École doctorale IAEM Lorraine

# Decentralized control design for synchronization of multi-agent systems with guaranteed individual costs

## THÈSE

présentée et soutenue publiquement le 3 décembre 2021
pour l'obtention du

**Doctorat de l'Université de Lorraine**

**Spécialité Automatique, Traitement du signal et des Images, Génie informatique**

par


Jomphop Veetaseveera


**Composition du jury**

| | | |
|---|---|---|
| *Présidente :* | I. QUEINNEC | Directeur de Recherche CNRS - LAAS |
| *Rapporteurs :* | L. BUSONIU | Professeur des Universités - Université Technique de Cluj-Napoca<br>Department of Automation |
| | Y. HAYEL | Maître de Conférences (HDR)- Avignon Université<br>CERI |
| *Encadrants :* | I.-C. MORĂRESCU | Professeur des Universités - Université de Lorraine<br>CRAN |
| | V.S. VARMA | Chargé de Recherche CNRS - CRAN |


**Centre de Recherche en Automatique de Nancy CNRS - UMR 7039**
**Université de Lorraine**
**2 avenue de la Forêt de Haye, 54516 Vandoeuvre-lès-Nancy**
**Tél. + 33 (0)3.83.59.59.59**

# Remerciements

Avant d'entamer une nouvelle aventure, je tiens à graver ici mes dernières pensées, mais surtout à remercier les personnes qui ont contribué de près ou de loin à ma réussite. Sans l'énergie et le soutien de ces personnes, ce chapitre de ma vie n'aurait jamais vu le jour. Rien ne me prédisposait à emprunter ce chemin, néanmoins, je ressentais ce profond besoin de sortir des sentiers battus. Tout me restait à faire et à découvrir.

La genèse de tout ça ? Selon moi, elle se situerait pendant le stage de fin d'études. Grâce à Romain, j'ai eu la chance de découvrir la recherche en milieu académique. Cette période a été décisive, car je me retrouve aujourd'hui avec un doctorat en poche et une aventure faite d'innombrables rencontres ! Il est important de souligner que j'essaye encore aujourd'hui d'appliquer tes précieux conseils.

Cette thèse a été réalisée au sein du CRAN, UMR 7039, à l'Université de Lorraine entre avril 2018 et décembre 2021. Le projet s'est déroulé sous la direction de M. Irinel-Constantin Morărescu et la co-direction de M. Vineeth Satheeskumar-Varma.

En premier lieu, je remercie Constantin et Vineeth de m'avoir fait confiance pour mener à bien ce projet de recherche. Constantin, tu t'es montré très compréhensif et patient, mais également humain. Quant à Vineeth, ton enthousiasme au quotidien m'a beaucoup aidé. Ton bureau jouxtant le mien, tu as été très disponible pour répondre à mes nombreuses questions d'ordre scientifique tout comme pour discuter sur des sujets plus variés.

Dans l'ordre naturel des choses, une thèse est généralement couronnée d'une soutenance. Ainsi, je remercie chaleureusement les membres du jury d'avoir accepté de rapporter et d'examiner mon travail de thèse. Je salue leurs expertises et l'intérêt qu'ils ont porté à mes travaux, merci à Isabelle Queinnec pour l'honneur qu'elle m'a faite en présidant mon jury, et merci à Lucian Busoniu et Yezekael Hayel de m'avoir remis en question. Grâce à vous, je continue d'approfondir mes connaissances.

Mes amis collègues doctorants, Prisca, Tommaso, Mathieu, Daniele, Sifeddine, Simone, Danh, Maxime, Elena, Bouchra, Viviana, Gustave et Olivier, je n'oublie pas de vous remercier. Grâce à vous, ces trois longues années auront été plus beaucoup agréables et plus supportables. En thèse, *"on est payé peu mais on rigole bien"*, disent-ils.

Enfin, merci à mes autres collègues du CRAN et notamment Christine, Clarice, Maya qui sont toujours souriantes et prêtes à m'aider pour le moindre problème.

Pierre (excellent maître de stage ou pas), il ne faut pas m'en vouloir mais je ne comprends pas toujours tes blagues. Ce que je retiens ? Nos longs débats, un trip à vélo au top et qu'à mon humble avis, tu n'es pas fait pour *AoE2*.

Harry *aka* le couzzin, *Argent, argent ? Yé pas argent, yé trop cher !* On discutait beaucoup de tout, mais surtout de rien et ça me faisait un bien fou de me vider l'esprit. Encore merci pour ces franches rigolades, ta bonne humeur va me manquer.

Thomas, Romain et Robin, la fin du lycée me semblait hier. Un plaisir de vous voir connectés si tard dans la nuit, combien de parties avons-nous enchaîné sur *Starcraft 2* ? Facilement plus de 2000 et toujours aucun *Master* parmi nous ?

The Doctoriales took place at the end of April 2021, and what a happy coincidence it was that we met ! Zarina, you are a ball of energy and always in a good mood. Now, I know when to cut my hair and plant the flowers.

Bikash, more than a teamwork, an unexpected but fruitful duo. One of my goals was to publish a TAC/Automatica paper with a friend and it is done now. I learned a lot from this exchange and our endless nonsensical discussions.

*Dik era ouy*, Gakku *aka* my sister from another mother, with whom am I going to have my coffee breaks ? Who will I eat my croissants with ? But above all, how will I pinch your fluffy cheeks ? One day, I will go to Kazakhstan and visit you ! Don't forget to invite me to your wedding. Gaukhar, Bikash, Zarina, you have breathed new life into this end of thesis and have been a great support to me, thank you.

Mes parents, je vous remercie pour m'avoir laissé tracer ma route, et en particulier ma mère pour m'avoir toujours encouragé à suivre ma voie. Les valeurs qui m'ont été transmises sont bien plus inestimables que tous les biens matériels que je n'ai jamais eus. Jonathan et Clarice, souvenez-vous que je suis très fier de vous.

Les DeVer, quelle famille ! Alice, Amandine, Jacques et Vincent, des soirées jeux de société aux repas express vitesse grand V, merci pour tous ces agréables moments passés ensemble, et encore plein d'autres à venir. Marguerite, Louise, Gustave et Simon *aka* Bébé2, c'est un bonheur de vous voir grandir. Mes tours de passe-passe ne fonctionnent pas aussi bien qu'avant mais je garde espoir.

Yaya, Sophie et Gilles, je vous côtoie depuis très longtemps, et quel honneur mais par dessus-tout quelle chance de vous connaître. Accueilli comme un fils sous votre toit, de manière quasi-inconditionnelle, vous avez toujours fait en sorte que je me sente bien dans votre famille (bientôt la mienne également), et cela n'a pas de prix à mes yeux. Merci pour tout.

À ma raison de me lever chaque jour,
À la prunelle de mes yeux, ce sourire délicieux,
Toi, ce ventre gargouillant et ce tempérament changeant,
Cet improbable et inexplicable *bordel* naturel qui me rend fou.
Grâce à ton soutien et à ta tendresse, rien ne me parait impossible,
Ni insurmontable, pas même les fracas du temps ni la distance, rien.
Toujours à me pousser vers le haut, tu étais là derrière chacun de mes pas.

Chacune de mes victoires, je te les dois.

# Contents

# Notation

| | | |
|---|---|---|
| $\mathbb{N}$ | - | *set of non-negative integers.* |
| $\mathbb{R}$ | - | *set of real numbers.* |
| $\mathbb{R}_+$ | - | *set of non-negative real numbers.* |
| $\mathbb{R}^{n\times m}$ | - | *set of $n \times m$ real matrices.* |
| $\mathbb{R}^n$ | - | *set of $n$ dimension real vectors.* |
| $\mathbf{I}_n$ ($\mathbf{0}_{n,m}$ resp.) | - | *$n \times n$ identity matrix ($n \times m$ zero matrix resp.).* |
| $\mathbb{1}_n$ | - | *column vector in $\mathbb{R}^n$ whose elements are all equal to one.* |
| $diag(P_1, P_2, ..., P_n)$ | - | *block diagonal matrix having the matrix $P_1, \dots, P_n$ on its main diagonal.* |
| $P^{-1}$ | - | *inverse of matrix $P$.* |
| $P > \mathbf{0}$ ($P < \mathbf{0}$ resp.) | - | *positive (negative resp.) definite matrix $P$.* |
| $P \geq \mathbf{0}$ ($P \leq \mathbf{0}$ resp.) | - | *positive semi-definite matrix (negative semi-definite resp.) $P$.* |
| $P < Q$ | - | *negative definite matrix $P - Q$.* |
| $P^\top$ | - | *transpose of matrix $P$.* |
| $x^\top$ | - | *transpose of vector $x$.* |
| $\\|P\\|$ | - | *Euclidean norm of matrix $P \in \mathbb{R}^{n\times n}$.* |
| $\\|x\\|$ | - | *Euclidean norm of vector $x \in \mathbb{R}^n$.* |
| $(x, y)$ | - | *stands for $[x^\top y^\top]^\top$ where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$.* |
| $\lambda_{max}(P)$ ($\lambda_{min}(P)$ resp.) | - | *maximum (minimum resp.) eigenvalue of the square matrix $P$.* |
| $\nu(P)$ | - | *measure of the square matrix $P$ as $\nu(P) = \frac{1}{2}\lambda_{max}(P + P^\top)$.* |
| $P \otimes Q$ | - | *Kronecker product of matrices $P$ and $Q$.* |
| $\mathscr{N}(P)$ | - | *kernel of matrice $P$.* |
| $\mathscr{R}(P)$ | - | *image of matrice $P$.* |
| $P^\perp$ | - | *denotes a matrix such that $\mathscr{N}(P^\perp) = \mathscr{R}(P)$ and $P^\perp P^{\perp\top} > 0$.* |

# Glossary

| | | |
|---|---|---|
| ARE | - | Algebraic Riccati Equation |
| LMI | - | Linear Matrix Inequality |
| LQR | - | Linear Quadratic Regulator |
| NE | - | Nash Equilibrium |
| SE | - | Satisfaction Equilibrium |
| SPT | - | Singular Perturbation Theory |
| TSS | - | Time-Scale Separation |

# Introduction

Apart from Chuck Noland [1] and Wilson, few people can claim to be able to escape the networks. Ubiquitous in all strata of our society, networks in the broadest sense have gradually infiltrated our daily lives, and now occupy every landscape. Either it is an electrical grid, a social network, a flock of birds in formation, or the spread of a pandemic, all these interconnected systems show a great degree of interdependence. Given this enthusiasm in large-scale physical and societal phenomena, *interconnected dynamical systems* have captured the attention of the scientific community in recent decades, [Cao et al., 1997; Wooldridge, 2009; Mesbahi and Egerstedt, 2010]. Both theoretically and also practically, *multi-agent systems* and *networked systems* are proving to be the most efficient and adequate way to model the dynamics of large-scale complex systems. In order to prevent and solve tomorrow's problems, the analysis and understanding of such systems seem inevitable, [Baillieul and Antsaklis, 2007; Lamnabhi-Lagarrigue et al., 2017].

A multi-agent system is a set of entities or agents often described by two fundamental aspects : the dynamics of the agents, and the interactions between them. Commonly, such an agent can be represented by a mobile robot, person, vehicle, bird, etc. Each agent collaborates with its fellow neighbors to carry out its assigned task. This coordination leads the system as a whole to a common objective, called *consensus* or *synchronization*, [DeGroot, 1974; Vicsek et al., 1995].

The consensus problem appears in various disciplines such as biology [Pavlopoulos et al., 2011], sociology [Hegselmann et al., 2002; Lorenz, 2007; Blondel et al., 2009], social networks [Blondel et al., 2009] or engineering [Martinez et al., 2007; Anderson et al., 2008; Bullo et al., 2009]. As for the multi-agent systems synchronization, either natural or artificial, we can cite the behavior of birds flock, school of fish or coordination of unmanned aerial vehicles, [Cortes et al., 2004; Blondel et al., 2005; Olfati-Saber, 2006; Tanner et al.,

1. In the movie *(Cast Away)*, FedEx employee Chuck Noland (Tom Hanks) is called to an emergency flight on Christmas Eve. Unfortunately, his cargo plane crashes, leaving him alone on a Pacific Ocean isolated island. For four years, he will maintain an unbreakable friendship with Wilson, a volleyball from the cargo.

2007; Leonard et al., 2007]. The applications are many and various : platooning of vehicles, exploring, patrolling, alignment of satellites, distributed sensor network, drones formation, tracking of a leader, etc. In general, control algorithms are applied to these systems to obtain the desired behavior. The purpose is to oil the wheels of the coordination between the agents.

In the control community, a very large number of works deal with problems related to the design of control laws achieving the coordination of multi-agent systems. Different approaches are suggested but two in particular hold our attention. Either, the studies focus on the dynamics of the agents, or on the type of communication and topology of the network. The first characteristic intrinsically related to the agents depends only on the nature of the agent itself. Several contributions consider multi-agent systems with linear dynamics [Jadbabaie et al., 2003; Xiao and Boyd, 2004; Moreau, 2005; Ren and Beard, 2005], non-linear dynamics [Das and Lewis, 2010; Li et al., 2012; Isidori et al., 2014; Su et al., 2015], non-holonomic robots [Strogatz, 2004; Lin et al., 2005], or coupled oscillators [Dörfler and Bullo, 2014]. As for the second characteristic, it depends firmly on the type of interactions between the agents. Widely addressed in the literature, methods oriented on network properties propose to model interactions by graphs with fixed or time-varying topology [Hong et al., 2006; Tanner et al., 2007; Ren, 2007; Scardovi and Sepulchre, 2009], with time delay [Seuret et al., 2008; Xiao and Wang, 2008] or limited communication ability [Dimarogonas et al., 2011].

In consensus problems, as the implemented algorithms are sensitive to the different types of interconnections, the analysis focuses on the network structure. For the synchronization, the primary objective is to coordinate the whole system. Thus, the approaches are focusing on the control design while taking into account the dynamics of the agents.

However, what is exactly the consensus or synchronization of multi-agent systems ?

## Consensus

"*Consensus means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents*", [Olfati-Saber et al., 2007]. Originated in management science and statistics, DeGroot provided a formal study of consensus problem for groups of individuals and described the concept of statistical consensus, [DeGroot, 1974]. The work studies the evolution of the opinion in a team, where each individual's opinion is represented by a probability distribution. In the context of control systems, a consensus occurs when the agent's states converge to one common point through interactions. The state or output of each agent represents the information that needs to be coordinated between them, and mainly depends on the agent's features. For instance, it can be the agent's position, velocity, voltage, opinion, shared resource, etc. In the problem of rendezvous, the state corresponds to the physical position of the agents.

Consider a network of $N$ interconnected agents. Each agent $i$ is associated with a state value $x_i \in \mathbb{R}^{n_x}$ where $i \in \mathcal{V} = \{1, ..., N\}$, and has the following state dynamics

$$\dot{x}_i(t) = u_i(t), \quad \forall i \in \mathcal{V}, \tag{1}$$

where $u_i \in \mathbb{R}^{n_u}$ is the control or consensus protocol, it defines a rule that governs the flow of information among agents in close proximity. One possible protocol is the famous continuous-time consensus algorithm from [Olfati-Saber and Murray, 2003, 2004]. In these seminal works, the authors posed and solved the consensus problem under various assumptions on the network topology, communication time-delays, and information flow. Their contributions are mainly inspired by [Fax, 2001; Fax and Murray, 2004]. The consensus protocol is described as follows

$$u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_i(t) - x_j(t)), \quad \forall i \in \mathcal{V}, \tag{2}$$

where $\mathcal{N}_i$ is the set of agents *neighboring* the agent $i$, i.e. whose information is available to agent $i$, and the interconnection weight $a_{ij}$ is the *strength* of the link between the $i$-th and $j$-th agent. We understand that the state of each agent is driven toward the states of its neighbors at each time.

Then, applying the protocol (2) for any initial condition $x_0$, consensus is said to be achieved if all the agents reach a final value such that

$$\lim_{t \to +\infty} x_i(t) = x^*, \quad \forall i \in \mathcal{V}, \tag{3}$$

where $x^*$ is the consensus value.

Finally, by collecting the state of all agents in a single vector $x = (x_1, ..., x_N) \in \mathbb{R}^{N.n_x}$, we denote the *agreement* or *consensus* dynamics by

$$\dot{x}(t) = -(\mathcal{L} \otimes I_{n_x})x(t), \tag{4}$$

where the Laplacian matrix $\mathcal{L}$ describes the topology of the network by giving the connections between the agents, and the symbol $\otimes$ stands for the Kronecker product. The matrix form facilitates the analysis of some network properties such as the rate of convergence or the connectivity.

## Synchronization

The process of synchronization, closely related to consensus, is skillfully explained in [Su et al., 2015] as "*In a multi-agent system, consensus means agents reaching an agreement regarding a certain quantity of interest that depends on the state of all agents, while synchronization defines the correlated-in-time behavior among different agents*".

In the mid 1990s, Vicsek introduced the heading synchronization and the flocking problem by investigating on the emergence of self-ordered motion in particles systems, [Vicsek et al., 1995]. The authors propose a discrete-time model of particles all moving in the plane with the same speed but with different headings. The headings of each agent evolves according to a local rule based on the average of its heading and those of its neighbors. Afterwards, [Jadbabaie et al., 2003] completed the previous work by providing a theoretical framework. Later, [Moreau, 2005] and [Ren and Beard, 2005] extended the theory by dealing with directed information flow. First, necessary and/or sufficient conditions on the communication topology were provided in [Moreau, 2005] to guarantee that the network reach the consensus. Then, assuming that certain assumptions are hold, the consensus was proved to be achieved asymptotically under dynamically changing interaction topologies, [Ren and Beard, 2005].

To better understand the difference between consensus and synchronization, we illustrate with a basic example. Consider a network of $N$ interconnected agents having linear dynamics, which are described by

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad \forall i \in \mathcal{V} = \{1, ..., N\}, \tag{5}$$

where $x_i \in \mathbb{R}^{n_x}$ represents the state, $u_i \in \mathbb{R}^{n_u}$ the control, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ correspond to the internal dynamics and control matrix, respectively.

Then, assuming that the controls $u_i$ are designed to accomplish the network objective, we say that the synchronization is achieved if the norm difference between all agents converge to zero such that

$$\lim_{t \to +\infty} \|x_i(t) - x_j(t)\| = 0, \quad \forall i, j \in \mathcal{V}, \tag{6}$$

for any initial condition $x_0$.

One of the major difference between the two processes is that the agents posses an internal dynamics. The individual agent dynamics can be linear, nonlinear, and can only depend on the state and control of the agent itself but also on those of the other agents. We say that the dynamics of the network are *homogeneous* if the dynamics of all the agents are identical, otherwise it is *heterogeneous*. Furthermore, the form of the control rely mainly on the global objective of the system. For instance, the goal can be to maintain a formation, to follow a leader, to cover an area, to keep a minimum security distance with other agents, etc.

By taking these dynamics into consideration, the dimensionality of the problem changes and it radically affects the control design of the network. A network with homogeneous dynamics allows the agents to synchronize on a common feature. On the contrary, the synchronization is achieved on several distinct features if the network has heterogeneous dynamics. A high degree of heterogeneity thus makes synchronization between agents more complex.

For the attention of the readers, the following surveys give a general overview of the progress and advances made in the field of consensus and synchronization of multi-agent systems until 2017, [Ren et al., 2005; Chen and Wang, 2005; Olfati-Saber et al., 2007; Murray, 2007; Cao et al., 2012; Antonelli, 2013; Oh et al., 2015; Qin et al., 2016; Ge et al., 2017] Moreover, short introductions to the field can be found in [Ren et al., 2007; Olfati-Saber et al., 2007; Martinez et al., 2007; Dorri et al., 2018; Chen et al., 2019].

## Centralized versus distributed

The multiple controls established in the literature are classified into two categories : *centralized* versus *distributed* control. The control architecture is said to be *centralized* when at least one agent communicates with all the other agents or requires global information, i.e. from the entire system. Conversely, the control is *distributed* if the task is performed via local interactions, i.e. each agent only exchanges information with its neighbors.

In the centralized form, a single monolithic, expensive and complex computer have to manage the coordination mission for the whole group of agents. This configuration can present several drawbacks : 1) a high computational load proportional to the size of the network, i.e. the number of agents, 2) a limited application scope for large-scale systems due to scalability issue, 3) each agent's controller is susceptible to failure of the central processing unit, 4) require to communicate with the entire network.

From another perspective, the distributed control design offer major advantages such as robustness, speed of execution, low operational costs, higher fault tolerance and strong adaptivity. The distributed approach also appears to be more promising in the presence of uncertainty such as communication noises or packet loss. In the long run, the purpose would be to replace the single machine with a fleet of less powerful but more affordable autonomous systems, that can achieve the same or better through their coordination.

These books among others presented distributed algorithms and applications for multi-agent systems [Lynch, 1996; Ren and Beard, 2008; Shamma, 2008; Bullo et al., 2009; Qu, 2009; Ren and Cao, 2011; Parker et al., 2016].

## Objectives

In the literature, several contributions investigate on a global cost minimization during the control design of the network, but only a few consider individuals costs. We denote a global cost by a function that takes into account the effort of the entire network to achieve the global objective. On the contrary, the individual cost is the effort related to one agent or a small densely connected group of agents in the network.

In applications with physical networked systems, the individual costs may be of practical interest when the agents have limited communication (short wireless signal ranges, narrow bandwidths) and operating capacity (fuel, battery, computation resource), [Anastasi et al., 2009; Goldenberg et al., 2004; Dimarogonas et al., 2011]. For instance, consider the scenario of automatic cruise control on highways, where each vehicle desires to follow the vehicle in front of it (global objective of synchronization). At the same time as accomplishing their tasks, we also want to ensure that their fuel consumption is not too excessive (individual costs minimization). The reduction or limitation of the fuel consumption can be seen as some performance constraints from technical specifications. In such applications, considering a global cost might not be fair to each vehicle. Moreover, the choice between a global cost and individual costs may significantly impact the control strategy.

The advantage of constraining or minimizing such a cost is clear, but theoretical results in this direction are crucially missing in the literature on multi-agent systems. Only a few results address the control design problem for the synchronization of agents while minimizing a cost. These reasons then lead us to focus in this direction. **The main objectives of this thesis are the control design and the analysis of synchronization algorithms for multi-agent systems taking into account communication constraints, while ensuring that the state and control costs are below a given bound.** The analysis is carried out with particular attention on multi-agent systems with homogeneous linear dynamics and clustered networks, with fixed topology in both cases. In the first case, an individual cost per agent is considered during the control design whereas, a cluster cost for the second case. Distributed control methods are provided, but the control gains may be designed in a centralized manner depending on the cases.

## Structure of the thesis

This manuscript is structured in three main chapters. In the following, we give a brief summary of each of them. A general conclusion and perspectives are presented at the end of the manuscript.

### Chapter 1 : Preliminaries

Basic concepts and a review of theoretical tools used throughout this thesis are presented. In the first section, the State-of-the-art develops the contributions about distributed control design from the literature. The consideration of a global or individual cost during the synthesis of such controllers is also discussed. Then, the second section recalls some notions of graph theory that are the basis for the analysis of interactions in multi-agent systems. Finally, concepts from game theory and singularly perturbed systems are presented.

### Chapter 2 : Decentralized control for guaranteed individuals costs

This chapter deals with the design of decentralized control aiming at synchronizing a network while considering an individual cost per agent. In order to facilitate the analysis, the synchronization problem is first reformulated into a stabilization problem. Then, we use the game theoretic notion of satisfaction equilibrium to guarantee, if feasible, a certain level of performance. Conditions in the form of linear matrix inequalities (LMI) are provided to check if a given gain profile is a satisfaction equilibrium, i.e. if all individual costs are bounded by a given threshold. Moreover, based on the output feedback control, a second result allows us to synthesize the gain of an agent assuming the gains of the other agents are known. Finally, an algorithm synthesizing the gain profiles corresponding to the satisfaction equilibria is also presented.

### Chapter 3 : Distributed composite control for clustered networks

We present a distributed control design for clustered networks, in which connections within the cluster are dense and between clusters are sparse. Our goal is to provide a computationally efficient method to design control strategies that guarantee a certain bound on the cost for each cluster. Based on network structure and singular perturbation theory, we apply time-scale separation methods to decouple the original system into fast (intra-cluster) and slow (inter-cluster) dynamics. Subsequently, we synthesize a distributed controller composed of two terms : one responsible for intra-cluster synchronization, and the other performing inter-cluster synchronization. The internal control does not require a high computational effort since it is obtained by an analytical expression. As for the external control, it is designed using a satisfaction equilibrium approach. In summary, the internal (fast) and external (slow) controls are designed independently of each other and ensure a satisfactory cost for each cluster. Furthermore, we show that the internal control only affects the cluster cost for a short period of time. Finally, numerical simulations emphasize the trade-off between the control performance and computational feasibility to obtain the required controller by comparing the strategy proposed in Chapter 2, Chapter 3 and in [Rejeb et al., 2018]. Despite being less effective the controller in Chapter 2, we must keep in mind that the composite control in Chapter 3 suits better for large-scale networks and presents an essential benefit in computation loads and time than. On the other hand, while the solution in [Rejeb et al., 2018] is computationally very fast, we observe that our strategy significantly outperforms the one in [Rejeb et al., 2018].

# Publications

## International journals

- J. Veetaseveera, V. S. Varma, I. C. Morărescu and J. Daafouz, "Decentralized control for guaranteed individual costs in a linear multi-agent system : A satisfaction equilibrium approach", *IEEE Control Systems Letters, Vol. 3(4), pages 918-923, 2019*, Presented at CDC 2019, Nice, France.
- J. Veetaseveera, B. Adhikari, V. S. Varma and I. C. Morărescu, "Distributed Composite Control of Clustered Networks", *2021 (under preparation)*.

## International conferences

- J. Veetaseveera, V. S. Varma and I. C. Morărescu, "A dynamic game formulation for control of opinion dynamics over social networks" *IEEE International conference on Network Games Control and Optimization (NetGCOOP), 2020*.
- V. S. Varma, J. Veetaseveera, R. Postoyan and I. C. Morărescu, "Distributed gradient methods to reach a Nash equilibrium in potential games", *IEEE Conference on Decision and Control (CDC), 2021*.

Chapter 1

# Preliminaries



This introductory chapter aims to provide the necessary prerequisites for a clear understanding of the problems presented in this thesis, as well as a short review of the literature. The Section 1.1 is dedicated to the agreement and synchronization control protocols for multi-agents systems in continuous time. Especially, we state some innovative works that tend to optimize an energy function related to the coordination process. In Section 1.2, we provide mathematical tools and concepts useful for the further developments presented in this manuscript. First, we give few definitions on graph theory and recall some notions such as undirected and directed graphs. Thereafter, we briefly introduce the notions of *Nash equillibrium* and *satisfaction equillibrium* from game theory. Finally, the singular perturbation theory is explained which focus on the decoupling of two-time scale systems into slow and fast subsystems. The fundamental definitions and results are given in the case of static networks.

## 1.1 State-of-the-art

Over the last two decades, technological advances have sparked a deep interest in the scientific communities for the research on consensus and synchronization of multi-agent systems. The distributed control design is one of the most widely addressed problem, extensively studied and analyzed across different applications and settings.

Interests in distributed controls date back to 1973. [Wang and Davison, 1973] stabilized homogeneous linear multi-agent systems by applying several local controllers. The authors provided a necessary and sufficient condition for the existence of local control law with dynamic compensation. The feedback control depend partially on outputs. Later, motivated by parallel and distributed computations over a network of processors, [Tsitsiklis, 1984] investigated on the distributed decision-making for asynchronous agreement problems. Then, notable contributions such as [Tsitsiklis et al., 1986; Bertsekas and Tsitsiklis, 1989] appeared. These works will serve as a framework for the analysis of distributed computational models. After a leap in time, [Fax and Murray, 2004] provided stability analysis tools for homogeneous linear multi-agent systems. Each subsystem applies a local control using the average information obtained from neighbors.

Thereafter, the theory of distributed control and optimization for networked systems have developed rapidly and yielded fruitful results. In this section, we will focus on the synthesis of distributed protocols and algorithms considering, either a *global* or *individual* cost during the coordination process.

In the following, we briefly present some works related to the synchronization of multi-agent systems investigating a cost function during the control design. In [Borrelli and Keviczky, 2008], the authors introduce a distributed optimal control problem with a global cost in the multi-agent framework. The problem is well posed when the dynamics and the initial state for all agents are perfectly known. However, due to the information structure imposed by the graph the solution stated therein is a *Non-deterministic Polynomial-time hard (NP-hard) problem*. Thus, they provide a distributed sub-optimal control design.

The collective behavior problem for swarms of identical mobile agents is addressed in [Yang et al., 2008]. An approach based on decentralized joint estimation and control problem is applied to coordinate the group. The agents communicate with their neighbors and estimates the global objective performance via distributed estimators, the estimations are required to implement each local controller.

Dedicated to consensus on networks with time-varying topology, [Nedic et al., 2010] presents a distributed projected sub-gradient algorithm. A global cost related to the consensus is considered, and it is expressed as the sum of individual agent costs. The algorithm requires each agent to perform a local computation, i.e. each agent minimizes its own cost and operates projections on its constraint set.

Later in [Jaleel et al., 2014; Ali and Wardi, 2015], the problem of synchronization is addressed as an optimal control problem. The authors derived a control law optimizing a mobility and communication energy in a centralized manner. In the same vein, [Hassan and Shamma, 2016] proposed a decentralized energy-aware control design. Sub-optimal policies inspired from approximate dynamic programming are provided, but no analytical results allowing the computation of the gain control were given. In these three previous works, a global cost was considered while designing the control.

More recently, in [Rejeb et al., 2018], the authors present the design of a decentralized control strategy that allows singularly perturbed multi-agent systems to achieve synchronization with global performance guarantees. Additionally, they assume that all agents utilize the same gain, which can be restrictive and even undesirable in some cases. Close to the previous work, [Jiao et al., 2019b,a] provided a distributed sub-optimal control for undirected multi-agent systems, where agents have homogeneous linear dynamics. The objective is to achieve a consensus over the network while the associated global quadratic cost is smaller than an a priori given upper bound. Two design methods are provided, it requires the solution of an Riccati inequality and the knowledge of smallest nonzero and the largest eigenvalue of the graph Laplacian.

All of these works mentioned above, concerning consensus or synchronization, have addressed distributed control with a global cost, but not with individual costs. Generally, the individual cost is a local cost represented by the sum of a state cost and a control cost related to one agent. Consequently, we provide some references dealing with coordination of multi-agents systems while taking into account individuals costs. In [Bauso et al., 2006], the authors consider consensus protocols for networks with fixed topology and undirected information flow. The agents have only access to their neighbors' state to reach a group consensus value, which depends on all the agents' initial state. They present a non-linear protocol designed by imposing individual objective to each agent, a Lyapunov approach is used to prove the asymptotic convergence to the consensus.

Employing a game theory approach, [Semsar-Kazerooni and Khorasani, 2009] presented a semi-decentralized optimal control strategy based on minimization of individual costs. The goal is to design a control strategy accomplishing a consensus over a common value, while considering a team cost expressed as a combination of individual costs. The authors used the concept of Pareto-efficient solution and Nash-bargaining solution to guarantee the minimum individual cost. Linear matrix inequality are provided to solve the minimization problem, and to ensure that the controls only use local information.

In [Li et al., 2021], a consensus over networks with fixed topology is performed with a distributed optimal control. It is proved that solving the optimal distributed control problem for all the agents, while considering local quadratic costs, is equivalent to solve a globally optimal problem. Moreover, the Laplacian matrix associated with the optimal solution turned out to be a complete graph. Thus, the globally optimal distributed control problem is solved for protocols with specified distributed structure.

Although these works are tackling the distributed control design problem while optimization individual costs, it does not address the synchronization of multi-agent systems, in which agents posses internal dynamics. Recently, *Reinforcement Learning (RL)* techniques were applied to solve the distributed optimal control problem for multi-agent systems. Generally, it requires the solution of the *Hamilton-Jacobi-Bellman (HJB)* equation, which is a nonlinear partial differential equation that is almost impossible to be solved analytically. Based on game theoretic concepts, control theory and estimation methods, they introduce some interesting contributions suggesting off-line or online algorithms.

In [Vamvoudakis et al., 2012], a multi-agent formulation for the online solution of team games is considered. The authors developed the notion of *graphical games* for dynamical systems, and present the *Interactive Nash Equilibrium* concept associated with this kind of games. They provide a cooperative policy iteration algorithm, in which the dynamics and cost functions of each agent depend only on local neighbor information. In case the neighbors of each agent do not update their policies, the algorithm converges to the *best response*. Otherwise, the algorithm converges to the cooperative *Nash equilibrium* when all agents update their policies simultaneously.

An RL technique is introduced in [Luo et al., 2014] to address the model-free nonlinear optimal control problem. The authors overcome the complexity related to the solution of the HJB equation by using real system data rather than a system model. They propose a data-based approximate policy iteration method based on off-line RL method. A model-free policy iteration algorithm is derived and its convergence is proved.

[Modares et al., 2016] consider an output synchronization of heterogeneous linear multi-agent systems. The authors formulated the output synchronization problem as an optimal control problem. Then, they apply a model-free off-policy reinforcement learning algorithm to solve the optimal output synchronization problem online in real time.

Based on graphical games, [Li et al., 2017] develops an off-policy RL algorithm to solve optimal synchronization of multi-agent systems. A prescribed control policy, called behavior policy, is applied to each agent to generate and collect data for learning. An off-policy HJB equation is derived for each agent to learn the value function for the policy under evaluation, called target policy, and find an improved policy, simultaneously. Finally, an off-policy RL algorithm is presented that is implemented in real time and gives the approximate optimal control policy for each agent using only measured data. It is shown that the optimal distributed policies found by the proposed algorithm satisfy the global Nash equilibrium and synchronize all agents to the leader.

In the decentralized control design paradigm, each system is able to design and implement its own control law without the help of a central entity. Nevertheless, as soon as each agent has its own individual cost, standard optimization or optimal control approaches cannot be directly applied. As reported in the literature, the only access to the local information can make the optimal control techniques unusable or make the problem NP-hard. Furthermore, the distributed controllers designed via a RL technique avoid the difficulty related to the HJB equation by approximating the solution. Usually, the approximated solution is estimated thanks to given data of the real system, or collect directly the data by stimulating the system with different input signals. However, these data are not always available or the access to the systems is impossible.

In this context, we are inspired by notions in game theory, specifically that of *satisfaction equilibrium* and *satisfaction games* introduced in [Ross and Chaib-draa, 2006]. A set of actions are said to be in satisfaction equilibrium when the individual cost for each agent is upper-bounded by a given threshold. This notion was applied to wireless networks in order to guarantee a satisfactory quality of service [Perlaza et al., 2012]. Note that we use game theory as an inspiration for some concepts and formalism, but not for the mathematical tools. Thus, we relax the problem of individual costs minimization to ensure that each individual cost is guaranteed to be lower than a given threshold. Our *first objective* is to provide analytical results allowing to compute a control gain that can be implemented in a distributed manner. The multi-agent system then achieves the synchronization with local or individual performance guarantees. Presented in Chapter 2, the algorithm performs well with a small network but is computationally demanding with a large-scale network, Section 2.3.5.

For this reason, the *second objective* is focused on the synchronization of large-scale networks. Examples of such networks include small-world networks [Watts and Strogatz, 1998], power systems [Chow, 1982], [Romeres et al., 2013], wireless sensor networks [Mytum-Smithson, 2005], social networks [Wasserman and Faust, 1994], etc. The goal is to provide a design of effective controllers with low computational load. One effective method to address the synchronization of the large-scale networks is model reduction. Based on the *Singular Perturbation Theory* (*SPT*), the model reduction is achieved by exploiting the time-scale properties of clustered networks. The purpose is to decrease the size of the system state while approximating its overall dynamic behavior.

Classical results in [Chow and Kokotović, 1985], [Chow, 1982] develop a simplified model using SPT on the networks of linear interconnected systems with diffusive coupling. These results have been extended in [Biyik and Arcak, 2008] to networks of non-linear systems interacting over fixed undirected networks. In [Romeres et al., 2013], the authors relax the requirement of some assumptions in [Chow and Kokotović, 1985] and also extend the results for the weighted graphs. Furthermore in [Martin et al., 2016], a model reduction using averaging theory is obtained for the time-varying directed graphs.

It is noteworthy that most of the existing works deals with the synchronization analysis without internal dynamics for the agents. In other words, they do not consider the control design for the synchronization of clustered networks. A particular setup for the synchronization of clustered networks is considered in [Boker et al., 2015]. However, none of the previously mentioned works consider the synchronization problem under requirements of costs optimization. These requirements are on one hand timely and on the other induce a high computational load preventing the design of (sub-)optimal controllers in a centralized manner.

## 1.2 Theoretical notions of interest

### 1.2.1 Graph Theory

Nowadays, the growing research in intelligent swarms, opinion dynamics, collective motion in biology, game theory or parallelization in optimization theory has received a significant attention in the literature. Among several notions appearing in these studies, the ones of *unity* and *group* especially draw our attention. An appropriate theory characterizing these concepts in a straightforward and clear manner makes use of the *Graph Theory*.

Throughout this thesis, we denote a single dynamical system by an *agent* and an ensemble by a *network* or *graph*. An agent is a dynamical system that interacts with other agents to form a network, and these interconnections are determined according to some specific communication topology that describe the concept of *neighborhood* for each agent. The literature is illustrated by various types of interactions such as information flows, the influence of social networks, chemical reactions between cells, the rules of a game or parallel calculations.

Whatever the nature of interactions, the study of graph theory is still a fundamental component for the analysis of networks and *multi-agent systems* (*MAS*). Graph theory is unavoidable and convenient to model the different communication topologies or concepts such as *neighborhood* and *connectivity*. To this purpose, we will rely on the definitions given in [Godsil and Royle, 2001] and [Mesbahi and Egerstedt, 2010] about graphs. In addition, short tutorials for graph theory can be found in [Olfati-Saber and Murray, 2004; Ren et al., 2007].

#### Definitions and concepts

***Definition* 1.** *A graph $\mathcal{G}$ is described by the couple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of $N$ vertices or nodes $\mathcal{V} = \{1, ..., N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of edges or pairs of distinct vertices $(i, j)$, $i \neq j$ which* *represents the flow of information from $i$ to $j$. We assume that $\mathcal{G}$ has no self-loops, i.e. for all $i \in \mathcal{V}$, $(i,i) \notin \mathcal{E}$.*

Despite its mathematical nature, a graph can be conveniently represented by a graphical scheme. Each element of $\mathcal{V}$ is symbolized by a circle while the edges are pictured by *lines* and *arrows* in the undirected and directed graphs, respectively. In the representation of digraphs, for $(i,j) \in \mathcal{E}$, the tail and the head of an arrow are the vertices $i$ and $j$, respectively.

When graphs are used to model MAS, we assimilate each vertex/node to an agent and an edge between two vertices indicates that two agents are communicating with each other. Moreover, the size of a network or the order of a graph $\mathcal{G}$ is given by $|\mathcal{V}| = N$.

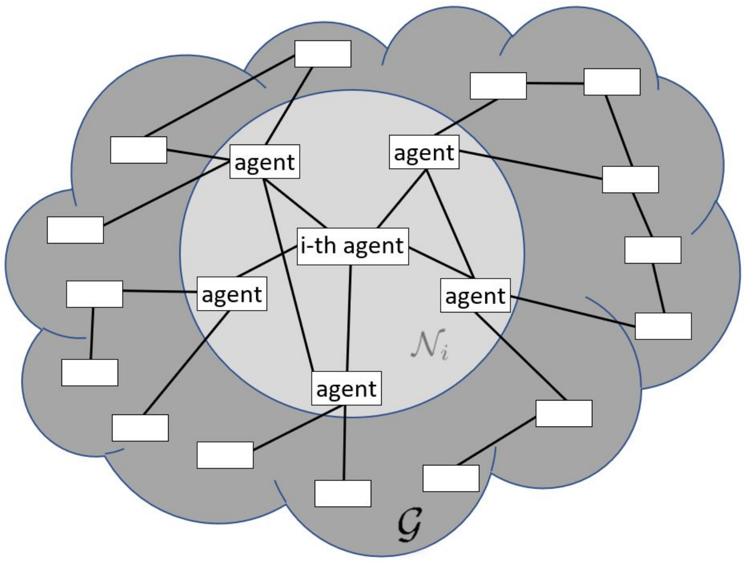


FIGURE 1.1 – [Antonelli, 2013] Network from the agent $i$ point of view. The cloud $\mathcal{G}$ represents the whole network whereas the light grey circle $\mathcal{N}_i$ corresponds to the agent $i$ and its neighbors. Black lines are connections between agents.

The concept of proximity among the agents is described by the *neighborhood*. The set of neighbors of an agent $i$ is defined as

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}, \tag{1.1}$$

it represents the set of all the agents connected to the agent $i$, i.e. that can take information from it. A representation of a network with a neighborhood is illustrated in Fig 1.1.

When the connection between two agents represents an information flow, one may wonder if the flow is bidirectional or unidirectional. Following that, the concepts of directed and undirected graphs naturally address the question.

***Definition* 2.** *A graph $\mathcal{G}$ is* ***undirected*** *if the edges are bidirectional, i.e. for all $i, j \in \mathcal{V}$, $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$ ; it is called* ***directed*** *graph or* ***digraph*** *otherwise.*

***Definition* 3.** *A path of length* ***p*** *in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is given by a sequence of distinct vertices*

$$i_0, i_1, ..., i_p, \tag{1.2}$$

*such that for $k = 0, 1, ..., p-1$, the vertices $i_k$ and $i_{k+1}$ are adjacent. In this case, $i_0$ and $i_p$ are called the end vertices of the path ; the vertices $i_1$,...,$i_p$ are the inner vertices. Moreover, two nodes $i$ and $j$ belonging to $\mathcal{G}$ are* ***connected nodes*** *if there exists at least a path in $\mathcal{G}$ from $i$ to $j$.*

***Definition* 4.** *An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is* ***connected*** *if, for any node $i, j \in \mathcal{V}$, there is a path joining $i$ and $j$, i.e. $i_0 = i$ and $i_p = j$. If this is not the case, the graph is called* ***disconnected***.

***Definition* 5.** *An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is* ***complete*** *if and only if for any node $i, j \in \mathcal{V}$, the couple $(i, j) \in \mathcal{E}$.*

***Definition* 6.** *A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be* ***connected*** *if there exists a node $i \in \mathcal{V}$ such that for any node $j \in \mathcal{V} \backslash \{i\}$, there is a path from $i$ to $j$. The node $i$ is called the* ***root*** *node of the graph.*

***Definition* 7.** *A* ***strongly connected*** *digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is such that for any couple $(i, j) \in \mathcal{E}$, there exits a path from the node $i$ to $j$.*

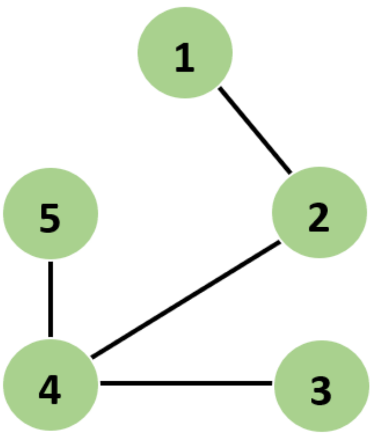


FIGURE 1.2 – Connected graph

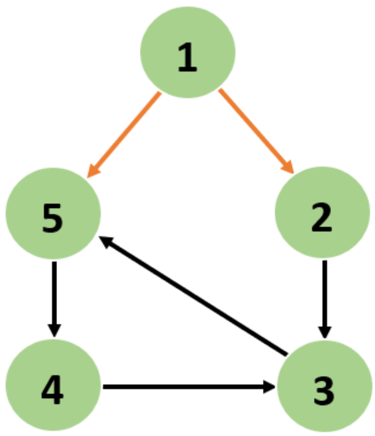


FIGURE 1.3 – Connected digraph

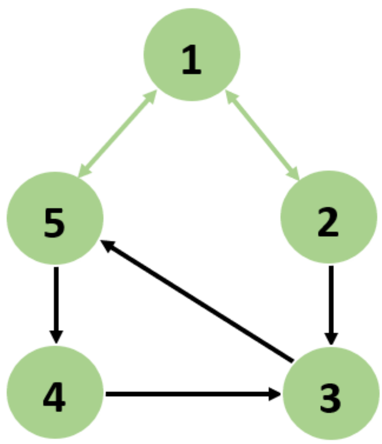


FIGURE 1.4 – Strongly connected digraph

Figure 1.2 is an example of a connected graph whereas Figure 1.3 represents a digraph with one root node. From the root node, any other node can be reached but there is no way to reach the node 1 from other nodes. In red are the directional communications leaving the root node. The Figure 1.4 illustrates the strong connectivity.

***Definition* 8.** *The **adjacency matrix** $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is the symmetric matrix encoding of the adjcency relationships in the graph $\mathcal{G}$, in that*

$$[a_{ij}] = \begin{cases} 1 & if\ (i,j) \in \mathcal{E}, \\ 0 & otherwise. \end{cases} \tag{1.3}$$

***Definition* 9.** *The graph **Laplacian matrix** of the graph $\mathcal{G}$ is defined as $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N\times N}$ such that*

$$[l_{ij}] = \begin{cases} \displaystyle\sum_{j=1,\ i\neq j}^{N} a_{ij}, & for\ i = j\ , \\ -a_{ij}, & if\ i \neq j. \end{cases} \tag{1.4}$$

*For an undirected graph $\mathcal{G}$, the Laplacian matrix $\mathcal{L}$ is symmetric and the sum of each row are null.*

***Remark* 1.** *[Godsil and Royle, 2001] Consider a connected undirected graph $\mathcal{G}$ and let denote the eigenvalues associated with the Laplacian matrix $\mathcal{L}$ by $\lambda_i$, in ascending order,*

$$\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N. \tag{1.5}$$

*Then, the following statements hold :*

1. *$\lambda_1 = 0$ is a simple eigenvalue of $\mathcal{L}$ associated to the eigenvector $\mathbb{1}_N$.*
2. *$\lambda_2 > 0$ is the second smallest eigenvalue and $\mathcal{L}$ semi-positive definite. Moreover, $\lambda_2$ is called the algebraic connectivity of the graph. It expresses the rate of convergence during a consensus process.*
3. *There exists an orthonormal matrix $T \in \mathbb{R}^{N\times N}$, i.e. $T^\top T = TT^\top = \mathbf{I}_N$, such that*

$$T\mathcal{L}T^\top = \Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_N). \tag{1.6}$$

### 1.2.2 Game Theory

In this section, we provide some useful definitions and concepts from game theory. Most of the material is based on [Başar and Olsder, 1998] and [Lasaulce and Tembine, 2011]. This section does not aim to present game theory in details but only gives a very brief overview of the field.

A little history to whet your appetite. As its name suggests, the basic concepts of game theory arose from the study of games such as chess and checkers. However, it rapidly became clear that these techniques can be applied to almost all type of interactions that occur between players. John von Neumann and Oskar Morgenstern, mathematician and geneticist respectively, published in 1944 the book *Theory of Games and Economic Behavior*

[Morgenstern and Von Neumann, 1953]. Based on the work of Emile Borel, the authors lay the foundations of the game theory we know today. This act structured the research of the first generation pioneers, including Robert Aumann, John Harsanyi, John Nash, Reinhard Selten and Lloyd Shapley. Afterwards, each of them have advanced the research in their own way. In the 50s, John Nash extended the work of von Neumann by considering games with more than two players (for *non-zero sum games*), and it is from him that came the so-called *Nash equilibrium*. Later, in 2012 Lloyd Shapley and Alvin Roth were awarded the Nobel Prize in Economics. Their work focuses on how best to match offer and demand in a market, with applications in organ donation and education. It is noteworthy that the Nobel Prize in Economics is mainly awarded for results in pure economics or social sciences.

Game theory has revolutionized microeconomics but also impelled new research directions in other scientific disciplines. For example, John Maynard Smith and George Price published in 1973 the paper named *The Logic of Animal Conflict* [Smith and Price, 1973]. The authors consider a situation of inter-species conflict. A natural selection operates a sort between different behaviors. The selection is stimulated by mechanisms of interactions and asexual reproduction. The interactions are modeled as a *non-cooperative game*. Individuals do not have the power to change their behavior over time, they are programmed to deploy the strategy inherited from their parent. The advantage/utility gained in an interaction measures the ability of that individual to adapt to the environment. So, what does exactly game theory mean ? And what is a game ?

**What is game theory ?**

The game theory provides a mathematical framework and concepts that enables to model and analyze the *interactions* between several *players* who can have conflicting or common *interests*. It develops tools, methods, and language allowing a coherent analysis of the decision-making processes when there are more than one player. Particularly, it studies the strategy design problems that optimizes the welfare of a player during an encounter. A natural question one may ask is *What is the best rational thing a player can do ?* How can a strategy be designed to have certain properties that are desirable or necessary for players when it is applied ?

From this, we can see the game theory as a decision-making problem. Particularly, it investigates these interactions through a *game* with different strategical scenarios and outcomes. By interactions, we mean the impact of the player's action on the others. In other words, the player's decision does not depend only on its own actions but also on those taken by the other players, i.e. they are interdependent. Thus, the overall outcome will critically depend on the choices made by all players. The meaning of a player is very broad : it can be a human being, a machine or a plant, a company, etc. Frequently, a player or a decision-maker is facing a situation and has to choose between a range of rational actions. They must strategically reason in order to achieve their objectives or optimize their outcomes. As for the interest, depending on the game and the rules, each player can cooperate for a common goal or compete for its self-interest. It is represented by a cost/reward/payoff/utility function that has to be optimized. Finally, a game in the sense of game theory, is described by a set of players and their possibilities to play the game according to the rules, i.e. their set of strategies.

In order to present some key concepts, let consider two situations where game theory could be used as a natural tool for predicting or comprehending the outcomes depending on the interactive scenarios. The most well known game is probably the Prisoner's Dilemma. In this social dilemma, the individuals interests are opposed to collective interests. These situation are quite well common in modern society, whether it is the problem of saving energy, paying the television licence fee or, more generally, participating in the financing of public goods.

Example 1 : Suspected of having committed a theft, two suspects are arrested. Unable to convict them because of lack of evidence, the police discuss separately with each suspect and offer them the same deal. If only one of them confesses and denounces his accomplice, he will be released. The betrayer goes free and the silent receives the sentence of eight years. If each betrays the other, each will receive a four-year sentence. If both stay silent, they will receive a short sentence of one year. Each prisoner must choose between to betray the other or to remain silent. So the question this dilemma poses is : How will the prisoners act?

| | | $Player\ 2$ | |
|---|---|---|---|
| | | $Cooperate$ | $Defect$ |
| $Player\ 1$ | $Cooperate$ | $1,1$ | $8,0$ |
| | $Defect$ | $0,8$ | $4,4$ |

TABLE 1.1 – Prisoner's Dilemma outcomes

In this game represented by two players $\{Player\ 1, Player\ 2\}$, each player has the following strategy set $\{Cooperate, Defect\}$. Each action associated with a punishment allows them to choose the best one. For example, if $Player\ 1$ defects and $Player\ 2$ cooperates, they have a punishment of 0 and 8 years respectively. The goal is to minimized the time spending in the jail.

$Player\ 1$ does better by playing $Defect$ no matter what $Player\ 2$ does. If $Player\ 2$ cooperates, $Player\ 1$ gets 0 by defecting and 1 by cooperating. Moreover, if $Player\ 2$ defects, $Player\ 1$ gets 4 by defecting and 8 by cooperating. Likewise, $Player\ 2$ does better by defecting no matter what $Player\ 1$ does.

Example 2 : In the Braess Paradox [Braess, 1968; Braess et al., 2005], the author considers a road network as an example. The drivers wish to travel from point $A$ to $B$ illustrated in Fig. 1.5. We represent the number of drivers on a given link $i$ by $x_i$, where $i = 1, 2, 3, 4$. Depending on the link $i$, the travel time in minute is given by $f(x_i)$ or $g(x_i)$. In the original work, the author chose the total number of cars to be $x = 6$, $f(x_i) = x_i + 50$ and $g(x_i) = 10x_i$. In the Scenario 1, drivers can travel on one of two paths : top path or bottom path. The optimal situation or an equilibrium situation is reached when $x_1 = x_2 = 3$ cars. The travel times related to this equilibrium point are $D_{1,3} = D_{2,4} = 83$ minutes. In Scenario 2, the drives have three possible paths instead of two after the addition of a new road. One justification for constructing such a link is to redirect traffic away from link $1$ if that route becomes too congested. Defining the cost function associated with the new road as $h(x_i) = x_i + 10$, the new equilibrium is obtained for $x_1 = 4$, $x_2 = 2$, and $x_5 = 2$. The travel times become $D_{1,3} = D_{2,4} = D_{1,5,4} = 92$, which are worse than in the Scenario 1. Although the drivers have more choice, the situation is not improved by adding a new route.

Several real-life examples illustrate this paradox. After investments in the road network in 1969, the traffic situation in Stuttgart, Germany, did not improve until a portion of the newly constructed route was blocked to traffic [Knödel, 2013]. In 1990, the closure of 42nd Street in New York City, decreased the traffic congestion in the region [Kolata, 1990]. In Seoul, there were three tunnels and one of them had to be shut down from 2003 to restore a river and a park. People participating in the project noticed that the traffic flow improved as a result of the adjustment [Vidal, 2006]. In 2012, the same phenomenon was observed when a road was closed as a result of an accident rather than as part of an urban project. A bridge in Rouen was destroyed in an accident ; other bridges were utilized more in the next two years, although the total number of automobiles crossing bridges was reduced.

**Where is it applied ?**

Apart from fields such that economics, sociology or biology, game theory has been widely applied in various fields like wireless communications [Lasaulce and Tembine, 2011], network security [Alpcan and Başar, 2010], automatic-control [Başar and Olsder, 1998], distributed optimization [Yang and Johansson, 2010], [Li and Marden, 2013] or learning [Syed and Schapire, 2008], [Lanctot et al., 2017].

Consider the water management problem, in which game theory could propose a strategy favoring a efficient water distribution, rapid access to the resource or control of the world market. Whenever resources are involved in a conflict or cooperation, game theory

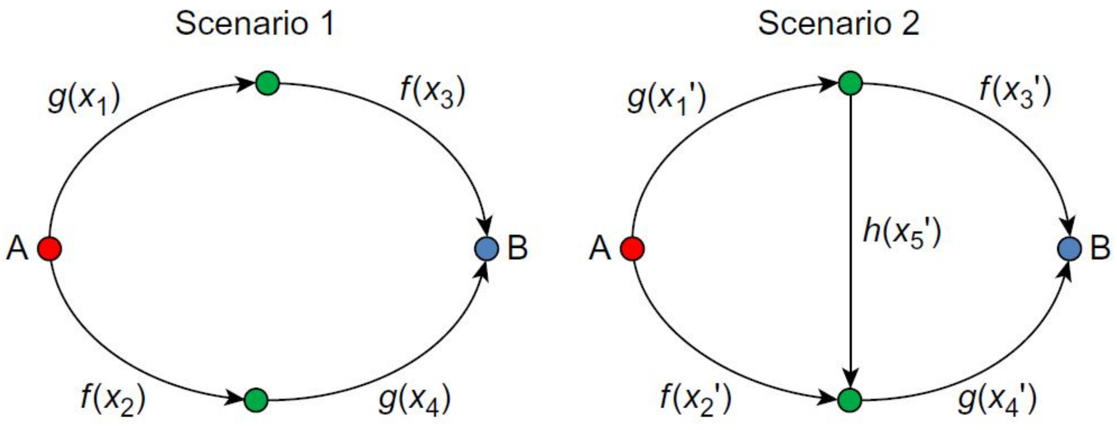


FIGURE 1.5 – [Lasaulce and Tembine, 2011]The routing scenarios considered by Braess

can be used either to explain existing behaviors or to improve further strategies. In [Madani, 2010], the game theory is used to analyze the conflicts related water systems management. The problem is modeled by water resource non-cooperative games. The authors discuss about the behaviors of stakeholders in different periods of the conflict.

Furthermore, as soon as a network is concerned, the interactions taking place inside can often be modeled as a game : the agents are players that compete or form coalitions to get some benefit. *Fortunately*, this is the case of multi-agent systems. The main motivation to formulate a network in a game is the interdependence between the actions of agents. In the case where a group of individuals has to achieve a goal, whether common or individual, the approach to designing such strategies can be radically different. The effect is further accentuated when resources are shared or not.

### Definitions and concepts

There are thee main types of representation of a game : *the normal or strategic form*, *the extensive form* and *the coalitional form*, but only the strategic form will be tackled. For more details and explanations, readers are referred to [Başar and Olsder, 1998] and [Lasaulce and Tembine, 2011] as complete resources.

***Definition* 10.** *A **strategic form game** is an ordered triplet*

$$\mathcal{G} = (\mathcal{V}, \{\mathcal{K}_i\}_{i\in\mathcal{V}}, \{J_i\}_{i\in\mathcal{V}}), \tag{1.7}$$

*where $\mathcal{V} = \{1, ..., N\}$ is the set of players, $\mathcal{K}_i$ is the set of strategies of player $i$, and $J_i$ is a utility/cost function of player $i$.*

In both game theory and multi-agent systems, the strategic form is the most widely used and the most simple representation. In general, it better suits to mathematical analysis and is well adapted to discrete and continuous strategy sets. In fact, the most common form relies on the existence of a cost/utility/reward function for each player. This allows us to define the satisfaction equilibrium and the Nash equilibrium as follows.

***Definition* 11.** *Let $\mathcal{G}$ be a strategic game and $f_1$,...,$f_N$ be $N$ set-valued satisfaction functions. The strategy profile $\mathbf{K}^* = (K_1^*, ..., K_N^*)$ is a* ***Satisfaction Equilibrium*** *(**SE**) if and only if*

$$\forall i \in \mathcal{V}, K_i^* \in f_i(K_{-i}^*), \tag{1.8}$$

*where $K_{-i}^* := (K_1^*, ..., K_{i-1}^*, K_{i+1}^*, ..., K_N^*)$ denotes the reduced profile with the component $K_i^*$ removed.*

We say that a player is satisfied if its current action satisfies his own constraints and thus an equilibrium is reached when all players are simultaneously satisfied. Since players are assumed to be selfish, we understand that a player satisfying his own constraints has no interest in changing his strategy. Moreover, if there is an equilibrium of satisfaction, it may not be unique.

It is also noteworthy that if for all $i \in \mathcal{V}$, the function $f_i$ is defined such that $f_i(K_{-i}) = \{K_i \in \mathcal{K}_i : J_i(K_i, K_{-i}) \geq \gamma_i\}$ where $\gamma_i$ is the minimum utility level required by player $i$, Definition 11 coincides with the definition of SE provided in [Ross and Chaib-draa, 2006].

The Nash equilibrium is a fundamental solution concept for the strategic form games and was extensively studied in the literature.

***Definition* 12.** *Let $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times ... \times \mathcal{K}_N$. Let $J_i : \mathcal{K} \to \mathbb{R}$, $i \in \mathcal{V}$. The profile $\mathbf{K}^*$ is a **Nash equilibrium** (**NE**) if*

$$\forall i \in \mathcal{V}, \forall K_i \in \mathcal{K}_i,\ J_i(K_i^*, K_{-i}^*) \geq J_i(K_i, K_{-i}^*). \tag{1.9}$$

Given the strategies adopted by the other players, a Nash equilibrium refers to a situation in which none of the players can find a better game strategy. Moreover, similarly to an SE we can have several NE.

One straightforward way for a player to choose its action is to optimize its utility function given the other players' strategies. The basic notion of best response is defined as follows.

***Definition* 13.** *The **Best Response** (**BR**) of player $i$ to the reduced strategy profiles $K_{-i}$ is the correspondence given by*

$$BR_i(K_{-i}) = arg \max_{K_i \in \mathcal{K}_i} J_i(K_i, K_{-i}), \tag{1.10}$$

*where $\mathcal{V} = \{1, ..., N\}$ is the set of players and $\mathcal{K}_i$ is the set of strategies of player $i$.*

Given the strategies of other players, the best response is the strategy producing the most favorable immediate outcome for the current player. By introducing the following function

$$\begin{aligned} BR : \mathcal{K} &\rightarrow \mathcal{K} \\ \mathbf{K} &\rightarrow BR_1(K_{-1}) \times ... \times BR_N(K_{-N}), \end{aligned} \tag{1.11}$$

one possible characterization of an NE is such that, a stragegy profile $\mathbf{K}^*$ is an NE if and only if $\mathbf{K}^* \in BR(\mathbf{K}^*)$. Implicitly, seeking for an NE can be seen as a fixed point problem.

### 1.2.3 Singularly perturbed systems

Generally, the mathematical results proposed by control theory are applied to a physical model of a real system. From there, any prior simplification of the equations may seem welcome. However, do not these simplifications call into question the validity of the conclusions they have established ? Indeed, the real system is often far more complex than the mathematical model.

This section aims to give some results on two time-scale systems and their approximations, which are less complex systems with essentially only one time-scale. This allows us to justify the relevance of reduced models on which we are able to provide stability or robustness results. The singular perturbation theory enables to take into account the presence of different time scales in a system. It leads to approaches based on the decoupling of dynamics such that the separation of slow and fast variables. The theory of singular perturbations provides a range of analysis tools based on the separation of time scales and the solution approximations of the original system by decoupling slow and fast dynamics, for example [Teel et al., 2003], [Nešić and Teel, 2001]. For this purpose, we will rely on [Kokotović et al., 1999] for existing definitions and results. We only provide results for linear time-invariant systems. However, I would like to give you a bit of history before we start.

The story starts in 1904 at the Third International Congress of Mathematicians in Heidelberg, Ludwig Prandtl would have presented his paper *On fluid motion with small friction* on fluid dynamical boundary layer, and from there the singular perturbation concept was born. However, the paper published in the wrong journal have delayed the development of the theory. The result would not surface until some twenty years later.

While singular perturbation theory, which has meanwhile become a traditional tool in fluid dynamics and nonlinear mechanics, encompasses a wide variety of dynamical phenomena with slow and fast modes, its assimilation into control theory came much later. It took almost fifty years for singular perturbation theory and control theory to meet. Extensively studied in the mathematical literature in [Tikhonov, 1948, 1952], [Levinson, 1950], [Vasil'eva, 1963], [Wasow, 1965], [Hoppensteadt, 1967, 1971], [O'Malley, 1971] and many others talented mathematicians, the singular perturbation model of finite-dimension was the first model to be used in control and systems theory.

In the 1950s, Tikhonov laid the foundations of singular perturbation theory in automatic-control. Afterwards, Kokotovic and Khalil gathered the most significant and useful results into two works that later became references in terms of nonlinear systems and singular perturbed systems, [Khalil and Grizzle, 2002], [Kokotović et al., 1999]. The analysis of these systems has become a subject of constantly evolving research.

For readers wishing to find more details, here are surveys from 1975 to 2012 : 1975 [Kokotovic et al., 1976], (1976-1983) [Saksena et al., 1984], (1984-2001) [Naidu, 2002], (2002-2012) [Zhang et al., 2014] and 2001 for Guidance and Control of Aerospace Systems [Naidu and Calise, 2001].

In order to study the two time-scale properties of a single system, we consider the following standard perturbation form as

$$\begin{cases} \dot{x}(t) = A_{11}x(t) + A_{12}z(t), \\ \epsilon\dot{z}(t) = A_{21}x(t) + A_{22}z(t), \end{cases} \tag{1.12}$$

where $x \in \mathbb{R}^{n_x}$ and $z \in \mathbb{R}^{n_z}$ represent the slow and fast system respectively, $\epsilon$ is a small positive scalar and $A_{11}$, $A_{12}$, $A_{21}$, $A_{22}$ are matrices of appropriate dimensions. The scalar $\epsilon$ represents all the small parameters to be neglected. Moreover, all the matrices in (1.12) are assumed to be constant and independent of $\epsilon$.

Operating in two different time-scales $t$ and $\tau = \epsilon t$, the fast dynamics emerges during the transient phase while the steady state is governed by the slow dynamics. By setting $\epsilon = 0$ in (1.12), if there exists $A_{22}^{-1}$, then the equation (1.12) yields

$$z_s(t) = -A_{22}^{-1}A_{21}x_s(t). \tag{1.13}$$

Substituting (1.13) in (1.12) yields the reduced slow model as

$$\dot{x}_s(t) = A_0 x_s(t), \tag{1.14}$$

where $x_s(0) = x(0)$ and $A_0 = A_{11} - A_{12}A_{22}^{-1}A_{21}$.

As for the boundary layer describing the fast dynamics of $z(t)$, it is given by

$$\epsilon\dot{z}_f(t) = A_{22}z_f(t), \tag{1.15}$$

where the initial condition $z_f(0) = z(0) + A_{22}^{-1}A_{21}x_s(0)$.

The following theorem provides the approximation conditions of the solution of the singularly perturbed system (1.12) from the analysis of the slow and fast dynamics separately.

***Theorem* 1.** *(Theorem 5.1, Chapter 2, [Kokotović et al., 1999]) If the matrix $A_{22}$ is Hurwitz, there exists an $\epsilon^* > 0$ such that, for all $\epsilon \in (0, \epsilon^*]$ and for all $t \in [0, T]$, the states of the original system (1.12) starting from any bounded initial conditions $x(0)$ and $z(0)$, $\|x(0)\| < c_1$ and $\|z(0)\| < c_2$, where $c_1$ and $c_2$ are constants independent of $\epsilon$, are approximated by*

$$\begin{cases} x(t) = x_s(t) + \mathcal{O}(\epsilon), \\ z(t) = -A_{22}^{-1} A_{21} x_s(t) + z_f(t) + \mathcal{O}(\epsilon), \end{cases} \tag{1.16}$$

*where $x_s(t)$ and $z_f(\tau)$ are the solutions of the reduced slow (1.14) and fast (1.13), respectively. Moreover, if $A_0$ is Hurwitz, then the approximation (1.16) holds for any time $t > 0$.*

The following asymptotic stability result follows.

***Theorem* 2.** *[Kokotović et al., 1999] If $A_{22}$ is non-singular and if the matrices $A_0$ and $A_{22}$ are Hurwitz, then there exists $\epsilon^*$ such that for all $\epsilon \in (0, \epsilon^*]$, the system (1.12) is asymptotically stable.*

In the following, we present the methodology for designing a state feedback control law for a singularly perturbed system in continuous time.

Consider the singularly perturbed system under a control law $u(t) \in \mathbb{R}^m$

$$\begin{cases} \dot{x}(t) = A_{11} x(t) + A_{12} z(t) + B_1 u(t), \\ \epsilon \dot{z}(t) = A_{21} x(t) + A_{22} z(t) + B_2 u(t). \end{cases} \tag{1.17}$$

After decoupling the slow and fast dynamics, the reduced system is

$$\begin{cases} \dot{x}_s(t) = A_0 x(t) + B_0 u(t), \quad x_s(0) = x(0), \\ \epsilon \dot{z}_s(t) = -A_{22}^{-1}(A_{21} x_s(t) + B_2 u_s(t)), \end{cases} \tag{1.18}$$

where $A_0 = A_{11} - A_{12} A_{22}^{-1} A_{21}$ and $B_0 = B1 - A_{12} A_{22}^{-1} A_{21} B_2$. The vectors $x_s$, $z_s$ and $u_s$ are the slow parts of the corresponding variables $x$, $z$ and $u$ in the original system (1.18). The boundary layer is described by

$$\begin{cases} \epsilon \dot{z}_f(t) = A_{22} z_f(t) + B_2 u_f(t), \\ z_f(0) = z(0) - z_s(0), \end{cases} \tag{1.19}$$

where $z_f(t) = z(t) - z_s(t)$ et $u_f(t) = u(t) - u_s(t)$ correspond to the fast component of the variables $z$ and $u$, respectively.

Chapter 2

# Decentralized control for guaranteed individual costs



This chapter presents our results on decentralized controls for multi-agent systems with homogeneous linear dynamics. The objective is to provide a control strategy design that can be implemented in a decentralized manner, and allows the network to achieve synchronization while ensuring that the individual costs are, if feasible, bounded by a certain given threshold.

In Section 2.1, we characterize the system and motivate the advantage of bounding an individual cost. The Section 2.2 explains how the concept of SE is applied to our framework. Then, in Section 2.3, we reformulate the synchronization problem to stabilization problem via a change of variable. Afterwards, we provide conditions in the form of LMIs to check if a given set of control gains are in SE, i.e. all individual costs are upper-bounded by the imposed threshold. In the section 2.2, we An algorithm is also proposed in order to synthesize gains that are in SE. Furthermore, we briefly address the issue of the algorithm complexity. The Section 2.4 shows a special case when the graph is complete. Finally, the Section 2.5 illustrates the results with numerical examples.

The results of this chapter corresponds to the publication [Veetaseveera et al., 2019].

***Remark* 2.** The

## 2.1 Problem statement

### 2.1.1 Linear agent dynamics

Consider a network of $N$ agents, where the interactions are described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, ..., N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. We identify each agent with its index $i \in \mathcal{V}$ and assign to each agent $i$ a state $x_i \in \mathbb{R}^{n_x}$. The individual agents dynamics are described by

$$\begin{cases} \dot{x}_i(t) = Ax_i(t) + Bu_i(t) \\ y_i(t) = \sum\limits_{j \in \mathcal{N}_i} x_j(t) - x_i(t) \end{cases}, \quad \forall i \in \mathcal{V}, \tag{2.1}$$

where $x_i, y_i \in \mathbb{R}^{n_x}$ are respectively the state and the output of the agent $i$, $u_i \in \mathbb{R}^{n_u}$ the control, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$. The control $u_i$ applied by the $i$-th agent depends directly on the output $y_i$. Let $x(t) = (x_1(t), ..., x_N(t)) \in \mathbb{R}^{N \cdot n_x}$ and $u(t) = (u_1(t), ..., u_N(t)) \in \mathbb{R}^{N \cdot n_u}$ be respectively the global state of the network and the control at time $t \in \mathbb{R}_+$. Furthermore, the initial condition is denoted by $x_0 = x(0)$.

### 2.1.2 Individual costs

The control objective is to asymptotically synchronize the set of agents, i.e. to ensure that for all $i, j \in \mathcal{V}$, $\lim\limits_{t \to +\infty} \|x_j(t) - x_i(t)\| = 0$. To achieve this common goal, each agent tends to provide a certain control effort that has to be limited in many real applications. The decision between a global cost and individual costs to be minimized might be critical in some circumstances.

In contrast with the existing works, our objective is to guarantee that the individual costs incurred by *each agent* does not exceed a given threshold during the process of synchronization. Furthermore, in the special case of complete graphs, we also provide a control optimizing each individual cost.

To this end, we consider a per-agent cost and propose two cost expressions. The first cost concerns the connected graphs while the second one is for complete graphs. Moreover, we recall that we are seeking for a decentralized control policy, it means that each agent only has access to the information on the relative state with respect to its neighbourhood $\mathcal{N}_i$, i.e. $y_i$ and does not have access to overall network state $x$.

## Case of connected graphs

For all $i \in \mathcal{V}$, we define an individual satisfactory cost $J_i^{SE}$ associated with the agent $i$ as follows

$$J_i^{SE} = \int_0^{+\infty} \sum_{j \in \mathcal{N}_i} \|x_j(t) - x_i(t)\|^2 + u_i^\top(t) R_i u_i(t) \, \mathrm{d}t, \tag{2.2}$$

where the positive definite matrix $R_i \in \mathbb{R}^{n_u \times n_u}$ represents the *weight* given to the control action $u_i$. The quadratic cost (2.2) is equivalent to the energy required by each agent to establish the network synchronization. The first term corresponds to the state cost or how far is an agent $i$ with respect to his neighborhood. As for the control part, it represents the action energy necessary for an agent $i$ to be synchronized.

## Case of complete graphs

Considering the particular case of complete graphs, we provide a second expression for the individual cost $J_i^{NE}$, for all $i \in \mathcal{V}$, such that

$$J_i^{NE} = \int_0^{+\infty} \left\| \sum_{j \in \mathcal{N}_i} x_j(t) - x_i(t) \right\|^2 + u_i^\top(t) R u_i(t) \, \mathrm{d}t, \tag{2.3}$$

where $R \in \mathbb{R}^{n_u \times n_u}$ is a positive definite matrix. In (2.2), the state part represents exactly the sum of the squared difference between an agent $i$ and its neighbors whereas in (2.3), some crossing terms appear. The cost (2.3) has two drawbacks, it does not represent well the proximity between an agent and its neighborhood, and in some cases can be conservative for an optimization problem. The state term can be null even though the agent $i$ is not synchronized with its neighbors. However, the all-to-all connection topology can be of interest if we consider the computational loads and complexity during the control procedure. This point will be developed later in a further section.

The notation $SE$ and $NE$ refer respectively to the *Satisfaction equilibrium* and *Nash equilibrium* that will be discussed in the upcoming sections. The next section explains how game theory allows us to reformulate the problem and to deal with a set of individual costs.

## 2.2 Reformulation of the problem via game theory

In the context of decentralized control design, two main reasons limit the direct application of the standard optimization techniques. First, the communication constraints induced by the decentralized control design restrict the access to the overall state $x$ of the network. The agents only have information from their neighbours and not from the whole network. The control $u_i$ must be designed independently of $x_0$ and must depend only on $y_i$. Secondly, the notion of optimality related to the set of cost functions seems unsuitable. It is straightforward to understand with the following example that the concept of *optimality* has to be *redefined*, *extended* or *reformulated*.

We do not optimize one global cost anymore but we consider a set of $N$ individual costs. It results in a non-convex optimization problem. To illustrate the inherent issue related to the set of cost functions, let us denote

$$\begin{cases} u^{\mathbf{A}} = (u_1^{\mathbf{A}}, ..., u_N^{\mathbf{A}}) \\ u^{\mathbf{B}} = (u_1^{\mathbf{B}}, ..., u_N^{\mathbf{B}}) \end{cases} \quad \text{and} \quad \begin{cases} J(u^{\mathbf{A}}) = (J_1^{\mathbf{A}}, ..., J_N^{\mathbf{A}}) \\ J(u^{\mathbf{B}}) = (J_1^{\mathbf{B}}, ..., J_N^{\mathbf{B}}) \end{cases}, \tag{2.4}$$

where $u^{\mathbf{A}}$ and $u^{\mathbf{B}}$ are the control stategy $\mathbf{A}$ and $\mathbf{B}$, respectively. The set of costs $J^{\mathbf{A}}$ and $J^{\mathbf{B}}$ are obtained by applying the stategy $\mathbf{A}$ and $\mathbf{B}$ to (2.1), respectively. Moreover, the individual costs in $J^{\mathbf{A}}$ and $J^{\mathbf{B}}$ are defined based on (2.2).

Let the control $u^{\mathbf{A}}$ and $u^{\mathbf{B}}$ be designed such that the following inequalities hold

$$\begin{cases} J_i^A \leq J_i^B, & \forall i = 1, ..., p, \\ J_i^A \geq J_i^B, & \forall i = p+1, ..., N. \end{cases} \tag{2.5}$$

Since the costs rely on the initial condition $x_0$, it is natural that some costs $J_i^A$ might be smaller than $J_i^B$ and vice versa. In the case we aim to minimize the costs, the strategy $\mathbf{A}$ is more efficient for $i = 1, ..., p$, but the strategy $\mathbf{B}$ performs better for $i = p+1, ..., N$. In an optimization problem, we observe that any control is better than the other. A piece is missing to the puzzle. Unfortunately, the classical tools of optimal control cannot be directly applied in our context.

Our problem involves a set of $N$ agents associated with individual costs. Since each agent desires to reduce its own cost, we are in the framework of a game. Besides the general information structure, the agents can also use information from past actions and take this into account for their current and future actions. This case corresponds to a differential game with incomplete information [Lasaulce and Tembine, 2011]. Nonetheless, the research on differential games remains open for a general information structure.

As mention in Chapter 1, we are inspired by [Ross and Chaib-draa, 2006] which introduced the notion of satisfaction games. The SE concept was applied to wireless networks in order to guarantee a satisfactory quality of service [Perlaza et al., 2012]. Furthermore, we also apply the concept of NE for complete graphs. In [Marden et al., 2007], the authors establish a relationship between cooperative control problems, such as the consensus problem, and game theoretic methods. A learning algorithm for finding a NE is also provided. Generally, cooperative control problems involve agents seeking to collectively accomplish a global objective.

In the following, we first reformulate the decentralized control design problem for multi-agent systems as a satisfaction game. Then, in the subsequent section, we apply results from LQR control with static output feedback to provide conditions on achieving a satisfaction equilibrium. Finally, we deal with the complete graphs by providing an analytical result for the control and conclude the chapter with numerical illustrations.

Concerning the controller, we are restricted to designing $u_i$ based on the output $y_i$. In this setting, we search for controllers that are of the static-output feedback type. The following assumption is valid for both connected and complete graphs.

***Assumption* 1.** The controller for agent $i \in \mathcal{V}$ is of the form $u_i = K_i y_i$, where $K_i \in \mathcal{K}_i$ with $\mathcal{K}_i = \mathbb{R}^{n_u \times n_x}$.

With Assumption 1, the control strategy for an agent $i$ is fully defined by the choice of gain $K_i$.

### 2.2.1 Satisfaction Equilibrium approach

Before presenting our satisfaction game, we define a gain profile as $\mathbf{K} = (K_1, ..., K_N) \in \mathcal{K}$, where $\mathcal{K} = \mathcal{K}_1 \times ... \times \mathcal{K}_N$. We also use $K_{-i} := (K_1, ..., K_{i-1}, K_{i+1}, ..., K_N)$ to denote the profile of gains for all agents except $i$. To emphasize its $i$-th component, we write the vector $\mathbf{K}$ as $(K_i, K_{-i})$ with a slight abuse of notation. Under Assumption 1, we define the satisfaction game as follows.

**Satisfaction game**

The satisfaction game in standard form is defined by the ordered triplet

$$\mathcal{G} = (V, \{\mathcal{K}_i\}_{i\in\mathcal{V}}, \{f_i\}_{i\in\mathcal{V}}), \tag{2.6}$$

where

- $\mathcal{V} = \{1, ..., N\}$ is the set of agents and we identify each agent $i$ as player $i$,
- $\mathcal{K}_i = \mathbb{R}^{n_u \times n_x}$ is the set of actions or gains $K_i$ applied by player $i$. The control applied by each player is given according to Assumption 1,
- A player $i$ is said to be satisfied when his action $K_i \in f_i(K_{-i})$ considering the actions of the other players $K_{-i}$ are given. Here, $f_i(K_{-i})$ is called as the satisfaction function, and is defined as follows

$$f_i(K_{-i}) := \{K_i \in \mathcal{K}_i | J_i^{SE}(K_i, K_{-i}) < \gamma\|x_0 - \mathbb{1}_N \otimes x_i(0)\|^2\}, \tag{2.7}$$

where $\gamma > 0$ is a given threshold. The satisfaction function $f_i$ of the player $i$ describes the set of actions which guarantees that its cost is upper-bounded by $\gamma$ given the actions of all the other players. The cost $J_i$ may represent the investment of agent $i$ in an auction, or the fuel consumption when the agent $i$ is a car. The expression of the norm $\|x_0 - \mathbb{1}_N \otimes x_i(0)\|^2$ is explained later in the next Section 2.3.

***Remark* 3.** The dependence of the cost (2.2) on the gain $K_i$ and the gains $K_{-i}$ is more explicit. It reflects that agents influence each other. We understand that choosing a single gain is not sufficient, we need to choose a gain profile such that all the agents are satisfied.

In our satisfaction game, when all players satisfy

$$J_i^{SE} \leq \gamma\|x_0 - \mathbb{1}_N \otimes x_i(0)\|^2, \tag{2.8}$$

we say that the players are in a satisfaction equilibrium. Next, it is important to note that the cost $J_i$ will always depend on the global initial state vector $x_0$. However, this dependency can be removed by treating the case in which all the other agents start in a unit ball radius around each agent. The resulting cost can then be scaled up for other initial conditions.

***Remark* 4.** To simplify the expression in the following, we scale the initial condition such that $\|x_0 - \mathbb{1}_N \otimes x_i(0)\|^2 \leq 1$, for all $i \in \mathcal{V}$. As usual, the individual costs scales with the initial condition and under the normalization above we get a satisfaction game when $J_i^{SE} \leq \gamma$, for all $i \in \mathcal{V}$. Thus, the satisfaction function (2.7) becomes

$$f_i(K_{-i}) := \{K_i \in \mathcal{K}_i | J_i^{SE}(K_i, K_{-i}) < \gamma\}, \quad \forall i \in \mathcal{V}. \tag{2.9}$$

**Satisfaction equilibrium**

The introduction of the game (2.6) allows us to define our satisfaction equilibrium as follows.

***Definition* 14.** Let $\mathcal{G}$ be a *strategic game* and $f_1$,...,$f_N$ be $N$ set-valued satisfaction functions. The strategy profile $\mathbf{K}^* = (K_1^*, ..., K_N^*)$ is an SE if and only if

$$\forall i \in \mathcal{V}, K_i^* \in f_i(K_{-i}^*).$$

According to Definition 14, the agents reach the satisfaction equibrilium when $J_i^{SE}(\mathbf{K}^*) \leq \gamma$, for all $i \in \mathcal{V}$. Once the players are at the SE, none of them have a particular interest to change their current actions as each player has achieved the desired bound on his cost. They are assumed to be careless of the satisfaction of other players.

***Remark* 5.** First, the Definition 14 implies that the SE is not an optimal solution. Indeed, we do not minimize the individual costs anymore but we just bound them. In other words, we are seeking for a gain $K_i$ given $K_{-i}$ such that $J_i^{SE} \leq \gamma$ for the agent $i$. Secondly, if the SE exists it may not be unique. The notion of equilibrium in game theory is far away from the one we are used in control theory.

In the sequel, we explain how such a strategy profile $\mathbf{K}^*$ can be designed thanks to the control theory.

## 2.3 Satisfactory control design

In this section, we transform the synchronization problem into a stabilization one. Inspired by the sub-optimal output control design in [Iwasaki et al., 1994], we perform a change of variables and recast the cost (2.2) in a new variable. Then, we provide conditions to check in the form of LMIs and propose a control design for a gain $K_i^*$ given the other gains $K_{-i}^*$. Finally, we end this section with a brief explanation of the iterative algorithm.

### 2.3.1 Change of variables

In order to check if $\mathbf{K}^*$ is an SE, we reduce the synchronization problem to a stabilization problem by a change of variables on (2.1). For further purpose, for each agent $i \in \mathcal{V}$, we introduce the error state vector as follows

$$\chi_i = (x_1 - x_i, \dots, x_{i-1} - x_i, x_{i+1} - x_i, \dots, x_N - x_i), \tag{2.10}$$

where $\chi_i \in \mathbb{R}^{(N-1)\times n_x}$. The stabilization of $\chi_i$ at the origin corresponds to the synchronization of the agent $x_i$ with the network. Indeed, one may notice that, $\forall i \in \mathcal{V}, \lim\limits_{t\to+\infty} \|\chi_i(t)\| = 0$ $\Leftrightarrow \forall i, j \in \mathcal{V}, \lim\limits_{t\to+\infty} \|x_j(t) - x_i(t)\| = 0$.

Let us give some useful notation for the dynamics of the error state (2.10). Denote by $\mathcal{L}_{-i}$, the Laplacian matrix $\mathcal{L}$ without the $i$-th row and column, and by $\mathcal{L}_i$, the $i$-th row of $\mathcal{L}$. We can now write the dynamics for $\chi_i$ as

$$\dot{\chi}_i = \mathcal{A}_i(K_{-i})\chi_i + \mathcal{B}_i u_i, \tag{2.11}$$

where $\mathcal{A}_i = (\mathbf{I}_{n-1} \otimes A) - (\mathbf{I}_{n-1} \otimes B)\mathrm{diag}(K_{-i})(\mathcal{L}_{-i} \otimes \mathbf{I}_{n_x})$ and $\mathcal{B}_i = -(\mathbb{1}_{n-1} \otimes B)$.

The term $\mathrm{diag}(K_{-i})$ is not a control action but it represents the behaviour of the network governed by the other agents actions. For agent $i$, the control action is simply $u_i = K_i y_i$ as it can not control the other agents.

In order to rewrite the cost (2.2), let us define the auxiliary variables as

$$\begin{cases} z_i = \mathcal{C}_i \chi_i + \mathcal{D}_i u_i, \\ y_i = \mathcal{F}_i \chi_i, \\ u_i = K_i y_i, \end{cases} \tag{2.12}$$

where

$$\mathcal{C}_i = \begin{pmatrix} \mathrm{diag}(\mathcal{L}_{i:\mathrm{red}}) \otimes \mathbf{I}_{n_x} \\ \mathbf{0}_{n_u \times (N-1)n_x} \end{pmatrix}, \quad \mathcal{D}_i = \begin{pmatrix} \mathbf{0}_{(N-1)n_x \times n_u} \\ R_i \end{pmatrix} \quad \text{and} \quad \mathcal{F}_i = -(\mathcal{L}_{i:red} \otimes \mathbf{I}_{n_x}). \tag{2.13}$$

Since the $N$-th block of $\mathcal{C}_i$ is $0$ by definition, the $N$-th block of $z_i$ will contain the weighted control $R_i u_i$. Finally, we also use $\mathcal{L}_{i:\mathrm{red}}$ to denote the row matrix $\mathcal{L}_i$ with the $i$-th column removed. The cost for any agent $i$ can now be written in terms of $z_i$ as

$$J_i^{SE} = \int_0^{+\infty} \|z_i(t)\|^2 \,\mathrm{d}t. \tag{2.14}$$

### 2.3.2 LMIs conditions for an SE

Under the new variables, the problem of asymptotic synchronization of the system (2.1) now becomes stabilization of the system (2.11). This allows us to establish the following result.

***Proposition* 3.** *Let a gain profile $\mathbf{K}^*$ be given. The following statements are equivalent when Assumption 1 holds.*

1. *The gain profile $\mathbf{K}^*$ is an SE of the satisfaction game $\mathcal{G}$ that stabilizes (2.11)-(2.12) for all $i \in \mathcal{V}$.*

2. *For all $i \in \mathcal{V}$, there exists a positive-definite matrix $P_i > 0$ such that*

$$\begin{cases} P_i\mathcal{A}_{i,cl}(\mathbf{K}^*) + \mathcal{A}_{i,cl}(\mathbf{K}^*)^\top P_i + \mathcal{C}_{i,cl}^\top \mathcal{C}_{i,cl} < 0, \\ P_i - \gamma \mathbf{I}_{(n-1)n_x} < 0, \end{cases} \tag{2.15}$$

*where $\mathcal{A}_{i,cl}(\mathbf{K}^*) = (\mathcal{A}_i(K_{-i}^*) + \mathcal{B}_i K_i^* \mathcal{F}_i)$, $\mathcal{C}_{i,cl} = (\mathcal{C}_i + \mathcal{D}_i K_i^* \mathcal{F}_i)$ are respectively the closed-loop matrices for $\chi_i$ and $z_i$.*

***Proof***:

Lemma 1 in [Iwasaki et al., 1994] states that, when $K_i^*$ is given and $\|\chi_i(0)\| \leq 1$ is known, the following are equivalent

1. The gain $K_i^*$ stabilizes the system (2.11)-(2.12) and yields the quadratic cost $J_i^{SE}(K_i^*, K_{-i}^*) < \gamma$

2. There exists $P_i > 0$ such that

$$P_i\mathcal{A}_{i,cl}(\mathbf{K}^*) + \mathcal{A}_{i,cl}(\mathbf{K}^*)^\top P_i + \mathcal{C}_{i,cl}^\top \mathcal{C}_{i,cl} < 0, \quad \text{and} \quad \|P_i\| < \gamma. \tag{2.16}$$

We develop the proof in two steps. First we prove the direct implication, then we show the inverse relation.

Proof of $1) \Rightarrow 2)$

Let $\mathbf{K}^*$ be an SE of $\mathcal{G}$ that stabilizes (2.11)-(2.12) for all $i \in \mathcal{V}$. By Definition 14, for all $i \in \mathcal{V}$, one has $K_i^* \in f_i(K_{-i}^*)$ yielding $J^{SE}(K_i^*, K_{-i}^*) < \gamma$, for all $i \in \mathcal{V}$. We rewrite (2.11)-(2.12) in closed-loop form as

$$\dot{\chi}_i = \mathcal{A}_{i,cl}(\mathbf{K}^*)\chi_i,$$

with the cost given by

$$J_i^{SE} = \int_0^{+\infty} \|\mathcal{C}_{i,cl}\chi_i(t)\|^2 \, \mathrm{d}t < \gamma,$$

for all $i \in \mathcal{V}$. Since $\mathbf{K}^*$ is given, the matrices $\mathcal{A}_{i,cl}(\mathbf{K}^*)$ are fixed and known. From Remark 4, we have $\|\chi_i(0)\| \leq 1$ and applying Lemma 1 in [Iwasaki et al., 1994] one obtains the existence of matrices $P_i$ satisfying (3.56).

Proof of $2) \Rightarrow 1)$

Suppose now that (3.56) holds for all $i \in \mathcal{V}$. Again, from Lemma 1 in [Iwasaki et al., 1994], one has that $K_i^*$ stabilizes (2.11)-(2.12) and $J^{SE}(K_i^*, K_{-i}^*) < \gamma$ for all $i \in \mathcal{V}$. Therefore, $K_i^* \in f_i(K_{-i}^*)$, for all $i \in \mathcal{V}$ which means that $\mathbf{K}^*$ is an SE of $\mathcal{G}$ given a $\gamma$. ■

To summarize, we have provided LMIs conditions to test if a given $\mathbf{K}$ is an SE of the game $\mathcal{G}$ for the performance bound $\gamma$. The given $\mathbf{K}$ is a satisfaction equilibrium of $\mathcal{G}$ if and only if it satisfies the condition 2) for all $i \in \mathcal{V}$. However, the inability to find matrices $P_i$ does not imply that they do not exist, this may arise due to numerical issues with the LMI solver. Finally, for a given $\gamma$, $\mathbf{K}$ may not be unique and it is possible to have several gains which are satisfaction equilibria. Therefore, we define the set of satisfaction equilibria as

$$\mathcal{K}^* = \{\mathbf{K}^* : \forall i \in \mathcal{V}, K_i^* \in f_i(K_{-i}^*)\}. \tag{2.17}$$

In the next subsection, we provide a method which allows us to synthesize the gains for a given thereshold $\gamma$.

### 2.3.3 Synthesis of a gain profile $\mathbf{K} \in \mathcal{K}^*$

In this section, we first present conditions that allow us to generate the satisfaction function $f_i(K_{-i})$ based on the results in [Iwasaki et al., 1994]. In the following proposition, for a given set of gains $K_{-i}$, we find a synchronizing gain $K_i$ under certain conditions as described below.

***Proposition* 4** (Based on ***Theorem 1*** in [Iwasaki et al., 1994])**.** *Let the set of gains $K_{-i}$ be given. Consider the sets*

$$\begin{aligned} \mathscr{X}_i(K_{-i}) &:= \left\{ X \in \mathbb{R}^{(N-1).n_x \times (N-1).n_x} : \begin{bmatrix} \mathcal{B}_i \\ \mathcal{D}_i \end{bmatrix}^{\perp} \begin{bmatrix} \mathcal{A}_i X + X\mathcal{A}_i^\top & X\mathcal{C}_i^\top \\ \mathcal{C}_i X & -\mathbf{I}_{(N-1).n_x} \end{bmatrix} \begin{bmatrix} \mathcal{B}_i \\ \mathcal{D}_i \end{bmatrix}^{\perp\top} < 0 \right\}, \\ \mathscr{Y}_i(K_{-i}) &:= \left\{ Y \in \mathbb{R}^{(N-1).n_x \times (N-1).n_x} : \mathcal{F}_i^{\top\perp}(Y\mathcal{A}_i + \mathcal{A}_i^\top Y + \mathcal{C}_i^\top \mathcal{C}_i)\mathcal{F}_i^{\top\perp\top} < 0, Y - \gamma \mathbf{I}_{(N-1).n_x} < 0 \right\}. \end{aligned} \tag{2.18}$$

*Under Assumption 1, if no $P > 0$ exists such that $P^{-1} \in \mathscr{X}_i(K_{-i})$ and $P \in \mathscr{Y}_i(K_{-i})$, then $f_i(K_{-i}) = \emptyset$, which implies that we can not find a suitable $K_i$ such that $(K_i, K_{-i})$ is a satisfaction equilibrium. Otherwise, the satisfaction function for the game $\mathcal{G}$ is given by*

$$\begin{aligned} f_i(K_{-i}) = \{ & -\rho_i \mathscr{B}_i^\top \Phi_i(K_{-i})\mathscr{C}_i^\top (\mathscr{C}_i \Phi_i(K_{-i})\mathscr{C}_i^\top)^{-1} + \rho_i S_i(K_{-i})^{1/2} M_i (\mathscr{C}_i \Phi(K_{-i})\mathscr{C}_i^\top)^{-1/2} : \\ & P > 0, P^{-1} \in \mathscr{X}(K_{-i}), P \in \mathscr{Y}(K_{-i}), \|M_i\| < 1 \}, \end{aligned} \tag{2.19}$$

*where*

$$\mathscr{B}_i := \begin{bmatrix} P\mathcal{B}_i \\ \mathcal{D}_i \end{bmatrix}, \quad \mathscr{C}_i := \begin{bmatrix} \mathcal{F}_i & 0 \end{bmatrix}, \quad Q_i(K_{-i}) := \begin{bmatrix} P\mathcal{A}_i(K_{-i}) + \mathcal{A}_i(K_{-i})^\top P & \mathcal{C}_i^\top \\ \mathcal{C}_i & -\mathbf{I}_{(N-1).n_x} \end{bmatrix},$$
$$\Phi_i(K_{-i}) := (\rho_i \mathscr{B}_i \mathscr{B}_i^\top - Q_i(K_{-i}))^{-1}, \quad S_i(K_{-i}) = \rho_i \mathbf{I}_{n_x} - \mathscr{B}_i^\top [\Phi_i - \Phi_i \mathscr{C}^\top (\mathscr{C}_i \Phi_i \mathscr{C}_i^\top)^{-1} \mathscr{C}_i \Phi_i] \mathscr{B}_i,$$
$$\rho_i(K_{-i}) := \rho_{min}(K_{-i}) + p, \quad \rho_{i,min}(K_{-i}) := max\{0, \lambda_{max}[\mathscr{B}_i^+ (Q_i(K_{-i}) -$$
$$Q_i(K_{-i}) \mathscr{B}_i^{\perp\top} (\mathscr{B}_i^\perp Q_i(K_{-i}) \mathscr{B}_i^{\perp\top})^{-1} \mathscr{B}_i^\perp Q_i(K_{-i})) \mathscr{B}_i^{+\top}]\},$$

*with $p \in \mathbb{R}_{\geq 0}$ an arbitrary non-negative scalar.*

***Proof***: By definition of the satisfaction function, we have $f_i(K_{-i}) = \{K_i | J_i^{SE}(K_i, K_{-i}) < \gamma\}$. Therefore, for a given $K_{-i}$, the subset $f_i$ is the set of gains resulting in a cost bounded by $\gamma$.

Moreover, we know that the Theorem 1 in [Iwasaki et al., 1994] provides conditions on the existence of a stabilizing static output feedback gain such that the LQ cost is bounded by a given factor $\gamma$, when $\chi_i(0) = Ww_0$ with $||w_0|| = 1$. From Remark 4, we have that $\|\chi_i(0)\| \leq 1$.

Then, once we rewrite the synchronization problem as (2.11) and (2.12), for a given $K_{-i}$, finding $K_i$ which results in $J_i^{SE}(K_i, K_{-i}) < \gamma$ is transformed into a problem of static output feedback with a bounded LQ cost.

Since we take $\|Ww_0\| \leq 1$, the condition $W^\top Y W - \gamma \mathbf{I} < 0$ as required in [Iwasaki et al., 1994] is satisfied if $Y - \gamma \mathbf{I}_{(N-1).n_x} < 0$. Applying the theorem, we get that the following are equivalent

- There exists $K_i$ stabilizing (2.11)-(2.12) such that $J_i^{SE}(K_i, K_{-i}) < \gamma$,
- There exists $P > 0$ such that $P^{-1} \in \mathscr{X}(K_{-i})$ and $P \in \mathscr{Y}(K_{-i})$ and $K_i$ is given by $-\rho \mathscr{B}^\top \Phi(K_{-i}) \mathscr{C}^\top (\mathscr{C} \Phi(K_{-i}) \mathscr{C}^\top)^{-1} + \rho S(K_{-i})^{1/2} M_i (\mathscr{C} \Phi(K_{-i}) \mathscr{C}^\top)^{-1/2}$.

Therefore, if no such $P$ exists, the satisfaction function is the empty-set. Otherwise, it can be written as in (2.19). ■

***Remark* 6.** It is noteworthy that the dynamics $\mathcal{A}_i$ depends on the set of gains $K_{-i}$. Furthermore, the matrix $M_i$ is a parameter in the algorithm chosen such that $\|M_i\| < 1$.

### 2.3.4 Symmetric case

If all agents have the same structure of $y_i$ and the dynamics are identical even if the indices are permuted the computation load is reduced by a factor of N. This type of graphs is called in the literature prime w.r.t. the modular decomposition, see [Brandstadt et al., 1999]. This means that the agents react the same way, so it allows us to look for a symmetric satisfaction equilibrium, i.e. we look for $K_0$ such that $(\mathbb{1}_N^\top \otimes K_0) \in \mathcal{K}$. For a given $K_0$, we can apply Proposition 3 to get that $(\mathbb{1}_N^\top \otimes K_0) \in \mathcal{K}^*$ if and only if there exists $P > 0$ such that

$$P\mathcal{A}_{i,cl}(\mathbb{1}_N^\top \otimes K_0) + \mathcal{A}_{i,cl}(\mathbb{1}_N^\top \otimes K_0)^\top P + \mathcal{C}_{i,cl}^\top \mathcal{C}_{i,cl} < 0 \quad \text{and} \quad \|P\| < \gamma. \tag{2.20}$$

for any $i \in \mathcal{V}$. Due to symmetry, we have $A_{i,cl} = A_{j,cl}$ and $C_{i,cl} = C_{j,cl}$ for any $i, j \in \mathcal{V}$. Therefore instead of having to solve $N$ LMIs, we just need to solve one. We can also simplify the synthesis with the following result.

***Corollary* 1.** *Let the network be prime w.r.t. the modular decomposition. Under Assumption 1, an SE for dynamics (2.1) with individual costs (2.2) is given by $(\mathbb{1}_N^\top \otimes K_0)$ if and only if there exists $P > 0$, $||M|| < 1$ such that*

$$\begin{aligned}
&P^{-1} \in \mathscr{X}_i(\mathbb{1}_{N-1}^\top \otimes K_0), P \in \mathscr{Y}_i(\mathbb{1}_{N-1}^\top \otimes K_0) \\
&K_0 = -\rho_i \mathscr{B}_i^\top \Phi_i(\mathbb{1}_{N-1}^\top \otimes K_0)\mathscr{C}_i^\top(\mathscr{C}_i\Phi_i(\mathbb{1}_{N-1}^\top \otimes K_0)\mathscr{C}_i^\top)^{-1} \\
&\quad +\rho_i S_i(\mathbb{1}_{N-1}^\top \otimes K_0)^{1/2} M(\mathscr{C}_i\Phi_i(\mathbb{1}_{N-1}^\top \otimes K_0)\mathscr{C}_i^\top)^{-1/2}
\end{aligned} \tag{2.21}$$

*with $\mathscr{X}_i, \mathscr{Y}_i, \mathscr{C}_i, \Phi_i, S_i, \rho$ as defined in Proposition 4, for all $i \in \mathcal{V}$.*

***Proof***: The above result is obtained by directly applying Proposition 4 and by exploiting the fact that $K_i = K_0$ and $K_{-i} = (\mathbb{1}_{n-1}^\top \otimes K_0)$. Additionally, the closed-loop dynamics are identical for all $i \in \mathcal{V}$. ■

Corollary 1 implies that for $K_0$ can be found as a solution to a fixed point equation. The Algorithm 1 developed in the next Section 2.3.5, can be used in a simpler manner to find a suitable $K_0$ by fixing all agents outside an arbitrary agent $i$ to have some gain $K_0$ and then finding $K_i$ using Proposition 4. Then, $K_0$ is updated to the $K_i$ and the process is repeated until $(\mathbb{1}_N^\top \otimes K_0) \in \mathcal{K}^*$.

The main motivation for considering this special case is that the number of iterations does not scale with $N$ as it does in the general case.

### 2.3.5 Sequential Satisfactory Response Algorithm (SSRA)

**How does this algorithm work ?**

In [Iwasaki et al., 1994], the Section 4 provides a scaled min-max algorithm which generates a matrix $P > 0$ satisfying the conditions in Proposition 4, if the problem has a solution. It is noteworthy that our design remains centralized although its implementation is distributed. Indeed, in order to find a $\mathbf{K}$ which satisfies Proposition 3, we need to find the gains $K_i$ for all agents.

To do that, we propose an algorithm which we call the *Sequential Satisfaction Response* (*SSR*) in order to achieve this task. This iterative algorithm is inspired by the sequential best response algorithm that is commonly found in the literature on game theory [Lasaulce and Tembine, 2011]. In contrast with the classical best response algorithm where the actions of players must converge over iterations, our algorithm only needs to satisfy $\mathbf{K} \in \mathcal{K}^*$ to succeed.

- ***Step 0 :*** (Initialization)
  Set $k = 1$, the maximum number of iterations $k_{max}$ and
  $K^0 = (K_1^0, ..., K_N^0)$ synchronizing the system.
- ***Step 1 :*** (Check stopping criterion)
  If the $N$ LMIs in Proposition 3 are satisfied, then **stop**.
  OR
  If $k > k_{max}$, then **stop**.
- ***Step 2 :***
  If ***Step 1*** not satisfied, then compute the gain $K_i^k$ given
  $K_{-i}^{k-1}$ according to Proposition 4. Solve the two convex
  optimizations problems with the scaled min/max algorithm.
- ***Step 3 :*** Set $k = (k \mod N) + 1$ and return to ***Step 1***.

**Algorithm 1:** Sequential satisfaction response

The Algorithm 1 applies the scaled min-max algorithm in [Iwasaki et al., 1994] repeatedly to find $P > 0$ satisfying the conditions in Proposition 4 given $K_{-i}$. This results in a gain for player $i$ which satisfies cost requirement. The player index is then updated to the next player and this procedure is repeated until $\mathbf{K}$ is a satisfaction equilibrium. The first player index $i$ is chosen arbitrarily.

Even if the algorithm fails to find a gain for a player $i$, it does not immediately stop and the player index is updated. However, if all $N$ players consecutively fail to find a gain $K_i$, the algorithm is stopped. Indeed, it is possible that the algorithm never stops as only a certain number of players are satisfied with this set cycling or not changing. We have no theoretical guarantee that this algorithm will find a satisfaction equilibrium even if $|\mathcal{K}^*| > 0$. In case of the algorithm failing, a larger $\gamma$ may be considered in order to find a satisfaction equilibrium for the game with the larger $\gamma$. This equilibrium may then be used as input for initializing the algorithm with the smaller $\gamma$.

Future works will explore improving the algorithm by adapting $\gamma$, but theoretical results are hard to obtain due to the conditions for finding the gains for just one player being non-convex and requiring the min-max algorithm. This difficulty is inherited from the problem of static output feedback design for linear systems and is not related to the multi-agent systems.

**What is the algorithm complexity ?**

Consider a network of $N$ agents with their respective dynamics. Our aim is to design a synchronizing gain profile $\mathbf{K} = (K_1, ..., K_N)$ satisfying the cost constraints (2.14).

In ***Step 1***, we first check $N$ LMIs. At ***Step 2***, we compute in the best case at least $2N$ gains $K_i^k$. To solve the convex optimization problems, we use the SeDuMi software. The complexity of solving an optimization problem is of order $O(N^{4.5})$ [Labit et al., 2002]. In our case the matrix size of LMIs during ***Step 2*** depends also on the number of agents, i.e. $N$ and the computational effort increases with increase in $N$. Since we solve at least $2N$ LMI, using SeDuMi the complexity of the **Algorithm 1** is $O(2N.N^{4.5})$. It is also important to note that the complexity of the algorithm depends on the choice of the solver.

For large scale networks, obtaining the gains via the general algorithm may be computationally infeasible as both the system matrix dimensions and the iterations scale with $N$, or the algorithm will require significant resources in terms of memory and time.

## 2.4 Analysis over complete graphs

Since designing the control $u_i$ is equivalent to choose a proper gain $K_i$, we apply the NE concept to select a gain profile $\mathbf{K}^*$ for the whole network achieving the synchronization. Similarly to the Section 2.2-2.3, we first define the strategic game and perform a change of variables. Then, we provide an analytical result deriving from the LQR control.

### Nash Equilibrium approach

Based on the Chapter 1, the strategy game in standard form related to the cost (2.3) is defined by the triplet as follows

$$\mathcal{G} = \left(\mathcal{V}, \{\mathcal{K}_i\}_{i\in\mathcal{V}}, \left\{J_i^{NE}\right\}_{i\in\mathcal{V}}\right), \tag{2.22}$$

where $\mathcal{V} = \{1, ..., N\}$ is the set of players, $\mathcal{K}_i = \mathbb{R}^{n_u \times n_x}$ represents to the set of gains $K_i$ applied by player $i$ and $J_i^{NE}$ corresponds to (2.3). For $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times ... \times \mathcal{K}_N$, we say that the profile $\mathbf{K}^* \in \mathcal{K}$ is a Nash equilibrium to the game (2.22) if

$$\forall i \in \mathcal{V}, \forall K_i \in \mathcal{K}_i,\ J_i^{NE}(K_i^*, K_{-i}^*) \leq J_i^{NE}(K_i, K_{-i}^*). \tag{2.23}$$

For this game, we recall that if a player unilaterally plays an action different from the Nash one while the remaining players maintain their Nash actions, then the player in question will inevitably increase his cost it aims to reduce. The players don't want anymore to satisfy their individual constraints but they desire to optimize their own costs.

In an all-to-all connection graph, each agent is aware of all the other agents and due to the symmetry in the communication structure, their actions might be similar. Thus, we assume that one possible equilibrium is when all the players apply the same best response, Definition 13.

***Assumption* 2.** Each agent applies a strategy $K^*$ such that it is the best response to all the other agents also applying $K^*$.

In this case, the Nash equilibrium will be the same for every agents and will correspond to the optimal gain.

**Change of variables and optimal control design**

On the purpose of studying the synchronization, we introduce an error variable $\delta_i$ corresponding to each agent $i$ as

$$\delta_i(t) = \sum_{j\in\mathcal{N}_i^{int}} x_i(t) - x_j(t), \quad \forall i \in \mathcal{V}. \tag{2.24}$$

Then, the error dynamics derived from (2.1)-(2.24) is described by

$$\dot{\delta}_i(t) = A\delta_i(t) + (N-1)Bu_i(t) - \sum_{j\in\mathcal{N}_i^{int}} Bu_j(t). \tag{2.25}$$

Under Assumption 2 and due to the complete topology, the last term (2.25) can be written as

$$\sum_{j\in\mathcal{N}_i^{int}} Bu_j(t) = BK^* \sum_{j\in\mathcal{N}_i^{int}} x_i(t) - x_j(t) = -u_i(t). \tag{2.26}$$

Therefore, the error dynamics (2.25) takes the following form,

$$\dot{\delta}_i(t) = A\delta_i(t) + NBu_i(t). \tag{2.27}$$

Now, we recast the cost (2.3) into the error variable $\delta_i$ as follows

$$J_i^{NE} = \int_{t_0}^{+\infty} \delta_i^\top(t)\delta_i(t) + u_i^\top(t)Ru_i(t)\,\mathrm{dt}, \qquad \forall i \in \mathcal{V}, \tag{2.28}$$

where $R$ is a positive definite matrix.

***Proposition* 5.** *Consider any complete graph of $N$ agents and assume that the pair $(A, NB)$ is stabilizable. If there exists an equilibrium such that Assumption 2 holds, then the gain expression is given by*

$$K^* = NR^{-1}B^\top P^*, \tag{2.29}$$

*where the positive definite matrix $P^*$ is the solution of the Algebraic Riccati Equation (ARE)*

$$P^*A + A^\top P^* - N^2P^*BR^{-1}B^\top P^* + \mathbf{I}_{n_x} = 0. \tag{2.30}$$

*Furthermore, the feedback control with the gain (2.29) stabilizes the error system (2.27) while optimizing the NE cost (2.28).*

Although the requirement of a complete graph is restrictive, Proposition 5 has the advantage of finding an NE by just solving a Riccati equation, and it is computationally efficient even for large graphs. Thus, this motivates us to consider clustered networks where the control inside clusters is designed by using Riccati equations while the control between clusters are based on the results from Section 2.3.

## 2.5 Numerical illustrations

In this section, we provide some numerical examples to illustrate the effectiveness of the algorithm proposed in this manuscript. Without any loss of generality, we consider only the following simple agent dynamics (2.1) with $A = 1$, $B = 1$ and $R_i = 1$ for all $i$. We use $K_i^*$ and $J_i^{SE*}$ for the control gains and the corresponding costs obtained using the proposed strategy. On the other hand, $K_i^o$ and $J_i^{SEo}$ are the control gains and the corresponding costs obtained using the strategy in [Rejeb et al., 2018]. For both graphs, we provide the a posteriori values of cost functions obtained by implementing the controller gains profile $\mathbf{K}$. The costs are computed by using the initial condition $x_0$ and the state of the network. Through the second graph, we show that the constraints are still satisfied independently of the initial conditions.

### 2.5.1 Ring directed graph

Let us first consider a simple ring directed graph with 5 agents described by the following Laplacian matrix

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.31}$$

For the simulation, we use $\gamma = 1.3$ and the results of the proposed algorithm are summarized in the table below.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $x_i(0)$ | 0.67 | 0.67 | 0.31 | 0.01 | 0.02 |
| $K_i^*$ | 3.54 | 3.54 | 3.52 | 3.51 | 3.60 |
| $J_i^{SE*}$ | 0.46 | 0.64 | 0.33 | 0.51 | 0.95 |

We note that, due to the regularity of the graph the five gains are quite similar and the costs are bounded by $\gamma$. The synchronizing trajectories of the 5 agents are plotted in Fig. 2.1 and the corresponding control inputs are in Fig. 2.2.

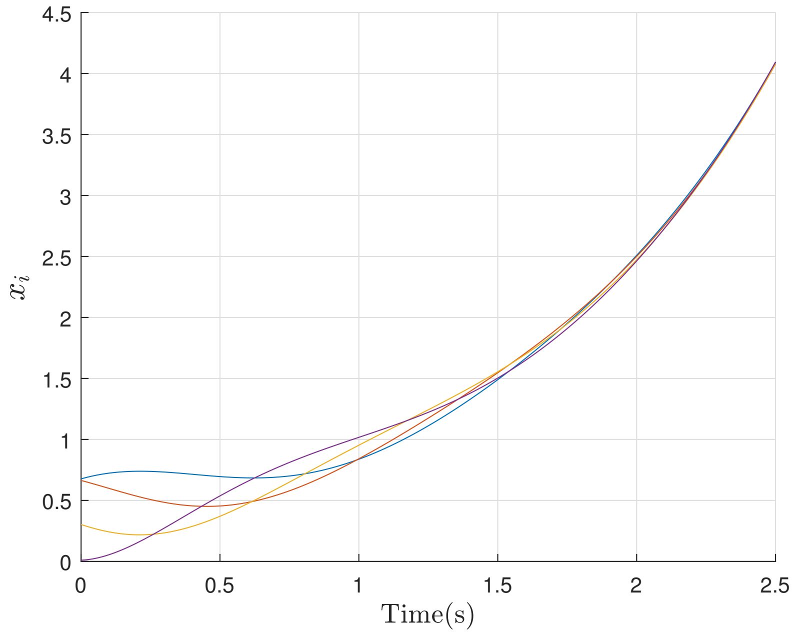


FIGURE 2.1 – Trajectories of the agents for the ring directed graph given by (2.31).

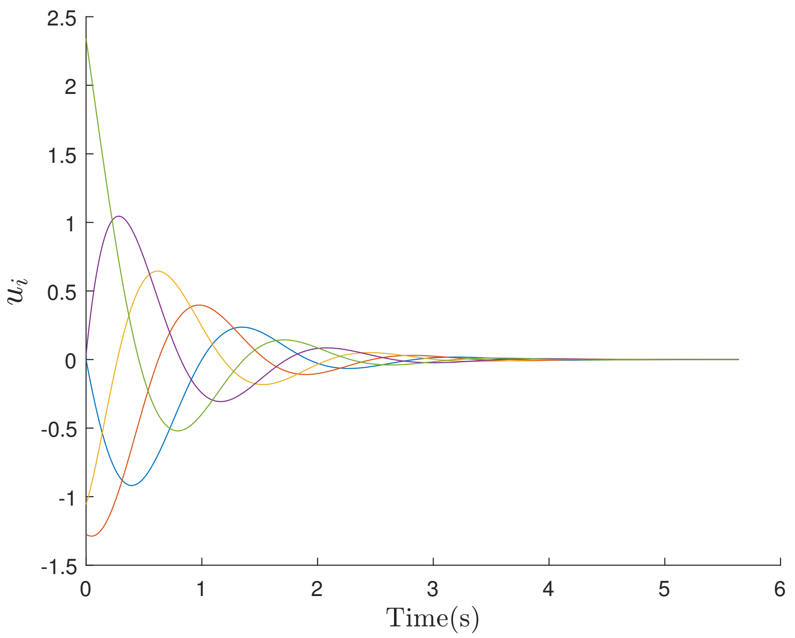


FIGURE 2.2 – Controls $u_i$ for the ring directed graph given by (2.31).

### 2.5.2 Undirected graph : comparison with [Rejeb et al., 2018]

In the following we consider an undirected graph with 8 agents in order to compare our design strategy with the one proposed in [Rejeb et al., 2018]. It is noteworthy that [Rejeb et al., 2018] proposes a decentralized control achieving synchronization with global performance guarantees. The control is applied to singularly perturbed multi-agent systems interacting over networks represented by undirected fixed graphs. It is imposed that all agents applies the same control gain. Moreover, the synchronizing control design is reduced to a stabilizing control design for an uncertain system with bounded uncertainties. These uncertainties are related to the bounds on the maximum and minimum (non-zero) eigenvalues of the Laplacian. Solving a Riccati equation, one finds a common gain for all agents ensuring that the global cost is upper-bounded.

In order to highlight the improvements that we can obtain by using the approach proposed, in this simulation, we consider a graph in which the agent centralities are very different. For the example above, the controller gains are almost similar meaning that the strategy proposed in [Rejeb et al., 2018] might provide good results. However if the undirected graph $G$ is associated to the following Laplacian

$$L = \begin{pmatrix} 7 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 2 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 2 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 2 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 7 \end{pmatrix}, \tag{2.32}$$

one has no reason to apply the same controller gain to all the agents. For the simulation, we use $\gamma = 0.6$ and the results of both strategies are summarized in the table below. We test 3 different initial conditions by varying $x_0$. We use $\mathbf{K}^* = (0.39, 1.30, 1.30, 1.30, 1.30, 1.30, 1.30, 0.39)$ found using Algorithm 1, and $\mathbf{K}^o = (3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09)$ based on the results in [Rejeb et al., 2018].

| $\gamma = 0.6$ | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| Case 1 | $x_i(0)$ | 0.50 | 0.1 | 0.4 | 0.1 | 0.4 | 0.4 | 0.1 | 0.04 |
| | $J_i^{SE*}$ | 0.41 | 0.13 | 0.10 | 0.13 | 0.10 | 0.10 | 0.13 | 0.35 |
| | $J_i^{SEo}$ | 1.17 | 0.14 | 0.11 | 0.14 | 0.11 | 0.11 | 0.14 | 0.88 |
| Case 2 | $x_i(0)$ | 0.04 | 0.53 | 0.04 | 0.46 | 0.04 | 0.46 | 0.04 | 0.53 |
| | $J_i^{SE*}$ | 0.35 | 0.21 | 0.18 | 0.12 | 0.18 | 0.12 | 0.18 | 0.40 |
| | $J_i^{SEo}$ | 0.80 | 0.27 | 0.21 | 0.15 | 0.21 | 0.15 | 0.21 | 1.05 |
| Case 3 | $x_i(0)$ | 0.44 | 0.47 | 0.03 | 0.07 | 0.02 | 0.54 | 0.06 | 0.54 |
| | $J_i^{SE*}$ | 0.16 | 0.05 | 0.33 | 0.26 | 0.35 | 0.10 | 0.28 | 0.29 |
| | $J_i^{SEo}$ | 0.45 | 0.15 | 0.34 | 0.26 | 0.36 | 0.26 | 0.28 | 1.10 |

TABLE 2.1 – Undirected graph : Individual cost $J_i^{SE}$ for different initial conditions, $\gamma = 0.6$

In Table 2.1, $J_i^{SE*}$ and $J_i^{SEo}$ are respectively the costs incurred by agent $i$ using our strategy and the one in [Rejeb et al., 2018]. Since [Rejeb et al., 2018] bounds a global cost, we can compare their global cost to a total cost evaluated as $n\gamma$. The guaranteed overall bounds are $n\gamma = 4.8$, for our strategy and $10.8$ for the one in [Rejeb et al., 2018]. As seen from the table, each of individual costs are bounded by $\gamma$ for several possible initial conditions.

## 2.6 Conclusion

We study the problem of static-output feedback synchronization in a multi-agent system, which guarantees individual performance bounds. This problem is modeled as a satisfaction game and we seek gains that are in satisfaction equilibrium, i.e. the cost associated to each agent is upper-bounded by a given $\gamma$. In this context, we provide conditions in the form of LMIs which can verify if a given set of gains are in satisfaction equilibrium. We provide a method to generate the gain for a certain agent when the gains for the other agents are known and this is used in an iterative algorithm which can synthesize a satisfaction equilibrium. Numerical examples illustrate our algorithm and compare our results with a previous result found in the literature.

Unfortunately, solving the LMIs using SeDuMi yields a complexity of order $O(2N.N^{4.5})$. Even though the complexity of the algorithm varies with the choice of the solver, it is mainly scaled with the number of agents $N$. Thus, computing the gains via the general algorithm may be computationally intractable in some cases. For this reason, we aim to provide a computational efficient control algorithm for large-scale networks in the next chapter.

Chapter 3

# Distributed composite control for clustered networks

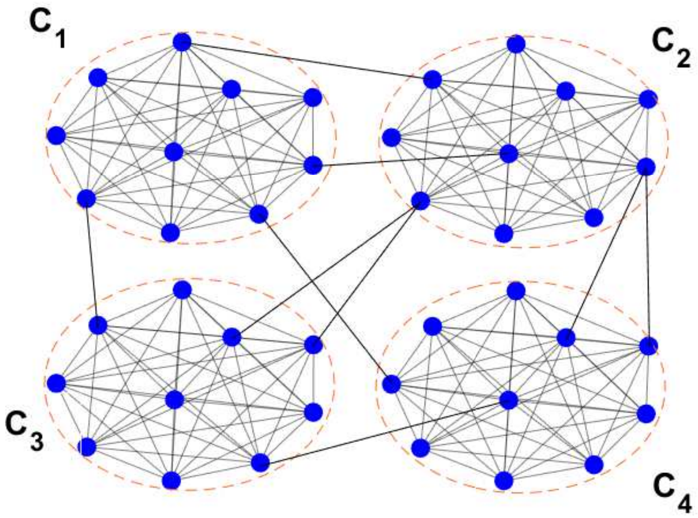


FIGURE 3.1 – A network partitioned into 4 clusters represented by the circles. The blue points corresponds to the agents while the black lines are the links between them.

This chapter considers clustered networks, in which connections inside the cluster are dense and between clusters are sparse [Holland and Leinhardt, 1971; Watts and Strogatz, 1998]. We address the problem of distributed composite control design by exploiting the network structure. The objective is to provide a computationally efficient method to design control strategies. At the same time, we also guarantee that the cost of each cluster is bounded by a given threshold when applying the proposed control law. By separating the control design and the cost optimization to the cluster-level, the approach aims to significantly reduce the problem's complexity and the computational effort necessary to obtain the controller.

In a clustered network, the interconnections are dense inside the clusters and sparse between them. This results in a fast convergence inside the cluster towards a local agreement and then slowly towards the global consensus. The approach relies on this network property to divide the control design problem into computationally tractable subproblems.

Provided that connections are much denser inside clusters than between clusters, we show that the network exhibits a slow (inter-cluster) and a fast (intra-cluster) dynamics that can be decoupled, via *Time-Scale Separation* (*TSS*) techniques. The fast variables represent the synchronization error inside the clusters whereas the slow variables represent the aggregate behavior of the agent states within each cluster. The long term behavior of the network depends on this slow dynamics.

Decoupling these dynamics, we can approximate the behavior of the original system while independently designing the controllers for the intra and inter clusters synchronization. For each agent present in the network, we define a composite control as a sum of the internal and external control. The two controls are designed independently at the cluster-level and employ a simplified model that significantly reduces the computational load and the control design complexity. The internal control, related to the fast dynamics, achieves the local consensus inside the clusters while minimizing a local cost. As the connections are dense inside the clusters, the internal control design is described analytically assuming that clusters are characterized by all-to-all (complete topology) connections. This assumption is only made for the design purpose and it is not required to be satisfied in practice. As for the external control, it synchronizes all the clusters employing a satisfaction equilibrium technique in Chapter 2. The only requirement for the global synchronization is the connectivity of the graph representing the inter-cluster network.

Finally, applying the results from *Singular Perturbation Theory (SPT)* [Kokotović et al., 1999], we show that the closed-loop response due to the composite control is close to that of the approximate models. In addition to the distributed control design, we also provide an approximation of the cluster cost after a short period of time required to synchronize the agents inside clusters.

The chapter is organized as follows. The model with the objectives are stated in Section II. The time-scale modeling is described in detail in Section III. Then, the internal and external design procedures are developed in Section IV. In Section V, we provide an approximation of the cluster cost. Finally, numerical results are presented in Section VI before concluding in Section VII. To make the chapter easily readable, some proofs are included in the Appendix.

## 3.1 Problem Statement

### 3.1.1 System Model

Consider a network of $N$ agents, where the interactions are described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, 2, ..., N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. We assume the network to be partitioned into $m$ non-empty clusters $\mathcal{C}_1, \ldots, \mathcal{C}_m \subset \mathcal{V}$. Let us denote by $\mathcal{M} = \{1, 2, \ldots, m\}$, the set of clusters while $n_k$ represents the cardinality of the cluster $\mathcal{C}_k$ such that $N = \sum_{k=1}^{m} n_k$.

Each agent in the network is identified by a couple $(k, i) \in \mathcal{C}_k$, where $k \in \mathcal{M}$ refers to the cluster $\mathcal{C}_k$ and $i = 1, ..., n_k$ the index of the agent. The state dynamics $x_{k,i}$ of an agent $(k, i) \in \mathcal{C}_k$ is given by

$$\dot{x}_{k,i}(t) = Ax_{k,i}(t) + Bu_{k,i}(t), \tag{3.1}$$

where $x_{k,i} \in \mathbb{R}^{n_x}$, $u_{k,i} \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$.

For each cluster $\mathcal{C}_k$, let $x_k(t) = (x_{k,1}(t), \ldots, x_{k,n_k}(t)) \in \mathbb{R}^{n_k \cdot n_x}$ denote the vector collecting the states of the cluster and $u_k(t) = (u_{k,1}(t), \ldots, u_{k,n_k}(t)) \in \mathbb{R}^{n_k \cdot n_u}$ the cluster control. Thus, the cluster dynamics takes the following form

$$\dot{x}_k(t) = (\mathbf{I}_{n_k} \otimes A)x_k(t) + (\mathbf{I}_{n_k} \otimes B)u_k(t), \quad \forall k \in \mathcal{M}. \tag{3.2}$$

The agents in the network may have connections with other agents in the same or different clusters. Throughout the Chapter 3, an *internal* connection is any connection between two agents from the same cluster, and an *external* connection refers to any connection between two agents from different clusters.

### 3.1.2 Objective

The main objective is to design a distributed control synchronizing the network with a sub-optimal cost. The network is said to be asymptotically synchronized when $\lim\limits_{t\to+\infty} \|x_{k,i}(t) - x_{l,j}(t)\| = 0$, for all $(k,i) \in \mathcal{C}_k$, $(l,j) \in \mathcal{C}_l$ and $k, l \in \mathcal{M}$.

Pertaining to our control design, we propose an individual composite control for an agent $(k,i) \in \mathcal{C}_k$ and $k \in \mathcal{M}$, as

$$u_{k,i}(t) = u_{k,i}^{int}(t) + u_{k,i}^{ext}(t), \tag{3.3}$$

where

$$\begin{cases} u_{k,i}^{int}(t) = -K_k^{int} \sum\limits_{(k,j)\in\mathcal{N}_{k,i}} x_{k,i}(t) - x_{k,j}(t), \\ u_{k,i}^{ext}(t) = -K_k^{ext} \sum\limits_{(l,p)\in\mathcal{N}_{k,i}} x_{k,i}(t) - x_{l,p}(t). \end{cases} \tag{3.4}$$

In other words, we apply a common internal gain for all agent's to synchronize with other agents in the same cluster and a common external gain to synchronize with external agents. The notation $(k,j) \in \mathcal{N}_{k,i}$ represents the neighbors of the agent $(k,i)$ in the same cluster $\mathcal{C}_k$ whereas $(l,p) \in \mathcal{N}_{k,i}$ indicates the neighbors belonging to a different cluster, with $(l,p) \neq (k,i)$. Thus, the composite cluster control is

$$u_k(t) = u_k^{int}(t) + u_k^{ext}(t), \quad \forall k \in \mathcal{M}, \tag{3.5}$$

where $u_k^{int}(t) = (u_{k,1}^{int}(t), ..., u_{k,n_k}^{int}(t))$ and $u_k^{ext}(t) = (u_{k,1}^{ext}(t), ..., u_{k,n_k}^{ext}(t))$ correspond to the internal and external cluster control, respectively.

The cluster cost $J_k$ associated with the cluster $\mathcal{C}_k$, for $k \in \mathcal{M}$, is defined as

$$J_k = \int_0^{+\infty} x_k^\top(t)(\mathcal{L}_k^{int} \otimes \mathbf{I}_{n_x})x_k(t) + x^\top(t)(\mathcal{L}_k^{ext} \otimes \mathbf{I}_{n_x})x(t) + u_k^\top(t)R_k u_k(t)\,\mathrm{d}t, \tag{3.6}$$

where the internal Laplacian $\mathcal{L}_k^{int} \in \mathbb{R}^{n_k \times n_k}$ captures the connections inside $\mathcal{C}_k$, and the external Laplacian $\mathcal{L}_k^{ext} \in \mathbb{R}^{N \times N}$ expresses the external connections between $\mathcal{C}_k$ and the neighboring clusters.

***Remark* 7.** The following vector $x_k = (x_{k,1}, \dots, x_{k,n_k}) \in \mathbb{R}^{n_k \cdot n_x}$ representing the states of the cluster $\mathcal{C}_k$ should not be confused with the previous form $x_i \in \mathbb{R}^{n_x}$ referring to the state of the agent $i \in \mathcal{V}$, in Chapter 2. The same applies to the various controls. The cost $J_k$ is now a *group* cost and is assimilated to the cluster $\mathcal{C}_k$, unlike the individual cost $J_i$ associated with the agent $i$.

By substituting the individual control (3.4) into (3.6), we recast the cost (3.6) as a sum of the internal and external cost, and a composite term as

$$\begin{aligned} J_k = &\underbrace{\int_0^{+\infty} x_k^\top(t)(\mathcal{L}_k^{int} \otimes \mathbf{I}_{n_x})x_k(t) + {u_k^{int}}^\top(t)(\mathbf{I}_{n_k} \otimes R_k)u_k^{int}(t)\,\mathrm{d}t}_{J_k^{int}} \\ &+ \underbrace{\int_0^{+\infty} x^\top(t)(\mathcal{L}_k^{ext} \otimes \mathbf{I}_{n_x})x(t) + {u_k^{ext}}^\top(t)(\mathbf{I}_{n_k} \otimes R_k)u_k^{ext}(t)\,\mathrm{d}t}_{J_k^{ext}} \\ &+ \underbrace{2\int_0^{+\infty} {u_k^{ext}}^\top(t)(\mathbf{I}_{n_k} \otimes R_k)u_k^{int}(t)\,\mathrm{d}t}_{J_k^{cross}}. \end{aligned} \tag{3.7}$$

The internal control $u_k^{int}$ represents the effort required to the local agreement, whereas the external control $u_k^{ext}$ is the energy necessary to synchronize the agents between the clusters.

In order to remove the crossed term $J_k^{cross}$ and to reduce the optimization of (3.6) to the optimization of the internal and external cost only, we consider the following inequality

$$J_k \leq 2(J_k^{int} + J_k^{ext}). \tag{3.8}$$

Thus, the main goal is to decouple the control design (3.5) such that the internal control design is related to the minimization of the internal cost $J_k^{int}$, and the external control is designed by imposing a prescribed satisfactory level on the external cost $J_k^{ext}$. Additionally, we prove that the cluster cost $J_k(T, +\infty)$ is approximated only by external cost $J_k^{ext}(T, +\infty)$, where $T > 0$ is a finite time after which the agents inside the clusters are synchronized. The notation $J_k(T, +\infty)$ stands for the cost evaluated during the time interval $[T, +\infty]$.

## 3.2 Time-Scale Modeling

In this section, we provide a procedure to decouple the closed-loop network dynamics into two subsystems, evolving on different time scales. First, we perform a coordinate transformation to exhibit the collective dynamics of the network : the average and the synchronization error dynamics. Then, we apply the TSS techniques to decouple the collective dynamics into slow and fast subsystems. In two time-scale, the slow variable corresponds to the average while the fast variable to the synchronization error.

For further analysis, let us denote the Laplacian of the network by $\mathcal{L} \in \mathbb{R}^{N\times N}$ such that $\mathcal{L} = \mathcal{L}^{int} + \mathcal{L}^{ext}$, [Chow and Kokotović, 1985]. The internal Laplacian of the network is defined as $\mathcal{L}^{int} = \mathrm{diag}(\mathcal{L}_1^{int}, ..., \mathcal{L}_m^{int})$ and the external Laplacian of network $\mathcal{L}^{ext}$ corresponds to the connections between agents from different clusters.

### 3.2.1 Coordinate transformation

Following [Panteley and Loria, 2017], [Adhikari et al., 2021], we introduce the coordinate transformation for the cluster $\mathcal{C}_k$ as

$$\overline{x}_k(t) = (T_k^\top \otimes \mathbf{I}_{n_x})x_k(t), \quad \forall k \in \mathcal{M}, \tag{3.9}$$

where the orthonormal matrix $T_k$, i.e. $T_k^\top T_k = T_k T_k^\top = \mathbf{I}_{n_k}$, is obtained by a Jordan decomposition of the symmetric Laplacian $\mathcal{L}_k^{int}$. Referring to the algebraic properties of the Laplacian matrix in Chapter 1, it satisfies

$$\mathcal{L}_k^{int} = T_k \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_k^{int} \end{bmatrix} T_k^\top, \quad \forall k \in \mathcal{M}, \tag{3.10}$$

where $\Lambda_k^{int} = \mathrm{diag}(\lambda_{k,2}^{int}, \dots, \lambda_{k,n_k}^{int}) \in \mathbb{R}^{(n_k-1)\times(n_k-1)}$ collects the $n_k - 1$ eigenvalues of $\mathcal{L}_k^{int}$. Moreover, the matrix $T_k$ is expressed as

$$T_k = \begin{bmatrix} v_{k,1} & V_k \end{bmatrix}, \quad \forall k \in \mathcal{M}, \tag{3.11}$$

where $v_{k,1}^\top = \frac{1}{\sqrt{n_k}}\mathbb{1}_{n_k}^\top$ is the eigenvector associated with the 0 eigenvalue and the matrix $V_k \in \mathbb{R}^{n_k\times(n_k-1)}$ is constituted of the eigenvectors corresponding to the nonzero eigenvalues of $\mathcal{L}_k^{int}$. Furthermore, it can be verified that,

$$v_{k,1}^\top V_k = 0 \quad \text{and} \quad V_k^\top V_k = \mathbf{I}_{n_k-1}.$$

According to (3.9) and (3.11), the collective state of each cluster is

$$\overline{x}_k(t) = \begin{bmatrix} y_k(t) \\ \xi_k(t) \end{bmatrix} := \begin{bmatrix} (v_{k,1}^\top \otimes \mathbf{I}_{n_x})\, x_k(t) \\ (V_k^\top \otimes \mathbf{I}_{n_x}) x_k(t) \end{bmatrix} = \begin{bmatrix} H_k^\top x_k(t) \\ Z_k^\top x_k(t) \end{bmatrix}, \quad \forall k \in \mathcal{M} \tag{3.12}$$

where $H_k^\top = (v_{k,1}^\top \otimes \mathbf{I}_{n_x})$ and $Z_k^\top = (V_k^\top \otimes \mathbf{I}_{n_x})$. The first component $\frac{y_k}{\sqrt{n_k}}$ is regarded as an average of the respective agents' states in the cluster $\mathcal{C}_k$. As for the second component, $\xi_k(t) = (\xi_{k,1}(t), \ldots, \xi_{k,n_k}(t)) \in \mathbb{R}^{(n_k-1).n_x}$ is the projection of the synchronization error,

$$e_k(t) = x_k(t) - (v_{k,1} \otimes \mathbf{I}_{n_x}) y_k(t), \tag{3.13}$$

onto the subspace orthogonal to $v_{k,1}$, i.e. $e_k(t) = Z_k \xi_k(t)$.

Thanks to (3.12) and (3.13), the vector $x_k$ is expressed in terms of $y_k$ and $\xi_k$ as follows,

$$x_k(t) = H_k y_k(t) + Z_k \xi_k(t), \quad \forall k \in \mathcal{M}. \tag{3.14}$$

Finally, using the transformation (3.9) on the overall network, the compact form is

$$\begin{cases} y(t) = H^\top x(t) \quad \text{and} \quad \xi(t) = Z^\top x(t), \\ x(t) = H y(t) + Z\xi(t), \end{cases} \tag{3.15}$$

where $x(t) = (x_1(t), \ldots, x_m(t)) \in \mathbb{R}^{N.n_x}$ is the vector collecting the states, $y(t) = (y_1(t), \ldots, y_m(t)) \in \mathbb{R}^{m.n_x}$ represents the scaled average and $\xi(t) = (\xi_1(t), \ldots, \xi_m(t)) \in \mathbb{R}^{(N-m).n_x}$ the synchronization error. We also denote $H = \mathrm{diag}(H_1, ..., H_m)$ and $Z = \mathrm{diag}(Z_1, ..., Z_m)$.

### 3.2.2 Collective dynamics

Gathering the cluster dynamics (3.2) results in the network dynamics as follows

$$\dot{x}(t) = (\mathbf{I}_N \otimes A) x(t) + (\mathbf{I}_N \otimes B) u(t), \tag{3.16}$$

where $u(t) = (u_1(t), \ldots, u_m(t)) \in \mathbb{R}^{N.n_u}$. Substituting the control (3.5) into (3.16), it yields the closed-loop dynamics

$$\dot{x}(t) = \left[ (\mathbf{I}_N \otimes A) - (\mathbf{I}_N \otimes B) K^{int} (\mathcal{L}^{int} \otimes \mathbf{I}_{n_x}) - (\mathbf{I}_N \otimes B) K^{ext} (\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x}) \right] x(t), \tag{3.17}$$

where

$$\begin{cases} K^{int} = \mathrm{diag}((\mathbf{I}_{n_1} \otimes K_1^{int}), \ldots, (\mathbf{I}_{n_m} \otimes K_m^{int})), \\ K^{ext} = \mathrm{diag}((\mathbf{I}_{n_1} \otimes K_1^{ext}), \ldots, (\mathbf{I}_{n_m} \otimes K_m^{ext})). \end{cases} \tag{3.18}$$

Finally, using (3.15) and (3.17), the collective dynamics is recasted as follows

$$\begin{cases} \dot{y}(t) = \bar{A}_{11}y(t) + \bar{A}_{12}\xi(t), \\ \dot{\xi}(t) = \bar{A}_{21}y(t) + (\bar{A}_{22}^1 + \bar{A}_{22}^2)\xi(t), \end{cases} \tag{3.19}$$

where

$$\begin{cases} \bar{A}_{11} = (\mathbf{I}_m \otimes A) - H^T(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes I_{n_x})H, \\ \bar{A}_{12} = -H^\top(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})Z, \\ \bar{A}_{21} = -Z^\top(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})H, \\ \bar{A}_{22}^1 = -Z^\top(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})Z, \\ \bar{A}_{22}^2 = (\mathbf{I}_{N-m} \otimes A) - (\mathbf{I}_{N-m} \otimes B)K_{N-m}^{int}(\Lambda^{int} \otimes \mathbf{I}_{n_x}), \\ K_{N-m}^{int} = \mathrm{diag}((\mathbf{I}_{n_1-1} \otimes K_1^{int}), \dots, (\mathbf{I}_{n_m-1} \otimes K_m^{int})), \\ \Lambda^{int} = \mathrm{diag}(\Lambda_1^{int}, \dots, \Lambda_m^{int}). \end{cases} \tag{3.20}$$

### 3.2.3 Time-Scale Separation

To study the time-scale behavior and analyze the synchronizing behavior, we define the network parameters as follows

$$\begin{cases} \mu^{ext} = \|(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})\|, \\ \mu^{int} = \min_{k \in \mathcal{M}} \|(\Lambda_k^{int} \otimes BK_k^{int})\|, \\ \epsilon = \dfrac{\mu^{ext}}{\mu^{int}}. \end{cases} \tag{3.21}$$

The network parameter $\epsilon$ is the ratio of the strength of the controls between and within the clusters. A small $\epsilon$ means that the dense connections inside the cluster coupled with the internal control is stronger that the sparse connections between the clusters coupled with the external control. The role of the controllers is not to change the strength of the communications between and within the clusters. In term of structure, a small $\epsilon$ is equivalent to a sparse connections between the clusters and a dense connections within them. This ratio should be small enough compare to $1$ for the two time-scale separation to occur, it is a standard assumption in SPT. The norm $\|.\|$ in (3.21) corresponds the 2-norm for a matrix.

In the absence of agents' internal dynamics, the authors in [Chow and Kokotović, 1985], [Martin et al., 2016], are able to express the closed-loop dynamics (3.17) into a *Standard Singular Perturbation Form* (*SSPF*). Upon analyzing the orders of the state matrices of the collective dynamics (3.19), it appears that the closed-loop network dynamics cannot be expressed in SSPF without the knowledge of the state matrix $A$.

***Assumption* 3.** The state matrix $A$ satisfies the following

$$\|A\| = \mathcal{O}(\mu^{ext}). \tag{3.22}$$

The assumption on the order of the matrix $A$ is necessary for representing the dynamics (3.19) in SSPF. However, we note that since $\mu^{ext}$ depends on $K^{ext}$, we can always choose $K^{ext}$ sufficiently large such that the assumption 3 is satisfied. In the following lemma, we analyze the order of the matrices in equation

***Lemma* 6.** *Under Assumption 3, the matrices in (3.20) satisfy the following conditions,*

- $\|\bar{A}_{11}\|$, $\|\bar{A}_{12}\|$, $\|\bar{A}_{21}\|$, $\|\bar{A}^1_{22}\|$ *are of order* $\mathcal{O}(\epsilon\mu^{int})$
- $\|\bar{A}^2_{22}\|$ *is of order* $\mathcal{O}(\mu^{int})$

***Proof***: From [Laub, 2005], we know that $\|(A \otimes B)\| = \|A\|.\|B\|$ for any matrix $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$. In addition, we have $\|H\| = \|H^\top\| = 1$ and $\|Z\| = \|Z^\top\| = 1$.

We only prove the order for the matrices $\bar{A}_{11}$, $\bar{A}_{12}$ and $\bar{A}^2_{22}$. From the Assumption 3, there exists a strictly positive constant $c_1 \in \mathbb{R}$ such that $\|A\| = c_1\mu^{ext}$. It follows that,

$$\begin{aligned} \|\bar{A}_{11}\| &= \|(\mathbf{I}_m \otimes A) - H^T(\mathbf{I}_n \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})H\| \\ &\leq \|A\| + \|H^T\|.\|(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})\|.\|H\| \\ &= (c_1+1)\mu^{ext} \\ &= (c_1+1)\epsilon\mu^{int}. \end{aligned} \tag{3.23}$$

The bounds of $\bar{A}_{12}$, $\bar{A}_{21}$ and $\bar{A}^1_{22}$ are derived similarly. That is why we only prove for $\bar{A}_{12}$,

$$\begin{aligned} \|\bar{A}_{12}\| &= \|H^\top(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})Z\| \\ &= \|H^\top\|.\|(\mathbf{I}_N \otimes B)K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})\|.\|Z\| \\ &\leq \mu^{ext} = \epsilon\mu^{int}. \end{aligned} \tag{3.24}$$

Then, we lower-bound the matrix $\bar{A}^2_{22}$ such that

$$\begin{aligned} \|\bar{A}^2_{22}\| &= \|(\mathbf{I}_{N-m} \otimes A) - (\mathbf{I}_{N-m} \otimes B)K^{int}_{n-m}(\Lambda^{int} \otimes \mathbf{I}_{n_x})\| \\ &\geq |\|A\| - \|(\mathbf{I}_{N-m} \otimes B)K^{int}_{N-m}(\Lambda^{int} \otimes \mathbf{I}_{n_x})\||. \end{aligned} \tag{3.25}$$

From (3.21)-(3.22), we understand that the second norm in (3.25) is much larger than the first one. Thus, by taking the difference between the largest value of the first term and a value less than $\|(\mathbf{I}_{N-m} \otimes B)K^{int}_{N-m}(\Lambda^{int} \otimes \mathbf{I}_{n_x})\|_2$, it yields a lower-bound as

$$\|\bar{A}^2_{22}\| \geq |c_1\epsilon\mu^{int} - \mu^{int}| = |1 - c_1\epsilon|\mu^{int}, \tag{3.26}$$

where $\mu^{int} = \min_{k \in \mathcal{M}} \|(\Lambda_k^{int} \otimes BK_k^{int})\|_2$.

Finally, through the bounds (3.23), (3.24) and (3.26), we conclude the proof. ■

As the consequence of the Assumption 3, all the matrices in (3.20) are of order $\mathcal{O}(\epsilon\mu^{int})$ except $A_{22}^2$, which is of order $\mathcal{O}(\mu^{int})$. Furthermore, due the standard assumption in SPT the network parameter $\epsilon$ is small, or equivalently we have $\mu^{ext} \ll \mu^{int}$. In other words, the dynamics of the aggregate state $y$ is much smaller than the dynamics of the synchronization error $\xi$. For this reason, the variable $y$ behaves as a slow variable and the variable $\xi$ behaves as a fast variable.

Thereafter, to reveal the evolution of the system on different time-scale, we define a fast time-scale $t_f = \mu^{int}t$ and a slow time-scale $t_s = \epsilon t_f$. The matrices $\bar{A}_{ij}$ in (3.20) are also re-scaled as follows,

$$\begin{cases} A_{11} = \dfrac{\bar{A}_{11}}{\epsilon\mu^{int}}, \qquad A_{12} = \dfrac{\bar{A}_{12}}{\epsilon\mu^{int}}, \qquad A_{21} = \dfrac{\bar{A}_{21}}{\epsilon\mu^{int}}, \\ A_{22}^1 = \dfrac{\bar{A}_{22}^1}{\epsilon\mu^{int}}, \qquad A_{22}^2 = \dfrac{\bar{A}_{22}^2}{\mu^{int}}. \end{cases} \tag{3.27}$$

### 3.2.4 Fast Dynamics

Performing the time re-scale $t_f = \mu^{int}t$, we obtain the fast dynamics as follows,

$$\begin{cases} \dfrac{d\hat{y}}{dt_f}(t_f) = \epsilon A_{11}\hat{y}(t_f) + \epsilon A_{12}\hat{\xi}(t_f), \\ \dfrac{d\hat{\xi}}{dt_f}(t_f) = \epsilon A_{21}\hat{y}(t_f) + (\epsilon A_{22}^1 + A_{22}^2)\hat{\xi}(t_f). \end{cases} \tag{3.28}$$

Then, setting $\epsilon = 0$, we have $\frac{d\hat{y}}{dt_f} = 0$ implying $\hat{y}$ is constant and the decoupled fast dynamics is

$$\frac{d\hat{\xi}_f}{dt_f}(t_f) = A_{22}^2\hat{\xi}_f(t_f). \tag{3.29}$$

The fast dynamics (3.29) in original time-scale $t$ is

$$\dot{\xi}_f(t) = (\mathbf{I}_{N-m} \otimes A)\xi_f(t) + (\mathbf{I}_{N-m} \otimes B)u_f(t), \tag{3.30}$$

where $u_f(t) = -K_{N-m}^{int}(\Lambda^{int} \otimes \mathbf{I}_{n_x})\xi_f(t)$.

### 3.2.5 Slow Dynamics

The collective dynamics in slow time-scale $t_s = \epsilon t_f$ is

$$\begin{cases} \dfrac{d\tilde{y}}{dt_s}(t_s) = A_{11}\tilde{y}(t_s) + A_{12}\tilde{\xi}(t_s), \\ \epsilon\dfrac{d\tilde{\xi}}{dt_s}(t_s) = \epsilon A_{21}\tilde{y}(t_s) + (\epsilon A_{22}^1 + A_{22}^2)\tilde{\xi}(t_s). \end{cases} \tag{3.31}$$

Setting $\epsilon = 0$ in (3.31) yields $\tilde{\xi}_s(t_s) = 0$ and the decoupled slow dynamics is,

$$\frac{d\tilde{y}_s}{dt_s}(t_s) = A_{11}\tilde{y}_s(t_s). \tag{3.32}$$

Since we have $t_s = \epsilon t_f = \epsilon\mu^{int}t$, the slow dynamics is

$$\dot{y}_s(t) =(\mathbf{I}_m \otimes A)y_s(t) + (\mathbf{I}_m \otimes B)u_s(t), \tag{3.33}$$

where $u_s(t) = -H^\top K^{ext}(\mathcal{L}^{ext} \otimes \mathbf{I}_{n_x})Hy_s(t)$.

Under **Assumption** 3, the closed-loop network dynamics (3.16) is reformulated into the collective dynamics (3.19) namely, the average and error dynamics. Afterward, the average and error dynamics (3.19) are decoupled into the slow (3.33) and fast dynamics (3.30) using TSS and SPT. It is noteworthy that the decoupled fast (3.29) and slow (3.32) dynamics are an approximation of the synchronization error and average dynamics (3.19), respectively.

### 3.2.6 Singular Perturbation Approximation

***Assumption* 4.** There exists an internal gain $K^{int}$ and an external gain $K^{ext}$ such that the network dynamics (3.17) is synchronized.

***Remark* 8.** Although, we assume the existence of the synchronizing internal and external gain, it will be ensured by a design protocol presented in Section 3.3 that such gains exist. The internal and external gains are designed independently and the obtained internal gain is optimal while the external gain is sub-optimal.

Under Assumption 3 and 4, the following theorem provides an approximation of the original system depending on the slow and fast subsystems. Applying the individual control (3.3) on the actual network (3.16), we show that the closed-loop response is close to that obtained from the approximated models. The proof follows Theorem 1 (from Theorem 5.1, Chapter 2, [Kokotović et al., 1999]).

***Theorem* 7.** *Under the Assumption 3, if the matrix $A_{22}^2$ is Hurwitz, there exists an $\epsilon^* > 0$ such that, for all $\epsilon \in (0, \epsilon^*]$, the original variables in (3.19) starting from any bounded initial conditions $y(t_0)$ and $\xi(t_0)$, are approximated for all finite time $t \geq t_0$ by*

$$\begin{cases} y(t) = y_s(t) + \epsilon\Psi(\epsilon)\xi_f(t_f), \\ \xi(t) = \xi_f(t_f) - \Omega(\epsilon)y_s(t) - \epsilon\Omega(\epsilon)\Psi(\epsilon)\xi_f(t_f), \end{cases} \tag{3.34}$$

*where $y_s(t) \in \mathbb{R}^{m.n_x}$ and $\xi_f(t_f) \in \mathbb{R}^{(N-m).n_x}$ have the respective the slow dynamics (3.33) and the fast dynamics (3.29). The terms $\epsilon\Psi(\epsilon)\xi_f(t_f)$ and $\Omega(\epsilon)y_s(t) + \epsilon\Omega(\epsilon)\Psi(\epsilon)\xi_f(t_f)$ are of order $\mathcal{O}(\epsilon)$. The approximation of the functions $\Omega$ and $\Psi$ are*

$$\begin{cases} \Omega(\epsilon) = \epsilon(A_{22}^2)^{-1}A_{21} + \mathcal{O}(\epsilon^2), \\ \Psi(\epsilon) = A_{12}(A_{22}^2)^{-1} + \epsilon((A_{22}^2)^{-1}A_{11}A_{12}(A_{22}^2)^{-1} - A_{12}) + \mathcal{O}(\epsilon^2). \end{cases} \tag{3.35}$$

***Proof***: See Appendix 4. ■

In (3.34), we notice that the approximation of $\xi$ depends on fast variable $\xi_f$ and the slow variable $y_s$, but the slow variable may not be stable. Then, as a auxiliary result we prove the following lemma which ensures the exponential stability of $\xi$ provided that $\xi_f$ is exponentially stable.

***Lemma* 8.** *The exponential stability of the fast dynamics (3.30) and the external error dynamics (3.50) implies the exponential stability of the error dynamics in (3.19).*

***Proof***: See Appendix 4. ■

Although, we assume the existence of the gain that stabilizes the slow and fast subsystems to prove the above theorem, in the next section we explain in detail the design of the fast (internal) and slow (external) gain. The internal and external gains are designed independently. The obtained internal gain $K^{int}$ is optimal while the external gain $K^{ext}$ is sub-optimal.

## 3.3 Design procedure

In this section, we explain the procedure to design the internal and external control independent of each other. The internal control design is related to the fast subsystem whereas the external control to the slow one. Although the control design is based on their respective reduced subsystems, we can still apply it to the collective dynamics (3.19). The purpose is to design an internal gain using the local information to asymptotically synchronize the agents inside the clusters. As for the external control, it must achieve the synchronization between the clusters while bounding a cost given a threshold.

The two next assumptions aim to model the graph structure. Due to the intensive communications between the agents inside the clusters, we approximate the interconnections inside the clusters by assuming all-to-all connections.

***Assumption* 5.** The internal graphs are complete for all clusters.

This assumption greatly reduces the computational effort required to obtain the control; it allows us to decouple the control design into agents' level and to obtain an analytical expression. However, it is imposed only for the control design purpose and the obtained controller can be implemented even if the Assumption 5 does not hold.

***Remark* 9.** Under Assumption 5, the non-zero eigenvalues of the internal Laplacian $\mathcal{L}_k^{int}$ are $\lambda_{k,i}^{int} = n_k$, for $i = 2, ..., n_k$ and for all $k \in \mathcal{M}$.

To ensure the synchronization of the entire network, we also assume that no cluster is isolated.

***Assumption* 6.** The graph of clusters is connected.

In the following, we first address the internal control by giving an analytical gain expression under Assumption 5. The key idea is to break down the control design at the cluster's level into the agent's level.

### 3.3.1 Internal (Fast) Control Design

As the fast variable $\xi_f$ is an approximation of the synchronization error $\xi$ inside the clusters, it is still relevant to consider the fast subsystems (3.30) for the internal control design. We denote by $\xi_{f,k} \in \mathbb{R}^{n_k \cdot n_x}$ the components of $\xi_f = (\xi_{f,1}, ..., \xi_{f,m})$ corresponding to the $k$-th cluster. For each cluster $\mathcal{C}_k$, for $k \in \mathcal{M}$, we have the following dynamics

$$\begin{cases} \dot{\xi}_{f,k}(t) = (\mathbf{I}_{n_k-1} \otimes A)\xi_{f,k}(t) + (\mathbf{I}_{n_k-1} \otimes B)u_{f,k}(t), \\ u_{f,k}(t) = -(\Lambda_k^{int} \otimes K_k^{int})\xi_{f,k}(t), \end{cases} \tag{3.36}$$

where $u_{f,k} \in \mathbb{R}^{n_k \cdot n_u}$ is the $k$-th component of $u_f = (u_{f,1}, ..., u_{f,m})$ corresponding to the $k$-th cluster control.

The cluster cost associated with the cluster $\mathcal{C}_k$ takes the form

$$J_{f,k} = \int_0^{+\infty} \xi_{f,k}^\top(t)(\Lambda_k^{int} \otimes \mathbf{I}_{n_x})\xi_{f,k}(t) + u_{f,k}^\top(t)(\mathbf{I}_{n_k-1} \otimes R_k)u_{f,k}(t)\,\mathrm{d}t\,. \tag{3.37}$$

Furthermore, the dynamics (3.36) and the cost (3.37) can be decoupled into $n_k - 1$ components. For each cluster $\mathcal{C}_k$, let denote the fast subsystems and the associated control by $\xi_{f,k} = (\xi_{f,k,2}, ..., \xi_{f,k,n_k})$ and $u_{f,k} = (u_{f,k,2}, ..., u_{f,k,n_k})$, respectively.

Then, for $i = 2, ..., n_k$ and for all $k \in \mathcal{M}$, the dynamics are

$$\begin{cases} \dot{\xi}_{f,k,i}(t) = A\xi_{f,k,i}(t) + n_k B u_{f,k,i}(t), \\ u_{f,k,i}(t) = -K_k^{int}\xi_{f,k,i}(t), \end{cases} \tag{3.38}$$

and the associated individual cost is

$$J_{f,k,i} = \int_0^{+\infty} n_k \xi_{f,k,i}^\top(t)\xi_{f,k,i}(t) + u_{f,k,i}^\top(t) n_k^2 R_k u_{f,k,i}(t)\,\mathrm{d}t\,. \tag{3.39}$$

Thus, the cost (3.37) is the sum of individual costs (3.39) as follows,

$$J_{f,k} = \sum_{i=2}^{n_k} J_{f,k,i}, \quad \forall k \in \mathcal{M}. \tag{3.40}$$

***Remark* 10.** The decoupling of (3.36) into $n_k - 1$ subsystems (3.38) is not only limited to all-to-all connections only. In case where the Laplacian eigenvalues can be characterized in terms of $n_k$ (for example, star graph), similar decoupling can be achieved.

***Remark* 11.** It is noteworthy that the gain $K_k^{int}$ is same for all the agents belonging to the same cluster $\mathcal{C}_k$. As a result, the rewriting of (3.37) into (3.40) reduces the computational effort for the control design. Indeed, one can solve only one optimization problem (3.38)-(3.39) for each cluster, it is equivalent to optimizing the cluster cost (3.37).

Finally, we apply the LQR-control to stabilize (3.38) while minimizing the cost (3.39).

***Lemma* 9.** *Under Assumption 6, if the pair $(A, n_k B)$ is stabilizable and $(A, (n_k^2 R_k)^{1/2})$ is detectable, then there exists a gain $K_k^{int}$ stabilizing (3.38) and minimizing (3.39) such that*

$$K_k^{int} = \frac{R_k^{-1}}{n_k} B^\top P_k^{int}, \quad k \in \mathcal{M}, \tag{3.41}$$

*where $P_k^{int}$ is the solution of the Algebraic Riccati Equation*

$$P_k^{int} A + A^\top P_k^{int} - P_k^{int} B R_k^{-1} B^\top P_k^{int} + n_k \mathbf{I}_{n_x} = 0. \tag{3.42}$$

*The internal control at agent's level are, for all $i \in \mathcal{C}_k$ and $k \in \mathcal{M}$,*

$$u_{k,i}^{int}(t) = -\frac{R_k^{-1}}{n_k} B^\top P_k^{int} \sum_{(k,j)\in\mathcal{N}_{k,i}} (x_{k,i}(t) - x_{k,j}(t)). \tag{3.43}$$

Once the local consensus is achieved, the external behavior is basically the dynamics of the average states formed by each cluster. It is safe to assume that the agents have merged into a single node. Therefore, the number of nodes representing the external network equals the number of clusters.

### 3.3.2 External (Slow) Control Design

The graph of agents connecting the clusters, or the external graph, is only connected. Thus, the previous procedure cannot be adopted to design the external control. To achieve the synchronization between the clusters, we propose a method based on Chapter 2. First, we define the average slow variable. Then, the synchronization problem is transformed to a stabilization problem using a change of variable. Finally, we design the control to stabilize the system while bounding an associated cost.

We recall that a block-diagonal matrix with the entries $P_1$,...,$P_N$ on the diagonal is written as $P = \mathrm{diag}(P_1, ..., P_N)$. In addition, we denote by $P_{-k} = \mathrm{diag}(P_1, ..., P_{k-1}, P_{k+1}, ..., P_N)$ the block-matrix with the $k$-th block removed. However, if the matrix $P \in \mathbb{R}^{N\times N}$, then we denote by $P_{-k} \in \mathbb{R}^{(N-1)\times(N-1)}$ the matrix $P$ with its $k$-th row and column removed.

#### Average slow variables

After the transformation (3.12), we recall that the slow variable in (3.19) is a scaled average. Thus, we redefine the average from (3.32) as follows,

$$\overline{y}(t) = (W \otimes \mathbf{I}_{n_x})y_s(t), \tag{3.44}$$

where $\overline{y} = (\overline{y}_1, \ldots, \overline{y}_m)$ and $W = \mathrm{diag}\left(\frac{1}{\sqrt{n_1}}, \ldots, \frac{1}{\sqrt{n_m}}\right)$. Then, the closed-loop average dynamics is

$$\dot{\overline{y}}(t) = \left[(\mathbf{I}_m \otimes A) - (\mathbf{I}_m \otimes B)\overline{K}^{ext}(\overline{\mathcal{L}}^{ext} \otimes \mathbf{I}_{n_x})\right]\overline{y}(t), \tag{3.45}$$

where

$$\begin{cases} \overline{\mathcal{L}}^{ext} = \begin{pmatrix} \sum_{l=2}^{m} \frac{a_{1l}^{ext}}{n_1} & -\frac{a_{12}^{ext}}{n_1} & \cdots & -\frac{a_{1m}^{ext}}{n_1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{m1}^{ext}}{n_m} & -\frac{a_{m2}^{ext}}{n_m} & \cdots & \sum_{l=1}^{m-1} \frac{a_{ml}^{ext}}{n_m} \end{pmatrix} \in \mathbb{R}^{m\times m}, \\ \overline{K}^{ext} = \mathrm{diag}\left(K_1^{ext}, ..., K_m^{ext}\right), \end{cases} \tag{3.46}$$

are the average Laplacian matrix and the external gain related to (3.45), respectively. The weights $a_{kl}^{ext}$ stand for the total number of connections between cluster $\mathcal{C}_k$ and $\mathcal{C}_l$.

The average dynamics of each cluster $\mathcal{C}_k$, for $k \in \mathcal{M}$, is

$$\begin{cases} \dot{\overline{y}}_k(t) = A\overline{y}_k(t) + B\overline{u}_k^{ext}(t), \\ \overline{u}_k^{ext}(t) = -K_k^{ext} \sum_{l \in \mathcal{N}_{\mathcal{C}_k}} \frac{a_{kl}^{ext}}{n_k}(\overline{y}_k(t) - \overline{y}_l(t)). \end{cases} \tag{3.47}$$

The average cost associated with each cluster $\mathcal{C}_k$, $k \in \mathcal{M}$, is defined as

$$\overline{J}_k^{ext} = \int_0^{+\infty} \sum_{l \in \mathcal{N}_{\mathcal{C}_k}} \frac{a_{kl}^{ext}}{n_k}(\overline{y}_k(t) - \overline{y}_l(t))^2 + n_k \sum_{i=1}^{n_k} \widehat{u}_{k,i}^{ext\top}(t) R_k \widehat{u}_{k,i}^{ext}(t)\,\mathrm{d}t, \tag{3.48}$$

where $\widehat{u}_{k,i}^{ext} = -K_k^{ext} \sum_{l \in \mathcal{N}_{C_k}} \frac{a_{(k,i)\leftrightarrow C_l}^{ext}}{n_k}(\overline{y}_k - \overline{y}_l)$, for $i \in C_k$, and $a_{(k,i)\leftrightarrow \mathcal{C}_l}^{ext}$ is the total number of connections between the $i$-th agent belonging to $\mathcal{C}_k$ and the cluster $\mathcal{C}_l$. The control $\widehat{u}_{k,i}^{ext}$ is the external control (3.4) expressed in the average variable $\overline{y}$. In addition, we have the relation $\overline{u}_k^{ext} = \sum_{i=1}^{n_k} \widehat{u}_{k,i}^{ext}$ and $a_{kl}^{ext} = \sum_{i=1}^{n_k} a_{(k,i)\leftrightarrow \mathcal{C}_l}^{ext}$.

***Remark* 12.** Although the cluster have merged into a single node, the agents still apply the control (3.4) rather than the average control $\overline{u}_k^{ext}$. Thus, we do not consider the average control $\overline{u}_k^{ext}$ directly into the cost. In fact, we express the control (3.4) in average variables $\overline{y}$ and apply it in individual manner as in equation (3.48).

### Change of Variables

To study the consensus between the clusters, we define the external error variable from Chapter 2 such that

$$Y_k := \begin{pmatrix} \overline{y}_1 - \overline{y}_k \\ \vdots \\ \overline{y}_{k-1} - \overline{y}_k \\ \overline{y}_{k+1} - \overline{y}_k \\ \vdots \\ \overline{y}_m - \overline{y}_k \end{pmatrix}, \quad \forall k \in \mathcal{M}. \tag{3.49}$$

Then, the corresponding external error dynamics are

$$\dot{Y}_k(t) = \mathbf{A}_k Y_k(t) + \mathbf{B}_k \overline{u}_k^{ext}(t), \quad \forall k \in \mathcal{M}, \tag{3.50}$$

where

$$\begin{cases} \mathbf{A}_k = (\mathbf{I}_{m-1} \otimes A) - (\mathbf{I}_{m-1} \otimes B)\overline{K}_{-k}^{ext}(\overline{\mathcal{L}}_{-k}^{ext} \otimes \mathbf{I}_{n_x}), \\ \mathbf{B}_k = -(\mathbb{1}_{m-1} \otimes B). \end{cases}$$

Here, $\overline{K}^{ext}_{-k} = \mathrm{diag}(K^{ext}_1, \dots, K_{k-1}, K_{k+1}, \dots, K_m)$ is not a control action but it represents the behavior of the network.

Upon close inspection at the structure of the external Laplacian $\mathcal{L}^{ext}$ we see that it has the following form,

$$\mathcal{L}^{ext} = \begin{pmatrix} \mathcal{L}^{ext}_{1,1} & \mathcal{L}^{ext}_{1,2} & \dots & \mathcal{L}^{ext}_{1,m} \\ \mathcal{L}^{ext}_{2,1} & \mathcal{L}^{ext}_{2,2} & \dots & \mathcal{L}^{ext}_{2,m} \\ \vdots & \vdots & & \vdots \\ \mathcal{L}^{ext}_{m,1} & \mathcal{L}^{ext}_{m,2} & \dots & \mathcal{L}^{ext}_{m,m} \end{pmatrix} \in \mathbb{R}^{N \times N}, \tag{3.51}$$

where $\mathcal{L}^{ext}_{p,q} \in \mathbb{R}^{n_p \times n_q}$ for $p, q \in \mathcal{M}$. We denote by $\mathcal{L}^{ext}_{k,row} \in \mathbb{R}^{n_k \times N}$ the $k$-th row of the block-matrix (3.51) for all $k \in \mathcal{M}$. It describes the connections of the cluster $\mathcal{C}_k$ with the rest of the agents in the network. The matrix $\mathcal{L}^{ext}_{k,red} \in \mathbb{R}^{n_k \times (N - n_k)}$ is obtained by removing the $\mathcal{L}^{ext}_{k,k}$ block from the $\mathcal{L}^{ext}_{k,row}$. For example, $\mathcal{L}^{ext}_{2,row} = [\mathcal{L}^{ext}_{2,1} \ \mathcal{L}^{ext}_{2,2} \ \dots \ \mathcal{L}^{ext}_{2,m}]$ and $\mathcal{L}^{ext}_{2,red} = [\mathcal{L}^{ext}_{2,1} \ \mathcal{L}^{ext}_{2,3} \ \dots \ \mathcal{L}^{ext}_{2,m}]$.

Then, we express the external cost (3.48) in terms of new variables as

$$\overline{J}^{ext}_k = \int_0^{+\infty} Y_k^\top(t) Q^{ext}_{k,1} Y_k(t) + Y_k^\top(t) \frac{Q^{ext}_{k,2}}{n_k} Y_k(t)\, \mathrm{d}t \tag{3.52}$$

where

$$\begin{cases} Q^{ext}_{k,1} = \left( \mathrm{diag}\left( \frac{a^{ext}_{k,1}}{n_k}, \dots, \frac{a^{ext}_{k,k-1}}{n_k}, \frac{a^{ext}_{k,k+1}}{n_k}, \dots, \frac{a^{ext}_{k,m}}{n_k} \right) \otimes \mathbf{I}_{n_x} \right), \\ Q^{ext}_{k,2} = U^\top_{-k} (\mathcal{L}^{ext\top}_{k,red} \mathcal{L}^{ext}_{k,red} \otimes K^{ext\top}_k R_k K^{ext}_k) U_{-k}, \\ \quad U = (\mathrm{diag}(\mathbb{1}_{n_1}, \dots, \mathbb{1}_{n_m}) \otimes \mathbf{I}_{n_x}), \\ \quad R_k > 0. \end{cases} \tag{3.53}$$

The matrices $Q^{ext}_{k,1}$ and $Q^{ext}_{k,2}$ simplify the expression (3.48) such that

$$\begin{cases} Y_k^\top Q^{ext}_{k,1} Y_k = \sum\limits_{l \in \mathcal{N}_{C_k}} \frac{a^{ext}_{kl}}{n_k} (\overline{y}_k - \overline{y}_l)^2, \\ Y_k^\top \frac{Q^{ext}_{k,2}}{n_k} Y_k = n_k \sum\limits_{i=1}^{n_k} \widehat{u}^{ext\top}_{k,i} R_k \widehat{u}^{ext}_{k,i}. \end{cases} \tag{3.54}$$

## Control Design

To design the external control, we use the satisfaction equilibrium approach proposed in Chapter 2. Given the external error dynamics (3.50), it characterizes the external gain profile synchronizing the network in such a way that each cost (3.48) is bounded,

$$\overline{J}_k^{ext} \le \gamma^{ext} \|Y_k(0)\|^2, \quad \text{for } k \in \mathcal{M}. \tag{3.55}$$

The term $\|Y_k(0)\|$ represents the initial condition of the cluster $\mathcal{C}_k$ while $\gamma^{ext}$ is a given threshold. The proposition is stated as follows.

***Proposition* 10** (Prop 1, Chapter 2)**.** *Let a gain profile* $\overline{K}^{ext} = \operatorname{diag}\left(K_1^{ext}, ..., K_m^{ext}\right)$ *be given. The following statements are equivalent*

1. *The gain profile* $\overline{K}^{ext}$ *is an SE of the satisfaction game* (3.50) *for all* $k \in \mathcal{M}$.
2. *For all* $k \in \mathcal{M}$, *there exists a positive-definite matrix* $P_k^{ext} > 0$ *such that*

$$\begin{cases} P_k^{ext} \mathbf{A}_{k,cl}(K_k^{ext}) + \mathbf{A}_{k,cl}^\top(K_k^{ext}) P_k^{ext} + \mathbf{Q}_k^{ext}(K_k^{ext}) < 0, \\ \qquad\qquad\qquad\qquad P_k^{ext} - \gamma^{ext} \mathbf{I}_{(m-1).n_x} < 0, \end{cases} \tag{3.56}$$

*where*

$$\begin{cases} \mathbf{A}_{k,cl}(K_k^{ext}) = \mathbf{A}_k + \mathbf{B}_k K_k^{ext} (F_k \otimes \mathbf{I}_{n_x}), \\ F_k = \left( \dfrac{a_{k,1}^{ext}}{n_k}, ..., \dfrac{a_{k,k-1}^{ext}}{n_k}, \dfrac{a_{k,k+1}^{ext}}{n_k}, ..., \dfrac{a_{k,m}^{ext}}{n_k} \right), \\ \mathbf{Q}_k^{ext} = \left( Q_{k,1}^{ext} + \dfrac{Q_{k,2}^{ext}}{n_k} \right). \end{cases} \tag{3.57}$$

### 3.3.3 Algorithm complexity

To illustrate how the design procedure reduces the computational load, we briefly explain the algorithm stated below. Consider a network of $m$ agents (which is the number of cluster in our case) with their respective dynamics. We aim to design a synchronizing gain profile $K^{ext} = (K_1^{ext}, ..., K_m^{ext})$ satisfying the cost constraints. The algorithm is as follows :

**Data :** $A, B$ and $n_k, k \in \mathcal{M}$ ;
**Set :** iterations $iter = 1$, maximum number of iterations $iter_{max}$, $0 < \epsilon^* \ll 1$ and $\mathbf{K}^{ext}(0) = (K_1^{ext}(0), ..., K_m^{ext}(0))$ initial gain profile synchronizing the system ;
**Calculate :** $P_k^{int}$ and $K_k^{int}$ using equation (3.42) and (3.41) for all $k \in \mathcal{M}$, respectively ;
**while** LMIs (3.56) not satisfied **OR** $iter \le iter_{max}$ **do**
  $K^{ext}(iter+1) \leftarrow \alpha^{ext} K^{ext}(iter), \quad \alpha^{ext} \in \mathbb{R}_+ \setminus \{0\}$ ;
  **Calculate :** $\epsilon$ ;
  **if** $\epsilon > \epsilon^*$ **then**
    $K_k^{int} \leftarrow \dfrac{\epsilon}{\epsilon^*} K_k^{int}$ ;
  **end if**
**end while**

**Algorithm 2:** Sequential Satisfaction Algorithm

In the above algorithm, we first calculate the internal gain by solving the algebraic riccati equation (3.42). To design the external gain, we start with an initial gain profile that synchronizes the network. Then, to obtain a sub-optimal gain, we multiply the gain from the previous iteration with a scalar $\alpha^{ext} \in \mathbb{R}_+ \setminus \{0\}$ and check if satisfies the LMI (3.56). One approach could be to start with high gain and decrease $\alpha^{ext}$ until the condition (3.56) is not satisfied and use the smallest gain that satisfied the condition. Furthermore, we should also make sure the network parameter $\epsilon$ is small so that control design using time-scale separation holds. Thus, to ensure this, we multiply the internal gain $K_k^{int}$ with $\epsilon/\epsilon^*$ to obtain the new internal gain such that $\epsilon \leq \epsilon^*$.

In algorithm 2, the computational complexity to obtain the internal gain is of order $\mathcal{O}(m)$ since we need to solve $m$ AREs, one for each cluster. The size of the LMIs depends on $n_k, k \in \mathcal{M}$. However for the external gain we just need to check that LMI (3.56) is satisfied for all $k \in \mathcal{M}$. Moreover, if the algorithm successfully converge to stabilizing internal gain $K^{int}$ and satisfy the LMI conditions (3.56) for synchronizing external gain $K^{ext}$, then they will satisfy the Assumption 4.

## 3.4 Cost approximation

The second objective of this chapter is to provide an approximation of the cluster cost. In the following, we prove that the cluster cost $J_k$ can be approximated only by $n_k \bar{J}_k^{ext}$ during the time interval $[T, +\infty)$. The time $T > 0$ is chosen such that the synchronization error inside the clusters is bounded by $\epsilon$.

### 3.4.1 Internal error bound

The necessity of the internal error bound arises in the approximation of the cluster cost. During the control design, we remind that the internal consensus is considered to be achieved before designing the external control. Thus, we need to characterize an error bound in finite time $T$, at which the cluster is very close to the internal consensus.

After, the internal control design $u_k^{int}$, the closed-loop fast dynamics is

$$\dot{\xi}_{f,k}(t) = \left[(\mathbf{I}_{n_k-1} \otimes A) - (\Lambda_k^{int} \otimes BK_k^{int})\right] \xi_{f,k}(t), \tag{3.58}$$

and it yields

$$\xi_{f,k}(t) = e^{Cl_{f,k}t}\xi_{f,k}(0), \tag{3.59}$$

where $Cl_{f,k} = (\mathbf{I}_{n_k-1} \otimes A) - (\Lambda_k^{int} \otimes BK_k^{int})$. Then, taking the norm on both sides and with the measure of the matrix, we obtain

$$\begin{aligned}\|\xi_{f,k}(t)\| &= e^{\nu(Cl_{f,k})t}\|\xi_{f,k}(0)\| \\ &\leq e^{\nu(Cl_f)t}\|\xi_{f,k}(0)\| \end{aligned} \tag{3.60}$$

where $\nu(Cl_f) = \max\limits_{k\in\mathcal{M}} \nu(Cl_{f,k})$.

Afterwards, as an internal error bound, we choose the smallest $T \geq 0$ such that

$$\|\xi_{f,k}(T)\| \leq e^{\nu(Cl_f)T} \max_{k\in\mathcal{M}} \|\xi_{f,k}(0)\| \leq \epsilon. \tag{3.61}$$

This bound characterizes the local consensus inside each cluster in the finite time $T$, it yields

$$\|\xi_{f,k}(t)\| \leq \epsilon e^{\nu(Cl_f)(t-T)} \qquad \forall k \in \mathcal{M}, \tag{3.62}$$

and

$$\|\xi_f(t)\| \leq \epsilon\sqrt{n-m}.e^{\nu(Cl_f)(t-T)}. \tag{3.63}$$

### 3.4.2 Approximation of the cluster cost

In this section, we prove that for the time $t \in [T, +\infty)$, the cluster cost $J_k$ is approximated by $n_k$ times the average cost, as $n_k\overline{J}_k^{ext}$. The motivation is derived from the fact that the internal dynamics converges rapidly to the consensus and the dominating network behaviour is illustrated by the external dynamics.

***Proposition* 11.** *During the time interval $[T, +\infty)$, the following approximation holds,*

$$J_k(T, +\infty) = n_k\overline{J}_k^{ext}(T, +\infty) + \mathcal{O}(\epsilon), \quad \forall k \in \mathcal{M}. \tag{3.64}$$

***Proof***: See Appendix 4.

## 3.5 Simulation

In this section, we provide some numerical examples to illustrate the effectiveness of the control procedure and the cost approximation through three scenarios. The agents dynamics are given by (3.1), where the dynamics state and control matrices are

$$A = \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix}, \quad \theta = 30 \text{ radians}, \quad B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{3.65}$$

The external connections between the agents in different clusters are generated using

Erdos-Renyi [Erdos et al., 1960] random graph generator. In the Scenario 1, we consider a graph $\mathcal{G}_1$ with four clusters $(m = 4)$. The total number of agents in the graph $\mathcal{G}_1$ is $n_{\mathcal{G}_1} = 630$ agents. Moreover, we impose the threshold $\gamma^{ext} = 0.8$ for the average cost (3.48).

The Scenario 2 is slightly different from Scenario 1, we keep the same parameters for the simulation. However, the internal graphs are not complete anymore but just connected, we use graphs where the connections are dense inside. We remind that the Assumption 5 is only needed for the design of the internal control. It is still practical for clusters with dense connections.

For the last Scenario 3, we compare the composite control with the method in Chapter 2 but also the control in [Rejeb et al., 2018] through the graph $\mathcal{G}_1$ and $\mathcal{G}_2$. The details of the simulations are present in Table 3.1 and 3.2.

In the tables, $n_k$ represent the number of agents in cluster $\mathcal{C}_k$, $error(k) = \frac{|J_k - n_k \bar{J}_k^{ext}|}{J_k} \times 100$, is the error percentage between the total cost and the external cost after time $T$, and $K^{ext}$ and $K^{int}$ are the respective external and internal gains.

## Scenario 1

The Figure 3.2 represents the synchronization of the agents in graph $\mathcal{G}_1$ with only 299 external connections between the clusters. For the graph $\mathcal{G}_1$, the network parameter is $\epsilon_1 = 0.06$. In the figure, we can observe the four branches appearing and merging into one. Each branch represents the local agreement within the clusters. Next, Figure 3.3 illustrates the cost approximation for the cluster $\mathcal{C}_4$ by comparing the total cluster cost $J_4$ and the external cost $n_4 \bar{J}_4^{ext}$, after finite time $T = 2s$.

| $\epsilon = 0.06$, $\gamma = 0.8$ | | | | Gain | |
|---|---|---|---|---|---|
| | $n_k$ | $J_k(\times 10^4)$ | $error(k)$ | $K^{int}$ | $K^{ext}$ |
| $\mathcal{C}_1$ | 120 | 8.966 | $0.45\%$ | $[1.5352, -0.1102]$ | $[0.85, 0.16]$ |
| $\mathcal{C}_2$ | 140 | 5.768 | $0.86\%$ | $[1.5349, -0.1114]$ | $[1.17, 0.22]$ |
| $\mathcal{C}_3$ | 170 | 18.950 | $0.24\%$ | $[1.5346, -0.1128]$ | $[0.59, 0.11]$ |
| $\mathcal{C}_4$ | 200 | 6.405 | $0.65\%$ | $[1.5344, -0.1137]$ | $[1.05, 0.2]$ |

TABLE 3.1 – Scenario 1 : Network with 630 agents and 299 external connections.

## Scenario 2

In this scenario, we consider the graph where the clusters has dense interconnections instead of the all-to-all connections compared to Scenario 1. We consider the network with same number of agents inside each cluster as Scenario 1. The internal gains are designed

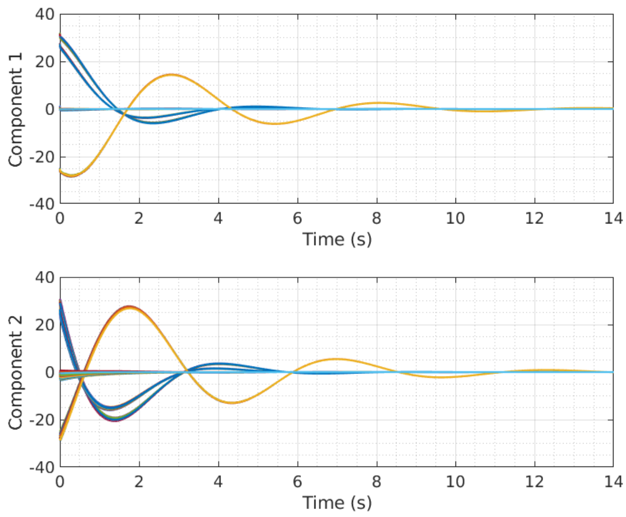


FIGURE 3.2 – State error between the first agent of the cluster $\mathcal{C}_2$ and the network, with the graph $\mathcal{G}_1$.

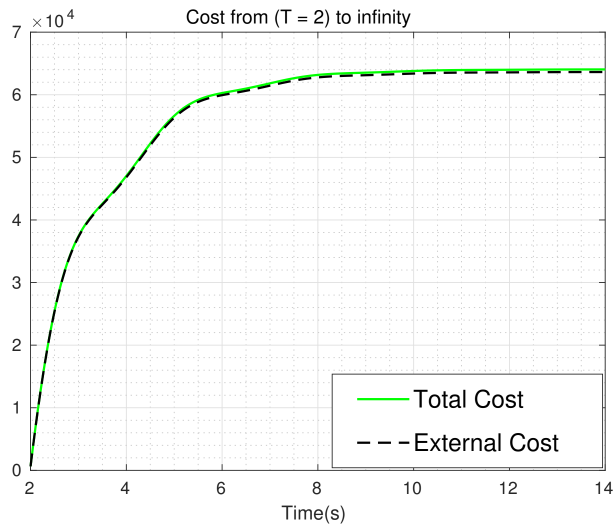


FIGURE 3.3 – State error between the first agent of the cluster $\mathcal{C}_2$ and the network, with the graph $\mathcal{G}_2$.

assuming the clusters has all-to-all connections and the gain is applied to the network with connected intra-cluster connections.

| $\epsilon = 0.06$, $\gamma = 0.8$ | | | | Gain | |
|---|---|---|---|---|---|
| | $n_k$ | $J_k(\times 10^4)$ | $error(k)$ | $K^{int}$ | $K^{ext}$ |
| $\mathcal{C}_1$ | 120 | 8.983 | 0.64% | $[1.5352, -0.1102]$ | $[0.85, 0.16]$ |
| $\mathcal{C}_2$ | 140 | 5.780 | 1.07% | $[1.5349, -0.1114]$ | $[1.17, 0.22]$ |
| $\mathcal{C}_3$ | 170 | 1.8.975 | 0.37% | $[1.5346, -0.1128]$ | $[0.59, 0.11]$ |
| $\mathcal{C}_4$ | 200 | 6.415 | 0.81% | $[1.5344, -0.1137]$ | $[1.05, 0.2]$ |

TABLE 3.2 – Scenario 2 : graph $\mathcal{G}_1$ without complete graphs for the internal graphs but with 299 external connections.

## Scenario 3

In the last scenario, we consider a network of $m = 4$ clusters with $n_k = 10$ agents in each. We recall that $\gamma^{ext} = 1$ is chosen for both controls. A comparison is done between the composite control proposed in Chapter 3 and the satisfactory control approach proposed in Chapter 2. The design procedure in Chapter 2 needs 13752 seconds (3.8 hours) to compute the gains for $n = 40$ agents while the composite design requires 13 seconds. However, we can observe an incontestable difference in performance on the cluster costs due to satisfactory control, as shown in Table 3.3. This emphasize the trade-off between the computing time/resources to obtain the required controller. Despite being less effective, we have to keep in mind that the composite control suits better for large-scale networks and present a important benefit in term of computation loads and time.

| | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{C}_3$ | $\mathcal{C}_4$ |
|---|---|---|---|---|
| $n_k$ | 10 | 10 | 10 | 10 |
| $J_k$ | 17204 | 5452 | 6943 | 16949 |
| $J_k^{NE}$ | 10164 | 3303 | 3080 | 9714 |

TABLE 3.3 – Graph $\mathcal{G}_3$ : $n_{\mathcal{G}_3} = 40$, $\gamma_3^{ext} = 1$ and 4 external connections.

Next, we compare the strategy in [Rejeb et al., 2018] with the composite control. In [Rejeb et al., 2018], every agents apply the same gain independently of their neighborhoods and aim to bound a global cost. Applying the control [Rejeb et al., 2018] on the graph $\mathcal{G}_1$, it results in a cluster cost which we label by $J_k^{\dagger}$. From Table 3.4, we observe that our strategy significantly outperforms the strategy in [Rejeb et al., 2018], the first cluster cost obtained via the composite control is 20 times smaller. One may observe the same for the other clusters.

| | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{C}_3$ | $\mathcal{C}_4$ |
|---|---|---|---|---|
| $n_k$ | 120 | 140 | 170 | 200 |
| $J_k(\times 10^6)$ | 0.385 | 0.269 | 0.689 | 0.262 |
| $J_k^{\dagger}(\times 10^6)$ | 6.7 | 8.1 | 16.7 | 20.5 |

TABLE 3.4 – Comparison of the cost based on the graph $\mathcal{G}_1$.

## 3.6 Conclusion

In this chapter, we propose a distributed composite control design strategy for the clustered network. Using coordinate transformation, the network dynamics is transformed into standard singular perturbation form and decoupled into slow and fast dynamics using time-scale separation. This decoupling of the network dynamics also decouple the control into fast (internal) and slow (external). The internal control is responsible for intra-cluster synchronization while the external synchronizes the network while satisfying the impose cost criterion. This independent design greatly reduces the computational effort required to obtain the control. Finally, we show that the cluster cost is approximated only by the external cost after short period of time.

Numerical simulations show that while the strategy proposed in Chapter 2 outperforms the strategy proposed in Chapter 3 (about 50% lower costs), the computation load is 1000 times more with merely $N = 40$ agents. Furthermore, the computations become infeasible for large-scale networks. On the other hand, while the solution in [Rejeb et al., 2018] is computationally very fast, the control performance is extremely poor compared to our solution. Thus, we demonstrate that the solution proposed in Chapter 3 offers an interesting trade-off between control performance and computational feasibility.

Chapter 4

# Final conclusions

## Overview

This thesis is dedicated to synchronization algorithms for *multi-agent systems (MAS)* that consider individual costs during the coordination process. The analysis is carried out with particular attention on MAS with homogeneous linear dynamics and clustered networks, with fixed topology in both cases. We have exploited concepts from game theory [Perlaza et al., 2012] as well as singular pertubation tools for time-scale separation [Kokotović et al., 1999] for this purpose. Furthermore, we take into account communication constraints in the network, i.e. the controls of each agent only use information coming from its neighborhood. Distributed control protocols are provided, but the control gains may be designed in a centralized manner depending on the cases.

**In Chapter 2**, we deal with the problem of output feedback synchronization for MAS, which guarantees individual performance bounds. The synchronization problem is first recast into a stabilization problem to make the analysis easier. Then, the problem is modeled as a satisfaction game and we seek gains that are in satisfaction equilibrium (SE), i.e. the cost associated to each agent is upper-bounded by a given threshold. Conditions in the form of linear matrix inequalities are provided to check if a given gain profile is an SE. Moreover, based on the output feedback control, a second result allows us to synthesize the gain of an agent assuming the gains of the other agents are known.

**In Chapter 3**, we present a distributed composite control design that synchronizes clustered networks, while guaranteeing a certain bound on the cluster costs. The approach aims to significantly reduce the problem's complexity and the computational effort necessary to obtain the controllers for large-scale networks. Based on the network structure, we apply time-scale separation techniques to decouple the original system into fast (intra-cluster) and slow (inter-cluster) dynamics. The fast variables correspond to the cluster synchronization error, whereas the slow variables is the aggregate behavior of the agent states inside each cluster. Then, the control design is broken into independent designs of slow and fast controllers, that aims at enhancing the synchronization process. Only one controller is derived per cluster for all the agents inside, irrespective of the number of agents in the cluster. This drastically reduces the computational effort and the complexity of the control design. Finally, we propose sub-optimal control strategies as the internal cost is minimized while the external cost is satisfactory for each cluster. The composite controller for each agent is synthesized as the sum of the internal and external control. Furthermore, we show that the internal control only affects the cluster cost for a short period of time compared to the synchronization time of the full network.

According to the simulations, even though the control strategy in Chapter 2 outperforms the control protocol in Chapter 3, the computation load related to the satisfactory control is substantially higher than the composite control, and is even infeasible for large-scale networks. On the other hand, the analytical result in [Rejeb et al., 2018] provides a very fast computation but presents an extremely poor control performance compared to the composite control. As a result, the composite control offers an interesting trade-off between control performance and computing efficiency.

# Perspectives

The results of the work developed in this thesis lead to various possible extensions that can be scientifically relevant.

## Short term

- For both homogeneous linear MAS and clustered networks, extending the study to weighted graphs can enlarge the range of possible applications as the communication topology will be more realistic. For example, the choice of directed threshold graphs to represent the energy flow in a electrical grid can be relevant to investigate on the behavior of the network.

  ***Definition* 15.** *An directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be threshold if there exists an injective weight function on the vertices $w(x) : \mathcal{V}(\mathcal{G}) \to \mathbb{R}$ and a threshold value $t \in \mathbb{R}$ such that $\overrightarrow{xy} \in \mathcal{E}$ if and only if $|w(x)| + |w(y)| > t$ and $w(x) > w(y)$, where $|.|$ is the absolute value.*

  The term $\overrightarrow{xy}$ represents the edge from $x$ to $y$ and the inequality $w(x) > w(y)$ expresses the fact that the flow is moving to $x$ to $y$.

- In clustered networks, it could be of interest to relax the assumption of complete graph (for internal graphs) to connected graph that can be characterized by its eigenvalues. We recall that if the eigenvalues of the internal Laplacian are known, then the internal cluster cost (3.37) can be separated into $n_k$ individual costs (3.39). The purpose is to reduce the computations while considering a communication topology that fit better to the reality. For instance, such a graph could be a strongly $k$-regular graph or threshold graph.

***Definition* 16.** *In a $k$-regular graph $\mathcal{G}$, each node is connected to $k$ other nodes.*

***Definition* 17.** ([Godsil and Royle, 2001])
*Let $\mathcal{G}$ be a regular graph that is neither complete nor empty. Then $\mathcal{G}$ is said to be strongly regular with parameters*

$$(N, k, \mathbf{a}, \mathbf{c}), \tag{4.1}$$

*where $N$ is the total number of nodes. If it is $k$-regular, every pair of adjacent nodes has $\mathbf{a}$ common neighbors and every pair of distinct non-adjacent nodes has $\mathbf{c}$ common neighbors.*

***Proposition* 12.** *([Godsil and Royle, 2001])*
*Let $\mathcal{G}$ be a strongly regular graph with parameters $(N, k, \mathbf{a}, \mathbf{c})$. Then, $\mathcal{G}$ has the three following eigenvalues such that*

— $k$ *with multiplicity* $1$,

— $\theta = \dfrac{(\mathbf{a}-\mathbf{c})+\sqrt{\Delta}}{2}$ *with multiplicity* $m_\theta = \dfrac{1}{2}\left((N-1) - \dfrac{2k+(N-1)(\mathbf{a}-\mathbf{c})}{\sqrt{\Delta}}\right)$,

— $\tau = \dfrac{(\mathbf{a}-\mathbf{c})-\sqrt{\Delta}}{2}$ *with multiplicity* $m_\tau = \dfrac{1}{2}\left((N-1) + \dfrac{2k+(N-1)(\mathbf{a}-\mathbf{c})}{\sqrt{\Delta}}\right)$,

*where* $\Delta = (\mathbf{a}-\mathbf{c})^2 + 4(k-\mathbf{c})$.

***Definition* 18.** Let $d_1 \geq d_2 \geq ... \geq d_N$ be the ordered degree of the graph $\mathcal{G}$. Then, the conjugate sequence of $d_1, d_2, ..., d_N$ is defined as

$$d_j^* = |\{d_i : d_i \geq j\}|_{card},\ j = 1, 2, ..., \tag{4.2}$$

where $|.|_{card}$ is the cardinality of the set.

***Theorem* 13.** *(Theorem 10.8, [Bapat, 2010]) Let $\mathcal{G}$ be a threshold graph with $\mathcal{V}(\mathcal{G}) = 1, 2, ..., N$. Let $\mathcal{L}(\mathcal{G})$ be the Laplacian and $d_1, ..., d_N$ the degree sequence of $\mathcal{G}$, then $d_1^*, ..., d_N^*$ are the eigenvalues of $\mathcal{L}(\mathcal{G})$.*

## Long term

- In homogeneous linear MAS, the synchronization problem with cost optimization can be formulated as a potential game.

  ***Definition* 19.** (Exact potential game, Lasaulce and Tembine [2011])
  *The game $\mathcal{G}$ is an exact potential game if there exists a function $\Phi$ such that for all $i \in \mathcal{V}$ and for all $K_i$, $K_i' \in \mathcal{K}$,*

  $$J_i(K_i, K_{-i}) - J_i(K_i', K_{-i}) = \Phi(K_i, K_{-i}) - \Phi(K_i', K_{-i}), \tag{4.3}$$

  *where $J_i(K_i, K_{-i})$ is the individual cost resulting from the actions $K_i$ and $K_{-i}$ associated with the player $i$ and the other players, respectively.*

  Thanks to the relation (4.3), one may choose a potential function $\Phi$ as a global cost and expresses it as a linear combination of individuals costs. Thus, the minimization of the global cost will be equivalent to the minimization each individual cost. Furthermore in the case of potential games, the Nash equilibrium can be found by applying the Best Response Algorithm in which each player chooses its Best Response turn-by-turn.

- The control protocols proposed in the thesis are derived from model-based design and it can be limited by an intensive computational effort or restrictive due to some required information on the whole network. In this context, data-driven techniques from *Reinforcement Learning* seem promising to reduce the computational burden and to propose a flexible control design. To the best of our knowledge, few control design based on data-driven methods address the problem of consensus. Thus, it can be interesting to propose a model-free approach to derive a distributed control.

# Appendix

## Lemma 8

***Proof***: Integrating the error dynamics in (3.19), we obtain

$$\begin{aligned}
\xi(t) &= e^{\bar{A}_{22}t}\xi(0) + \int_0^t e^{\bar{A}_{22}(t-\tau)}\bar{A}_{21}y(\tau)\,\mathrm{d}\tau \\
&= e^{\bar{A}_{22}t}\xi(0) + \int_0^t e^{\bar{A}_{22}(t-\tau)}\bar{A}_{21}(y_s(\tau) + \epsilon\Psi(\epsilon)\xi_f(\tau))\,\mathrm{d}\tau \\
&= e^{\bar{A}_{22}t}\xi(0) + \int_0^t e^{\bar{A}_{22}(t-\tau)}Z^T MY(\tau)\,\mathrm{d}\tau + \epsilon\int_0^t e^{\bar{A}_{22}(t-\tau)}\bar{A}_{21}\Psi(\epsilon)\xi_f(\tau)\,\mathrm{d}\tau
\end{aligned} \tag{1}$$

where $M = diag(M_1, ..., M_m)$ and $M_k = (\mathcal{L}^{ext}_{k,red} \otimes BK^{ext}_k)U_{-k}$. By taking norm on both sides, we have

$$\begin{aligned}
\|\xi(t)\| \leq \|e^{\bar{A}_{22}t}\|.\|\xi(0)\| + \|Z^T M\| \int_0^t \|e^{\bar{A}_{22}(t-\tau)}\|.\|Y(\tau)\|\,\mathrm{d}\tau \\
+ \epsilon\|\bar{A}_{21}\Psi(\epsilon)\| \int_0^t \|e^{\bar{A}_{22}(t-\tau)}\|.\|\xi_f(\tau)\|\,\mathrm{d}\tau
\end{aligned} \tag{2}$$

Also, from the design of internal and external control we know that, for all $t \geq 0$,

$$\begin{cases} Y(t) = e^{\mathbf{A}_{cl}t}Y(0) \\ \xi_f(t) = e^{\bar{A}^2_{22}t}\xi_f(0) \end{cases} \Rightarrow \begin{cases} \|Y(t)\| \leq e^{\nu(\mathbf{A}_{cl})t}\|Y(0)\| \\ \|\xi_f(t)\| \leq e^{\nu(\bar{A}^2_{22})t}\|\xi_f(0)\| \end{cases} \tag{3}$$

where $\mathbf{A}_{cl} = diag(\mathbf{A}_{1,cl}, ..., \mathbf{A}_{m,cl})$ is the closed-loop dynamics of the external error (3.50). Then, it follows that

$$\begin{aligned}
\|\xi(t)\| \leq e^{\nu(\bar{A}_{22})t}\|\xi(0)\| + \|Z^T M\|.\|Y(0)\| \int_0^t e^{\nu(\bar{A}_{22})(t-\tau)}e^{\nu(\mathbf{A}_{cl})\tau}\,\mathrm{d}\tau \\
+ \epsilon\|\bar{A}_{21}\Psi(\epsilon)\|.\|\xi_f(0)\| \int_0^t e^{\nu(\bar{A}_{22})(t-\tau)}e^{\nu(\bar{A}^2_{22})\tau}\,\mathrm{d}\tau\,.
\end{aligned} \tag{4}$$

By integrating the second term in (4), we have

$$\begin{aligned} \|Z^T M\|.\|Y(0)\| \int_0^t e^{\nu(\bar{A}_{22})(t-\tau)} e^{\nu(\mathbf{A}_{cl})\tau}\, \mathrm{d}\tau &= \|Z^T M\|.\|Y(0)\| e^{\nu(\bar{A}_{22})t} \int_0^t e^{(\nu(\mathbf{A}_{cl})-\nu(\bar{A}_{22}))\tau}\, \mathrm{d}\tau \\ &= \frac{\|Z^T M\|.\|Y(0)\|}{\nu(\mathbf{A}_{cl}) - \nu(\bar{A}_{22})} \left[ e^{\nu(\mathbf{A}_{cl})t} - e^{\nu(\bar{A}_{22})t} \right] \end{aligned} \tag{5}$$

In the same manner, we obtain

$$\epsilon \|\bar{A}_{21}\Psi(\epsilon)\|.\|\xi_f(0)\| \int_0^t e^{\nu(\bar{A}_{22})(t-\tau)} e^{\nu(\bar{A}_{22}^2)\tau}\, \mathrm{d}\tau = \frac{\epsilon \|\bar{A}_{21}\Psi(\epsilon)\|.\|\xi_f(0)\|}{\nu(\bar{A}_{22}^2) - \nu(\bar{A}_{22})} \left[ e^{\nu(\bar{A}_{22}^2)t} - e^{\nu(\bar{A}_{22})t} \right] \tag{6}$$

Finally, we have

$$\|\xi(t)\| \leq \mathbf{C}_1 e^{\nu(\mathbf{A}_{cl})t} + \epsilon \mathbf{C}_2 e^{\nu(\bar{A}_{22}^2)t} + \left( \|\xi(0)\| - \mathbf{C}_1 - \epsilon \mathbf{C}_2 \right) e^{\nu(\bar{A}_{22})t}, \tag{7}$$

where $\mathbf{C}_1 = \frac{\|Z^T M\| \|Y(0)\|}{\nu(\mathbf{A}_{cl}) - \nu(\bar{A}_{22})}$ and $\mathbf{C}_2 = \frac{\|\bar{A}_{21}\Psi(\epsilon)\| \|\xi_f(0)\|}{\nu(\bar{A}_{22}^2) - \nu(\bar{A}_{22})}$. Moreover, we know that $\nu(\bar{A}_{22}^2) < \nu(\bar{A}_{22}) < \nu(\mathbf{A}_{cl}) < 0$. Thus, we conclude that $\xi$ converges exponentially to zero and the rate of convergence can be bounded as

$$\|\xi(t)\| \leq \|\xi(0)\| e^{\nu(\mathbf{A}_{cl})t}. \tag{8}$$

■

# Theorem 7

***Proof***: The proof follows the reasoning in Theorem 1 (from Theorem 5.1, Chapter 2, [Kokotović et al., 1999]). In the book, via similarity transformation, the authors express and decouple the original slow and fast variables into the approximated variables.

The singularly perturbed system dynamics (3.31) is slightly different from the one in the book. Thus, we adapt their results to our system model to obtain the approximation results. The similarity transformations [Kokotović et al., 1999] for the decoupling of the dynamics are

$$\begin{bmatrix} y \\ \xi \end{bmatrix} = \begin{bmatrix} I_{m.n_x} & \epsilon\Psi(\epsilon) \\ -\Omega(\epsilon) & I_{n_x(n-m)} - \epsilon\Omega(\epsilon)\Psi(\epsilon) \end{bmatrix} \begin{bmatrix} y_s \\ \xi_f \end{bmatrix} \tag{9}$$

$$\text{and} \quad \begin{bmatrix} y_s \\ \xi_f \end{bmatrix} = \begin{bmatrix} I_{m.n_x} - \epsilon\Psi(\epsilon)\Omega(\epsilon) & -\epsilon\Psi(\epsilon) \\ \Omega(\epsilon) & I_{n_x(n-m)} \end{bmatrix} \begin{bmatrix} y \\ \xi \end{bmatrix}, \tag{10}$$

where the functions $\Omega$ and $\Psi$ should satisfy the following,

$$\begin{cases} R(\Omega(\epsilon),\epsilon) = \epsilon A_{21} - \epsilon A_{22}^{1}\Omega(\epsilon) - A_{22}^{2}\Omega(\epsilon) + \epsilon\Omega(\epsilon)A_{11} - \epsilon\Omega(\epsilon)A_{12}\Omega(\epsilon) = 0, \\ S(\Psi(\epsilon),\epsilon) = \epsilon A_{11}\Psi(\epsilon) + A_{12} - \epsilon A_{12}\Omega(\epsilon)\Psi(\epsilon) - \epsilon\Psi(\epsilon)A_{22}^{1} - \Psi(\epsilon)A_{22}^{2} - \epsilon\Psi(\epsilon)\Omega(\epsilon)A_{12} = 0. \end{cases} \tag{11}$$

The approximation of $\Omega$ and $\Psi$, obtained with the Taylor development w.r.t. $\epsilon$, are

$$\begin{cases} \Omega(\epsilon) = \epsilon(A_{22}^{2})^{-1}A_{21} + \mathcal{O}(\epsilon^{2}), \\ \Psi(\epsilon) = A_{12}(A_{22}^{2})^{-1} + \epsilon((A_{22}^{2})^{-1}A_{11}A_{12}(A_{22}^{2})^{-1} - A_{12}) + \mathcal{O}(\epsilon^{2}). \end{cases} \tag{12}$$

From Lemma 8, we know that $\xi(t)$ and $\xi_f(t_f)$ converge to zero exponentially fast as $t$ and $t_f$ tend to $+\infty$, respectively. Thus, we can claim that $\Omega(\epsilon)y_s(t)$ has an exponential decrease to zero w.r.t. $t$.

Finally, from the above transformation (9)-(10) and (12), we obtain the approximations (3.34).

■

# Proposition 3.64

***Proof***: The cost $J_k$ is split into the sum of the internal and external costs and composite term as follows,

$$\begin{aligned} J_k(T,+\infty) = & \underbrace{\int_T^{+\infty} x_k^\top(\mathcal{L}_k^{int}\otimes I_{n_x})x_k + u_k^{int\top}(I_{n_k}\otimes R_k)u_k^{int}\,\mathrm{dt}}_{J_k^{int}} \\ & + \underbrace{\int_T^{+\infty} x^\top(\mathcal{L}_k^{ext}\otimes I_{n_x})x + u_k^{ext\top}(I_{n_k}\otimes R_k)u_k^{ext}\,\mathrm{dt}}_{J_k^{ext}} \\ & + \underbrace{2\int_T^{+\infty} u_k^{ext\top}(I_{n_k}\otimes R_k)u_k^{int}\,\mathrm{dt}}_{J_k^{comp}}. \end{aligned} \tag{13}$$

Then, we bound the internal and external costs from time $T$ to infinity. We proceed similarly with the composite term.

**Internal cost**

For all $k \in \mathcal{M}$ and for $t \geq T$,

$$J_k^{int}(T, +\infty) = \int_T^{+\infty} x_k^\top (\mathcal{L}_k^{int} \otimes I_{n_x}) x_k + u_k^{int\top} (I_{n_k} \otimes R_k) u_k^{int} \,\mathrm{dt}\,. \tag{14}$$

Substituting $x_k = H_k y_k + Z_k \xi_k$ from equation (3.14) into (14) and with $H_k^\top (\mathcal{L}_k^{int} \otimes I_{n_x}) = 0$, it yields

$$\begin{aligned}
J_k^{int}(T, +\infty) &= \int_T^{+\infty} \xi_k^\top Z_k^\top \left( (\mathcal{L}_k^{int} \otimes I_{n_x}) + (\mathcal{L}_k^{int\top} \mathcal{L}_k^{int} \otimes K_k^{int\top} R_k K_k^{int}) \right) Z_k \xi_k \,\mathrm{dt} \\
&= \int_T^{+\infty} \xi_k^\top \left( (\Lambda_k^{int} \otimes I_{n_x}) + \left( \Lambda_k^{int2} \otimes P_k^{int\top} B \frac{R_k^{-1}}{n_k^2} B^\top P_k^{int} \right) \right) \xi_k \,\mathrm{dt} \\
&= \int_T^{+\infty} \xi_k^\top \left( (\Lambda_k^{int} \otimes I_{n_x}) + \left( I_{n_k - 1} \otimes P_k^{int\top} B R_k^{-1} B^\top P_k^{int} \right) \right) \xi_k \,\mathrm{dt} \\
&= \int_T^{+\infty} \xi_k^\top \mathbf{M}_1 \xi_k \,\mathrm{dt}
\end{aligned} \tag{15}$$

where $\mathbf{M}_1 = (\Lambda_k^{int} \otimes I_{n_x}) + \left( I_{n_k - 1} \otimes P_k^{int\top} B R_k^{-1} B^\top P_k^{int} \right)$. Taking the norm on both sides,

$$J_k^{int}(T, +\infty) \leq \|\mathbf{M}_1\| \int_T^{+\infty} \|\xi_k(t)\|^2 \,\mathrm{dt} \tag{16}$$

where

$$\begin{aligned}
\|\mathbf{M}_1\| &\leq \|\Lambda_k^{int}\| + \|P_k^{int\top} B R_k{}^{-1} B^\top P_k^{int})\| \\
&= n_k + \|P_k^{int} A + A^\top P_k^{int} + n_k I_{n_x}\| \\
&\leq 2(n_k + \Lambda_{max}(n_k)\|A\|) = \mathbf{M}_2.
\end{aligned} \tag{17}$$

Then, it follows,

$$\begin{aligned}
J_k^{int}(T, +\infty) &\leq \mathbf{M}_2 \int_T^{+\infty} \|\xi_k(t)\|^2 \,\mathrm{dt} \\
&\leq \mathbf{M}_2 \int_T^{+\infty} \|\xi(t)\|^2 \,\mathrm{dt}\,.
\end{aligned} \tag{18}$$

From Lemma 8 and equation (8), we have $\|\xi(t)\| \leq \|\xi(T)\| e^{\nu(\mathbf{A}_{cl})(t-T)}$, for all $t \in [T, +\infty)$. Then, with $\nu(\mathbf{A}_{cl}) < 0$, we have,

$$\int_T^{+\infty} \|\xi(t)\|^2 \,\mathrm{dt} \leq -\frac{\|\xi(T)\|^2}{2\nu(\mathbf{A}_{cl})} = \mathbf{C}_3 \|\xi(T)\|^2, \tag{19}$$

where $\mathbf{C}_3 = -\frac{1}{2\nu(\mathbf{A}_{cl})}$.

Thus, from (18)-(19) and the approximation of $\xi$ in equation (3.34),

$$\begin{aligned} J_k^{int}(T,+\infty) &\leq \mathbf{M}_2\mathbf{C}_3\|\xi_f(T)+\mathcal{O}(\epsilon)\|^2 \\ &\leq \mathbf{M}_2\mathbf{C}_3\left(\|\xi_f(T)\|^2+2\mathcal{O}(\epsilon)\|\xi_f(T)\|+\mathcal{O}(\epsilon^2)\right). \end{aligned} \tag{20}$$

Finally, replacing $\|\xi_f(T)\| \leq \epsilon\sqrt{n-m}$ from (3.63) in (20) we have

$$J_k^{int}(T,+\infty) \leq \mathbf{M}_2\mathbf{C}_3(n-m)\epsilon^2+\mathcal{O}(\epsilon^2)=\mathcal{O}(\epsilon^2). \tag{21}$$

__External cost__

First, we recast the collective external control (3.4) in the external error variable $Y_k$, as follows

$$\begin{aligned} u_k^{ext}(t) &= -(I_{n_k}\otimes K_k^{ext})(\mathcal{L}_{k,row}^{ext}\otimes I_{n_x})x(t) \\ &= -(\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext})(Hy(t)+Z\xi(t)) \\ &= -(\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext})(Hy_s(t)+\epsilon H\Psi(\epsilon)\xi_f(t_f)+Z\xi(t)) \\ &= (\mathcal{L}_{k,red}^{ext}\otimes K_k^{ext})U_{-k}Y_k(t)-(\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext})(\epsilon H\Psi(\epsilon)\xi_f(t_f)+Z\xi(t)), \end{aligned} \tag{22}$$

where $\mathcal{L}_{k,row}^{ext}$ is the $k$-th block-row of $\mathcal{L}^{ext}$ and $\mathcal{L}_{k,red}^{ext}$ is obtained by removing the $\mathcal{L}_{k,k}^{ext}$ block from $\mathcal{L}_{k,row}^{ext}$. Then, it yields

$$\begin{aligned} u_k^{ext\top}(t)(I_{n_k}\otimes R_k)u_k^{ext}(t) &= Y_k^\top(t)Q_{k,2}^{ext}Y_k(t)+\epsilon^2\xi_f^\top(t_f)D_1\xi_f(t_f)+\xi^\top(t)D_2\xi(t) \\ &\quad -\epsilon Y_k^\top(t)D_3\xi_f(t_f)-Y_k^\top(t)D_4\xi(t)+\epsilon\xi^\top(t)D_5\xi_f(t_f), \end{aligned} \tag{23}$$

where

$$\begin{cases} Q_{k,2}^{ext} = U_{-k}^\top(\mathcal{L}_{k,red}^{ext\top}\mathcal{L}_{k,red}^{ext}\otimes K_k^{ext\top}R_kK_k^{ext})U_{-k}, \\ D_1 = \Psi(\epsilon)^\top H^\top(\mathcal{L}_{k,row}^{ext\top}\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext\top}R_kK_k^{ext})H\Psi(\epsilon), \\ D_2 = Z^\top(\mathcal{L}_{k,row}^{ext\top}\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext\top}R_kK_k^{ext})Z, \\ D_3 = 2U_{-k}^\top(\mathcal{L}_{k,red}^{ext\top}\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext\top}R_kK_k^{ext})H\Psi(\epsilon), \\ D_4 = 2U_{-k}^\top(\mathcal{L}_{k,red}^{ext\top}\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext\top}R_kK_k^{ext})Z, \\ D_5 = 2Z^\top(\mathcal{L}_{k,row}^{ext\top}\mathcal{L}_{k,row}^{ext}\otimes K_k^{ext\top}R_kK_k^{ext})H\Psi(\epsilon). \end{cases} \tag{24}$$

Secondly, let consider the state part in the external cost. To simplify the expression, we use $(\mathcal{L}_k^{ext}\otimes I_{n_x})Hy_s(t) = -(\mathcal{L}_{k,col}^{ext}\otimes I_{n_x})U_{-k}Y_k(t)$ where $\mathcal{L}_{k,col}^{ext}$ is the matrix $\mathcal{L}_k^{ext}$ with its $k$-th block-column removed. Then, we obtain

$$\begin{aligned} x^\top(t)(\mathcal{L}_k^{ext}\otimes I_{n_x})x(t) &= (\star)^\top(\mathcal{L}_k^{ext}\otimes I_{n_x})(Hy_s(t)+\epsilon H\Psi(\epsilon)\xi_f(t_f)+Z\xi(t)) \\ &= n_kY_k^\top(t)Q_{k,1}^{ext}Y_k(t)+\epsilon^2\xi_f(t_f)^\top M_1\xi_f(t_f)+\xi^\top(t)M_2\xi(t) \\ &\quad -\epsilon Y_k^\top(t)M_3\xi_f(t_f)-Y_k^\top(t)M_4\xi(t)+\epsilon\xi^\top(t)M_5\xi_f(t_f) \end{aligned} \tag{25}$$

where

$$\begin{cases} M_1 = \Psi(\epsilon)^\top H^\top (\mathcal{L}_k^{ext} \otimes I_{n_x}) H \Psi(\epsilon) \\ M_2 = Z^\top (\mathcal{L}_k^{ext} \otimes I_{n_x}) Z \\ M_3 = 2U_{-k}^\top (\mathcal{L}_{k,col}^{ext\top} \otimes I_{n_x}) H \Psi(\epsilon) \\ M_4 = 2U_{-k}^\top (\mathcal{L}_{k,col}^{ext\top} \otimes I_{n_x}) Z \\ M_5 = 2Z^\top (\mathcal{L}_k^{ext} \otimes I_{n_x}) H \Psi(\epsilon). \end{cases} \tag{26}$$

Then, replacing (23) and (25) into the expression of (13), we get

$$\begin{aligned} J_k^{ext}(T, +\infty) &= n_k \int_T^{+\infty} Y_k^\top(t) Q_{k,1}^{ext} Y_k(t) + Y_k^\top(t) \frac{Q_{k,2}^{ext}}{n_k} Y_k(t)\,\mathrm{d}t + \Delta_1 \\ &= n_k \overline{J}_k^{ext}(T, +\infty) + \Delta_1, \end{aligned} \tag{27}$$

where $\Delta_1 = \Delta_1^1 + \Delta_1^2 + \Delta_1^3 + \Delta_1^4 + \Delta_1^5$ and

$$\begin{cases} \Delta_1^1 = \epsilon^2 \displaystyle\int_T^{+\infty} \xi_f(t_f)^\top \left(M_1 + D_1\right) \xi_f(t_f)\,\mathrm{d}t, \\ \Delta_1^2 = \displaystyle\int_T^{+\infty} \xi^\top(t) \left(M_2 + D_2\right) \xi(t)\,\mathrm{d}t, \\ \Delta_1^3 = -\epsilon \displaystyle\int_T^{+\infty} Y_k^\top(t) \left(M_3 + D_3\right) \xi_f(t_f)\,\mathrm{d}t, \\ \Delta_1^4 = - \displaystyle\int_T^{+\infty} Y_k^\top(t) \left(M_4 + D_4\right) \xi(t)\,\mathrm{d}t, \\ \Delta_1^5 = \epsilon \displaystyle\int_T^{+\infty} \xi^\top(t) \left(M_5 + D_5\right) \xi_f(t_f)\,\mathrm{d}t\,. \end{cases} \tag{28}$$

We recall that from the design of internal and external control, we have for all $t \geq 0$

$$\begin{cases} Y(t) = e^{\mathbf{A}_{cl} t} Y(0) \\ \xi_f(t) = e^{\bar{A}_{22}^2 t} \xi_f(0) \end{cases} \Rightarrow \begin{cases} \|Y(t)\| \leq e^{\nu(\mathbf{A}_{cl}) t} \|Y(0)\| \\ \|\xi_f(t)\| \leq e^{\nu(\bar{A}_{22}^2) t} \|\xi_f(0)\| \end{cases} \tag{29}$$

where $\mathbf{A}_{cl} = diag(\mathbf{A}_{1,cl}, ..., \mathbf{A}_{m,cl})$ is the closed-loop dynamics of the external error. Also, we have $\nu(\mathbf{A}_{cl}) < 0$, $\nu(\bar{A}_{22}^2) < 0$ and $\xi_f(t) = e^{\bar{A}_{22}^2 t} \xi_f(0) = e^{A_{22}^2 t_f} \xi_f(0) = \xi_f(t_f)$.

**Bound for $\Delta_1^1$**

$$\begin{aligned} \Delta_1^1 &\leq \epsilon^2 \|M_1 + D_1\| \int_T^{+\infty} \|\xi_f(t_f)\|^2\,\mathrm{d}t \\ &\leq -\epsilon^2 \frac{\|M_1 + D_1\| \|\xi_f(0)\|^2}{2\nu(\bar{A}_{22}^2)} e^{2\nu(\bar{A}_{22}^2) T} = \mathcal{O}(\epsilon^2). \end{aligned} \tag{30}$$

**Bound for $\Delta_1^2$**

In the same manner as the internal cost, we have

$$\begin{aligned} \Delta_1^2 &\leq \|M_2 + D_2\| \int_T^{+\infty} \|\xi(t)\|^2 \,\mathrm{dt} \\ &\leq \mathbf{M}_3 \|M_2 + D_2\| \|\xi(T)\|^2 \\ &\leq \mathbf{M}_3 \|M_2 + D_2\| \left( \|\xi_f(T)\|^2 + 2\mathcal{O}(\epsilon)\|\xi_f(T)\| + \mathcal{O}(\epsilon^2) \right). \end{aligned} \tag{31}$$

Replacing $\|\xi_f(T)\| \leq \epsilon\sqrt{n-m}$ in the above inequality,

$$\Delta_1^2 \leq \mathbf{M}_3 \|M_2 + D_2\| \left( \epsilon^2(n-m) + 2\epsilon\mathcal{O}\sqrt{n-m} + \mathcal{O}(\epsilon^2) \right) = \mathcal{O}(\epsilon^2). \tag{32}$$

**Bound for $\Delta_1^3$**

It yields

$$\begin{aligned} \Delta_1^3 &\leq \epsilon \|M_3 + D_3\| \int_T^{+\infty} \|Y_k(t)\| \|\xi_f(t_f)\| \,\mathrm{dt} \\ &\leq \epsilon \|M_3 + D_3\| \|Y(0)\| \|\xi_f(0)\| \int_T^{+\infty} e^{\nu(\mathbf{A}_{cl})t} e^{\nu(\bar{A}_{22}^2)t} \,\mathrm{dt} \\ &= -\epsilon \frac{\|M_3 + D_3\| \|Y(0)\| \|\xi_f(0)\|}{\nu(\mathbf{A}_{cl}) + \nu(\bar{A}_{22}^2)} e^{\left(\nu(\mathbf{A}_{cl}) + \nu(\bar{A}_{22}^2)\right)T} \\ &= \mathcal{O}(\epsilon). \end{aligned} \tag{33}$$

Similarly, $\Delta_1^4$ and $\Delta_1^5$ are of order $\mathcal{O}(\epsilon)$. Finally, from (27) and bounds in (28) for $\Delta_1$, we obtain

$$J_k^{ext}(T, +\infty) = n_k \bar{J}_k^{ext}(T, +\infty) + \mathcal{O}(\epsilon). \tag{34}$$

**Composite term**

In order the simplify the expression, we rewrite the external control (22) as

$$u_k^{ext}(t) = -\mathbf{C}_4 Y_k(t) - \epsilon \mathbf{C}_5 \xi_f(t_f) - \mathbf{C}_6 \xi(t), \tag{35}$$

where $\mathbf{C}_4 = (\mathcal{L}_{k,red}^{ext} \otimes K_k^{ext}) U_{-k}$, $\mathbf{C}_5 = (\mathcal{L}_{k,row}^{ext} \otimes K_k^{ext}) H \Psi(\epsilon)$ and $\mathbf{C}_6 = (\mathcal{L}_{k,row}^{ext} \otimes K_k^{ext}) Z$. As for the internal control, from (3.14) and (3.4) we have

$$u_k^{int}(t) = (\mathcal{L}_k^{int} \otimes K_k^{int}) Z_k \xi_k(t) = \mathbf{C}_7 \xi_k(t). \tag{36}$$

Then, the bound for the composite term is

$$
\begin{aligned}
&J_k^{comp}(T,+\infty) = 2\left\|\int_T^{+\infty} u_k^{ext\top}(t) R_k u_k^{int}(t)\,\mathrm{dt}\right\| \\
&\leq 2\|R_k\| \int_T^{+\infty} \|u_k^{ext\top}(t)\|.\|u_k^{int}(t)\|\,\mathrm{dt} \\
&\leq 2\|R_k\| \int_T^{+\infty} \|-\mathbf{C}_4 Y_k(t) - \epsilon \mathbf{C}_5 \xi_f(t_f) - \mathbf{C}_6 \xi(t)\|.\|\mathbf{C}_7 \xi_k(t)\|\,\mathrm{dt} \\
&\leq 2\|R_k\| \Bigg( \|\mathbf{C}_4\|.\|\mathbf{C}_7\| \int_T^{+\infty} \|Y_k(t)\|.\|\xi_k(t)\|\,\mathrm{dt} \; + \epsilon \|\mathbf{C}_5\|.\|\mathbf{C}_7\| \int_T^{+\infty} \|\xi_f(t_f)\|.\|\xi_k(t)\|\,\mathrm{dt} \\
&\qquad + \|\mathbf{C}_6\|.\|\mathbf{C}_7\| \int_T^{+\infty} \|\xi(t)\|.\|\xi_k(t)\|\,\mathrm{dt} \Bigg).
\end{aligned}
\tag{37}
$$

With simple calculation it can be shown that the first integral in the above equation is of order $\mathcal{O}(\epsilon)$ and the second and the third integrals are of order $\mathcal{O}(\epsilon^2)$. This concludes our proof.

■

# Bibliographie


Adhikari, B., Morărescu, I. ., and Panteley, E. (2021). An emerging dynamics approach for synchronization of linear heterogeneous agents interconnected over switching topologies. *IEEE Control Systems Letters*, 5(1) :43–48.

Ali, U. and Wardi, Y. (2015). Multiple shooting technique for optimal control problems with application to power aware networks. *IFAC-PapersOnLine*, 48(27) :286–290.

Alpcan, T. and Başar, T. (2010). *Network security : A decision and game-theoretic approach*. Cambridge University Press.

Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks : A survey. *Ad hoc networks*, 7(3) :537–568.

Anderson, B. D., Yu, C., Fidan, B., and Hendrickx, J. M. (2008). Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine*, 28(6) :48–63.

Antonelli, G. (2013). Interconnected dynamic systems : An overview on distributed control. *IEEE Control Systems Magazine*, 33(1) :76–88.

Baillieul, J. and Antsaklis, P. J. (2007). Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1) :9–28.

Bapat, R. B. (2010). *Graphs and matrices*, volume 27. Springer.

Başar, T. and Olsder, G. J. (1998). *Dynamic noncooperative game theory*. SIAM.

Bauso, D., Giarré, L., and Pesenti, R. (2006). Non-linear protocols for optimal distributed consensus in networks of dynamic agents. *Systems & Control Letters*, 55(11) :918–928.

Bertsekas, D. and Tsitsiklis, J. (1989). *Parallel and distributed computation : numerical methods*. Prentice-Hall.

Biyik, E. and Arcak, M. (2008). Area aggregation and time-scale modeling for sparse nonlinear networks. *Systems & Control Letters*, 57(2) :142–149.

Blondel, V. D., Hendrickx, J. M., Olshevsky, A., and Tsitsiklis, J. N. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2996–3000. IEEE.

Blondel, V. D., Hendrickx, J. M., and Tsitsiklis, J. N. (2009). On krause's multi-agent consensus model with state-dependent connectivity. *IEEE transactions on Automatic Control*, 54(11) :2586–2597.

Boker, A. M., Nudell, T. R., and Chakrabortty, A. (2015). On aggregate control of clustered consensus networks. In *2015 American Control Conference (ACC)*, pages 5527–5532.

Borrelli, F. and Keviczky, T. (2008). Distributed lqr design for identical dynamically decoupled systems. *IEEE Transactions on Automatic Control*, 53(8) :1901–1912.

Braess, D. (1968). Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1) :258–268.

Braess, D., Nagurney, A., and Wakolbinger, T. (2005). On a paradox of traffic planning. *Transportation science*, 39(4) :446–450.

Brandstadt, A., Le, V. B., and Spinrad, J. (1999). *Graph Classes : A Survey*. SIAM Monographs on Discrete Mathematics and Applications.

Bullo, F., Cortés, J., and Martinez, S. (2009). *Distributed control of robotic networks*. Princeton University Press.

Cao, Y., Yu, W., Ren, W., and Chen, G. (2012). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1) :427–438.

Cao, Y. U., Kahng, A. B., and Fukunaga, A. S. (1997). Cooperative mobile robotics : Antecedents and directions. In *Robot colonies*, pages 7–27. Springer.

Chen, F., Ren, W., et al. (2019). On the control of multi-agent systems : A survey. *Foundations and Trends® in Systems and Control*, 6(4) :339–499.

Chen, Y. Q. and Wang, Z. (2005). Formation control : a review and a new consideration. In *2005 IEEE/RSJ International conference on intelligent robots and systems*, pages 3181–3186. IEEE.

Chow, J. and Kokotović, P. (1985). Time scale modeling of sparse dynamic networks. *IEEE Transactions on Automatic Control*, 30(8) :714–722.

Chow, Ed., J. H. (1982). Time-scale modeling of dynamic networks with applications to power systems. *Lecture Notes in Control and Information Sciences*.

Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2) :243–255.

Das, A. and Lewis, F. L. (2010). Distributed adaptive control for synchronization of unknown nonlinear networked systems. *Automatica*, 46(12) :2014–2021.

DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association*, 69(345) :118–121.

Dimarogonas, D. V., Frazzoli, E., and Johansson, K. H. (2011). Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5) :1291–1297.

Dörfler, F. and Bullo, F. (2014). Synchronization in complex networks of phase oscillators : A survey. *Automatica*, 50(6) :1539–1564.

Dorri, A., Kanhere, S. S., and Jurdak, R. (2018). Multi-agent systems : A survey. *Ieee Access*, 6 :28573–28593.

Erdos, P., Rényi, A., et al. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1) :17–60.

Fax, J. A. (2001). *Optimal and cooperative control of vehicle formations*. Ph.D. dissertation, California Institute of Technology.

Fax, J. A. and Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control*, 49(9) :1465–1476.

Ge, X., Yang, F., and Han, Q.-L. (2017). Distributed networked control systems : A brief overview. *Information Sciences*, 380 :117–131.

Godsil, C. and Royle, G. F. (2001). *Algebraic graph theory*, volume 207. Springer Science & Business Media.

Goldenberg, D. K., Lin, J., Morse, A. S., Rosen, B. E., and Yang, Y. R. (2004). Towards mobility as a network control primitive. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 163–174.

Hassan, J. and Shamma, J. (2016). Decentralized energy aware co-optimization of mobility and communication in multiagent systems. In *55th IEEE CDC*.

Hegselmann, R., Krause, U., et al. (2002). Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3).

Holland, P. W. and Leinhardt, S. (1971). Transitivity in structural models of small groups. *Comparative group studies*, 2(2) :107–124.

Hong, Y., Hu, J., and Gao, L. (2006). Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42(7) :1177–1182.

Hoppensteadt, F. (1967). Stability in systems with parameter. *Journal of Mathematical Analysis and Applications*, 18(1) :129–134.

Hoppensteadt, F. (1971). Properties of solutions of ordinary differential equations with small parameters. *Communications on Pure and Applied Mathematics*, 24(6) :807–840.

Isidori, A., Marconi, L., and Casadei, G. (2014). Robust output synchronization of a network of heterogeneous nonlinear agents via nonlinear regulation theory. *IEEE Transactions on Automatic Control*, 59(10) :2680–2691.

Iwasaki, T., Skelton, R., and Geromel, J. (1994). Linear quadratic suboptimal control with static output feedback. *Systems & Control Letters*, 23(6) :421–430.

Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6) :988–1001.

Jaleel, H., Wardi, Y., and Egerstedt, M. (2014). Minimizing mobility and communication energy in robotic networks : An optimal control approach. In *2014 American Control Conference*, pages 2662–2667. IEEE.

Jiao, J., Trentelman, H. L., and Camlibel, M. K. (2019a). Distributed linear quadratic optimal control : Compute locally and act globally. *IEEE Control Systems Letters*, 4(1) :67–72.

Jiao, J., Trentelman, H. L., and Camlibel, M. K. (2019b). A suboptimality approach to distributed linear quadratic optimal control. *IEEE Transactions on Automatic Control*, 65(3) :1218–1225.

Khalil, H. K. and Grizzle, J. W. (2002). *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ.

Knödel, W. (2013). *Graphentheoretische methoden und ihre anwendungen*, volume 13. Springer-Verlag.

Kokotović, P., Khalil, H. K., and O'Reilly, J. (1999). *Singular Perturbation Methods in Control : Analysis and Design*. Society for Industrial and Applied Mathematics.

Kokotovic, P. V., O'Malley Jr, R. E., and Sannuti, P. (1976). Singular perturbations and order reduction in control theory-an overview. *Automatica*, 12(2) :123–132.

Kolata, G. (1990). What if they closed 42d street and nobody noticed. *New York Times*, 25 :38.

Labit, Y., Peaucelle, D., and Henrion, D. (2002). Sedumi interface 1.02 : a tool for solving lmi problems with sedumi. In *Proceedings. IEEE International Symposium on Computer Aided Control System Design*, pages 272–277. IEEE.

Lamnabhi-Lagarrigue, F., Annaswamy, A., Engell, S., Isaksson, A., Khargonekar, P., Murray, R. M., Nijmeijer, H., Samad, T., Tilbury, D., and Van den Hof, P. (2017). Systems & control for the future of humanity, research agenda : Current and future roles, impact and grand challenges. *Annual Reviews in Control*, 43 :1–64.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., and Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. *arXiv preprint arXiv :1711.00832*.

Lasaulce, S. and Tembine, H. (2011). *Game theory and learning for wireless networks : fundamentals and applications*. Academic Press.

Laub, A. J. (2005). *Matrix analysis for scientists and engineers*, volume 91. Siam.

Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M., and Davis, R. E. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1) :48–74.

Levinson, N. (1950). Perturbations of discontinuous solutions of non-linear systems of differential equations. *Acta Mathematica*, 82(1) :71–106.

Li, J., Feng, T., Zhang, J., and Yan, F. (2021). Optimal distributed cooperative control for multi-agent systems with constrains on convergence speed and control input. *Neurocomputing*, 426 :14–25.

Li, J., Modares, H., Chai, T., Lewis, F. L., and Xie, L. (2017). Off-policy reinforcement learning for synchronization in multiagent graphical games. *IEEE transactions on neural networks and learning systems*, 28(10) :2434–2445.

Li, N. and Marden, J. R. (2013). Designing games for distributed optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7(2) :230–242.

Li, Z., Ren, W., Liu, X., and Fu, M. (2012). Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols. *IEEE Transactions on Automatic Control*, 58(7) :1786–1791.

Lin, Z., Francis, B., and Maggiore, M. (2005). Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on automatic control*, 50(1) :121–127.

Lorenz, J. (2007). Continuous opinion dynamics under bounded confidence : A survey. *International Journal of Modern Physics C*, 18(12) :1819–1838.

Luo, B., Wu, H.-N., Huang, T., and Liu, D. (2014). Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design. *Automatica*, 50(12) :3281–3290.

Lynch, N. A. (1996). *Distributed algorithms*. Elsevier.

Madani, K. (2010). Game theory and water resources. *Journal of Hydrology*, 381(3-4) :225–238.

Marden, J. R., Arslan, G., and Shamma, J. S. (2007). Connections between cooperative control and potential games illustrated on the consensus problem. In *2007 European Control Conference (ECC)*, pages 4604–4611. IEEE.

Martin, S., Morarescu, I.-C., and Nesic, D. (2016). Time scale modeling for consensus in sparse directed networks with time-varying topologies. *2016 IEEE 55th Conference on Decision and Control (CDC)*.

Martinez, S., Cortes, J., and Bullo, F. (2007). Motion coordination with distributed information. *IEEE control systems magazine*, 27(4) :75–88.

Mesbahi, M. and Egerstedt, M. (2010). *Graph theoretic methods in multiagent networks*. Princeton University Press.

Modares, H., Nageshrao, S. P., Lopes, G. A. D., Babuška, R., and Lewis, F. L. (2016). Optimal model-free output synchronization of heterogeneous systems using off-policy reinforcement learning. *Automatica*, 71 :334–341.

Moreau, L. (2005). Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on automatic control*, 50(2) :169–182.

Morgenstern, O. and Von Neumann, J. (1953). *Theory of games and economic behavior*. Princeton university press.

Murray, R. M. (2007). Recent research in cooperative control of multivehicle systems.

Mytum-Smithson, J. (2005). Wireless sensor networks : An information processing approach. *Sensor Review*, 25(2).

Naidu, D. (2002). Singular perturbations and time scales in control theory and applications : an overview. *Dynamics of Continuous Discrete and Impulsive Systems Series B*, 9 :233–278.

Naidu, D. S. and Calise, A. J. (2001). Singular perturbations and time scales in guidance and control of aerospace systems : A survey. *Journal of Guidance, Control, and Dynamics*, 24(6) :1057–1078.

Nedic, A., Ozdaglar, A., and Parrilo, P. A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4) :922–938.

Nešić, D. and Teel, A. R. (2001). Input-to-state stability for nonlinear time-varying systems via averaging. *Mathematics of Control, Signals and Systems*, 14(3) :257–280.

Oh, K.-K., Park, M.-C., and Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, 53 :424–440.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems : Algorithms and theory. *IEEE Transactions on automatic control*, 51(3) :401–420.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1) :215–233.

Olfati-Saber, R. and Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49 :1520–1533.

Olfati-Saber, R. and Murray, R. M. (2003). Consensus protocols for networks of dynamic agents.

O'Malley, R. (1971). Boundary layer methods for nonlinear initial value problems. *SIAM Review*, 13(4) :425–434.

Panteley, E. and Loria, A. (2017). Synchronization and dynamic consensus of heterogeneous networked systems. *IEEE Transactions on Automatic Control*, 62(8) :3758–3773.

Parker, L. E., Rus, D., and Sukhatme, G. S. (2016). Multiple mobile robot systems. In *Springer Handbook of Robotics*, pages 1335–1384. Springer.

Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R., and Bagos, P. G. (2011). Using graph theory to analyze biological networks. *BioData mining*, 4(1) :10.

Perlaza, S. M., Tembine, H., Lasaulce, S., and Debbah, M. (2012). Quality-of-service provisioning in decentralized networks : A satisfaction equilibrium approach. *IEEE Journal of Selected Topics in Signal Processing*, 6(2) :104–116.

Qin, J., Ma, Q., Shi, Y., and Wang, L. (2016). Recent advances in consensus of multi-agent systems : A brief survey. *IEEE Transactions on Industrial Electronics*, 64(6) :4972–4983.

Qu, Z. (2009). *Cooperative control of dynamical systems : applications to autonomous vehicles*. Springer Science & Business Media.

Rejeb, J. B., Morărescu, I.-C., and Daafouz, J. (2018). Control design with guaranteed cost for synchronization in networks of linear singularly perturbed systems. *Automatica*, 91 :89–97.

Ren, W. (2007). Multi-vehicle consensus with a time-varying reference state. *Systems & Control Letters*, 56(7-8) :474–483.

Ren, W. and Beard, R. W. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5) :655–661.

Ren, W. and Beard, R. W. (2008). *Distributed consensus in multi-vehicle cooperative control*, volume 27. Springer.

Ren, W., Beard, R. W., and Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1859–1864. IEEE.

Ren, W., Beard, R. W., and Atkins, E. M. (2007). Information consensus in multivehicle cooperative control. *IEEE Control systems magazine*, 27(2) :71–82.

Ren, W. and Cao, Y. (2011). *Distributed coordination of multi-agent networks : emergent problems, models, and issues*, volume 1. Springer.

Romeres, D., DÃ¶rfler, F., and Bullo, F. (2013). Novel results on slow coherency in consensus and power networks. In *2013 European Control Conference (ECC)*, pages 742–747.

Ross, S. and Chaib-draa, B. (2006). Satisfaction equilibrium : Achieving cooperation in incomplete information games. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 61–72. Springer.

Saksena, V. R., O'reilly, J., and Kokotovic, P. V. (1984). Singular perturbations and time-scale methods in control theory : survey 1976–1983. *Automatica*, 20(3) :273–293.

Scardovi, L. and Sepulchre, R. (2009). Synchronization in networks of identical linear systems. *Automatica*, 45(11) :2557–2562.

Semsar-Kazerooni, E. and Khorasani, K. (2009). A game theory approach to multi-agent team cooperation. In *2009 American Control Conference*, pages 4512–4518. IEEE.

Seuret, A., Dimarogonas, D. V., and Johansson, K. H. (2008). Consensus under communication delays. In *2008 47th IEEE Conference on Decision and Control*, pages 4922–4927. IEEE.

Shamma, J. (2008). *Cooperative control of distributed multi-agent systems*. John Wiley & Sons.

Smith, J. M. and Price, G. R. (1973). The logic of animal conflict. *Nature*, 246(5427) :15–18.

Strogatz, S. (2004). *Sync : The emerging science of spontaneous order*. Penguin UK.

Su, S., Lin, Z., and Garcia, A. (2015). Distributed synchronization control of multiagent systems with unknown nonlinearities. *IEEE Transactions on Cybernetics*, 46(1) :325–338.

Syed, U. and Schapire, R. E. (2008). A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pages 1449–1456.

Tanner, H. G., Jadbabaie, A., and Pappas, G. J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic control*, 52(5) :863–868.

Teel, A. R., Moreau, L., and Nesic, D. (2003). A unified framework for input-to-state stability in systems with two time scales. *IEEE Transactions on Automatic Control*, 48(9) :1526–1544.

Tikhonov, A. (1948). On the dependence of the solutions of differential equations on a small parameter. *Matematicheskii sbornik*, 64(2) :193–204.

Tikhonov, A. N. (1952). Systems of differential equations containing small parameters in the derivatives. *Matematicheskii sbornik*, 73(3) :575–586.

Tsitsiklis, J., Bertsekas, D., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9) :803–812.

Tsitsiklis, J. N. (1984). Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems.

Vamvoudakis, K. G., Lewis, F. L., and Hudas, G. R. (2012). Multi-agent differential graphical games : Online adaptive learning solution for synchronization with optimality. *Automatica*, 48(8) :1598–1611.

Vasil'eva, A. B. (1963). Asymptotic behaviour of solutions to certain problems involving non-linear differential equations containing a small parameter multiplying the highest derivatives. *Russian Mathematical Surveys*, 18(3) :13–84.

Veetaseveera, J., Varma, V. S., Morărescu, I., and Daafouz, J. (2019). Decentralized control for guaranteed individual costs in a linear multi-agent system : A satisfaction equilibrium approach. *IEEE Control Systems Letters*, 3(4) :918–923.

Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 75(6) :1226.

Vidal, J. (2006). Heart and soul of the city. *The guardian*, page 38.

Wang, S.-H. and Davison, E. (1973). On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, 18(5) :473–478.

Wasow, W. (1965). *Asymptotic expansions for ordinary differential equations*. Courier Dover Publications.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis : Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684) :440–442.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.

Xiao, F. and Wang, L. (2008). Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, 53(8) :1804–1816.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1) :65–78.

Yang, B. and Johansson, M. (2010). Distributed optimization and games : A tutorial overview. *Networked Control Systems*, pages 109–148.

Yang, P., Freeman, R. A., and Lynch, K. M. (2008). Multi-agent coordination by decentralized estimation and control. *IEEE Transactions on Automatic Control*, 53(11) :2480–2496.

Zhang, Y., Naidu, D. S., Cai, C., and Zou, Y. (2014). Singular perturbations and time scales in control theories and applications : An overview 2002–2012. *Int. J. Inf. Syst. Sci*, 9(1) :1–36.

# Résumé

À part Chuck Noland et Wilson, peu de personnes peuvent prétendre pouvoir échapper aux réseaux. Omniprésents dans toutes les strates de notre société, les réseaux au sens large ont progressivement infiltré notre quotidien et occupent désormais tous les paysages. Qu'il s'agisse d'un réseau électrique, d'un réseau social, d'une volée d'oiseaux en formation, ou de la propagation d'une pandémie, tous ces systèmes interconnectés montrent un grand degré d'interdépendance. Compte tenu de cet engouement pour les phénomènes physiques et sociétaux à grande échelle, les *systèmes dynamiques interconnectés* ont capté l'attention de la communauté scientifique au cours des dernières décennies, [Cao et al., 1997; Wooldridge, 2009; Mesbahi and Egerstedt, 2010]. Tant sur le plan théorique que pratique, le *systèmes multi-agents* et le *systèmes en réseau* s'avèrent être la manière la plus efficace et la plus adéquate de modéliser la dynamique des systèmes complexes à grande échelle. Afin de prévenir et de résoudre les problèmes de demain, l'analyse et la compréhension de ces systèmes semblent inévitables, [Baillieul and Antsaklis, 2007; Lamnabhi-Lagarrigue et al., 2017].

Un système multi-agents est un ensemble d'entités ou d'agents souvent décrits par deux aspects fondamentaux : la dynamique des agents, et les interactions entre eux. Communément, un tel agent peut être représenté par un robot mobile, une personne, un véhicule, un oiseau, etc. Chaque agent collabore avec ses voisins pour accomplir la tâche qui lui est assignée. Cette coordination conduit le système dans son ensemble vers un objectif commun, appelé *consensus* ou *synchronisation*, [DeGroot, 1974; Vicsek et al., 1995].

Le problème du consensus apparaît dans diverses disciplines comme la biologie [Pavlopoulos et al., 2011], la sociologie [Hegselmann et al., 2002; Lorenz, 2007; Blondel et al., 2009], les réseaux sociaux [Blondel et al., 2009] ou l'ingénierie [Martinez et al., 2007; Anderson et al., 2008; Bullo et al., 2009]. Quant à la synchronisation des systèmes multi-agents, qu'elle soit naturelle ou artificielle, on peut citer le comportement des oiseaux en vol, les bancs de poissons ou la coordination des drones, [Cortes et al., 2004; Blondel et al., 2005; Olfati-Saber, 2006; Tanner et al., 2007; Leonard et al., 2007]. Les applications sont nombreuses et variées : pelotonnage de véhicules, exploration, patrouille, alignement de satellites, réseau de capteurs distribués, formation de drones, suivi d'un leader, etc. En général, des algorithmes de contrôle sont appliqués à ces systèmes pour obtenir le comportement souhaité. Le but est de faciliter de la coordination entre les agents.

Dans la communauté du contrôle, un très grand nombre de travaux traitent des problèmes liés à la conception de lois de contrôle permettant la coordination de systèmes multi-agents. Différentes approches sont proposées mais deux en particulier retiennent notre attention. Soit, les études se concentrent sur la dynamique des agents, soit sur le type de communication et la topologie du réseau. La première caractéristique intrinsèquement

liée aux agents ne dépend que de la nature de l'agent lui-même. Plusieurs contributions considèrent les systèmes multi-agents avec une dynamique linéaire : [Jadbabaie et al., 2003; Xiao and Boyd, 2004; Moreau, 2005; Ren and Beard, 2005], une dynamique non-linéaire : [Das and Lewis, 2010; Li et al., 2012; Isidori et al., 2014; Su et al., 2015], les robots non-holonomes [Strogatz, 2004; Lin et al., 2005], ou les oscillateurs couplés [Dörfler and Bullo, 2014]. Quant à la seconde caractéristique, elle dépend fortement du type d'interactions entre les agents. Largement abordées dans la littérature, les méthodes orientées sur les propriétés des réseaux proposent de modéliser les interactions par des graphes à topologie fixe ou variable dans le temps [Hong et al., 2006; Tanner et al., 2007; Ren, 2007; Scardovi and Sepulchre, 2009], avec un délai [Seuret et al., 2008; Xiao and Wang, 2008] ou à capacité de communication limitée [Dimarogonas et al., 2011].

Dans les problèmes de consensus, les algorithmes mis en oeuvre étant sensibles aux différents types d'interconnexions, l'analyse se concentre sur la structure du réseau. Pour la synchronisation, l'objectif premier est de coordonner l'ensemble du système. Ainsi, les approches se concentrent sur la conception du contrôle tout en tenant compte de la dynamique des agents.

Dans la littérature, plusieurs contributions s'intéressent à la minimisation du coût global lors de la conception du contrôle du réseau, mais seules quelques-unes considèrent les coûts individuels. Nous désignons par un coût global, une fonction qui prend en compte l'effort de l'ensemble du réseau pour atteindre l'objectif commun. Au contraire, le coût individuel est l'effort lié à un agent ou à un petit groupe d'agents densément connectés dans le réseau.

Dans les applications de systèmes physiques en réseau, les coûts individuels peuvent présenter un intérêt pratique lorsque les agents ont une capacité de communication (courte portée des signaux sans fil, bande passante étroite) et de fonctionnement (carburant, batterie, ressources de calcul) limitée, [Anastasi et al., 2009; Goldenberg et al., 2004; Dimarogonas et al., 2011]. Par exemple, considérons le scénario du contrôle automatique de la vitesse des véhicules sur les autoroutes, où chaque véhicule souhaite suivre celui qui le précède (objectif commun de synchronisation). En même temps qu'ils accomplissent leurs tâches, nous voulons aussi nous assurer que leur consommation de carburant n'est pas trop excessive (minimisation des coûts individuels). La réduction ou la limitation de la consommation de carburant peut être considérée comme des contraintes de performance issues de spécifications techniques. Dans de telles applications, la prise en compte d'un coût global peut ne pas être équitable pour chaque véhicule. De plus, le choix entre un coût global et des coûts individuels peut avoir un impact significatif sur la stratégie de contrôle.

L'avantage de contraindre ou de minimiser un tel coût est clair, mais les résultats théoriques dans cette direction manquent cruellement dans la littérature sur les systèmes multi-agents. Seuls quelques résultats abordent le problème de la conception du contrôle pour la synchronisation des agents tout en minimisant un coût. Ces raisons nous amènent donc à nous concentrer dans cette direction. **Les objectifs principaux de cette thèse sont la conception de contrôle et l'analyse d'algorithmes de synchronisation pour les systèmes multi-agents prenant en compte les contraintes de communication, tout en assurant que les coûts d'état et de contrôle sont inférieurs à une borne donnée.** L'analyse est menée avec une attention particulière sur les systèmes multi-agents à dynamique linéaire homogène et les réseaux clusterisés, à topologie fixe dans les deux cas. Dans le premier cas, un coût individuel par agent est considéré lors de la conception du contrôle, alors qu'un coût de cluster dans le second cas. Des méthodes de contrôle distribuées sont fournies, mais les gains de contrôle peuvent être conçus de manière centralisée selon les cas.

Dans ce qui suit, nous donnons un résumé de la thèse et décrivons de brièvement les trois chapitres principaux. Une conclusion générale est présentée à la fin.

Les travaux de cette thèse portent sur la synthèse et l'analyse d'algorithmes de synchronisation pour des systèmes multi-agents avec une dynamique linéaire. Par synchronisation, nous voulons que les états de tous les agents évoluent sur la même trajectoire à partir d'un certain temps. En prenant en compte des contraintes de communication, nous proposons des architectures de commandes décentralisées, c-à-d qui n'utilisent que des informations locales. Chaque agent n'exploite que des données venant de son voisinage direct. Par exemple, en collectant les positions et les vitesses de ses voisins, l'agent en question sera capable de choisir la bonne trajectoire à suivre.

Dans une première partie, nous nous inspirons de la théorie des jeux pour proposer une loi de commande considérant un coût individuel de satisfaction par agent. Afin de faciliter l'analyse, le problème de synchronisation est d'abord reformulé en un problème de stabilisation. Ensuite, des conditions données sous forme d'inégalités matricielles linéaires permettent de vérifier si un profil de gains correspond à un équilibre de satisfaction ou non. Un ensemble de gains est un équilibre de satisfaction lorsque le coût individuel de chaque agent est borné par un seuil donné. Un algorithme itératif permettant de calculer les gains de chaque agent est présenté à la fin du Chapitre 2.

La seconde partie consacrée aux réseaux avec des clusters, se base sur la théorie des systèmes singulièrement perturbés pour présenter une loi de commande plus axée sur des réseaux de grandes envergures. L'objectif est de fournir une méthode efficace en termes de calcul pour concevoir des stratégies de contrôle qui garantissent une certaine performance sur le coût de chaque cluster. En utilisant une méthode de séparation d'échelles de temps, la conception de la loi de commande est séparée en deux parties : une commande interne et une commande externe. Les commandes internes sont liées aux synchronisations locales à l'intérieur de chaque cluster alors que les commandes externes vont synchroniser tout le réseau. Leurs conceptions se font indépendamment l'une de l'autre et tend à réduire les charges de calculs. De plus, nous montrons que la commande interne n'affecte le coût du cluster que pendant une courte période de temps. Les méthodes de conception des lois de commande sont expliquées dans le Chapitre 3.

## Chapitre 1 : Préliminaires

Les concepts de base et une revue des outils théoriques utilisés tout au long de cette thèse sont présentés. Dans la première section, l'état de l'art développe les contributions de la littérature sur la conception de commandes distribuées. La prise en compte d'une fonction coût global ou individuel lors de la synthèse de telles commandes est également discutée. Ensuite, la deuxième section rappelle quelques notions de théorie des graphes qui sont à la base de l'analyse des interactions dans les systèmes multi-agents. Enfin, des concepts issus de la théorie des jeux et des systèmes singulièrement perturbés sont présentés.

## Chapitre 2 : Commande décentralisée avec des garanties de coûts individuels

Ce chapitre traite de la conception d'une commande décentralisée visant à synchroniser un réseau tout en considérant un coût individuel pour chaque agent. En considérant une même dynamique linéaire pour tous les agents et un graphe de communication non-dirigé, nous associons à chaque agent une fonction d'énergie tenant en compte l'objectif de synchronisation mais aussi l'énergie de synchronisation utilisée par chaque agent. Afin de faciliter l'analyse, le problème de synchronisation est d'abord reformulé en un problème de stabilisation. Ensuite, nous utilisons la notion d'équilibre de satisfaction de la théorie des jeux pour garantir, si possible, un certain niveau de performance de la commande. Des conditions sous forme d'inégalités matricielles linéaires sont fournies pour vérifier si un profil de gains donné est un équilibre de satisfaction, c'est-à-dire si tous les coûts individuels sont limités par un seuil donné. De plus, sur la base d'une commande par retour de sortie, un deuxième résultat nous permet de synthétiser le gain d'un agent en supposant que les gains des autres agents sont connus. Enfin, un algorithme itératif synthétisant les profils de gain correspondant aux équilibres de satisfaction est également présenté. Une ouverture sur la puissance de calcul nécessaire à cette méthode est également faite pour les systèmes de grande taille.

### Chapter 3 : Commande composée distribuée pour un réseau avec des clusters

Nous présentons une conception de commandes distribuées pour les réseaux avec des clusters, dans lesquels les connexions au sein des clusters sont denses et entre les clusters sont éparses. Notre objectif est de fournir une méthode efficace en termes de calcul pour concevoir des stratégies de contrôle qui garantissent une certaine limite sur le coût pour chaque cluster. Sur la base de la structure du réseau et de la théorie des perturbations singulières, nous appliquons des méthodes de séparation d'échelle temporelle pour découpler le système original en dynamiques rapides (intra-cluster) et lentes (inter-cluster). Par la suite, nous synthétisons une commande distribuée composée de deux termes : l'un responsable de la synchronisation intra-cluster, et l'autre effectuant la synchronisation inter-cluster. La commande interne ne nécessite pas un effort de calcul élevé puisqu'il est obtenu par une expression analytique. Quant au contrôle externe, il est conçu en utilisant l'approche d'équilibre de satisfaction du Chapitre 2. En résumé, les commandes internes (rapides) et externes (lentes) sont conçues indépendamment l'un de l'autre et assurent un coût satisfaisant pour chaque cluster. En outre, nous montrons que le contrôle interne n'affecte le coût du cluster que pendant une courte période. Enfin, des simulations numériques soulignent le compromis entre la performance de la commande et les ressources de calculs nécessaires pour obtenir la commande requise en comparant la stratégie proposée dans le Chapitre 2, le Chapitre 3 et dans [Rejeb et al., 2018]. Bien que la commande du Chapitre 2 soit moins efficace, nous devons garder à l'esprit que la commande composée du Chapitre 3 convient mieux aux réseaux de grande échelle et présente un avantage essentiel en termes de charges de calcul et de temps. D'autre part, alors que la solution de [Rejeb et al., 2018] est très rapide à obtenir, nous observons que notre stratégie est nettement plus performante que celle de [Rejeb et al., 2018].

## Conclusion

Cette thèse est consacrée aux algorithmes de synchronisation pour les *systèmes multi-agents (SMA)* qui considèrent les coûts individuels pendant le processus de coordination. L'analyse est menée avec une attention particulière sur les SMA à dynamique linéaire homogène et les réseaux avec des clusters, avec une topologie fixe dans les deux cas.

Nous avons exploité à cette fin des concepts de la théorie des jeux [Perlaza et al., 2012] ainsi que des outils de la théorie des systèmes singulièrement perturbés pour la séparation des échelles de temps [Kokotović et al., 1999]. De plus, nous prenons en compte les contraintes de communication dans le réseau, c'est-à-dire que les contrôleurs de chaque agent n'utilisent que les informations provenant de son voisinage direct. Des protocoles de commandes distribuées sont fournis, mais les gains de commande peuvent être conçus de manière centralisée selon les cas.

Mots clés : systèmes multi-agents ; commandes décentralisées ou distribuées ; coûts individuels ; synchronisation ; théorie des jeux ; systèmes singulièrement perturbés ; réseaux avec clusters ; réduction des charges de calculs.