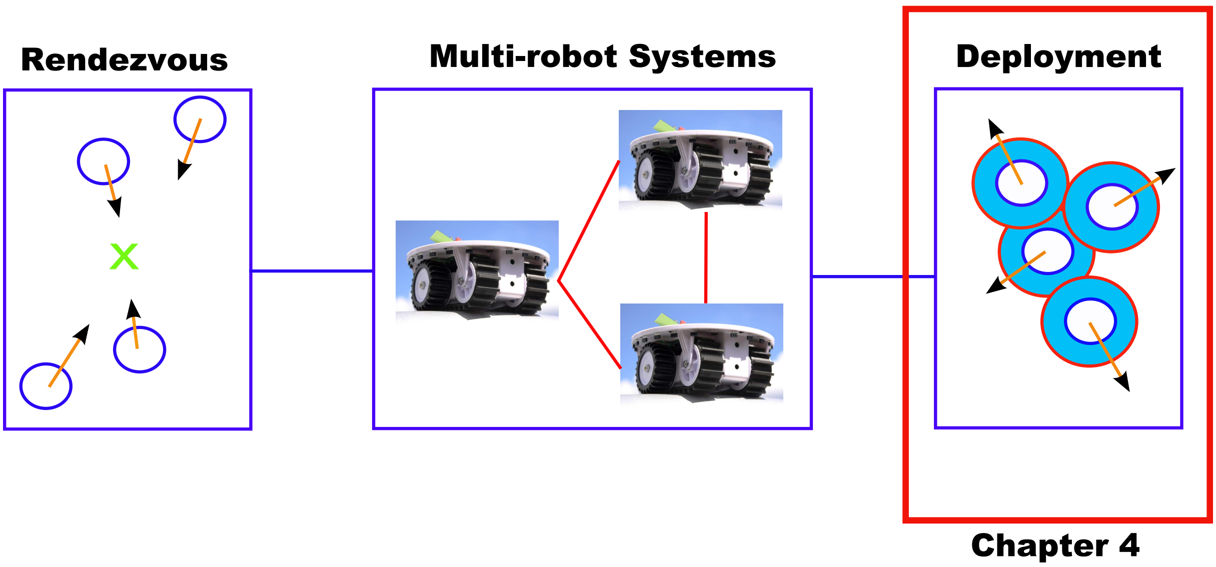


FIGURE B.18 – Contexte du Chapitre 4

Ce travail a été réalisé en collaboration avec KTH, le Royal Institute of Technology of Stockholm, en Suède. Il a également été soutenu par une bourse de mobilité internationale accordée par Grenoble INP.

B.4.2 Définition du problème et préliminaires

Cette section présente les définitions et les outils utilisés tout au long des développements qui suivent.

B.4.3 Description du système

Considérez N agents opérant sur un espace $W \subset \mathbb{R}^2$. Les dynamiques de chaque agent sont décrites comme :

$$\dot{q}_i = u_i, \quad i \in \mathcal{N} = \{1, ..., N\}, \tag{B.4.1}$$

où q_i représente la position de l'agent i, $q = [q_1, ..., q_N]^T$ la configuration du système, et u_i l'entrée de chaque agent. En plus, on considère que $q_i = [x_i, y_i]^T \in \mathbb{R}^2$ tel que x_i et y_i représentent la dynamique de q_i sur l'axe x et y, respectivement. Définissons maintenant un vecteur connectant deux agents (i, j) tel que :

$$q_{ij} = q_j - q_i,$$

et que β_{ij} représente le carré de la distance entre deux agents :

$$\beta_{ij} = \|q_j - q_i\|^2, \forall\, i, j \in \mathcal{N}.$$

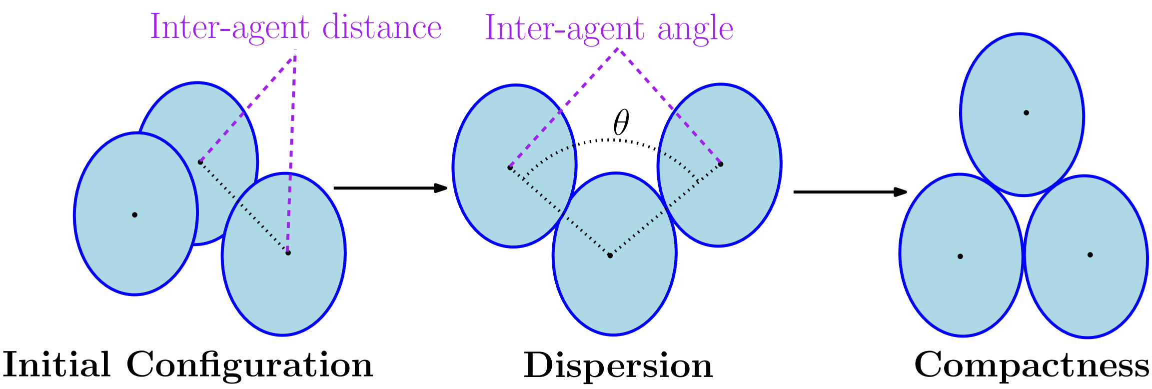


FIGURE B.19 – Objectifs du Chapitre 4 : De gauche à droite, la configuration initiale, la configuration souhaitée après contrôle de la dispersion et de la compacité. L'expression anglaise “inter-agent distance” doit être traduite, en français, par “distance inter-agents”, tandis que l'expression “inter-agent angle” concerne les angles créés entre les différents véhicules.

Dans la suite, on définit pour tout $i \in \mathcal{N}$ et $r > 0$:

$$\mathcal{N}_{i,r}(t) = \left\{ j \in \mathcal{N} \backslash \{i\} | \;\; \beta_{ij} < r^2 \right\},$$

comme le sous-ensemble de $\mathcal{N}$ regroupant tous les voisins de i, c'est-à-dire, tous les nœuds dans un rayon r de l'agent i tel que $|\mathcal{N}_{i,r}|$ représente le nombre de ses voisins. Pour tout agent i, définissons un graphe local tel que :

$$\mathcal{P}_{i,r}(t) = (\mathcal{N}_{i,r}(t), \Upsilon_i(t)),$$

où $\Upsilon_i(t) \subseteq \mathcal{N}_{i,r}(t) \times \mathcal{N}_{i,r}(t)$ est l'ensemble d'arêtes connectant, à l'instant t, l'agent i à tous $j \in \mathcal{N}_{i,r}$.

Dans la suite, on suppose que chaque agent a deux rayons de communication, voir Figure B.20. De plus, considérons $d_1 < d_2$ tel que :

$$d_1 = \xi d_2,$$

où $0 < \xi < 1$. Le plus petit rayon, d_1, limite la zone dans laquelle les distances inter-agents sont contrôlées, tandis que d_2 est utilisé pour établir le domaine dans lequel nous contrôlons les angles inter-agents. Une illustration d'un angle inter-agents est présentée dans la figure B.19. Les définitions suivantes s'imposent.

Définition B.1. *Un triplé $(i, j, k) \in \mathcal{N}^3$ est un* Triangle *si i, j, k sont des nœuds différents et*

$$i, k \in \mathcal{N}_{j,d_2}, \;\; i \notin \mathcal{N}_{k,d_2}.$$

Alors, un Triangle est un graphe connecté. De plus, l'agent central peut communiquer avec les deux autres agents, tandis que les deux autres ne peuvent pas communiquer entre eux. Remarquez que, en termes de notation, l'ordre des agents est important. Cela signifie que, lorsque les triangles sont discutés, un triplé (i, j, k) est centré en j.

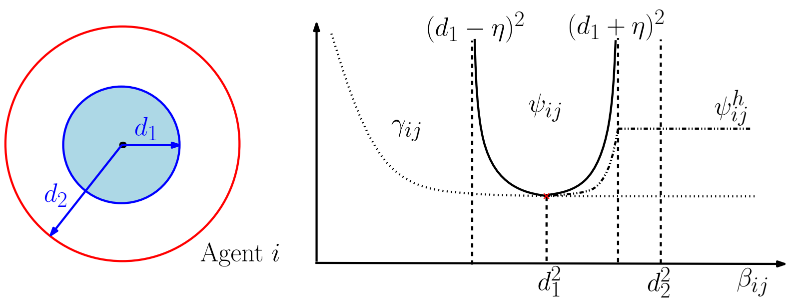


Figure B.20 – Représentation d'un agent i et de ses fonctions potentielles

Définition B.2. *Un triplé* $(i, j, k) \in \mathcal{N}^3$ *est un* Triangle Compacte *si* i, j, k *sont des nœuds différents et*

$$i, k \in \mathcal{N}_{j,d_2},\ i \in \mathcal{N}_{k,d_2}.$$

Alors un Triangle Compact est un graphe complet de trois agents.

Pour chaque Triangle (i, j, k), on définit l'angle inter-agents comme :

$$\theta_{ijk} = \arccos\left(\frac{<q_{ji}, q_{jk}>}{\|q_{ij}\| \cdot \|q_{jk}\|}\right), \tag{B.4.2}$$

dont un exemple est représenté dans la Figure B.19. Pour les algorithmes de dispersion, définissons maintenant l'ensemble de conditions initiales comme :

$$\mathcal{I}(d_1) = \left\{q \in W \mid \forall i \in \mathcal{N}, j \in \mathcal{N}_{i,d_2},\ \beta_{ij} \in]0, d_1^2]\right\},$$

et l'ensemble de configurations finales comme suit :

$$\mathcal{F}(d_1, \eta) = \left\{q \in W \mid \forall i \in \mathcal{N}, j \in \mathcal{N}_{i,d_2},\ (d_1 - \eta)^2 < \beta_{ij} < (d_1 + \eta)^2\right\}.$$

Considérons aussi :

$$\mathcal{D}(d_2, \Delta) = \{q \in W \mid \forall i \in \mathcal{N}, |\mathcal{N}_{i,d_2}| \leq \Delta\},$$

où Δ est un scalaire à définir et :

$$\mathcal{E}(d_1) = \left\{q \in W \mid \text{ pour tous } \textit{Triangles } (i, j, k),\ \beta_{ij} = \beta_{jk} = d_1^2\right\}.$$

Relativement au contrôle de la compacité de la formation, l'ensemble des conditions initiales est défini par :

$$\mathcal{F}'(d_1, d_2) = \mathcal{D}(d_2, 2) \cap \mathcal{E}(d_1).$$

B.4.4 Définition des fonctions potentielles

La synthèse des protocoles de dispersion est basée sur des fonctions potentielles. Dans la figure B.20 on peut trouver une représentation des deux fonctions, γ_{ij} et $\psi_{ij} \in C([0, +\infty))$. Ces fonctions sont définies avec les propriétés suivantes :

Définition de γ_{ij} pour une paire d'agents $(i,j) \in \mathcal{N}^2$:

- γ_{ij} est une fonction décroissante ;
- γ_{ij} tend vers $+\infty$ quand β_{ij} tend vers zéro ;
- $\frac{\partial \gamma_{ij}}{\partial \beta_{ij}} = 0$ si $\beta_{ij} \geq d_1^2$.

Définition de ψ_{ij} pour une paire d'agents $(i,j) \in \mathcal{N}^2$:

- ψ_{ij} tend vers $+\infty$ quand β_{ij} tend vers $(d_1 \pm \eta)^2$;
- $\psi_{ij} = \gamma_{ij}$ et $\frac{\partial \gamma_{ij}}{\partial \beta_{ij}} = 0$ si $\beta_{ij} = d_1^2$.

  où η est une constante positive qui définit un voisinage autour de d_1. Sa pertinence pour les lois de commande de la connectivité du graphe sera expliquée plus tard.

Une fonction ψ_{ij}^h, garantissant une transition appropriée vers ψ_{ij}, quand de nouvelles arêtes sont créées, peut aussi être définie avec des propriétés similaires aux précédentes. Cependant, une définition détaillée de cette fonction sera exclue de ce résumé bien qu'elle soit présentée dans le Chapitre 4.

B.4.5 Algorithmes de dispersion

Dans cette section, nous présentons un contrôleur pour la dispersion des agents avec maintien de la connectivité du graphe de communication. Le principe de la dispersion est illustré dans la Figure B.21.

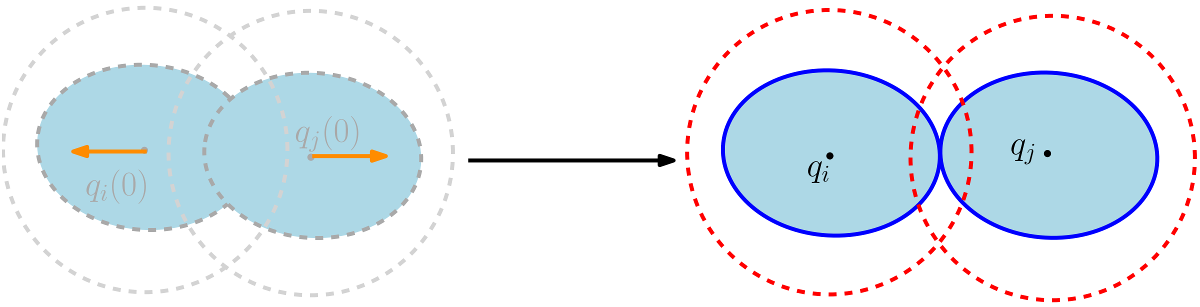


FIGURE B.21 – Principe de la dispersion

B.4.6 Synthèse des contrôleurs

Considérons un ensemble de N agents définis dans (B.4.1). Dans cette section, l'algorithme de dispersion, u_1, est donné par :

$$u_{i1} = -\sum_{j \in \mathcal{N}_{i,d_1}} \frac{\partial \gamma_{ij}}{\partial q_i} - \sum_{j \in \mathcal{N}^h_{i,d_1+\eta}} \frac{\partial \bar{\psi}_{ij}}{\partial q_i} - \sum_{j \in \mathcal{N}_{i,d_2}} \frac{\partial \psi_{ij}^h}{\partial q_i}, \tag{B.4.3}$$

où

$$\bar{\psi}_{ij}(t) = \begin{cases} 0, & \text{si } t < t^*_{ij}, \\ \psi_{ij}, & \text{sinon,} \end{cases}$$

et

$$t^*_{ij} = \min_{i\in\mathcal{N},\ j\in\mathcal{N}_{i,d_1}} \{t|\ \beta_{ij}(0) < d_1^2, \beta_{ij}(t^*_{ij}) = d_1^2\}.$$

En d'autres termes, t^*_{ij} correspond à l'instant où la distance entre deux agents i et j, initialement proches tel que $\dot{\beta}_{ij} > 0$, atteint pour la première fois la valeur d_1.

Il est à noter que l'équation (B.4.3) correspond à la somme des gradients négatifs des fonctions potentielles présentées précédemment. Ceci signifie donc que chaque agent est équipé d'une force répulsive et une attractive par rapport à ses voisins.

B.4.7 Algorithmes de contrôle de la compacité d'une formation

La loi de commande responsable du contrôle de la compacité de la formation, u_2, a pour but de minimiser les angles inter-agents et d'obtenir la formation la plus compacte possible. Pour un triplé (i, j, k), ceci peut être défini comme :

$$\mathcal{G}(d_1, d_2) = \left\{q \in W\ |\forall j \in \mathcal{N}, \exists (i,k) \in \mathcal{N}^2_{j,d_2} \quad \text{st.} \quad \beta_{ij} = \beta_{jk} = \beta_{ki} = d_1^2\right\}.$$

Alors, le triplé (i, j, k) est un *Triangle Compacte.*

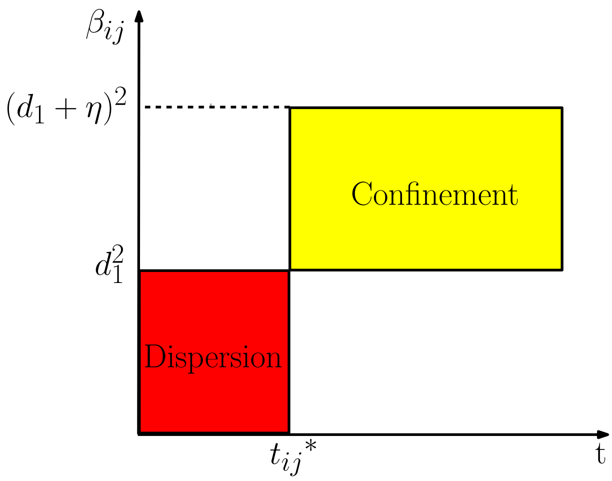


FIGURE B.22 – Structure de l'algorithme de dispersion. Le terme anglais “confinement” se réfère à la composante de maintien de la connectivité entre deux agents.

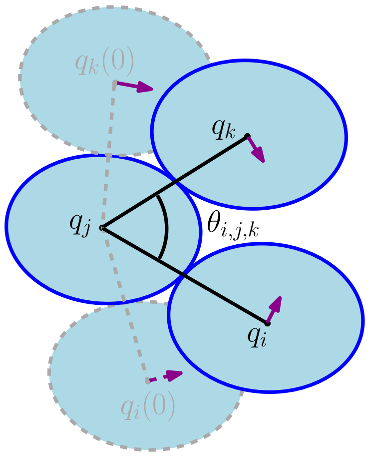


FIGURE B.23 – Principe du contrôle de la compacité d'une formation.

Synthèse des contrôleurs

Pour chaque agent i, la loi de commande est donnée par :

$$u_{i2} = -\sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}\setminus\{i\}} \sum_{k\in\mathcal{N}\setminus\{i,j\}} \chi_{ijk} R_{ot} q_{ji}, \tag{B.4.4}$$

où

$$\chi_{ijk} = \begin{cases} K[\theta_{ijk} - sign(\theta_{ijk})\frac{\pi}{3}], & \text{si } (i,k) \in \mathcal{N}^2_{j,d_2} \setminus \mathcal{N}^2_{j,d_1}, \\ 0, & \text{sinon}, \end{cases}$$

où K est un scalaire positif. Il est important de mentionner que la force résultante de cette loi de contrôle est appliquée dans la direction de $q_{ij}^{\perp}$, c'est à dire, dans une direction perpendiculaire à q_{ij}, comme illustré dans la Figure B.23. Ceci signifie donc que nous obtiendrons à terme un Triangle Compact avec tous les angles inter-agents égaux à $\pi/3$, voir Figure B.19.

B.4.8 Contrôleur séquentiel

Dans les sections précédentes, nous avons présenté les deux composantes nécessaires à l'approche séquentielle qui a également été proposée dans cette thèse. La structure de cette nouvelle loi de commande, qui rejoint ces deux parties, est visible dans la Figure B.24.

Pour chaque agent i, l'expression de ce contrôleur a la forme :

$$u_i = u_{i1} + u_{i2},$$

où $u = [u_1, \ldots, u_N]^T$. Il est démontré dans cette thèse que ce système est un système hybride qui doit alors être étudié en tant que tel. Une approche d'analyse de stabilité des systèmes hybrides a donc été proposée dans cette thèse. Cependant, cette approche, dans son ensemble, ainsi que l'analyse correspondante, sera exclue de ce résumé, bien qu'elle figure, en détail, dans le Chapitre 4.

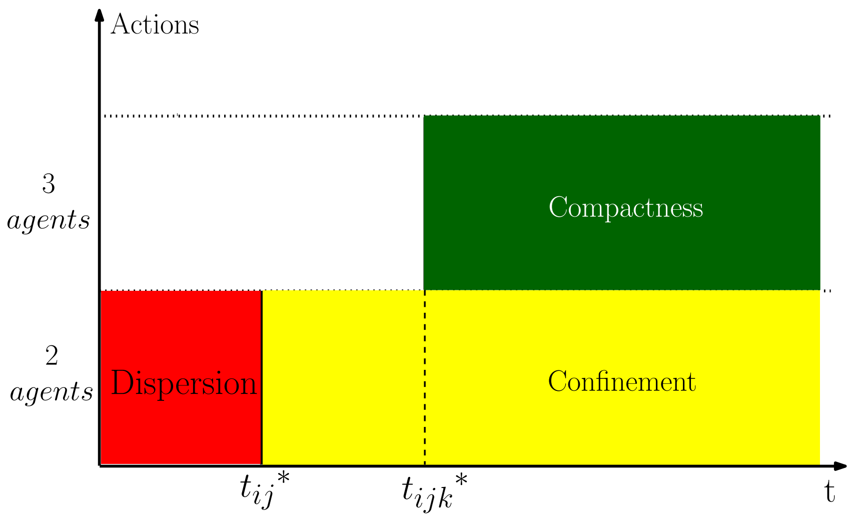


FIGURE B.24 – Structure du contrôleur séquentiel. Le terme anglais "compacteness" doit être traduit, en français, par compacité, tandis que "confinement" se réfère à la composante de maintien de la connectivité entre deux agents.

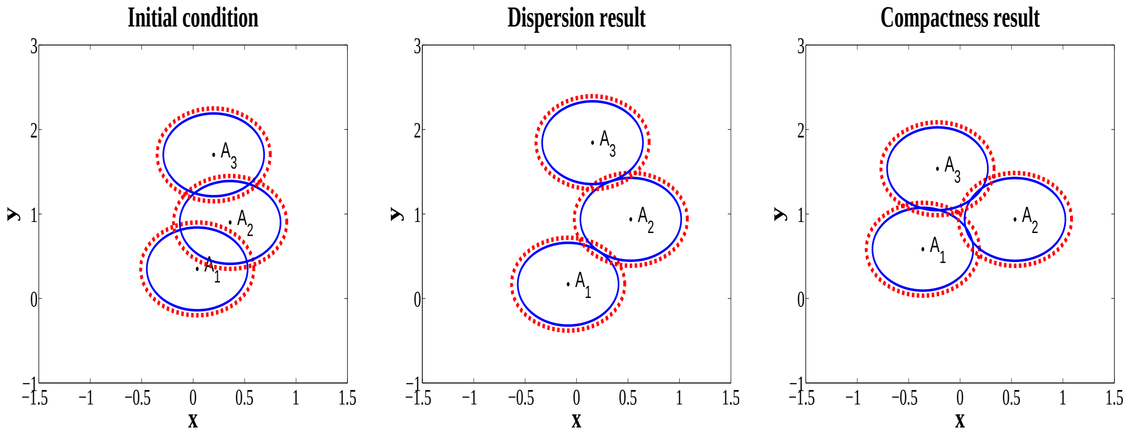
B.4.9 Résultats théoriques et de simulation

Cette section présente nos principales conclusions si l'on considère un ensemble de trois agents. Il est important de remarquer qu'on ne présente ici qu'une petite partie des résultats de cette thèse. Des formations de plus grande dimension et d'autres configurations particulières ont aussi été étudiées dans ce document. Pour toute information complémentaire, le lecteur est prié de consulter l'intégralité du Chapitre 4.

Le Théorème et le Lemme suivants présentent nos résultats pour un groupe de trois agents.

Théorème B.3. *(Rodrigues de Campos et al. [229]) Considérons $N = 3$ agents décrits en (B.4.1) et contrôlés par (B.4.3). Supposons une configuration initiale appartenant à $\mathcal{I}(d_1)$. Alors la configuration finale du système appartient à $\mathcal{F}(d_1, \eta)$.*

Lemme B.2. *(Rodrigues de Campos et al. [229]) Considérons $N = 3$ agents (i, j, k), décrits en (B.4.1) et contrôlés par (B.4.4). Supposons une configuration initiale appartenant à $\mathcal{F}'(d_1, d_2)$. Alors la configuration finale du système appartient à $\mathcal{G}(d_1, d_2)$.*



(a) (b) (c)

FIGURE B.25 – Résultats de simulation pour une configuration de trois agents

Pour valider nos développements théoriques, nous présentons par la suite les résultats issus de simulations. Pour pouvoir évaluer les différentes étapes de notre approche, nous fournissons ici les résultats de notre contrôleur séquentiel, composé, comme expliqué auparavant, d'un élément responsable pour la dispersion et d'un autre pour le contrôle de la compacité. On rappelle que

$$q_i = [x_i, y_i]^T \in \mathbb{R}^2, i \in \mathcal{N} = \{1, ..., N\}$$

est le vecteur de position de l'agent i dans un plan cartésien. Les conditions initiales pour un ensemble de trois robots ont été choisies suivant :

$$q^{3\ agents}(0) = \left[\ 0.04,\quad 0.35,\quad 0.36,\quad 0.90,\quad 0.20,\quad 1.70\ \right]^T.$$

La figure B.25 révèle l'évolution de la formation . En particulier, la Figure B.25(a) présente la configuration initiale tandis que la Figure B.25(b) montre le résultat de notre approche de dispersion. On peut facilement observer que tous les agents sont éloignés d'une distance égale d_1 entre eux. Finalement, sur la Figure B.25(c), on vérifie le résultat final de notre approche. Nous pouvons observer qu'un triangle équilatéral a été formé et que les angles inter-agents sont tous égaux à $\pi/3$. Cette formation est donc la plus compacte possible et satisfait les objectifs de contrôle détaillés précédemment.

B.4.10 Conclusions

Dans cette section nous avons considéré le déploiement compact d'un ensemble d'agents. La première contribution de ces travaux correspond à une extension d'une approche existante pour la dispersion, à laquelle on a ajouté une composante garantissant le maintien de la connectivité. Comme deuxième contribution de ce chapitre nous avons présenté une loi de commande qui permet le contrôle de la compacité de la formation. En effet, il semble qu'une approche qui permettrait d'agir directement sur les angles inter-agents, contrôlant ainsi la connectivité du graphe de communication, n'a pas encore été proposée dans la littérature.

Finalement, nous avons présenté des résultats de simulation 2D pour une configuration de trois agents, comme un cas particulier de l'ensemble des configurations étudiées dans cette thèse. Dans le cadre de recherches futures, nous avons l'intention de développer un algorithme généralisé pour toutes sortes de configurations initiales et de renforcer l'analyse de stabilité pour de tels cas.

B.5 Conclusions générales

Le but de cette section est de résumer les contributions de cette thèse et de présenter quelques perspectives pour de futures recherches. Cette thèse porte sur les stratégies de contrôle distribué pour des systèmes multi-robots. Plus précisément, elle avance des résultats pertinents concernant le contrôle d'un groupe de véhicules mobiles qui garantissent les meilleures propriétés de connectivité possibles. Un résumé des travaux de cette thèse figure dans la suite.

B.5.1 Algorithmes de consensus

Au cours d'une grande partie de cette thèse, une attention particulière est accordée aux algorithmes de consensus pour des agents hétérogènes représentant, par exemple, différents modèles ou générations de robots. Ces travaux ont été motivés par des applications extrêmement exigeantes intégrant des agents hétérogènes tels que les opérations de recherche, de sauvetage ou de surveillance, dans des configurations civiles ou militaires. Bien que plusieurs travaux considèrent des algorithmes de consensus pour des ensembles homogènes d'agents, seuls quelques travaux étudient le cas des agents hétérogènes. De plus, les résultats qui en découlent présentent des inconvénients majeurs tels que les besoins de calcul ou la complexité.

Plus précisément, une stratégie, où le protocole de consensus est découplé du système d'origine, est présentée dans le Chapitre 2. En d'autres termes, nous attribuons à chaque agent une variable de contrôle supplémentaire qui permet d'atteindre un consensus, toute en garantissant que la grandeur d'intérêt de chaque agent converge vers cette variable supplémentaire. Le grand avantage de la méthode proposée réside dans une analyse découplée entre les différents composants du système. Par conséquent, il a été possible d'étendre nos résultats à des situations plus complexes, en considérant, par exemple, des retards de communication ou des filtres distribués avec des références externes. De futures recherches devront inclure l'étude des systèmes non linéaires ou des algorithmes de consensus en temps-fini. L'extension de méthodes et les outils d'analyse devront aussi être étudiés.

D'un autre côté, nous nous sommes intéressés à savoir s'il serait possible d'améliorer les propriétés de convergence des protocoles traditionnels. Dans le Chapitre 3, on propose d'incorporer de la mémoire dans de nouveaux protocoles pour des systèmes du type simple et double intégrateur, c'est-à-dire, d'ajouter une composante échantillonnée de l'état à la loi de commande. De plus, de la mémoire partielle et globale sont utilisées et les performances sont observées. L'analyse de stabilité de ces nouveaux algorithmes est basée sur des conditions du type LMI. Cependant, la complexité d'une telle approche augmente considérablement pour les réseaux de grande dimension. Une évolution logique des résultats de cette thèse devrait inclure d'autres outils d'analyse, ainsi que l'étude des cas asynchrones.

B.5.2 Déploiement compact d'agents

Dans le Chapitre 4 nous avons considéré le déploiement compact d'un ensemble d'agents. La première contribution de ces travaux correspond à une extension d'une approche existante pour la dispersion, à laquelle on a ajouté une composante qui garantit le maintien de la connectivité. La deuxième contribution comprend une loi de commande permettant le contrôle de la compacité de la formation. Finalement, nous avons également présenté des résultats de simulation 2D pour une configuration de trois agents, comme un cas particulier de l'ensemble de configurations étudié dans cette thèse. Dans le cadre de recherches futures, nous avons l'intention de développer un algorithme généralisé pour toutes sortes de configurations initiales et de renforcer l'analyse de stabilité pour de tels cas.

Bibliography

[1] Special issue on sensor networks and applications. *Proceedings of the IEEE*, 91(8), August 2003.

[2] Special issue on networking and information theory. *IEEE Transactions on Information Theory*, 52(6), June 2006.

[3] H. Ando, Y. Oasa, Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.

[4] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson. Distributed algebraic connectivity estimation for adaptive event-triggered consensus. In *American Control Conference*, 2012.

[5] A. Arsie and E. Frazzoli. Efficient routing of multiple vehicles with no explicit communications. *International Journal of Robust and Nonlinear Control*, 18:154 –164, 2008.

[6] S. Bag, S. Spurgeon, and C. Edwards. Output feedback sliding mode design for linear uncertain systems. *IEEE Proceedings -Control Theory and Applications*, 144(3):209 –216, may 1997.

[7] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, December 1998.

[8] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato. Distributed perimeter patrolling and tracking for camera networks. In 49^{th} *IEEE Conference on Decision and Control, Atlanta, USA*, pages 2093–2098, 2010.

[9] M. Basiri, A. Bishop, and P. Jensfelt. Distributed control of triangular formations with angle-only constraints. *Systems & Control Letters*, 59(2):147 – 154, 2010.

[10] M. Batalin and G. Sukhatme. *Coverage, Exploration, and Deployment by a Mobile Robot and Communication Network*. Springer Berlin / Heidelberg, 2003.

[11] M. Batalin and G. Sukhatme. The analysis of an efficient algorithm for robot coverage and exploration based on sensor network deployment. In *IEEE International Conference on Robotics and Automation*, pages 3478 – 3485, 2005.

[12] M. Batalin and G. Sukhatme. The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment. *IEEE Transactions on Robotics and Automation*, 23(4):661–675, 2007.

[13] D. Bauso, L. Giarre, and R. Pesenti. Attitude alignment of a team of UAVs under decentralized information structure. In *IEEE Conference on Control Applications, Istanbul, Turkey*, volume 1, pages 486 – 491, 2003.

[14] D. Bauso and R. Pesenti. Distributed consensus protocols for coordinating buyers. In 42^{nd} *IEEE Conference on Decision and Control, Hawaii*, pages 588–592, 2003.

[15] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A feedback architecture for formation control. In *American Control Conference, Chicago, IL, USA*, volume 6, pages 4087–4091, 2000.

[16] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9:777–790, 2001.

[17] R. W. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 94(7):1306 –1324, 2006.

[18] C. Belta and V. Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875, 2004.

[19] M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2-3):89–99, 2002.

[20] D. P. Bertsekas and J. Tsitsiklis. *Parallell and Distributed Computation.* Upper Saddle River, 1989.

[21] S. Bhat and D. Bernstein. Continuous finite-time stabilization of the translational and rotational double integrators. *IEEE Transactions on Automatic Control*, 43(5):678 –682, 1998.

[22] S. Bhat and D. Bernstein. Finite-time stability of contin- uous autonomous systems. *SIAM J. Control and Optimization*, 38:751–766, 2000.

[23] N. Biggs. *Algebraic Graph Theory.* Cambridge University Press, 1994.

[24] A. Bishop. A very relaxed control law for bearing-only triangular formation control. In 18^{th} *IFAC World Congress, Milan*, 2011.

[25] A. Bishop, I. Shames, and B. Anderson. Stabilization of rigid formations with direction-only constraints. In 50^{th} *IEEE Conference on Decision and Control and European Control Conference*, 2011.

[26] P.-A. Bliman and G. Ferrari-Trecate. Average consensus problems in networks of agents with delayed communications. *Automatica*, 44:1985–1995, 2008.

[27] V. Borkar and P. Varaiya. Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control*, 27(3):650–655, 1982.

[28] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[29] L. Briñón-Arranz. *Cooperative control design for a fleet of AUVs under communication constraints*. PhD thesis, Université de Grenoble, Gipsa-Lab, 2011.

[30] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. Translation control of a fleet circular formation of AUVs under finite communication range. In 48^{th} *IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference, Shanghai, China*, pages 8345–8350, 2009.

[31] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. Contraction control of a fleet circular formation of auvs under limited communication range. In *IEEE American Control Conference, Baltimore, MD, USA*, 2010.

[32] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. General framework using affine transformations to formation control design. In 2^{nd} *IFAC Workshop on Distributed Estimation and Control in Networked Systems, Annecy, France*, 2010.

[33] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. Collaborative estimation of gradient direction by a formation of auvs. In 5^{th} *International ICST Conference on Performance Evaluation Methodologies and Tools, 2011 Cachan, France*, 2011.

[34] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. Collaborative estimation of gradient direction by a formation of auvs under communication constraints. In 50^{th} *IEEE Conference on Decision and Control, held jointly with the European Control Conference, Orlando, FL, USA*, 2011.

[35] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. Elastic formation control based on affine transformations. In *American Control Conference, San Francisco, CA, USA*, pages 3984–3989, 2011.

[36] A. M. Bruckstein, N. Cohen, and A. Efrat. Ants, crickets, and frogs in cyclic pursuit. Technical Report CIS 9105, Center for Intelligent Systems, Technion, Haifa, Israel, 1991.

[37] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks. Applied Mathematics Series*. Princeton University Press, 2009.

[38] K. Cai and H. Ishii. Quantized consensus and averaging on gossip digraphs. *IEEE Transactions on Automatic Control*, 56(9):2087 –2100, 2011.

[39] K. Cai and H. Ishii. Average consensus on arbitrary strongly connected digraphs with dynamic topologies. In *American Control Conference*, 2012.

[40] K. Cai and H. Ishii. Average consensus on general strongly connected digraphs. *Automatica* Submitted, 2012.

[41] C. Canudas de Wit, F. Garin, R. Fabbiano, P. Rouchon, and A. Rousseau. Source localization using poisson integral. In 3^{th} *IFAC workshop on distributed estimation and control in networked systems*, 2012.

[42] M. Cao, A. S. Morse, and B. Anderson. Reaching a consensus in a dynamically changing environment-convergence rates, measurement delays and asynchronous events. *SIAM Journal on Control and Optimization*, 47:601–623, 2008.

[43] Z. Cao, L. Xie, B. Zhang, S. Wang, and M. Tan. Formation constrained multi-robot system in unknown environments. In *IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, volume 1, pages 735–740, 2003.

[44] R. Carli and F. Bullo. Quantized coordination algorithms for rendezvous and deployment. *SIAM Journal on Control and Optimization*, 48(3):1251–1274, 2009.

[45] J. Chen, D. Sun, J. Yang, and H. Chen. Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme. *The International Journal of Robotics Research*, 29(6):727–747, May 2010.

[46] J. Chen, D. Xie, and M. Yu. Consensus problem of networked multi-agent systems with constant communication delay: stochastic switching topology case. *International Journal of Control*-To appear, 85(9):1248–1262, 2012.

[47] T. Chen and B. Francis. *Optimal sampled-data control systems.* Springer-Verlag, Berlin, Germany, 1995.

[48] Y. Q. Chen and Z. Wang. Formation control: a review and a new consideration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada*, pages 3181–3186, 2005.

[49] X. Cheng. On-line collision-free path planning for service and assembly tasks by a two-arm robot. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1523 –1528 vol.2, may 1995.

[50] N. Chopra and M. W. Spong. *Passivity-based control of multi-agent systems.* Springer, 2006.

[51] N. Chopra and M. W. Spong. Output synchronization of nonlinear systems with relative degree one. *Recent. Adv. Learning Control*, LNCIS371:51–64, 2008.

[52] N. Chopra and M. W. Spong. On exponential synchronization of kuramoto oscillators. *IEEE Transactions on Automatic Control*, 54:353–357, 2009.

[53] N. Chopra, D. Stipanovic, and M. W. Spong. On synchronization and collision avoidance for mechanical systems. In *American Control Conference,*, pages 3713 –3718, 2008.

[54] F. Chung. Spectral graph theory. In *CBMS Regional Conference Series in Mathematics*, 1997.

[55] T. Clutton-Brock, P. Brotherton, A. Russell, M. O'Riain, D. Gaynor, R. Kansky, A. Griffin, M. Manse, L.Sharpe, G. McIlrath, T. Small, A. Moss, and S. Monfort.

Cooperation, control, and concession in meerkat groups. *Science*, 291(5503):478–481, 2001.

[56] J. Cochran and M. Krstić. Source seeking with a nonholonomic unicycle without position measurements and with tuning of angular velocity Part I: Stability analysis. In 46^{th} *IEEE Conference on Decision and Control, New Orleans, LA, USA*, pages 6009–6016, 2007.

[57] J. Cochran, A. Siranosian, N. Ghods, and M. Krstić. Source seeking with a nonholonomic unicycle without position measurements and with tuning of angular velocity Part II: Applications. In 46^{th} *IEEE Conference on Decision and Control, New Orleans, LA, USA*, pages 1951–1956, 2007.

[58] L. Conradt and T. Roper. Group decision-making in animals. *Nature*, 421:155 – 158, 2003.

[59] L. Consolinia, F. Morbidib, D. Prattichizzo, and M. Tosques. On the control of a leader-follower formation of nonholonomic mobile robots. In 45^{th} *IEEE Conference on Decision and Control, San Diego, CA, USA*, pages 5992–5997, 2006.

[60] L. Consolinia, F. Morbidib, D. Prattichizzo, and M. Tosques. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44:1343–1349, 2008.

[61] J. Cortés and F. Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, 2005.

[62] J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51:1289–1298, 2006.

[63] J. Cortés, S. Martínez, T. Karataş, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[64] J. Cortés. Finite-time convergent gradient flows with applications to network consensus. *Automatica*, 42(11):1993 – 2000, 2006.

[65] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

[66] X. Cui, C. T. Hardin, R. K. Ragade, and A. S. Elmaghraby. A swarm approach for emission sources localization. In *16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA*, pages 424–430, 2004.

[67] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6(3):86–94, 1993.

[68] P. Davidsson. Agent based social simulation: A computer science view. *Journal of Artificial Societies and Social Simulation*, 5(1), 2002.

[69] G. de Castro and F. Paganini. Convex synthesis of controllers for consensus. In *American Control Conference*, volume 6, pages 4933 –4938, 2004.

[70] M. Defoort, T. Floquet, A. Kökösy, and W. Perruquetti. Sliding-mode formation control for cooperative autonomous mobile robots. *IEEE Transactions on Industrial Electronics*, 55:3944–3953, 2008.

[71] M. Defoort, A. Kokosy, T. Floquet, W. Perruquettic, and J. Palos. Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach. *Robotics and Autonomous Systems*, 57:1094–1106, 2009.

[72] J. P. Desai, J. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *IEEE International Conference on Robotics and Automation, Leuven, Belgium*, volume 4, pages 2864–2869, 1998.

[73] D. V. Dimarogonas and K. J. Kyriakopoulos. An inverse agreement control strategy with application to swarm dispersion. In 46^{th} *IEEE Conference on Decision and Control*, pages 6148 –6153, 2007.

[74] D. V. Dimarogonas and K. J. Kyriakopoulos. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52:916–922, 2007.

[75] D. V. Dimarogonas and K. J. Kyriakopoulos. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems. *Automatica*, 44:2648–2654, 2008.

[76] D. V. Dimarogonas and K. J. Kyriakopoulos. Inverse agreement protocols with application to distributed multi-agent dispersion. *IEEE Transactions on Automatic Control*, 54(3):657 –663, 2009.

[77] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. Zavlanos. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica*, 42:229–243, 2006.

[78] K. Do. Flocking for multiple elliptical agents with limited communication ranges. *Asian Journal of Control*, 27(5):931–942, 2011.

[79] K. Do and M. Lau. Practical formation control of multiple unicycle-type mobile robots with limited sensing ranges. *Journal of Intelligent & Robotic Systems*, 64:245–275, 2011.

[80] A. Doniec, N. Bouraqadi, M. Defoort, V. T. Le, and S. Stinckwich. Distributed constraint reasoning applied to multi-robot exploration. In 21^{st} *IEEE International Conference on Tools with Artificial Intelligence, Newark, NJ, USA*, pages 159–166, 2009.

[81] D. D. Dudenhoeffer and M. P. Jones. A formation behavior for large-scale micro-robot force deployement. In 32^{nd} *Conference on Winter simulation, Orlando, FL, USA*, pages 972–982, 2000.

[82] C. Edwards and S. K. Spurgeon. Sliding mode stabilization of uncertain systems using only output information. *International Journal of Control*, 62(5):1129–1144, 1995.

[83] C. Edwards and S. K. Spurgeon. *Sliding Mode Control: Theory and Applications.* London, U.K.: Taylor and Francis, 1998.

[84] D. B. Edwards, T. A. Bean, D. L. Odell, and M. J. Anderson. A leader-follower algorithm for multiple auv formations. *IEEE/OES Autonomous Underwater Vehicles*, pages 40–46, 2004.

[85] M. Egerstedt and X. Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17:947–951, 2001.

[86] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. In 45^{th} *IEEE Conference on Decision and Control*, 2006.

[87] Y. Fan, G. Feng, and Y. Wang. Combination framework of rendezvous algorithm for multi-agent systems with limited sensing ranges. *Asian Journal of Control*, 13(2):283–294, 2011.

[88] J. A. Fax and R. M. Murray. Graph laplacians and stabilization of vehicle formations. In 15^{th} *IFAC World Conference, Barcelona, Spain*, July 2002.

[89] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49:1465–1476, 2004.

[90] J. Feddema, C. Lewis, and D. Schoenwald. Decentralized control of cooperative robotic vehicles: theory and application. *IEEE Transactions on Robotics and Automation*, 18(5):852 – 864, 2002.

[91] X. Feng and W. Long. Reaching agreement in finite time via continuous local state feedback. In *Chinese Control Conference*, 2007.

[92] J. Ferber. *Multi-Agent System: An Introduction to Distributed Artificial Intelligence.* Addison Wesley Longman, 1999.

[93] C. Ferrari, E. Pagello, J. Ota, and T. Arai. Multirobot motion coordination in space and time. *Robotics and Autonomous Systems*, 25:219–229, 1998.

[94] M. Fiedler. *Algebraic connectivity of graphs.* Number no. 2. Czechoslovak Mathematical Journal, 1973.

[95] T. Fraichard. Trajectory planning in dynamic workspace: a state-time space approach. *Advanced Robotics*, 13(6-8):75–94, 1999.

[96] E. Frazzoli, M. A. Dahleh, and E. Feron. Robust hybrid control for autonomous vehicle motion planning. In 39^{th} *IEEE Conference on Decision and Control*, pages 821–826, 2000.

[97] E. Fridman. A refined input delay approach to sampled-data control. *Automatica*, 46(2):421–427, 2010.

[98] E. Fridman, A. Seuret, and J.-P. Richard. Robust sampled-data stabilization of linear systems: An input delay approach. *Automatica*, 40(8):1141–1446, 2004.

[99] A. Frommer and D. B. Szyld. On asynchronous iterations. *Journal of Computational and Applied Mathematics*, 123(1):201–216, 2000.

[100] H. Fujioka. Stability analysis of systems with aperiodic sample-and-hold devices. *Automatica*, 45(3):771–775, 2009.

[101] A. Ganguli, J. Cortés, and F. Bullo. Distributed deployment of asynchronous guards in art galleries. In *American Control Conference*, pages 1416–1421, 2006.

[102] V. Gazi and K. M. Passino. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–697, 2003.

[103] S. S. Ge, C.-H. Fua, and K. W. Lim. Multi-robot formations: queues and artificial potential trenches. In *IEEE International Conference on Robotics and Automation, New Orleans, LA, USA*, volume 4, pages 3345–3350, 2004.

[104] R. Ghabcheloo, A. Pascoal, C. Silvestre, and I. Kaminer. Coordinated path following control of multiple wheeled robots with directed communication links. In 44^{th} *IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, 2005.

[105] S. Glavaski, M. Chaves, R. Day, P. Nag, A. Williams, and W. Zhang. Vehicle networks: achieving regular formation. In *American Control Conference, Minneapolis, MN, USA*, volume 5, pages 4095–4100, 2003.

[106] C. Godsil and G. Royle. *Algebraic Graph Theory*, volume 207. Springer-Verlag, 2001.

[107] R. Goebel, R. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28 –93, 2009.

[108] M. F. Z. Goldgeier and P. S. Krishnaprasad. Control of small formations using shape coordinates. In *IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, pages 2510–2515, 2003.

[109] M. Greytak and F. S. Hover. Motion planning with an analytic risk cost for holonomic vehicles. In 48^{th} *IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference, Shanghai, China*, 2009.

[110] K. Gu, V.-L. Kharitonov, and J. Chen. *Stability of time-delay systems*. Birkhauser, 2003.

[111] V. Gupta, B. Hassibi, and R. M. Murray. Stability analysis of stochastically varying formations of dynamic agents. In 42^{nd} *IEEE Conference on Decision and Control,Maui, Hawai*, pages 504–509, 2003.

[112] H. Hao and P. Barooah. Improving convergence rate of distributed consensus through asymmetric weights. In *American Control Conference*, 2012.

[113] Y. Hatano and M. Mesbahi. Agreement over random networks. In 43^{rd} *IEEE Conference on Decision and Control*, pages 2010 – 2015, 2004.

[114] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control*, 50(7):1607–1613, 2004.

[115] B. Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72(1):329 –365, 1995.

[116] D. Helbing. Traffic and related self-driven many-particle systems. *Rev. Mod. Phys.*, 73:1067–1141, 2001.

[117] J. M. Hendrickx. *Graphs and Networks for the Analysis of Autonomous Agent Systems*. PhD thesis, Université catholique de Louvain, Ecole Polytechnique de Louvain, Département d'Ingénierie Mathématique, 2008.

[118] S. Hernandez and D. A. Paley. Three-dimensional motion coordination in a time-invariant flowfield. In 48^{th} *IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference, Shanghai, China*, pages 7043–7048, 2009.

[119] J. Hespanha. Uniform stability of switched linear systems: extensions of lasalle's invariance principle. *IEEE Transactions on Automatic Control*, 49(4):470 – 482, 2004.

[120] J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of IEEE*, 95(1):138 – 162, 2007.

[121] L. Hetel. *Robust stability and control of switched linear systems*. Phd thesis, Institut National Polytechnique de Lorraine - INPL, Nancy France, 2007.

[122] L. Hetel, J. Daafouz, and C. Iung. Stabilization of arbitrary switched linear systems with unknown time-varying delays. *IEEE Transactions on Automatic Control*, 51(10):1668–1674, 2006.

[123] Y. Hong, J. Huang, and Y. Xu. On an output feedback finite-time stabilization problem. *IEEE Transactions on Automatic Control*, 46(2):305 –309, 2001.

[124] Y. Hong and Z.-P. Jiang. Finite-time stabilization of nonlinear systems with parametric and dynamic uncertainties. *IEEE Transactions on Automatic Control*, 51(12):1950 –1956, 2006.

[125] T.-y. Huang, X.-b. Chen, W.-b. Xu, and W. Wang. A self-organized approach to collaborative handling of multi-robot systems. In *Advances in Swarm Intelligence*, volume 6729 of *Lecture Notes in Computer Science*, pages 82–90. Springer Berlin / Heidelberg, 2011.

[126] A. Isidori. *Nonlinear Control Systems*. Springer, 1995.

[127] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

[128] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[129] M. Ji and M. Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.

[130] U. Jönsson and C.-Y. Kao. Consensus of heterogeneous linear agents applied to a formation control problem. In 49^{th} *IEEE Conference on Decision and Control*, pages 6902 –6907, 2010.

[131] E. W. Justh and P. S. Krishnaprasad. Equilibria and steering laws for planar formations. *Systems & Control Letters*, 52(1):25–38, 2004.

[132] A. Kashyap, T. Basar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.

[133] H. Kim, H. Shim, and J. H. Seo. Output consensus of heterogeneous uncertain linear multi-agent systems. *IEEE Transactions on Automatic Control*, 56(1):200–206, 2011.

[134] T.-H. Kim and T. Sugie. Cooperative control for target-capturing task based on a cyclic pursuit strategy. *Automatica*, 43:1426–1431, 2007.

[135] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*, 51(6):949–962, 2006.

[136] D. J. Klein and K. A. Morgansen. Controlled collective motion for trajectory tracking. In *American Control Conference, Minneapolis, MN, USA*, pages 5269–5275, June 2006.

[137] D. Kostic, S. Adinandra, J. Caarls, N. V. D. Wouw, and H. Nijmeijer. Collision-free tracking control of unicycle mobile robots. In 48^{th} *IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference, Shanghai, China*, 2009.

[138] M. Krstić and H.-H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595–601, 2000.

[139] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, volume 39 of *Lecture Notes in Physics*, pages 420–422. Springer Berlin / Heidelberg, 1975.

[140] Y. Kuramoto. Collective synchronization of pulse-coupled oscillators and excitable units. *Physica D: Nonlinear Phenomena*, 50:15–30, 1991.

[141] A. Kwok and S. Martínez. Unicycle coverage control via hybrid modeling. *IEEE Transactions on Automatic Control*, 55:528–532, 2010.

[142] Y. Lan, Z. Lin, M. Cao, and G. Yan. A distributed reconfigurable control law for escorting and patrolling missions using teams of unicycles. In 49^{th} *IEEE Conference on Decision and Control, Atlanta, GA, USA*, pages 5456–5461, 2010.

[143] S. LaValle and S. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6):912 – 925, 1998.

[144] D. Lee and M. W. Spong. Agreement with non-uniform information delays. In *American Control Conference*, 2006.

[145] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Frantatoni, and R. E. Davis. Collective motion, sensor networks and ocean sampling. *Proceedings of the IEEE*, 95:48–74, 2007.

[146] I. Lestas and G. Vinnicombe. Scalable robust stability for nonsymmetric heterogeneous networks. *Automatica*, 43(4):714 – 723, 2007.

[147] M. A. Lewis and K.-H. Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4(4):387–403, 1997.

[148] J. Lin, A. S. Morse, and B. Anderson. The multi-agent rendezvous problem. In 42^{nd} *IEEE Conference on Decision and Control, Maui, Hawaii*, pages 1508 – 1513, 2003.

[149] J. Lin, A. S. Morse, and B. Anderson. The multi-agent rendezvous problem - the asynchronous case. In 43^{rd} *IEEE Conference on Decision and Control*, pages 1926 – 1931 Vol.2, 2004.

[150] J. Lin, A. S. Morse, and B. Anderson. The multi-agent rendezvous problem. Part 1: The synchronous case. *SIAM Journal on Control and Optimization*, 46:2096–2119, 2007.

[151] J. Lin, A. S. Morse, and B. Anderson. The multi-agent rendezvous problem. Part 2: The asynchronous case. *SIAM Journal on Control and Optimization*, 46:2120–2147, 2007.

[152] P. Lin and Y. Jia. Distributed rotating formation control of multi-agent systems. *Systems & Control Letters*, 59(10):587 – 595, 2010.

[153] P. Lin, Y. M. Jia, J. P. Du, and S. Y. Yuan. Distributed consensus control for second-order agents with fixed topology and time-delay. In 26^{th} *Chinese Control Conference*, 2007.

[154] Z. Lin, M. Broucke, and B. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, 2004.

[155] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on Automatic Control*, 50:121–127, 2005.

[156] K. Liu, V. Suplin, and E. Fridman. Stability of linear systems with general sawtooth delay. *IMA J. of Math. Control and Information*, 27(4):419–436, 2010.

[157] S. Liu, L. Xie, and H. Zhang. Distributed consensus for multi-agent systems with delays and noises in transmission channels. *Automatica*, 47(5):920 – 934, 2011.

[158] V. J. Lumelsky and K. R. Harinarayan. Decentralized motion planning for multiple mobile robots: the cocktail party model. *Autonomous Robots*, 4(1):121–135, 1997.

[159] J.-M. Luna, R. Fierro, C. Abdallah, and J. Wood. An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors. In 49^{th} *IEEE Conference on Decision and Control, Atlanta, GA, USA*, pages 1250–1256, 2010.

[160] J. Lygeros, K. Johansson, S. Simic, J. Zhang, and S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2 – 17, 2003.

[161] J. A. Marshall, M. E. Broucke, and B. A. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 2004.

[162] S. Martin, A. Fazeli, A. Jadbabaie, and A. Girard. Multi-agent flocking with random communication radius. In *American Control Conference*, 2012.

[163] S. Martin and A. Girard. Sufficient conditions for flocking via graph robustness analysis. In 49^{th} *IEEE Conference on Decision and Control*, 2010.

[164] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42:661–668, 2006.

[165] S. Martínez, J. Cortés, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27:75–88, 2007.

[166] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *The International Journal of Robotics Research*, 27:107–126, 2008.

[167] S. McCamish. Autonomous distributed control of simultaneous multiple spacecraft proximity maneuvers. *IEEE Transactions on Automation Science and Engineering*, 7:630– 644, 2010.

[168] R. Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197,198:143 – 176, 1994.

[169] L. Mirkin. Some remarks on the use of time-varying delay to model sample-and-hold circuits. *IEEE Transactions on Automatic Control*, 52(6):1009–1112, 2007.

[170] S. Monteiro and E. Bicho. A dynamical systems approach to behavior-based formation control. In *IEEE International Conference on Robotics and Automation, Washington D.C., USA*, volume 3, pages 2606–2611, 2002.

[171] B. J. Moore and C. Canudas-de-Wit. Formation control via distributed optimization of alignment error. In 48^{th} *IEEE Decision and Control held jointly with the 28th Chinese Control Conference, Shanghai, China*, pages 3075–3080, 2009.

[172] B. J. Moore and C. Canudas-de-Wit. Source seeking via collaborative measurements by a circular formation of agents. In *Prooceedings of the 2010 IEEE American Control Conference, Baltimore, USA*, pages 6417–6422, 2010.

[173] L. Moreau. Stability of continuous-time distributed consensus algorithms. In 43^{rd} *IEEE Conference on Decision and Control*, 2004.

[174] L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[175] Y. Moreno and A. F. Pacheco. Synchronization of kuramoto oscillators in scale-free networks. *Europhysics Letters*, 68(4), 2004.

[176] U. Münz. *Delay Robustness in Cooperative Control*. PhD thesis, Institut für Systemtheorie und Regelungstechnik, Universität Stuttgart, 2010.

[177] U. Münz, A. Papachristodoulou, and F. Allgöwer. Delay-dependent rendezvous and flocking of large scale multi-agent systems with communication delays. In 47^{th} *IEEE Conference on Decision and Control*, 2008.

[178] U. Münz, A. Papachristodoulou, and F. Allgöwer. Generalized nyquist consensus condition for high-order linear multi-agent systems with communication delays. In 48^{th} *IEEE Conference on Decision and Control*, pages 4765 –4771, 2009.

[179] U. Münz, A. Papachristodoulou, and F. Allgöwer. Delay robustness in consensus problems. *Automatica*, 46(8):1252 – 1265, 2010.

[180] R. M. Murray, J. Fax, R. Olfati-Saber, and D. Spanos. Consensus and cooperation in multi-agent networked systems. *Proceedings of IEEE*, pages 215 –233, 2007.

[181] S. Muthukrishnan, B. Ghosh, and M. H. Schultz. First and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory of Computing Systems*, 31(4):331–354, 1998.

[182] P. Naghshtabrizi, J. Hespanha, and A. Teel. Exponential stability of impulsive systems with application to uncertain sampled-data systems. *Systems & Control Letters*, 57(5):378–385, 2008.

[183] S. Nair and N. E. Leonard. Stable synchronization of mechanical system networks. *SIAM Journal on Control and Optimization*, 47(2):234 – 265, 2008.

[184] A. Nedic and A. Ozdaglar. Convergence rate for consensus with delays. *Journal of Global Optimization*, 47:437–456, 2010.

[185] P. O'Donnell and T. Lozano-Periz. Deadlock-free and collision-free coordination of two robot manipulators. In *IEEE International Conference on Robotics and Automation*, pages 484 –489 vol.1, may 1989.

[186] P. Ögren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49:1292–1302, 2004.

[187] Y. Oishi and H. Fujioka. Stability and stabilization of aperiodic sampled-data control systems: An approach using robust linear matrix inequalities. In *Joint* 48^{th} *IEEE Conference on Decision and Control and* 28^{th} *Chinese Control Conference*, pages 8142 – 8147, 2009.

[188] A. Okubo. Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds. *Advances in Biophysics*, 22(0):1 – 94, 1986.

[189] R. Olfati-Saber. Ultrafast consensus in small-world networks. In *American Control Conference*, pages 2371 – 2378, 2005.

[190] R. Olfati-Saber. Distributed Kalman filtering and sensor fusion in sensor networks. In *Networked embedded sensing and control*, volume 331 of *Lecture Notes in Control and Inform. Sci.*, pages 157–167. Springer, Berlin, 2006.

[191] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51:401–420, 2006.

[192] R. Olfati-Saber, J. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215 –233, 2007.

[193] R. Olfati-Saber and R. M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In 41^{st} *IEEE Conference on Decision and Control, Las Vegas, Nevada, USA*, 2002.

[194] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents whith switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[195] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In 44^{th} *IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, pages 6698–6703, 2005.

[196] J. Opderbecke, M. Drogou, and M. E. Bouhier. D8.1 description of the scientific mission scenario(s) to be investigated for the marine application. Technical report, Ifremer, FeedNetBack project, 2009.

[197] R. Oung, A. Ramezani, and R. D'Andrea. Feasibility of a distributed flight array. In 48^{th} *IEEE Conference on Decision and Control held jointly with the* 28^{th} *Chinese Control Conference, Shanghai, China*, 2009.

[198] D. A. Paley, N. E. Leonard, and R. Sepulchre. Collective motion: Bistability and trajectory tracking. In 43^{th} *IEEE Conference Decision and Control, Paradise Island, Bahamas*, 2004.

[199] D. A. Paley, N. E. Leonard, and R. Sepulchre. Oscillator models and collective motion: Splay state stabilization of self-propelled particles. In 44^{th} *IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, pages 3935–3940, December 2005.

[200] D. A. Paley, N. E. Leonard, and R. Sepulchre. Stabilization of symmetric formations to motion around convex loops. *Systems & Control Letters*, 57(3):209–215, 2008.

[201] D. A. Paley and C. Peterson. Stabilization of collective motion in a time-invariant flowfield. *Journal of Guidance, Control, and Dynamics*, 32:771–779, 2009.

[202] A. Papachristodoulou, A. Jadbabaie, and U. Münz. Effects of delay in multi-agent consensus and oscillator synchronization. *IEEE Transactions on Automatic Control*, 55(6):1471 –1477, 2010.

[203] M. Pavella and P. Murthy. *Transient stablity of power systems: Theory and Practise.* Communication and Control Engineering. John wiley and Sons, 1994.

[204] M. Pavone and E. Frazolli. Decentralized patern formation and path coverage. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):633 – 644, 2007.

[205] M. Peet, A. Papachristodoulou, and S. Lall. Positive forms and stability of linear time-delay systems. *SIAM Journal on Control and Optimization*, 47(6):3227–3258, 2009.

[206] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira. Sensing and coverage for a network of heterogeneous robots. In *47th IEEE Conference on Decision and Control*, pages 3947–3952, Cancun, Mexico, 2008.

[207] P.Miller. Swarm theory. *National Geographic*, 7:126–147, 2007.

[208] S. Poduri and G. Sukhatme. Constrained coverage for mobile sensor networks. In *IEEE International Conference on Robotics and Automation*, pages 165 – 171 Vol.1, 2004.

[209] M. Porfiri and D. Stilwell. Consensus seeking over random weighted directed graphs. *IEEE Transactions on Automatic Control*, 52(9):1767 – 1773, 2007.

[210] V. Preciado, M. Zavlanos, A. Jadbabaie, and G. Pappas. Distributed control of the laplacian spectral moments of a network. In *American Control Conference*, 2010.

[211] Z. Qu, J. Chunyu, and J. Wang. Nonlinear cooperative control for consensus of non linear and heterogeneous systems. In *IEEE Conference Decision and Control*, 2007.

[212] Z. Qu, C. Li, and F. Lewis. Cooperative control based on distributed estimation of network connectivity. In *American Control Conference*, 2011.

[213] R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In 43^{rd} *IEEE Conference on Decision and Control, Paradise Islands, The Bahamas*, pages 2453–2459, 2004.

[214] R.D'Andrea and G. Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478 – 1495, 2003.

[215] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27:171–194, May 1999.

[216] W. Ren. On consensus algorithms for double-integrator dynamics. In 46^{th} *IEEE Conference on Decision and Control*, 2007.

[217] W. Ren. On consensus algorithms for double-integrator dynamics. *IEEE Transactions on Automatic Control*, 53(6):1503–1509, 2008.

[218] W. Ren and E. Atkins. Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control*, 17:1002–1033, 2007.

[219] W. Ren and R. W. Beard. Consensus seeking in multi-agents systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50:655–661, 2005.

[220] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communication and Control Engineering. Springer, 2007.

[221] W. Ren, R. W. Beard, and E. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference, Portland, OR, USA*, pages 1859–1864, 2005.

[222] W. Ren, R. W. Beard, and E. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71 –82, 2007.

[223] W. Ren, R. W. Beard, and T. McLain. *Coordination Variables and Consensus Building in Multiple Vehicle Systems*, volume 309 of *Lecture Notes in Control and Information Sciences*. Springer Berlin / Heidelberg, 2005.

[224] W. Ren and Y. Cao. Convergence on sampled-data consensus algorithms for double-integrator dynamics. In 47^{th} *IEEE Conference on Decision and Control*, 2008.

[225] W. Ren and Y. Cao. Overview of recent research in distributed multi-agent coordination. In *Distributed Coordination of Multi-agent Networks*, Communications and Control Engineering, pages 23–41. Springer London, 2011.

[226] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. Kosmatopoulos. Adaptive-based distributed cooperative multi-robot coverage. In *American Control Conference, San Francisco, CA, USA*, pages 468–473, 2011.

[227] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In 14^{th} *annual conference on Computer graphics and interactive techniques, Anaheim, CA, USA*, 1987.

[228] G. Rodrigues de Campos, L. Briñón-Arranz, A. Seuret, and S.-I. Niculescu. On the consensus of heterogeneous multi-agent systems: a decoupling approach. In 3^{rd} *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2012.

[229] G. Rodrigues de Campos, D. V. Dimarogonas, A. Seuret, and K. H. Johansson. Distributed control strategy for multi-agent systems compact formations. *Journal paper under Preparation.*

[230] G. Rodrigues de Campos and A. Seuret. Improved stability of consensus algorithms for multi-agent systems using appropriated sampling. *Journal paper under Preparation.*

[231] G. Rodrigues de Campos and A. Seuret. Continuous-time double integrator consensus algorithms improved by an appropriate sampling. In 2^{nd} *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2010.

[232] G. Rodrigues de Campos and A. Seuret. Improved consensus algorithms using memory effects. In 50^{th} *IEEE Conference on Decision and Control and European Control Conference*, 2011.

[233] G. Rodrigues de Campos, I. Shames, and A. Bishop. Distributed labeling in autonomous agent populations. In 20^{th} *International Symposium on Mathematical Theory of Networks and Systems*, 2012.

[234] M. Rotkowitz and S. Lall. Affine controller parameterization for decentralized control over banach spaces. *IEEE Transactions on Automatic Control*, 51(9):1497 –1500, 2006.

[235] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Englewood Cliffs, 1995.

[236] L. Sabattini, C. Secchi, and C. Fantuzzi. Potential based control strategy for arbitrary shape formations of mobile robots. In *IEEE/RSJ International conference on Intelligent robots and systems, St. Louis, MO, USA*, pages 3762–3767, 2009.

[237] S. S. Sahyoun, S. M. Djouadi, and H. Qi. Dynamic plume tracking using mobile sensors. In *American Control Conference, Baltimore, MD, USA*, pages 2915–2920, 2010.

[238] A. Sarlette. *Geometry and Symmetries in Coordination Control.* PhD thesis, Université de Liège, Faculté des Sciences Appliquées, 2009.

[239] A. V. Savkin. Coordinated collective motion of groups of autonomous mobile robots: Analysis of vicsek's model. *IEEE Transactions on Automatic Control*, 49(6), 2004.

[240] K. Savla, G. Notarstefano, and F. Bullo. Maintaining limited-range connectivity among second-order agents. *SIAM Journal on Control and Optimization*, 2006.

[241] L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the n-torus. *Systems & Control Letters*, 56(5):335 – 341, 2007.

[242] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 44(11):2557–2562, 2009.

[243] M. Schwager, J. McLurkin, and D. Rus. Distributed coverage control with sensory feedback for networked robots. In *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, USA, 2006.

[244] R. Sepulchre. Consensus on nonlinear spaces. In 8^{th} *IFAC Symposium on Nonlinear Control Systems, Bologna, Italy*, 2010.

[245] R. Sepulchre. Consensus on nonlinear spaces. *Annual Reviews in Control*, 35:56–64, 2011.

[246] R. Sepulchre, D. A. Paley, and N. E. Leonard. Group coordination and cooperative control of steered particles in the plane. *Lecture Notes in control and Information Sciences*, 2006.

[247] R. Sepulchre, D. A. Paley, and N. E. Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 2007.

[248] R. Sepulchre, D. A. Paley, and N. E. Leonard. Stabilization of planar collective motion with limited communication. *IEEE Transactions on Automatic Control*, 53:706–719, 2008.

[249] A. Seuret. Stability analysis for sampled-data systems with a time-varying period. In *Joint* 48^{th} *IEEE Conference on Decision and Control and* 28^{th} *Chinese Control Conference*, pages 8130 – 8135, 2009.

[250] A. Seuret. A novel stability analysis of linear systems under asynchronous samplings. *Automatica*, 48(1):177 – 182, 2012.

[251] A. Seuret, D. V. Dimarogonas, and K. H. Johansson. Consensus under communication delays. In 47^{th} *IEEE Conference on Decision and Control*, pages 4922–4927, 2008.

[252] A. Seuret, C. Edwards, S. K. Spurgeon, and E. Fridman. Static output feedback sliding mode control design via an artificial stabilizing delay. *IEEE Transactions on Automatic Control*, 54(2):256–265, 2008.

[253] T. A. Severson and D. A. Paley. Distributed optimization for radar mission coordination. In *American Control Conference*, 2012.

[254] V. Sharma, M. Savchenko, E. Frazzoli, and P. Voulgaris. Time complexity of sensor-based vehicle routing. In S. Thrun, G. Sukhatme, S. Schaal, and O. Brock, editors, *Robotics Science and Systems*, pages 297–304. 2005.

[255] A. Simonetto, T. Keviczky, and R. Babuska. On distributed maximization of algebraic connectivity in robotic networks. In *American Control Conference*, 2011.

[256] A. Sinha and D. Ghose. Generalization of linear cyclic pursuit with application to rendezvous of multiple autonomous agents. *IEEE Transactions on Automatic Control*, 51(11):1819 –1824, 2006.

[257] R. Sipahi, F. Atay, and S.-I. Niculescu. Stability of traffic flow behavior with distributed delays modeling the memory effects of the drivers. *SIAM Journal on Applied Mathematics*, 68(3):738–759, 2007.

[258] J.-J. Slotine and W. Wang. A study of synchronization and group cooperation using partial contraction theory. In *Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Sciences*, pages 443–446. Springer Berlin / Heidelberg, 2005.

[259] S. L. Smith, M. Broucke, and B. Francis. Curve shortening and the rendezvous problem for mobile autonomous robots. *IEEE Transactions on Automatic Control*, 52(6):1154 –1159, 2007.

[260] S. L. Smith and F. Bullo. Target assignment for robotic networks: Asymptotic performance under limited communication. In *American Control Conference*, New York, 2007.

[261] D. Spanos, R. Olfati-Saber, and R. M. Murray. Approximate distributed kalman filtering in sensor networks with quantifiable performance. In 4^{th} *International Symposium on Information Processing in Sensor Networks, IPSN05*, pages 133–139, 2005.

[262] D. Spanos, R. Olfati-Saber, and R. M. Murray. Dynamic consensus on mobile networks. In 16^{th} *IFAC World Congress*, 2005.

[263] S. Spry and J. K. Hedrick. Formation control using generalized coordinates. In 43^{rd} *IEEE Conference on Decision and Control, Paradis Island, Bahamas*, pages 2441–2446, 2004.

[264] R. Srikant. *The mathematics of internet congestion control.* Communication and Control Engineering. Birkhäuser, 2004.

[265] S. F. Stan and A. Graesser. *Is It an agent, or just a program?: A taxonomy for autonomous agents*, volume 1193 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1997.

[266] J. C. Strikwerda. A probabilistic analysis of asynchronous iteration. *Linear Algebra and its Applications*, 349(1):125 – 154, 2002.

[267] S. H. Strogatz. From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica: Nonlinear Phenomena*, 143:1–20, 2000.

[268] S. H. Strogatz. *SYNC: the emerging science of spontaneous order*. Theia, 2003.

[269] H. Su, X. Wang, and G. Chen. Rendezvous of multiple mobile agents with preserved network connectivity. *Systems & Control Letters*, 59:313–322, 2010.

[270] K. Sugihara and I. Suzuki. Distributed motion coordination of multiple mobile robots. In 5^{th} *IEEE International Symposium on Intelligent Control*, 1990.

[271] Y. Suh. Stability and stabilization of nonuniform sampling systems. *Automatica*, 44(12):3222–3226, 2008.

[272] V. A. Sujan and S. Dubowsky. Visually guided cooperative robot actions based on information quality. *Autonomous Robots*, 19:89–110, 2005.

[273] F. Sun, J. Chen, Z.-H. Guan, L. Ding, and T. Li. Leader-following finite-time consensus for multi-agent systems with jointly-reachable leader. *Nonlinear Analysis: Real World Applications*, 13(5):2271–2284, 2012.

[274] Y. Sun and L. Wang. Consensus of multi-agent systems in directed networks with nonuniform time-varying delays. *IEEE Transactions on Automatic Control*, 54(7):1607–1613, 2009.

[275] Y. Sun, L. Wang, and G. Xie. Average consensus in networks of dynamic agents with switching topologies and multiple time-varying delays. *Systems & Control Letters*, 57(2):175–183, 2008.

[276] A. Tahbaz-Salehi. A necessary and sufficient condition for consensus over random networks. *IEEE Transactions on Automatic Control*, 53(3):791 – 795, 2008.

[277] H. Takahashi, H. Nishi, and K. Ohnishi. Autonomous decentralized control for formation of multiple mobile robots considering ability of robot. *IEEE Transactions on Industrial Electronics*, 51(6):1272–1279, 2004.

[278] K.-H. Tan and M. A. Lewis. Virtual structures for high-precision cooperative mobile robotic control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Osaka, Japan*, volume 1, pages 132–139, 1996.

[279] H. G. Tanner, A. Jadbabaie, and G. Pappas. Stable flocking of mobile agents, part I: Fixed topology. In 42^{nd} *IEEE Conference on Decision and Control, Maui, Hawaii*, pages 2010–2015, 2003.

[280] H. G. Tanner, A. Jadbabaie, and G. Pappas. Stable flocking of mobile agents, part II: Dynamic topology. In 42^{nd} *IEEE Conference on Decision and Control, Maui, Hawaii*, pages 2016–2021, 2003.

[281] H. G. Tanner, A. Jadbabaie, and G. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, 2007.

[282] H. G. Tanner, A. Jadbabaie, and G. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, 2007.

[283] H. G. Tanner, G. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, 2004.

[284] Y. Tian and C. Liu. Consensus of multi-agent systems with diverse input and communication delays. *IEEE Transactions on Automatic Control*, 53(9):2122–2128, 2008.

[285] Y. Tian and C. Liu. Robust consensus of multi-agent systems with diverse input delays and asymmetric interconnection perturbations. *Automatica*, 45(5):1374–1353, 2009.

[286] V.-H. Tran and S.-G. Lee. A stable formation control using approximation of translational and angular accelerations. *International Journal of Advanced Robotic Systems*, 8:65–75, 2011.

[287] W. Truszkowski, H. L. Hallock, C. Rouff, J. Karlin, J. Rash, M. Hinchey, and R. Sterritt. *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, chapter Swarms in Space Missions, pages 207–221. NASA Monographs in Systems and Software Engineering, 2009.

[288] J. Tsitsiklis. *Problems in Decentralized Decision Making and Computation.* PhD thesis, Department of EECS, Laboratory for Information and Decision Systems, MIT, 1984.

[289] J. Tsitsiklis and M. Athans. Convergence and asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, 29(1):42 – 50, 1984.

[290] J. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[291] N. Ukita and T. Matsuyama. Real-time cooperative multi-target tracking by communicating active vision agents. *Computer Vision and Image Understanding*, 97:137–179, 2005.

[292] J. van den Berg and M. Overmars. Prioritized motion planning for multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 430 – 435, 2005.

[293] E. Verriest and W. Michiels. Stability analysis of linear systems with stochastically varying delays. *Systems & Control Letters*, 58(10-11):783–791, 2009.

[294] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, 1995.

[295] L. Wang and F. Xiao. Finite-time consensus problems for networks of dynamic agents. *IEEE Transactions on Automatic Control*, 55(4):950 –955, 2010.

[296] X. Wang and Y. Hong. Finite-time consensus for multi-agent networks with second-order agent dynamics. In 17^{th} *World Congress The International Federation of Automatic Control*, 2008.

[297] L. G. Weiss. Autonomous robots in the fog of war. *IEEE Spectrum*, 2011.

[298] P. Wieland, J.-S. Kimb, and F. Allgöwer. On topology and dynamics of consensus among linear high-order agents. *International Journal of Systems Science*, 42:1831–1842, 2011.

[299] P. Wieland, R. Sepulchre, and F. Allgöwer. An internal model principle is necessary and sufficient for linear output synchronization. *Automatica*, 47:1068–1074, 2011.

[300] M. Wooldridge and N. R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10:115–152, 1995.

[301] F. Xiao and L. Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, 53(8):1804–1816, 2008.

[302] F. Xiao and L. Wang. Consensus protocols for discrete-time multi-agent systems with time-varying delays. *Automatica*, 44(10):2577–2582, 2008.

[303] F. Xiao, L. Wang, J. Chen, and Y. Gao. Finite-time formation control for multi-agent systems. *Automatica*, 45(11):2605 – 2611, 2009.

[304] L. Xiao and S. Boyd. Faster linear iterations for distributed averaging. *Systems & Control Letters*, 53:65–78, 2004.

[305] G. Xie and L. Wang. Consensus control for a class of networks of dynamic agents. *International Journal of Robust and Nonlinear Control*, 17(10-11):941–959, 2007.

[306] J. Xiong and J. Lam. Stabilization of linear systems over networks with bounded packet loss. *Automatica*, 43(1):80 – 87, 2007.

[307] W. Yang, A. Bertozzi, and X. Wang. Stability of a second order consensus algorithm with time delay. In 47^{th} *IEEE Conference onDecision and Control,*, pages 2926 –2931, 2008.

[308] J. Yu, S. LaValle, and D. Liberzon. Rendezvous without coordinates. In 47^{th} *IEEE Conference on Decision and Control*, 2008.

[309] J. Yu, S. LaValle, and D. Liberzon. Rendezvous without coordinates. *IEEE Transactions on Automatic Control*, 57(2):421–434, 2012.

[310] J. Yu and L. Wang. Group consensus in multi-agent systems with switching topologies and communication delays. *Systems & Control Letters*, 59(6):340–348, 2010.

[311] S. Zampieri. A survey of recent results in Networked Control Systems. In *Proc. of the* 17^{th} *IFAC World Congress*, pages 2886–2894, Seoul, Korea, July 2008.

[312] S. Zampieri and A. Seuret. D01.03 classification of networked control systems. Technical report, NeCS Team, INRIA Rône-Alpes, FeedNetBack project, August 2009.

[313] M. Zavlanos and G. Pappas. Controlling connectivity of dynamic graphs. In *Control and Decision Conference and European Control Conference*, pages 6388–6393, 2005.

[314] M. Zavlanos and G. Pappas. Distributed connectivity control of mobile networks. *IEEE Transactions on Robotics*, 24(6):1416–1428, 2008.

[315] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Transactions on Automatic Control*, 55:650–663, 2010.

[316] W. Zhang and M. Branicky. Stability of networked control systems with time-varying transmission period. In *Proceedings Of The Annual Allerton Conference On Communication Control And Computing*, volume 39, pages 1205–1214. The University; 1998, 2001.

[317] W. Zhang, M. Branicky, and S. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 1(21):84–99, 2001.

[318] Y. Zhang. Consensus of data-sampled multi-agent systems with random communication delay and packet loss. *IEEE Transactions on Automatic Control*, 55(4):939 – 943, 2010.

[319] Y. Zhang and Y.-P. Tian. Consentability and protocol design of multi-agent systems with stochastic switching topology. *Automatica*, 45(5):1195 – 1201, 2009.

[320] Y. Zheng and L. Wang. Consensus of heterogeneous multi-agent systems without velocity measurements. *International Journal of Control*, 85(7):906–9014, 2012.

[321] Y. Zheng and L. Wang. Finite-timeconsensus of heterogeneousmulti-agentsystems with and without velocitymeasurements. *Systems & Control Letters*, 61(8):871–878, 2012.

[322] Y. Zheng, Y. Zhu, and L. Wang. Consensus of heterogeneous multi-agent systems. *IET Control Theory and Applications*, 5:1881–1888, 2011.

[323] J. Zhu, Y.-P. Tian, and J. Kuang. On the general consensus protocol of multi-agent systems with double-integrator dynamics. *Linear Algebra and its Applications*, 431(5-7):701–715, 2009.

[324] M. Zhu and S. Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.

[325] W. Zhuang, Y. Zhang, and H. Tan. A decentralized control algorithm for maintaining communication links in mobile sensor networks. In *International Conference on Intelligent Computation Technology and Automation*, volume 1, pages 61 –64, 2010.