HAL open science

Decentralized control and analysis of cluster patterned networks

Marcos Cesar Bragagnolo

▶ To cite this version:

Marcos Cesar Bragagnolo. Decentralized control and analysis of cluster patterned networks. Automatic Control Engineering. Universite de Lorraine, 2015. English. ⟨NNT : 2015LORR0283⟩. ⟨tel-01318815v2⟩

HAL Id: tel-01318815

https://hal.science/tel-01318815v2

Submitted on 20 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autorisation HAL

HAL Authorization

École doctorale IAEM Lorraine

Decentralized control and analysis of cluster patterned networks

THÈSE

présentée et soutenue publiquement le 27 novembre 2015

pour l'obtention du

Doctorat de l'Université de Lorraine

(Mention Automatique, Traitement du Signal et des Images, Génie Informatique)

par

Marcos Cesar Bragagnolo

Composition du jury

Rapporteurs :	A. GIRARD	Maître de Conférences, Lab. Jean Kuntzmann
	W. MICHIELS	Professeur, KU Leuven
Examinateurs :	J. M. HENDRICKX	Maître de Conférences, U. C. de Louvain
	J. DAAFOUZ	Professeur, Université de Lorraine
Directeur de thèse:	P. RIEDINGER	Professeur, Université de Lorraine
Co-directeur de thèse:	I.-C. MORĂRESCU	Maître de Conférences, Université de Lorraine

Centre de Recherche en Automatique de Nancy
UMR 7039 CNRS – Université de Lorraine

2, avenue de la forêt de Haye 54516 Vandœuvre-lès-Nancy
Tél.+33 (0)3 83 59 59 59 Fax +33 (0)3 83 59 56 44

Mis en page avec la classe thesul.

I dedicate this thesis to my parents, my sisters, my family.

Acknowledgments

I would like to take this opportunity to acknowledge to those whose help and support have made this dissertation possible.

First and foremost I would like to thank my advisor Dr. Irinel-Constantin Morărescu for his help, support and patience during the three years of thesis. This thesis would not have been possible without his knowledge of the subject matter and his hard working on the development of my research skills. I would like to thank my thesis director Prof. Pierre Riedinger for giving me the opportunity to do this work, and all the assistance in my research.

I would like to thank Dr. Lucian Buşoniu and everyone in the Technical University of Cluj-Napoca for receiving me during my short stay in Romania.

I am grateful to Prof. Wim Michiels, Dr. Julien M. Hendrickx, Dr. Antoine Girard and Prof. Jamal Daafouz for agreeing to be in the committee of this thesis and for their valuable comments and discussions.

I would like to sincerely thank the whole team at the CRAN laboratory. In particular, Mme Carole Courrier and Mme Christine Pierson for their kind and immediate help in the administrative issues. I want to thank my colleagues and friends, Julien Louis, Vairis Caune, Gundars Korats, Maria Kholodtsova, Dorina Ionescu, Mohammed Hamid and Sharib Ali for their support and friendship during this thesis. I would also like to wish a good luck to the new PhD students in the lab.

And last, but not least, I would like to thank my girlfriend Rima Soboh for her patience and support during the last months of this thesis.

Contents

Resumé en Français **1**

Resumo em Português **7**

Chapter 1 Introduction **13**

1.1 General introduction 13

1.2 Consensus problem 15

1.2.1 Flocking 15

1.2.2 Formation 16

1.3 Clustered consensus problem 18

1.3.1 Consensus in leaderless clusters 19

1.3.2 Consensus in clusters with leaders 19

1.4 Structure of the thesis 20

1.5 Publications 21

Chapter 2 Preliminaries **23**

2.1 Graph theory 23

2.1.1 Graph-related matrices 25

2.2 Basic concepts on matrices and Linear Matrix Inequalities 27

2.2.1 Matrices definitions 27

2.2.2 Linear Matrix Inequalities 29

2.3 Reset systems 30

Chapter 3 Consensus in multiagent systems **33**

3.1 Motivation on consensus 33

3.2 Preliminaries on consensus 35

3.3 Static networks 38

3.3.1 Continuous 39
3.3.2 Discrete 40
3.4 Switching networks 40
3.4.1 Continuous 41
3.4.2 Discrete 42
3.5 Clustering of networks 44
3.5.1 Graph partitioning 44
3.5.2 Consensus in clustered networks 47
3.6 Thesis Contributions 49
3.6.1 Characterization of the consensus value 50
3.6.2 Stability in term of LMI 51
3.6.3 Control design 53

Chapter 4 Consensus on interconnected networks: stability and design 55
4.1 Problem formulation 56
4.2 Agreement behavior 57
4.2.1 Local agreements 58
4.2.2 Consensus value 58
4.3 Stability analysis 61
4.3.1 Parametric LMI condition 61
4.3.2 Decay rate analysis 63
4.4 Convergence toward a prescribed value 65
4.4.1 Convergence toward a prescribed value with a prescribed decay rate . . 66
4.5 Numerical implementation 67
4.5.1 Periodic resets 67
4.5.2 Quasi-periodic resets. 67
4.6 Illustrative examples 69
4.6.1 Small network analysis 69
4.6.2 Larger network analysis 73
4.7 Conclusion 75

Chapter 5 Consensus on interconnected networks of nonholonomic robots 77
5.1 Nonholonomic robots 77
5.2 Formation control and nonholonomic vehicles 79
5.2.1 Ground vehicles 79

5.2.2 Aerial Vehicles 79
5.2.3 Underwater Vehicles 80
5.3 Problem formulation 80
5.3.1 Robot dynamics and tracking error 81
5.3.2 Network topology and agreement dynamics 82
5.4 Consensus and tracking problems 84
5.4.1 Tracking for nonholonomic systems 85
5.5 Overall controller design 85
5.6 Simulation results 89
5.6.1 Small-scale example: Ellipse formation 89
5.6.2 Large-scale example: Three-leaf clover formation 91
5.7 Conclusion 92

Chapter 6 Conclusion **95**
6.1 Conclusions 95
6.2 Contributions 96
6.3 Future research 96

Bibliography **99**

Resumé en Français

Les systèmes multi-agents sont des systèmes qui comportent plusieurs sous-systèmes (par exemple, robots, satellites, neurones ...) qui collaborent pour accomplir une tâche. Ils agissent de façon décentralisée pour accomplir une tâche donnée, où chaque agent est responsable d'une partie de la tâche globale. Dans certains cas, un système multi-agents améliore la robustesse du système, ce qui signifie que si un agent ne parvient pas à remplir sa partie de la tâche, les autres agents peuvent prendre ses fonctions. Un problème particulièrement intéressant dans les systèmes multi-agents est le problème de consensus. Celui-ci exige que les agents s'accordent sur une certaine quantité d'intérêt. Un exemple de ce problème peut être la négociation dans un groupe de gens avec différentes opinions.

La plupart des résultats dans le probléme de consensus suppose que même si certaines connexions peuvent être perdues, le réseau reste connecté. Cependant, dans certains cas, cela est faux. Un grand réseau peut être divisé en plus petits groupes qui pourraient ne pas être connecté à tout moment à cause des transmissions défectueux, longue distance, le coût de la transmission, etc. Par conséquent, il est important de développer un algorithme permettant d'atteindre le consensus dans les réseaux partitionné en clusters non-connexes.

Objectifs et contributions

Motivé par la discussion précédente, nous étudions un nouveau système qui considère la séparation d'un réseau dans des réseaux plus petits. Dans cette thèse, nous avons les objectifs suivants:

- Proposer un nouveau protocol de consensus dans les réseaux partitionné en groupe;
- Fournir des outils pour caractériser la valeur de consensus du réseau;
- Prouver la stabilité de ce nouveau algorithme;
- Développer des outils de design pour atteindre une valeur de consensus et une vitesse de

convergence a priori définies;

- Appliquer les techniques développées à un système de robots non-holonomes.

Plan de la Thèse

Cette thèse est organisée de la façon suivante.

Chapitre 1: Introduction

Dans le premier chapitre, nous présentons de manière générale les systèmes multi-agents et le problème de consensus. Les systèmes multi-agents ont reçu beaucoup d'attention dans les dernières décennies car plusieurs disciplines peuvent bénéficier de l'étude de ces systèmes. Nous mettons en évidence leur utilisation en ingénierie (par exemple, la formation des robots [BA98, CMPT08], capteurs distribués [HGJ03], oscillateurs couplés [PJM10]) et la sociologie (par exemple les réseaux sociaux [SS02], la dynamique d'opinion [HK02]) pour ne citer que quelques-uns. En automatique leur utilisation est principalement axée sur: la réalisation de formation de robots mobiles [BA98, CMPT08], la synchronisation d'oscillateurs couplés [PJM10], la stabilité de réseaux énergétique [ZC12, PFR09].

Problème de consensus

Un problème souvent étudié dans le cadre des systèmes multi-agents est le consensus. Le but de ce problème est de parvenir à un accord sur une certaine quantité d'intérêts en partageant des informations via un réseau, avec une topologie prescrite. Plusieurs formulations du problème peuvent être trouvées dans la littérature: interaction via une topologie dynamique ou fixe, agents avec une dynamique continue ou discrète, agents avec une dynamiques linéaire ou non linéaires, interactions linéaires ou non linéaires entre les agents. Nous présentons quelques cas particuliers de consensus tels que l'alignement en vitesse et la réalisation d'une formation par des robots mobiles.

Alignement en vitesse

Ce problème considère un comportement commun dans la nature, souvent vu dans les groupes d'oiseaux. Il caractérise ce mouvement commun des oiseaux, lors du vol dans une formation

en V ou dans ce qui semble être un nuage. Dans ce cas, les agents ne cherchent pas à se mettre d'accord uniquement sur la vitesse absolue, mais aussi sur la direction de cette vitesse. Cela permet à tous les agents de se déplacer de façon synchronisée une fois que le consensus est atteint. L'un des premiers modèles a été proposé par Craig Reynolds en 1987, appelé le modèle de Boids [Rey87]. Plusieurs autres modèles ont été proposés par la suite. L'un de ces modèles est le modèle proposé par Vicsek et al. en 1995 [VCBJ+95]. D'autres modèles qui étudient l'alignement en vitesse sont présentés dans [CS07, OS06].

Formation

Un autre problème qui appartient à l'ensemble de problèmes de consensus est celui de la réalisation d'une formation. Celui-ci se traduit comme la réalisation d'une forme géométrique par un système multi-agents. Ces problèmes apparaissent dans un certain nombre de systèmes biologiques, tels que la forme en V que les oiseaux migrateurs accomplissent durant leur vol, afin de limiter la traînée tout en conservant la visibilité. Autrement dit, l'objectif de ce problème consiste à agencer tous les agents de telle sorte que la forme ou le motif soit réalisé dans l'espace. Le contrôle de la formation est particulièrement important dans le domaine de l'ingénierie. La réalisation de formations présentes des applications importantes dans la commande des flottes de drones (UAV) ou des sous-marins autonomes (AUV). Il est intéressant pour les AUV de garder la formation afin de surveiller une partie spécifique de l'océan [BASCdW11, BASCDW09], ou pour les drones afin de protéger un convoi [DRE09].

Problème de consensus en groupe

Alors que la plupart des résultats concernant le problème de consensus considèrent un réseau comme étant connexe, en réalité, cela pourrait ne pas être vrai. Plusieurs réseaux complexes peuvent être composés de sous-réseaux plus petits qui seront être appelés clusters à partir de maintenant. Dans ce travail, la loi de commande imposera des connexions inter-cluster, même si la partition en clusters impose seulement des connexions intra-cluster. L'objectif ici est d'atteindre le consensus dans le réseau entier et non seulement à l'intérieur de chaque cluster. Afin de réaliser cela nous considérons que chaque cluster contient un agent qui peu interagir de temps en temps à l'extérieur de son groupe. Cet agent est appelé leader.

Comme sous-ensemble du problème du au-dessus on peut formuler le consensus sans leaders. Dans ce cas, tous les agents ont une dynamique continue et le consensus est atteint à cause de la présence des interactions à l'intérieur de chaque cluster mais aussi entre ces clusters. Le rôle

des leaders dans notre étude est justement de négocier le consensus entre les différents accords locaux obtenus dans chacun des clusters. La tâche du leader à l'intérieur d'un réseau peut être variable en fonction de l'implémentation.

Chapitre 2: Préliminaires

Le Chapitre 2 introduit les concepts mathématiques fondamentaux pour une bonne compréhension de la thèse. Nous présentons d'abord les concepts de graphe dirigé et non dirigé. Toujours dans la théorie des graphes, nous montrons qu'il est possible d'associer certaines matrices à un graphe. Une attention particulière recevra la matrice de Laplace. Ensuite, nous rappelons quelques notions nécessaires sur les matrices, telles que la définition de matrices stochastiques, matrices symétriques et autres. Nous présentons également ici le concept d'inégalité matricielle linéaire (LMI). La dernière partie présente brièvement les systèmes à réinitialisation d'état, tout en donnant un exemple de ce genre des systèmes. Une attention particulière est portée à la classe spécifique des systèmes linéaires impulsifs. Un résultat clef garantissant que 0 est globalement uniformément asymptotiquement stable est rappelé.

Chapitre 3: Consensus dans les systèmes multi-agents

Ce chapitre fournit une vue d'ensemble de l'état de l'art sur la théorie du consensus. Il comporte de définitions et résultats fondamentaux dans le cas ou le réseau et statique ou dynamique et la dynamique de chaque agent et continue ou discrète. Entre autre, nous discutons la pertinence de considérer un réseau statique (sa topologie ne change pas au cours du temps) ou dynamique (topologie du réseau dépendante du temps ou de l'état) en fonction de l'application traitée. Finalement, nous nous intéressons au problème de partitionnement d'un graph en clusters. Nous présentons brièvement les principales méthodes de détection de clusters dans un graphe. Le chapitre ce termine par la formulation du problème traité dans ce manuscrit.

Chapitre 4: Consensus dans les reseaux interconnecté: stabilité et design

Ce chapitre contient les principaux résultats théoriques de la thèse. Dans un premier temps, nous allons introduire le cadre de l'étude. La motivation vient des études réalisées sur les réseaux sociaux [WF94], où il pourrait y avoir une séparation d'un grand groupe en plusieurs petits groupes. Un grand groupe ne pourrait pas être capable de converser et échanger des informations fréquemment, alors que les plus petits groupes pourraient être capables de le faire. Dans ce

travail, nous considérons que le réseau est fixé et connu a priori, des concepts sur la modularité des réseaux en cluster peuvent être utilisés pour définir ces divisions [BHT09, MG11, Kra97]. Un premier résultat fondamental pour la suite est la caractérisation de la valeur de consensus dans un réseau structuré en clusters où la dynamique interne au cluster est continue alors que les interactions inter-clusters produisent des impulsions. Il est remarquable que cette valeur ne dépende pas de la séquence des instants d'impulsion. Un deuxième résultat majeur est la caractérisation de la stabilité du consensus en terme de LMI. Cela implique d'un côté qu'il est facile de vérifier numériquement le critère de stabilité et de l'autre ça nous permet d'éteindre le résultat au design des interactions entre les clusters afin d'obtenir une valeur de consensus prescrite. La vitesse de convergence vers le consensus s'avère très importante dans plusieurs applications. Par conséquent, nous proposons aussi une méthodologie pour évaluer ainsi qu'accélérer cette vitesse.

Chapitre 5: Consensus dans les reseaux interconnecté des robots non-holonomes

Dans ce chapitre, nous faisons usage des résultats proposés dans le chapitre précédent. Précisément, nous abordons le problème du contrôle d'une flotte de systèmes non holonomes censés réaliser une formation. Ces robots mobiles forment un réseau structuré en plusieurs clusters et les communications inter-clusters sont rares à cause d'un coût énergétique élevé. D'abord, nous écrivons le problème en terme de consensus dans un réseau et nous montrons que la formation initialement prévue peut être réalisée puisque les robots sont capables d'atteindre une valeur de consensus global. Pour résoudre ce problème, nous le séparons en deux sous-problèmes. Le premier est lié a la génération de trajectoires désirées fondée sur un algorithme de consensus linéaire. Le second est le design d'un contrôleur local capable d'assurer la poursuite de trajectoire. Nous avons ensuite proposé un contrôleur général qui réunit à la fois les deux contrôleurs. Il est remarquable que cette loi de commande utilise seulement l'information locale fournie par le voisinage de chaque robot mobile.

Chapitre 6: Conclusion

Dans le Chapitre 6, nous présentons les conclusions de la thèse, nous énonçons ses contributions et nous proposons les lignes futures de la recherche pour le travail présenté dans cette thèse.

Publications

Le contenu de ce manuscrit a été validé dans les publications suivantes:

Journal

Reset strategy for consensus in networks of clusters. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger.*Automatica*. Accepté.

Conférences

LMI sufficient conditions for the consensus of linear agents with nearly-periodic resets. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. 2014 American Control Conference, Portland, USA.

Design of reset strategy for consensus in networks with cluster pattern. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. 8th European Nonlinear Dynamics Conference, Wien, Austria.

Planning methods for the optimal control and performance certification of general nonlinear switched systems. Lucian Buşoniu, Marcos Cesar Bragagnolo, Jamal Daafouz, Irinel-Constantin Morărescu. 54th IEEE Conference on Decision and Control, Osaka, Japan.

Chapitres de livres

Design and analysis of reset strategy for consensus in networks with cluster pattern Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. Springer series Advances in Dynamics and Delays: DELAYS IN NETWORKED CONTROL SYSTEMS. Editeurs: Alexandre Seuret, Laurentiu Hetel, Jamal Daafouz, Karl H. Johansson.

Implementation of consensus algorithms under harsh communication constraints. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Lucian Buşoniu, Pierre Riedinger. Handling Uncertainty and Networked Structure in Robot Control. Editeurs: Lucian Buşoniu, Levente Tamas.

Resumo em Português

Os sistemas multi-agentes são sistemas que consistem de agentes (por exemplo, robôs, satélites, neurônios ...) que colaboram para cumprir um objetivo. Ele proporcionam uma maneira decentralizada pra cumprir tal objetivo, onde cada agente é responsável por uma parte do objetivo global. Em certos casos, um sistema multi-agente proporciona ma melhora da robustez do sistema, sendo que se algum agente não complete a sua parte do objetivo, outros agentes podem suprir a sua função. Um problema particularmente interessante em sistemas multi-agentes é o problema de consenso. O problema de consenso exige que os agentes concordem com o valor de uma certa grandeza de interesse. Um exemplo que pode ser considerado nesse problema é um encontro de pessoas, onde elas devem concordar com uma certa opinião.

A grande maioria dos resultados em consenso supõe que mesmo se certa conexões entre agentes possam ser perdidas, a rede continua conectada. No entanto, em alguns casos, isso não é verdade. Uma grade rede pode ser dividida em pequenos grupos que podem não estar conectados a todo momento. (por exemplo, falha em transmissões, longa distância, custo de transmissão). Nesse caso é importante desenvolver um algoritmo que considere esta situação enquanto obtêm um consenso global.

Objetivos e contribuições

Considerando a discussão precedente, nós estudamos um novo sistema que considera a separação de uma rede em redes menores. Nesta tese, nós temos os seguintes objetivos:

- Propor um novo protocolo de consenso para redes particionadas em grupos;
- Fornecer ferramentas para caracterizar o valor de consenso da rede;
- Garantir a estabilidade desse novo algoritmo;
- Providenciar ferramentas para o design de rede de forma a atingir um valor de consenso e velocidade de convergência definidos a priori;

- Aplicar as técnicas desenvolvidas em um sistema de robôs não-holonômicos.

Estrutura da tese

Esta tese é organizada da seguinte maneira.

Capítulo 1: Introdução

No primeiro capítulo, nós apresentamos de maneira geral os sistemas multi-agentes e o problema do consensus. Os sistemas multi-agentes têm recebido grande atenção nas ultimas décadas com diversas disciplinas que podem se beneficiar do estudo desses sistemas. Nós evidenciamos a sua utilização na engenharia (por exemplo, a formação de robôs [BA98, CMPT08], sensores distribuídos [HGJ03], osciladores acoplados [PJM10]) e a sociologia (por exemplo as redes sociais [SS02], dinâmica de opinião [HK02]). Eu automação sua utilização é principalmente a seguinte: formação de robôs moveis [BA98, CMPT08], a sincronização de osciladores acoplados [PJM10], e a estabilidade de redes energéticas [ZC12, PFR09].

O problema de consenso

Um problema normalmente estudado em sistemas multi-agentes é o consenso. O objetivo desse problema é fazer como que os agentes concordem com o valor de uma certa grandeza de interesse, através do partilhamento de informação em uma rede de topologia fixa. Consenso foi estudado de diversas maneiras na literatura: com interações em topologia fixa ou dinâmica, agentes com dinâmicas continuas ou discretas, dinâmicas lineares ou não lineares, interações lineares ou não lineares entre os agentes. Nós apresentamos alguns casos em particular de consenso como o alinhamento em velocidade (flocking) e a formação no espaço de robôs moveis.

Alinhamento em velocidade

Este problema considera um comportamento comum na natureza, normalmente visto em grupos de aves. Ele caracteriza um movimento comum das aves, quando voam em uma formação em V ou quando formam um nuvem. Nesse caso, os agentes não procuram concordar apenas na sua velocidade absoluta, mas também com o sentido dessa velocidade. Isso permite que os agentes se movimentem de maneira sincronizada assim que o consenso seja atingido. Um dos

primeiros modelos a serem propostos foi o de Craig Reynolds em 1987, chamado de modelo de Boids [Rey87]. Um dos outros modelos possíveis foi proposto por Vicsek et al. em 1995 [VCBJ+95]. Outros modelos podem ser vistos em [CS07, OS06].

Formação

Outro problema que pertence a classe dos problemas de consenso é o da realização de uma formação. Neste problema procura-se realizar uma forma geométrica no espaço através de um sistema multi-agente. Formação aparece em um grande numero de sistema biológicos, como a forma em V que os pássaros migratórios mantém em seu voo, de forma a diminuir o arrasto enquanto mantêm a visibilidade. Em outras palavras, o objetivo desse problema é controlar todos os agentes de forma que uma forma geométrica seja realizada no espaço. O controle da formação é um problema particularmente importante na engenharia. A realização de uma formação apresenta aplicações importantes no controle de drones ou de submarinos autônomos. No caso dos AUVs, é interessante manter a formação para monitorar uma parte especifica do oceano [BASCdW11, BASCDW09], ou no caso dos drones para proteger um comboio [DRE09].

Problema de consenso em grupo

No momento a grande maioria dos resultados consideram o problema de consenso com uma rede sempre conectada (ou conexa), mas na realidade isso nem sempre é verdade. Várias redes complexas podem ser compostas de sub-redes menores, que serão chamadas de *clusters* à partir de agora. Nessa tese, a lei de controle irá impor conexões inter-cluster, mesmo que a divisão da rede faça com que existam apenas conexões intra-cluster. O objetivo aqui é de obter o consenso na rede inteira e não apenas no interior de cada cluster. Para atingir esse objetivo nós consideramos que cada cluster contém um agente que irá interagir de tempo em tempo com o exterior do seu grupo. Esse agente se chamará líder.

Um sub-grupo desse problema procura também atingir o consenso sem líderes. Nesse caso, todos os agentes tem uma dinâmica continua e o consenso é atingido não apenas por causa das interações no interior de cada grupo mas também entre os clusters. No nosso estudo o objetivo dos líderes é justamente de "negociar" o consenso entre os diferentes acordos locais obtidos em cada cluster. O líder, no entanto, pode apresentar funções diferentes dependendo da implementação.

Capítulo 2: Preliminares

O Capítulo 2 introduz os conceitos matemáticos fundamentais para uma boa compreensão da tese. Nós apresentamos os conceitos de grafo direcionado e não direcionado. Na teoria dos grafos, nós mostramos que é possível associar certas matrizes a um grafo. Uma atenção particular é reservada a matriz de Laplace. Em seguida, nós relembramos algumas noções necessárias de matrizes, como a definição de matrizes estocásticas, matrizes simétricas e outras. Nós apresentamos também o conceito de desigualdade matricial linear (LMI). A ultima parte apresenta brevemente a classe dos sistemas impulsivos lineares. Um resultado apresentado nessa parte garante que o valor 0 é globalmente uniformemente assintoticamente estável.

Capítulo 3: Consenso em sistemas multi-agentes

Esse capítulo fornece uma breve apresentação do estado da arte da teoria do consenso. Ele apresenta definições e resultados fundamentais nos casos de redes fixas ou dinâmicas e quando a dinâmica de cada agente é continua ou discreta. Nós apresentamos uma discussão sobre o uso de redes fixas (quando a topologia da rede não muda com o tempo) ou dinâmica (a topologia da rede depende do tempo ou dos seus estados) em função da aplicação estudada. Finalmente, nós apresentamos o problema de particionamento de um grafo em clusters. Nós apresentamos brevemente os principais métodos de detecção de clusters em um grafo. O capítulo termina com a formulação do problema nesse manuscrito.

Capítulo 4: Consenso em redes interconectadas: estabilidade e design

Esse capítulo contém os principais resultados teóricos da tese. A motivação vem de estudos realizados em redes sociais [WF94], onde pode-se observar uma separação de um grande grupo em vários pequenos grupos. Um grande grupo não é capaz de troca informações frequentemente, no entanto pequenos grupos são capazes de fazê-lo. Nesse trabalho, nós consideramos que a rede é fixa e conhecida a priori, os conceitos de modularidade de redes em cluster podem ser usados para definir estas divisões [BHT09, MG11, Kra97]. O primeiro resultado fundamental apresentado é a caracterização do valor de consenso em uma rede estruturada em clusters onde a dinâmica interna dos clusters é continua, enquanto as interações intra-clusters provocam uma resposta impulsiva. Nota-se que este valor não depende da sequência de instantes do impulso. Um segundo resultado importante é a caracterização da estabilidade do sistema através de uma LMI. Dessa forma é fácil de verificar numericamente o critério de estabilidade, além do mais, esta técnica nos permite realizar o design das interações entre os clusters para obter um valor

de consenso definido a priori. A velocidade de convergência do consenso é muito importante em várias aplicações. Desta forma, nós propomos também uma metodologia para acelerar essa velocidade.

Capítulo 5: Consenso em redes interconectadas de robôs não-holonômicos

Nesse capítulo, nós utilizamos os resultados propostos pelo capítulo anterior. Mais precisamente, nós consideramos o problema de controle de uma frota de sistemas não-holonômicos que procuram realizar uma formação. Esses robôs moveis formam uma rede estruturada em vários clusters e as comunicações inter-clusters são raras graças a um custo energético elevado. Desta forma, nós apresentamos o problema na forma de consenso em uma rede e mostramos que a formação prevista pode ser realizada porque os robôs são capazes de atingir um valor de consenso global. Para resolver esse problema, nós separamos o problema em dois sub-problemas. O primeiro é a geração de trajetórias desejadas, obtidas através de um algoritmo de consenso linear. O segundo é o design de um controlador local capaz de assegurar o seguimento dessa trajetória. Em seguida, nós propomos um controlador geral que reúne os dois controladores. Nota-se que esta lei de controle utiliza apenas a informação local fornecida pelos vizinhos de cada robô.

Capítulo 6: Conclusão

No Capítulo 6, nós concluímos a tese, apresentando suas contribuições enquanto propomos futuras linhas de pesquisa para o trabalho apresentado.

Publicações

O conteúdo dessa tese foi validado pelas seguintes publicações:

Revista científica

Reset strategy for consensus in networks of clusters. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger.*Automatica*. Aceito.

Conferências

LMI sufficient conditions for the consensus of linear agents with nearly-periodic resets. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. 2014 American Control Conference, Portland, USA.

Design of reset strategy for consensus in networks with cluster pattern. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. 8th European Nonlinear Dynamics Conference, Wien, Austria.

Planning methods for the optimal control and performance certification of general nonlinear switched systems. Lucian Buşoniu, Marcos Cesar Bragagnolo, Jamal Daafouz, Irinel-Constantin Morărescu. 54th IEEE Conference on Decision and Control, Osaka, Japan.

Capítulos de livros

Design and analysis of reset strategy for consensus in networks with cluster pattern Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. Springer series Advances in Dynamics and Delays: DELAYS IN NETWORKED CONTROL SYSTEMS. Editores: Alexandre Seuret, Laurentiu Hetel, Jamal Daafouz, Karl H. Johansson.

Implementation of consensus algorithms under harsh communication constraints. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Lucian Buşoniu, Pierre Riedinger. Handling Uncertainty and Networked Structure in Robot Control. Editores: Lucian Buşoniu, Levente Tamas.

Chapter 1

Introduction

1.1 General introduction

Multiagent systems have been broadly studied throughout the years and several disciplines can benefit from studying these systems. They can be used to model several interactions in engineering (e.g. formation of robots [BA98, CMPT08], distributed sensors [HGJ03], coupled oscillators [PJM10]) and sociology (e.g. social networks [SS02], opinion dynamics [HK02]) just to name a few. Control engineering community mainly focused on the application of multiagent systems in: formation of robots [BA98, CMPT08], coupled oscillators [PJM10], energy transmission [ZC12, PFR09]. However, there is a growing interest in sociology to use this formalism to represent interactions in a social network and study the opinion dynamics. Past studies in multiagent systems allow social networks to be modeled considering situations such as: group division inside a network, collaborative/competitive behavior.

While usually these agents have a collaborative behavior, competition is also possible. Collaborative networks are often seen in nature (e.g. a swarm of insects, a school of fish, a flock of birds, etc). The importance of studying these natural behaviors is to adapt what we can perceive of the natural world into techniques used in other fields. One of these techniques is called *flocking*, and it resembles the behavior of a school of fish or a flock of immigrating birds. Further details about flocking will be presented in Subsection 1.2.1. Not only collaborative dynamics can be seen in the natural world. Often are the cases where the agents have disagreeing views, one example is the predator-prey dynamics. The prey have an antagonistic relation with the predator, therefore having opposite goals (i.e. the goal of the prey is to stay together while staying away of the predator, while the predator wants to get closer to the prey [Lee06]).

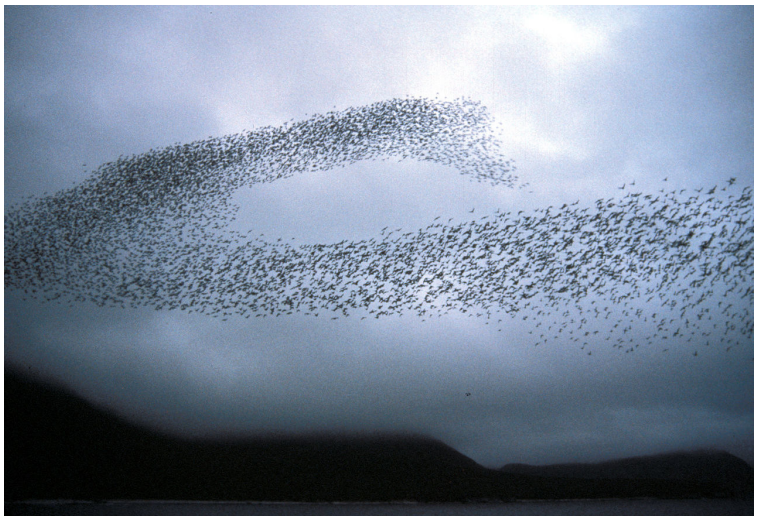

Figure 1.1: Example of a flock of birds, flying in synchrony.

In the engineering framework the main advantage of a multiagent system is the decentralized control. Instead of having a centralized control that is prone to faults, every agent is responsible at achieving some task. By collaborating with the remaining network, a main goal can be achieved. In some situations this may bring robustness to the systems because if an agent ceases to collaborate the other agents are still able to reach a main goal (e.g. robots scanning an area, group of sensors). However, in some other situations all the agents must function correctly, otherwise it is impossible to accomplish a main goal (e.g. collaborative transport of cargo, collaborative industrial robots in a assembly line).

A common sector where multiagent systems are important is the power networks. Applications of multiagent systems are mainly related to microgrids and smart grids. A microgrid is a type of power system, which is formed by the interconnection of small, modular generation to low voltage distribution systems. Microgrids can be connected to the main power network or be operated autonomously, similar to power systems of physical islands. Interested readers can find more information in: [DH04, LSW08, JCR08, DH07]. The smart grid is a power grid expected to exhibit some characteristics as: self-healing, consumer friendly, attack resistant, able to accommodate all generation and storage options and optimizes assets and operates efficiently. Further information about smartgrids and multiagent systems in smartgrids can be found for instance in: [ZC12, PFR09, HWK+07, HBA+13]. In the next subsection we will present a common problem in multiagent systems: the consensus problem.

1.2 Consensus problem

A problem often studied in multiagent systems framework is the consensus. The aim of this problem is to reach an agreement in a certain quantities of interest by sharing information over a network, oftenly with a prescribed topology. Several iterations of the problem can be seen in the literature: fixed or dynamically switching topology, continuous or discrete agent dynamics and linear or nonlinear agent dynamics. In some cases, the agreement of the agents is required to be the average of the initial conditions and this has been well studied in the literature [OSM04b]. However, this is not always the case and sometimes it is enough just to prove that agreement is reached. In some cases, given a fixed network topology and the initial conditions it is possible to define a priori the agreement value of a network [OSM04b, BMDR14].

Throughout the manuscript, when the agreement is not related to a specific variable (e.g. the heading of a mobile robot, or its position, etc), we will refer to the value of a certain agent as its state. There are several cases where the agreement on a certain variable can be studied as a specific case of the consensus problem. In the next Subsections we will present some of them such as the **flocking** and **formation** problems.

1.2.1 Flocking

This problem considers a common behavior in nature, often seen in flocks of birds (hence the name, **flocking**). It categorizes this common movement of birds, when they fly in a *v-formation* or in what seems to be a *cloud*. In this case the agents are not trying to agree on only the absolute velocity, but also in the heading of this velocity. This allows all agents to move in synchrony once consensus is reached. One of the first models were proposed by Craig Reynolds in 1987, named the *Boids Model* [Rey87]. One of the main contributions of Reynolds was to define a number of characteristics that the movement had to satisfy to be considered *flocking*. These characteristics are:

- **separation:** the agents should not collide and must keep a minimum distance from their neighbors;
- **cohesion:** the agents must stay close enough from each other;
- **alignment:** all the agents must have the same velocity (both in amplitude and direction).

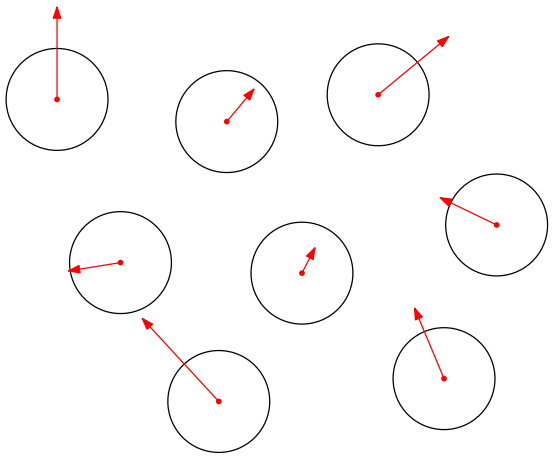

Figure 1.2: Agents with their velocities.

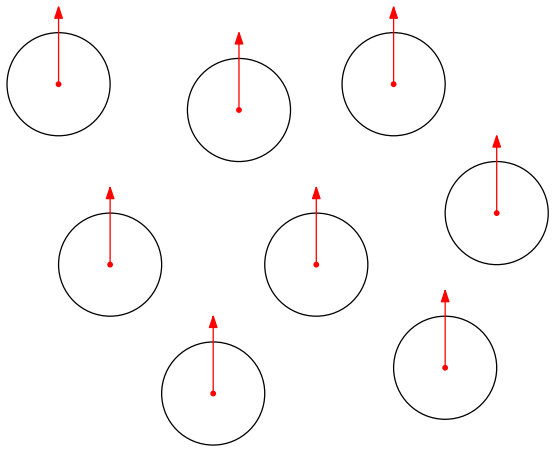

Figure 1.3: Agents aligned in flocking

Reynolds proposed a decentralized system to emulate the behavior of birds in a realistic manner. In the experiments, the agents had some limited information about their surroundings, therefore only communication with close neighbors was possible. His algorithm was implemented in discrete time, but we can present an equivalent system in continuous time:

$$\begin{cases} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= alignment_i(t) + repulsion_i(t) + attraction_i(t), \end{cases} \tag{1.1}$$

where $x_i(t), v_i(t) \in \mathbb{R}^d$ represents the position and velocity of agent i. The alignment factor ($alignment_i(t)$) can be seen as a case of consensus, where the agents must agree on the velocity. The attraction $attraction_i(t)$ acts over the agents when they are distant while $repulsion_i(t)$ acts when the agents are too close. For the sake of simplicity we show Figure 1.2, that represents a group of agents before consensus on the velocities is reached, and Figure 1.3, that shows the same group in flocking movement.

While being one of the first models, the Boids model is not the only one. Several other models were proposed later. One of these models is the **Vicsek Model** proposed by Vicsek et al. in 1995 [VCBJ+95]. This model considers that every agent has the same absolute velocity and at each step it assumes the average direction of the agents in its neighborhood, plus a random perturbation. Other models related to flocking are [CS07, OS06].

1.2.2 Formation

Another problem that belongs to this subset of consensus problems is the **formation** one. Formation is the realization of a geometrical form by a multiagent system. Formations appear in

a number of biological systems, such as the V-shape that migratory birds accomplish during flight, to limit drag while still maintaining visibility. Simply put, the objective of this problem is to move all agents in such a way that a shape or pattern is realized in space.

Formation control is specially important in the engineering field. Unmanned Aerial Vehicles (UAVs) and Autonomous Underwater Vehicles (AUVs) greatly benefit from studies in formation control. It is interesting for AUVs to keep formation to scan a particular part of the ocean [BASCdW11, BASCDW09], or for UAVs to protect a convoy [DRE09].

Several methods were studied in the past. One of them is by reaching consensus in the difference between the target formation and the position of the agents. Therefore, we can consider that the consensus is realized with this new variable:

$$\tau_i = q_i - \xi_i, \tag{1.2}$$

where $q_i = (x_i, y_i)^\top$ is the real position of the agent i, $\xi_i = (\mathbf{x}_i, \mathbf{y}_i)^\top$ is the position of the agent i in the formation and $\tau_i = (\tau_{i1}, \tau_{i2})^\top$ is the displacement vector. When all the agents agree to a certain value, i.e. $\tau_i = \tau_j = \tau_k$ ..., then the displacement vector of all agents is the same and the formation is realized somewhere in the Cartesian space. The change of variable can be visualized in Figure 1.4. This method is used frequently in the literature, and will be explored in more details in Chapter 4.

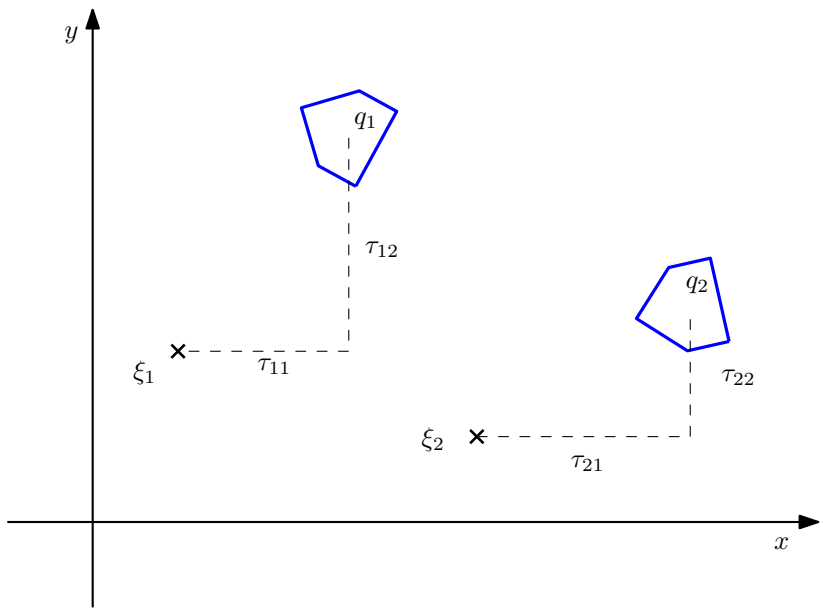

Figure 1.4: Change of variables in formation problems. The aim here is to achieve consensus in such a way that $\tau_{11} = \tau_{21}$ and $\tau_{12} = \tau_{22}$.

In the literature, formation control falls under three other main methods: leader-follower, behavior-based and the virtual structure models. In the leader-follower model a single agent of the net-

work is assigned the role of the leader, and the leader follows a predefined trajectory. The other agents, designed as followers, must maintain a desired distance and orientation to the leader. While this approach simplifies the overall control scheme of the network, the over-reliance on one agent can bring some problems. If the leader is incapable of keeping the mission goals, then the whole network fails. Nevertheless, this method is still appreciated due to its scalability and simplicity. Alternatively, some results use a virtual leader approach. The trajectory of this virtual leader is sent to the agents by some overseeing entity (e.g. the command room in a military operation).

The behavior-based method considers a combination of several desired behaviors (e.g. collision avoidance, formation keeping, target seeking), which are prescribed for each agent. By weighting the relative importance of each behavior, the result action of each robot can be derived. While robust, this method can have a difficult mathematical formalization. Therefore is not easy to guarantee the convergence of the formation.

The last method described here, is the virtual structure approach. In this approach the formation is considered as a virtual rigid structure. This means that the behavior of the system of agents is similar to a physical object. In this case, desired trajectories are not assigned to each one of the agents but to the formation as a whole. This allows to a very predictable behavior of the formation, but with the cost of a larger communication bandwidth.

1.3 Clustered consensus problem

While most results in consensus consider a network to be completely connected, in reality this might not be true. More complex networks can be composed by smaller sub-networks that will be called clusters from now on. Despite the separation into cluster, it is assumed the presence of inter-cluster connections, as well as the intra-cluster ones. Most of the results consider a purely collaborative network, where not only consensus intra-cluster must be reached, but also as a whole network. However, there might be the case where agents are collaborative inside their cluster but competitive with those in another cluster. In this case consensus cannot be reached for the whole network, and instead a particular case of agreement is reached, called bipartite consensus [Alt13, ZC12].

As seen previously, the inspiration for these problems often comes from nature. In [GKES89], the authors investigate the oscillatory response of neurons in the cat visual cortex. They demonstrate that spatially separated group of neurons can synchronize their oscillatory responses. Drawing inspiration from the works of Gray et al, [OK91] also investigates the synchronization of coupled oscillators. The authors consider different cases of synchronization: weakly coupled

clusters where the weights of the interactions between clusters is weaker than the one inside clusters, non-weakly coupled clusters where the interactions inter-cluster have the same weight as intra-cluster and indirect synchronization, a special case of the weakly coupled clusters.

To be more precise, in the following we will present two cases of the clustered consensus problem, the leaderless case and the case where clusters have one or more leaders.

1.3.1 Consensus in leaderless clusters

As subset of the clustered consensus problem stands the leaderless clusters. In this case, all agents have the same dynamics and have no particularity over the network topology. In some cases this problem is characterized as group consensus, where several sub-networks inside the main network reach also the same consistent value when communication channels exist not only inside the group but outside.

In [YW09] the authors present the idea of group consensus in multiagent systems and propose consensus protocols under some algebraic condition to ensure average group consensus under some assumptions. In [FXZ14] the authors investigate the group consensus problem on the framework of double-integrator dynamics, extending the results of [YW09].

One special case of the of the leaderless cluster, and more specifically the group consensus, is the bipartite consensus. While not naturally separated into clusters, the competitive and collaborative interactions provoke a polarization of the agents, where they assume antagonizing behavior. In [Alt13] a signed graph is considered and the authors prove that all agents converge to a consensus value that is the same for all agents, expect for the sign. This is in the case of single-integrator dynamics. In [ZC12] the authors present a solution to the bipartite consensus where the agents are modeled by linear time-invariant systems.

1.3.2 Consensus in clusters with leaders

There are cases in which the presence of a special agent called leader is required for supervisory or control aims. The task of the leader inside a network may vary depending on the implementation. In [BMDR14] the leader serves as a source of communication outside his own group. We assume there is a fixed communication network between leaders. In this case, the communication between leaders allows consensus over the entire network to be reached in a otherwise disconnected network. This approach is specially interesting when the communication between agents is difficult or costly. Overall network consensus can be reached even if there are less connections between agents, as long as each cluster has at least one leader and the leaders' network

remains connected. In Chapter 5 we present an application of this network related to formation control. The division of the network in groups allows each group to maintain a partial version of the complete formation.

The leader-following formation approach can also be considered in clustered consensus problem. The main point of failure in the leader-following approach is the over-reliance on the leader, as we discussed before. One possible solution is to define the complete formation in small subgroups, with one or more leaders. In the case of failure of one leader, the other agents could be absorbed into another cluster and keep the main goal. In [RS08] the consensus and leader-follower approach are integrated. This approach allows to minimize the use of communication channels while providing some robustness in the case of leader failure.

1.4 Structure of the thesis

The structure of the thesis is as follows. In Chapter 2 we present some preliminaries, mainly in graph theory. Preliminaries on reset systems and some concepts of matrices and Linear Matrix Inequalities (LMIs) are also presented, as this information is essential for the results in the next chapters. Chapter 3 presents a non exhaustive state of the art in consensus theory. We present the results of the literature for fixed and switching networks, in both continuous and discrete time. Then we relate our work with the state of art in clustering of networks, mainly results related to modularity of clusters and consensus in clustered networks. The chapter ends with the contributions of the thesis, such as characterization of the consensus value, stability and design for clustered networks.

Chapter 4 presents the main results in clustered networks. At first, we present the network topology where all the agents inside a network are divided into clusters and communication outside these clusters is done only at specific times in a special network called leader's network. Secondly, we provide a way to compute the consensus value given only the initial conditions of the system and the topology of the network. This leads to the next results, where the stability of the system is proven by means of a parametric LMI. Afterward, by modifying the LMI introduced during the stability proof we are capable of designing the leader's network to accomplish some specific goals, such as reaching a priori defined consensus value and reaching consensus quicker than some defined decay rate. Simulations on large scale and small scale systems are presented. We finish the chapter with a discussion of the results.

Chapter 5 presents results on formation control of mobile robots. By utilizing the topology presented in Chapter 4 we are capable of driving a number of mobile robots to a certain formation in the space. To keep this chapter self contained, we start with an introduction in mobile

robots with non-holonomic dynamics. Then, results on trajectory tracking of non-holonomic robots are presented, mainly the tracking control proposed in [PLLN98]. To solve the formation problem we separate the problem into two subproblems: trajectory tracking and consensus agreement between the agents. As to facilitate implementation on a real system, we propose a discretization of the network topology. Simulations are used to validate the results. The chapter ends with a discussion of results.

Chapter 6 provides concluding remarks and perspectives related to the work presented in the manuscript.

1.5 Publications

The following publications resulted from the research performed during the PhD:

Journal

Reset strategy for consensus in networks of clusters. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. To appear in *Automatica*.

Conferences

LMI sufficient conditions for the consensus of linear agents with nearly-periodic resets. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. Presented at 2014 American Control Conference, Portland, USA.

Design of reset strategy for consensus in networks with cluster pattern. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. Presented at 8th European Nonlinear Dynamics Conference, Vienna, Austria.

Planning methods for the optimal control and performance certification of general nonlinear switched systems. Lucian Buşoniu, Marcos Cesar Bragagnolo, Jamal Daafouz, Irinel-Constantin Morărescu. 54th IEEE Conference on Decision and Control, Osaka, Japan.

Book chapters

Design and analysis of reset strategy for consensus in networks with cluster pattern Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Jamal Daafouz, Pierre Riedinger. Springer series Advances in Dynamics and Delays: DELAYS IN NETWORKED CONTROL SYSTEMS. Editors: Alexandre Seuret, Laurentiu Hetel, Jamal Daafouz, Karl H. Johansson.

Implementation of consensus algorithms under harsh communication constraints. Marcos Cesar Bragagnolo, Irinel-Constantin Morărescu, Lucian Buşoniu, Pierre Riedinger. Handling Uncertainty and Networked Structure in Robot Control. Editors: Lucian Buşoniu, Levente Tamas.

Chapter 2

Preliminaries

The purpose of this chapter is to provide few tools for the next chapters. While this may seem like a break on the flow, specially after we motivated our work in relation to multiagent systems, these tools are essential for understanding the main results of the thesis. We start the chapter by providing few concepts on graph theory, beginning with the definition of graph. We introduce the concepts of undirected and directed graph. Still in graph theory we show that is possible to associate a graph to certain matrices, with special attention to the Laplacian matrix. Afterward some necessary concepts on matrices are introduced, such as the definitions of stochastic matrices, symmetric matrices and others. we also introduce here the concept of LMI. The last subsection briefly presents reset systems, while giving an example of this kind of systems. A special attention is payed to the particular class of impulsive linear systems. A key result guaranteeing 0 is globally uniformly asymptotically stable is recalled.

2.1 Graph theory

In multiagent systems, the agents and their interactions are commonly represented by graphs. It becomes necessary to introduce first a graph. In [GR13] a graph is defined as:

Definition 1 *A graph X consists of a vertex set $V(X)$ and an edge set $E(X)$ where an edge is an unordered pair of distinct vertices of X. A graph is considered undirected if there exist an edge connecting vertices v_i and v_j (denoted as $\{v_i v_j\}$) as well as edge connecting v_j and v_i, and it is directed otherwise.*

In the multiagents framework, the vertex set $V(X)$ is used to represent the agents while the edge set $E(X)$ represents the communication between pairs of agents. Starting now and throughout

the manuscript, we will use the notation $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to represent a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$.

Despite being inherently a set theoretic object, a graph can be represented graphically. At the moment, we present the graphical representation of an undirected graph. This graphical representation consists of "dots" that represent the vertices v_i and "lines" between v_i and v_j that represent the edge $\{v_i v_j\} \in \mathcal{E}$. If there is an edge $\{v_i v_j\}$, the vertices v_i and v_j are said to be *adjacent*. Figure 2.1 gives an example of an undirected graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$ and $\mathcal{E} = \{v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_1, v_5v_2, v_5v_3\}$.

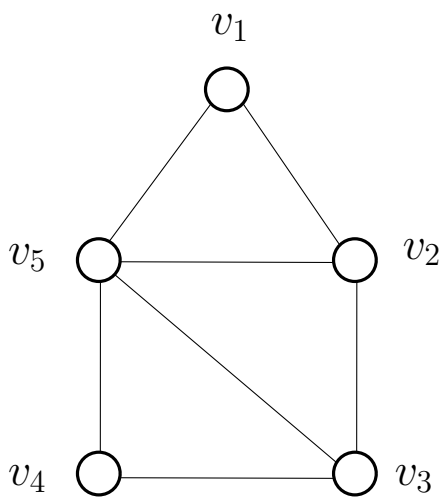

Figure 2.1: Example of an undirected graph with 5 vertices.

Taking advantage of this graphical representation, we also define the neighborhood $N(i) \subseteq \mathcal{V}$ of a vertex v_i as the set $\{v_j \in \mathcal{V} | v_i v_j \in \mathcal{E}\}$, or the set of all agents adjacent to v_i. In the case of undirected graphs, if $v_j \in N(i)$ then $v_i \in N(j)$. With the graphical representation the concept of *path* is clearer. A *path* in a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a union of edges $\bigcup_{k=1}^{p}(i_k, j_k)$ such that $i_{k+1} = j_k$, $\forall k \in \{1, \ldots, p-1\}$. Two nodes i, j are **connected** in a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if there exists at least a path in $\mathcal{G}$ joining i and j (i.e. $i_1 = i$ and $j_p = j$). A graph is said to be connected if any two nodes are connected.

Directed graphs

In some cases, the communication inside a network is done via unidirectional communication links. In these situations the network is represented by directed graphs (digraphs). Therefore, in the representation of the edges set $\mathcal{E}$, $v_i v_j \in \mathcal{E}$ is an influence from vertex v_i to vertex v_j but not the other way around. To properly separate directed graphs from undirected graphs, we denote

$\mathcal{D} = (\mathcal{V}, \mathcal{E})$ as a digraph. Graphically, it is common to represent the edges in directed graphs as arrows and, borrowing from the previous example, an edge $v_i v_j \in \mathcal{E}$ is represented by an arrow with the tail in v_i and the head in v_j. Therefore, for directed graphs we have another relation to connectivity. A digraph is said to be strongly connected when there is path connecting any two vertices in the graph. A tree is a connected directed graph with no cycles. A spanning tree of a graph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ is a tree visiting all the vertices of $\mathcal{D}$.

2.1.1 Graph-related matrices

As we have seen before, graphs are constructed to represent a relation between a finite number of objects. Graphs not only admit a graphical representation but can also be associated to certain matrices. In this subsection we will present some important matrices related to graphs.

Adjacency and Degree matrix

For an undirected graph $\mathcal{G}$ consider the neighborhood set $N(i)$ previously defined. The *degree*, $d(v_i)$, of a vertex v_i is the cardinality of the set $N(i)$, that is, the number of adjacent vertices of vertex v_i. The *degree matrix* of $\mathcal{G}$ is a diagonal matrix containing the vertex-degrees of $\mathcal{G}$ on the diagonal,

$$D(\mathcal{G}) = \begin{pmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_n) \end{pmatrix}, \tag{2.1}$$

with n as the number of vertices.

Additionally, we present the definition of *adjacency matrix*, $A(\mathcal{G})$, encoding the adjacency relations of graph $\mathcal{G}$, that is:

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } \{v_i v_j\} \in \mathcal{E}(\mathcal{G}) \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Laplacian matrix

A very important matrix associated with a graph is the *Laplacian matrix*. The Laplacian matrix not only provides a compact way to treat a graph but some of its properties are very important for consensus, and will be presented in Chapter 3 when we discuss consensus in detail.

The Laplacian of a graph can be defined in several ways. The most common way to define a Laplacian matrix is the following:

$$L(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G}), \tag{2.3}$$

where $D(\mathcal{G})$ is the degree matrix of graph $\mathcal{G}$ and $A(\mathcal{G})$ is adjacency matrix of graph $\mathcal{G}$. From Equation (2.3) comes a very important property of the Laplacian matrix: the sum of the rows. For every graph, the Laplacian that represents it will have sum of the rows equal to 0. Therefore, the Laplacian matrix has always an eigenvalue 0 associated to it. Moreover, the eigenvector of ones, denoted as $\mathbf{1}$, is eigenvector associated to the eigenvalue 0. Another important feature that is only available in undirected graphs is the symmetry of the Laplacian matrix. This happens due to the two-way communications in undirected graphs.

Matrices in directed graphs

For completeness, we consider the case of weighted digraphs, as this is kind of graph we will use in the upcoming chapters. For clarity, we denote the weighted digraph as $\mathcal{D}$. From [ME10] the adjacency matrix is

$$[A(\mathcal{D})]_{ij} = \begin{cases} w_{ij} & \text{if } \{v_i v_j\} \in \mathcal{E}(\mathcal{D}) \\ 0 & \text{otherwise,} \end{cases} \tag{2.4}$$

where w_{ij} is the weight associated to the influence of vertex v_i to vertex v_j. For the diagonal degree matrix we have:

$$[D(\mathcal{D})]_{ii} = d_{in}(v_i) \quad \forall i, \tag{2.5}$$

where $d_{in}(v)$ is the weighted in-degree of vertex v, that is:

$$d_{in}(v_i) = \sum_{j|\{v_i v_j\} \in \mathcal{E}(\mathcal{D})} w_{ij}. \tag{2.6}$$

Another way to define the degree matrix is through the adjacency matrix:

$$D(\mathcal{D}) = \mathbf{diag}(A(\mathcal{D})\mathbf{1}). \tag{2.7}$$

Some digraphs may retain properties similar to undirected graphs. To these graphs we give the name of **balanced digraphs**. The definition of balanced digraphs comes from Definition 3.14 of [ME10]:

Definition 2 *A digraph is called balanced if, for every vertex, the in-degree and out-degree are equal.*

Borrowing from Equation (2.3), we can define the Laplacian for the digraph $\mathcal{D}$ as:

$$L(\mathcal{D}) = D(\mathcal{D}) - A(\mathcal{D}). \tag{2.8}$$

The Laplacian matrix of a digraph presents some of the properties of its undirected counterpart. The sum of the rows is still equal to 0, and this property can be seen more easily thanks to the Equation (2.7). The Laplacian $L(\mathcal{D})$ retains the eigenvalue 0 and the eigenvector $\mathbf{1}$. However, since a digraph might not necessarily have a two way communication $L(\mathcal{D})$ is no longer symmetric.

2.2 Basic concepts on matrices and Linear Matrix Inequalities

In this section we present some definitions of certain kinds of matrices. We also present two theorems, the Perron-Frobenius theorem and the Gershgorin circle theorem. The section ends with a subsection about Linear Matrix Inequalities.

2.2.1 Matrices definitions

Stochastic matrix

A stochastic matrix (also named Markov matrix) is a matrix used to describe the transitions of probabilities, with each of its entries as a nonnegative real number. Stochastic matrices are square matrices with non-negative elements with the following property:

$$\sum_j P_{ij} = 1. \tag{2.9}$$

A stochastic matrix is said to be right stochastic if, for all its rows, the sum is equal to 1. On the other hand, a left stochastic matrix is a matrix with the sum of its columns equal to 1. Doubly stochastic matrices refer to stochastic matrices where both the columns and the rows sum up to 1. Due to the sum of rows/columns property, a stochastic matrix always has 1 as an eigenvalue.

Symmetric matrix

A symmetric matrix is a square matrix that is equal to its transpose. In other words, considering a matrix A, A is symmetric if

$$A = A^T. \tag{2.10}$$

Positive definite and negative definite matrices

A positive definite matrix is a special case of symmetric matrix. A symmetric $n \times n$ real matrix A is said to be positive (semi)definite if $z^T Az$ is positive (non-negative) for every non-zero column vector of n real numbers. Similarly, a negative definite matrix is defined by the same expression, but in this case $z^T Mz$ must be always negative.

Irreducible matrix

A matrix is said to be irreducible if it is not similar via a permutation to a block upper triangular matrix. In a special case related to graph theory, by replacing non-zero entries in the matrix by one, and viewing the matrix as the adjacency matrix of a directed graph, the matrix is irreducible if and only if such directed graph is strongly connected.

Perron-Frobenius theorem for non-negative matrices

We assume that matrix T is irreducible. The Perron-Frobenius theorem states:

Theorem 3 *Let T be irreducible,*

- *T has a positive (real) eigenvalue λ_{max} such that all other eigenvalues of T satisfy $|\lambda| \leq \lambda_{max}$.*
- *Furthermore λ_{max} has algebraic and geometric multiplicity one, and has an eigenvector x with $x > 0$.*
- *Any non-negative eigenvector is a multiple of x.*
- *More generally, if $y \geq 0$, $y \neq 0$ is a vector and μ is a number such that $Ty \leq \mu y$, then $y > 0$ and $\mu \geq \lambda_{max}$, where $\mu = \lambda_{max}$ if and only if y is a multiple of x.*
- *If $0 \leq S \leq T$, $S \neq T$ then every eigenvalue σ of S satisfies $|\sigma| < \lambda_{max}$.*
- *If T is primitive, then all other eigenvalues of T satisfy $|\lambda| < \lambda_{max}$.*

Kronecker product

The Kronecker product of two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$ with $a_{ij} = [A]_{ij}$ and $b_{ij} = [B]_{ij}$, denoted $A \otimes B$, is defined as the $np \times mq$ matrix

$$\begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ a_{21}B & \cdots & a_{2m}B \\ a_{31}B & \cdots & a_{3m}B \\ \vdots & \vdots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{bmatrix} \tag{2.11}$$

Gershgorin circle theorem

Let A be a complex $n \times n$ matrix, with entries a_{ij}. For $i \in \{1, \ldots, n\}$ let $R_i = \sum_{j\neq i} |a_{ij}|$ be the sum of the absolute values of the non-diagonal entries in the i-th row. Let $D(a_{ii}, R_i)$ be the closed disc centered at a_{ii} with radius R_i. Such a disc is called a Gershgorin disc.

Theorem 4 *Every eigenvalue of A lies within at least one of the Gershgorin discs $D(a_{ii}, R_i)$.*

2.2.2 Linear Matrix Inequalities

Generally, a Linear Matrix Inequality has the following form:

$$F(x) \triangleq F_0 + \sum_{i=1}^{m} x_i F_i > 0, \tag{2.12}$$

where $x = \mathbb{R}^m$ is the variable and symmetric matrices $F_i = F_i^T \in \mathbb{R}^{n\times n}$, $i = 0, ..., m$ are given. This means that the LMI (2.12) is a convex conxtraint in x, i.e., the set $\{x|F(x) > 0\}$ is convex.

To the special case of nonlinear inequalities, these can be converted to LMI with the aid of Schur complements. That is, the LMI:

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0, \tag{2.13}$$

where $Q(x) = Q(x)^T$, $R(x) = R(x)^T$, and $S(x)$ depend affinely on x, is equivalent to

$$R(x) > 0 \qquad Q(x) - S(x)R(x)^{-1}S(x)^T > 0. \tag{2.14}$$

Therefore, the set of nonlinear inequalities 2.14 can be represented as the LMI 2.13.

2.3 Reset systems

Reset control systems are a special type of hybrid systems in which the time evolution depends both on continuous dynamics between resets and the discrete dynamics corresponding to the reset instants. Reset systems were first introduced in [Cle58], where a nonlinear integrator based on a reset action was presented. Basically, the *Clegg Integrator* sets its output to zero whenever its input is zero, leading to a faster system response without excessive overshooting.

A special case of impulsive systems is given by linear systems with finite state jumps. In this case, the continuous-time linear system undergo a state change at some discrete instants of time. Such systems can be regarded as a special case of reset systems in the sense that the reset rule is done through a time condition instead of a state condition. To clarify this, we use the reset system from [HDTP13] as an example. Consider the following linear reset system:

$$\begin{cases} \dot{y}(t) = A_c y(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ y(t) = A_r y(t^-) & \forall t \in \mathcal{T} \\ y(0) = y_0 \in \mathbb{R}^n \end{cases} \tag{2.15}$$

where $x \in \mathbb{R}^n$ represents the system state and the matrices A_c and A_r are constant matrices. The set

$$\mathcal{T} = \{t_k \in \mathbb{R}_+ \mid t_k < t_{k+1},\ \forall k \in \mathbb{N}, t_k \text{ reset time}\} \tag{2.16}$$

represents the set of reset times. We assume that the reset times are not periodic, and instead take the form $t_{k+1} - t_k = \delta + \delta'$, where $\delta \in \mathbb{R}_+$ is the fixed period and $\delta' \in \Delta$ is a jitter belonging to the compact set $\Delta \subset \mathbb{R}_+$. Thus the set of reset times $\mathcal{T}$ belongs to the set of all admissible reset sequences associated with Δ:

$$\Phi(\Delta) \triangleq \left\{ \{t_k\}_{k\in\mathbb{N}}, t_{k+1} - t_k = \delta + \delta'_k, \delta'_k \in \Delta, \forall k \in \mathbb{N} \right\} \tag{2.17}$$

We present now the definition of Globally Uniformly Exponentially Stable (GUES):

Definition 5 *The equilibrium $y^* = \mathbf{0}_n$ of the system (2.15) is GUES with respect to the set of reset sequences $\Phi(\Delta)$ if there exist positive scalars c, λ such that for any $\mathcal{T} \in \Phi(\Delta)$, any $y_0 \in \mathbb{R}^n$, and any $t \geq 0$*

$$\|\varphi(t, y_0)\| \leq c e^{-\lambda t} \|y_0\| \tag{2.18}$$

where $\varphi(t, y_0)$ is the value of the solution at time $t \in \mathbb{R}^+$, with the initial condition $x_0 \in \mathbb{R}^n$, and the reset sequence $\mathcal{T} \in \Phi(\Delta)$.

The following theorem is an instrumental result from [HDTP13]:

Theorem 6 (Theorem 1 in [HDTP13]) *Consider the system (2.15) with the set of reset times $\mathcal{T} \in \Phi(\Delta)$. The equilibrium $y^* = \mathbf{0}_n$ is GUES if and only if there exists a positive function $V : \mathbb{R}^n \mapsto \mathbb{R}_+$ strictly convex,*

$$V(y) = y^\top S_{[y]} y,$$

homogeneous (of second order), $S_{[\cdot]} : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$, $S_{[y]} = S_{[y]}^\top = S_{[ay]} > 0$, $\forall x \neq 0, a \in \mathbb{R}$, $a \neq 0$, $V(0) = 0$, such that $V(y(t_k)) > V(y(t_{k+1}))$ for all $y(t_k) \neq 0$, $k \in \mathbb{N}$ and any of the possible reset sequences $\mathcal{T} \in \Phi(\Delta)$.

Chapter 3

Consensus in multiagent systems

This chapter provides an overall view of the state of art in consensus theory. We begin the chapter with by providing motivation for the consensus problem in an overview of consensus theory in static networks, i.e. networks that retain their topology through time. To improve presentation, we divide this initial section into static networks which the agents have continuous and discrete dynamics. We continue the chapter with contributions in the field of switching networks, once again making the distinction between continuous and discrete dynamics. Afterward, we present the results of the literature in the clustering of networks. The chapter ends with the contributions of the thesis and a discussion of the literature presented.

As stated in Chapter 1, the agreement (consensus) problem in multiagent systems is a very common one, applicable to many fields such as engineering, sociology, biology and economics. Agreement protocols are easily found in nature, with several animals agreeing in order to complete a collaborative task. To better elaborate the motivation for studying the consensus problem, we provide a brief section with the use of consensus problem in different areas.

3.1 Motivation on consensus

In the Introduction chapter we motivated the consensus problem with cases in the nature such as a flock of birds, school of fish, etc. However, the consensus problem can be seen in several other fields as well. In this subsection we motivate the study of consensus problem in several areas of knowledge.

Formation

Distributed aerospace systems, such as multiple spacecraft, fleets of autonomous rovers, and formations of unmanned aerial vehicles, provide an wide array of applications. It is envisioned that distributed aerospace technologies will enable the implementation of a spatially distributed network of vehicles that collaborate toward a single collective scientific, military, or civilian goal. Distributed aerospace systems lead to higher degrees of scalability and adaptability in response to changes in the mission goals and system capabilities.

An example of a multiple platform aerospace system is space-borne optical interferometry [Nas]. Space interferometers are composed of separated optical instruments, leading to a so-called sparse aperture. Although optical interferometers can, in principle, function on the earth's surface, there are many advantages in operating them in space. Space-borne interferometers have greater optical sensitivity and resolution, wider field of view, and greater detection capability. Another example is the formation of Unmanned Aerial Vehicles (UAVs) for scanning an area or protecting civil or military targets [DRE09]. Formation control is also important on Autonomous Underwater Vehicles (AUVs) [BASCdW11, BASCDW09], specially in sensoring, e.g. find the gradient of pollution in the ocean [LFPA06], rescue operation [KRS04], source localization [ZFBZ08].

Sensor networks

A wireless sensor network consists of spatially distributed autonomous devices that cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, or pressure. Each node in a sensor network is equipped with a wireless communication device as well as an energy source–such as a battery–that needs to be efficiently utilized. The size, cost, and fidelity of a single sensor node can vary greatly, often in direct correspondence with its energy use, computational speed, and the ease by which it can be integrated within the network. Each sensor exchanges information on its local measurements with other nodes in the network in order to reach an accurate estimate of the physical or environmental variable of interest. We note that the efficiency requirement on the utilization of the energy source for each sensor often dictates a geometry on the internode communication for the sensor network. Moreover, sensor networks are often divided in clusters where a leader send information from the cluster sporadically [HGJ03, SPNW06].

Social networks

Social networks are comprised of social entities, such as individuals and organizations, with

a given set of interdependencies. The interaction between these entities can assume a multitude of relations, such as financial, social, and informational [WF94]. Such networks are of great interest in a variety of fields, including theoretical sociology, organizational studies, and sociolinguistics. In fact, the structure of social networks has always been of fundamental importance for understanding these networks. More recently, the notion of manipulating the network structure has been contemplated as a viable means of altering the network behavior. For example, the concept of a change agent refers to a network entity that intentionally or indirectly causes or accelerates social, cultural, or behavioral change in the network.

Energy networks

Complex, large-scale energy systems, delivering electrical and mechanical energy from generators to loads via an intricate distribution network, are among the most useful engineered networked dynamic systems. These systems often consist of a heterogeneous set of dynamic systems, such as power electronics and switching logics, that evolve over multiple timescales. Dynamics, stability, and control of individual power system elements (e.g., synchronous machines) or their interconnections (e.g., multi-machine models) have extensively been examined in the literature. However, as the need for more efficient generation and utilization of energy has become prevalent, distributed and network architectures such as the "smart grid" have gained particular prominence [ZC12].

3.2 Preliminaries on consensus

In this section some preliminaries on consensus problem are presented. To provide an easy illustration, we take an example from sensor networks as it was presented in the previous section. Consider a group of sensors which task is to measure the temperature of an area. Although the temperature of each sensor will vary according to their position, what is required is a single value for the temperature related to this group. Therefore, all the sensors must communicate in a network and agree to a single value and to reach this objective, an agreement protocol must be proposed. Consider that the network does not change with time, but an agent communicates only with its neighbors. For the sake of simplicity, we first consider the simplest protocol, where the agents are in an undirected networks. Following our example of a group of n sensors, each sensor has the following agreement protocol:

$$\dot{x}_i = \sum_{j \in N(i)} (x_j(t) - x_i(t)), \quad \forall i = 1, ..., n, \tag{3.1}$$

where $x_i \in \mathbb{R}$ is the state of sensor i and $N(i)$ is the set of sensors connected to sensor i in the network, or "adjacent" to sensor i. However, the overall system can be represented as a whole:

$$\dot{x}(t) = -L(\mathcal{G})x(t), \tag{3.2}$$

where $L(\mathcal{G})$ is the Laplacian matrix (as shown in Equation (2.3)) of an undirected graph $\mathcal{G}$. As we can expect from the consensus protocol, the values of these agents are pulled towards each other and finally reach an weighted average of their initial conditions. We refer this weighted average to as *consensus* or *agreement*. Therefore, it is necessary to provide a formal definition of agreement. We do so by presenting the definition of agreement, which comes from [ME10]. The agreement in a graph can be defined as:

Definition 7 *The agreement set $\mathcal{A} \subseteq \mathbb{R}^n$ is the subspace $span\{\mathbf{1}_n\}$, that is,*

$$\mathcal{A} = \{x \in \mathbb{R}^n | x_i = x_j, \forall i, j \quad \in \{1, ..., n\}\}. \tag{3.3}$$

The Laplacian matrix of an undirected graph offers other interesting properties. Since an undirected graph generates a symmetric (and also positive semidefinite, due to the characteristics of the Laplacian) Laplacian matrix, its eigenvalues, are real and can be ordered:

$$\lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq ... \leq \lambda_n(\mathcal{G}), \tag{3.4}$$

with $\lambda_1(\mathcal{G}) = 0$. In addition to this result, a graph is said to be connected if $\lambda_2(\mathcal{G}) > 0$. The significance of λ_2 goes further than the connectivity of a graph. The second eigenvalue of L influences the speed of convergence of an undirected graph $\mathcal{G}$ and it is known as the *algebraic connectivity* of a graph $\mathcal{G}$. We state this formally using the Theorem 3.4 from [ME10]:

Theorem 8 *Let $\mathcal{G}$ be a connected graph. Then the (undirected) agreement protocol* (3.2) *converges to the agreement set* (3.3) *with a rate of convergence that is dictated by $\lambda_2(\mathcal{G})$.*

Another important result concerning the algebraic connectivity comes from [GR13], which takes into account the increase of λ_2 when an edge is added to the graph. We present the Lemma 13.6.1 of [GR13] below:

Lemma 9 *Let $\mathcal{X}$ be a graph and let $\mathcal{Y}$ be obtained from $\mathcal{X}$ by adding an edge joining two distinct vertices of $\mathcal{X}$. Then,*

$$\lambda_2(\mathcal{X}) \leq \lambda_2(\mathcal{Y}) \leq \lambda_2(\mathcal{X}) + 2. \tag{3.5}$$

Now that the undirected case has been presented, we focus on the consensus theory of directed graphs. From Chapter 2, we defined a directed graph as a graph where the influence between

agents is not symmetrical, or agent i influences agent j but agent j may not influence agent i. Therefore, considering a digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ we recall that this digraph can be represented by:

$$\dot{x}(t) = -L(\mathcal{D})x(t), \tag{3.6}$$

where $L(\mathcal{D})$ is the Laplacian associated to the digraph $\mathcal{D}$. Our goal now is to identify sufficient and necessary conditions on $\mathcal{D}$ that lead the system (3.6) to the agreement set. Since the Laplacian retains its properties of sum of the rows equal to 0, the smallest eigenvalue will be equal to 0. However, it is important to study the remaining eigenvalues of directed Laplacian. Recalling the Gershgorin circle theorem, we present the following Proposition from [ME10]:

Proposition 10 *Let $\mathcal{D}$ be a weighted digraph on n vertices. Then the spectrum of $L(\mathcal{D})$ lies in the region*

$$\{z \in C || z - \bar{d}_i n(\mathcal{D})| \leq \bar{d}_i n(\mathcal{D})\}, \tag{3.7}$$

where $\bar{d}_i n$ denotes the maximum (weighted) in-degree in $\mathcal{D}$. In other words, for every digraph $\mathcal{D}$, the eigenvalues of $L(\mathcal{D})$ have non-negative real parts.

This result allows a better understanding of the Laplacian matrix of digraphs and, moreover, leads to a statement about the consensus value of a digraph. We state this result about the consensus value in the following Lemma:

Lemma 11 *Let $\mathcal{D}$ be a strongly connected digraph and L the corresponding Laplacian matrix. Consider a network of agents whose collective dynamics is described by $\dot{x}(t) = -Lx(t)$. Let us also consider $L\mathbf{1} = \mathbf{0}$, $u^\top L = \mathbf{0}$ and $u^\top \mathbf{1} = 1$. Then, the agents asymptotically reach a consensus and the consensus value is given by $x^* = u^\top x(0)$. Moreover, the vector u defines an invariant subspace for the collective dynamics: $u^\top x(t) = u^\top x(0), \forall t \geq 0$*

Finally, this result permits a characterization of the consensus value depending only on the topology of the network and the initial conditions.

Discrete dynamics

The discrete dynamics of the agreement protocol has a strong relation with the theory of Markov chains. Therefore, the agreement protocol for discrete cases can be represented as stochastic matrix P. This means that the ijth entry of the matrix P, denoted p_{ij}, is the probability of a random variable X having a state i at time k, assumes state j at time $k+1$. So we can define the overall dynamics of the discrete network as

$$x(k+1) = Px(k), \tag{3.8}$$

where P is a stochastic matrix, which can be defined in a number of ways. Consider the Laplacian of a digraph previously defined, we can discretize its continuous dynamics by assuming a constant sampling interval δ. This leads to the following equation

$$x(k+1) = e^{-L(\mathcal{D})\delta}x(k). \tag{3.9}$$

The next task is to connect (3.8) and (3.9). To do so we present the following Proposition, which comes from Proposition 3.18 of [ME10]:

Proposition 12 *For all digraphs $\mathcal{D}$ and sampling intervals $\delta > 0$, one has*

$$e^{-L(\mathcal{D})\delta}\mathbf{1} = \mathbf{1} \quad \textit{and} \quad e^{-L(\mathcal{D})\delta} \geq 0; \tag{3.10}$$

that is, for all $\mathcal{D}$ and $\delta > 0$, $e^{-L(\mathcal{D})\delta}$ is a stochastic matrix.

Similarly, another way to define a discrete agreement matrix is by what is called a *Perron matrix*. Consider a integer $\epsilon \in (0, 1/2\Delta)$ where Δ is the biggest degree of an agent in the network. Equation (3.8) can be written as

$$x(k+1) = (I - \epsilon L)x(k). \tag{3.11}$$

For clarity, we provide as well a modified form of the agreement protocol for each agent

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N(i)} (x_j(k) - x_i(k)), \quad \forall i = 1, ..., n. \tag{3.12}$$

3.3 Static networks

Networks are said to be of static type when the topology does not change in time. These are the most simple cases studied in consensus and usually takes into consideration networks where network does not change or change very little, e.g. a network of sensors of a machine in a factory, or are less likely to suffer from faults. Another problem that greatly uses the idea of static networks is the distribution of work load in a cluster of processors. As these communication links do not change through time we can safely model it as a static network.

In this section we present results on both the continuous and discrete domains. We present results that provided common tools for the solution of consensus problems in both discrete and continuous time.

3.3.1 Continuous

Continuous interactions are used to model systems where the communications happen in continuous time. This means is the case in certain fields, such as biology. One of the examples that we can take from biology is a group of neurons transmitting information continuously between themselves. In another case pertaining the biology field is flocking. To simulate the movements of a flock of birds is necessary continuous sensing. In some other cases, discrete interactions that are fast enough are approached to a continuous interaction.

The work of [OSFM07] provided important tools both in the continuous and discrete field. In continuous consensus an important result is Theorem 1, which provide conditions for the consensus in directed graphs. We state the Theorem in the following:

Theorem 13 *Consider a network of n agents with topology $\mathcal{D}$ applying the following consensus algorithm:*

$$\dot{x}_i(t) = \sum_{j \in N(i)} a_{ij}(x_j(t) - x_i(t)), \qquad x(0) = z. \tag{3.13}$$

where a_{ij} is the weight of the opinion of agent i to agent j. Suppose $\mathcal{D}$ is a strongly connected digraph. Let L be the Laplacian of $\mathcal{D}$ with a left eigenvector $\gamma = (\gamma_1, ..., \gamma_n)$ satisfying $\gamma^T L = \mathbf{0}$. Then

- *a consensus is asymptotically reached for all initial states;*
- *the algorithm solves the consensus problem with the linear function $f(z) = (\gamma^T z)/(\gamma^T \mathbf{1})$, i.e., the consensus value is $\alpha = \sum_i \gamma_i z_i$ with $\sum_i \gamma_i = 1$;*
- *if the digraph is balanced (Definition 2), an average-consensus is asymptotically reached and $\alpha = (\sum_i z_i)/n$.*

In [OSM04a] the authors provide a series of tools for static networks. The authors establish a direct connection between the algebraic connectivity of the network and the performance (or negotiation speed) of a linear consensus protocol. This requires the generalization of the notion of algebraic connectivity of undirected graphs to digraphs. It turns out that balanced digraphs play a key role in addressing average-consensus problems. To help with the convergence analysis, disagreement functions are introduced. The authors consider the effect of time-delays in the convergence of a fixed network.

Other works focus on specific cases of fixed networks. While in most cases consensus is considered with simple dynamics such as the single integrator, in some cases more complex dynamics must be used. In [XW05] the authors consider a typical point mass model with continuous fixed networks. The consensus is achieved using a two step algorithm: a local speed feedback controller and the interactions from the neighbors. Some works consider leader following on

fixed network topology [NC10]. In [NC10], the authors present a network with a leader agent and instead on focusing into single or double integrator dynamics, the agents have a nth order linear system. Another interesting progress is in static networks with sampled control protocol [XLWJ09]. The main contributions of [XLWJ09] is to provide a framework for continuous fixed networks with sampled data, with sampling delay or not.

3.3.2 Discrete

Discrete static networks may not be commonly seen in the nature as their continuous counterpart, but are very important in other fields such as engineering and economics. In [OSFM07] not only tools in continuous dynamics but also on discrete dynamics were provided. Similar to the result in continuous part, we focus on one theorem, Theorem 2 which states conditions for the stability of a discrete directed graph. We state the theorem in the following:

Theorem 14 *Consider a network of n agents with topology $\mathcal{G}$ applying the following consensus algorithm:*

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N(i)} a_{ij}(x_j(k) - x_i(k)) \tag{3.14}$$

where $0 < \epsilon < 1/\Delta$ and Δ is the maximum degree of the network. Suppose $\mathcal{G}$ is a strongly connected digraph. Let P be the Perron matrix of $\mathcal{G}$ with a left eigenvector $\omega = (\omega_1, ..., \omega_n)$ satisfying $\omega^T P = \omega$. Then

- *a consensus is asymptotically reached for all initial states;*
- *the consensus value is $\alpha = \sum_i \omega_i x_i(0)$ with $\sum_i \omega_i = \mathbf{1}$;*
- *if the digraph is balanced (or P is doubly stochastic), an average-consensus is asymptotically reached and $\alpha = (\sum_i x_i(0))/n$.*

Another interesting point of research is related to the convergence rate of graphs. In [OT09] the authors investigate the convergence rate of static graphs with agents with discrete dynamics. The authors characterize the worst-case convergence on the special case of averaging algorithms. The authors provide the convergence rate as a function of the number n of agents as well as a lower bounds on the convergence time.

3.4 Switching networks

Switching networks refer to the case where connections inside a network change through time, for a number of different conditions. Networks might change due to proximity of the agents or

connections might be lost because of faults in the system. In some cases the network topology might be time-dependent or state-dependent. In fact, the formation problem discussed in the Introduction often uses switching networks to model the network topology. In reality the mobile robots (or UAVs, AUVs) have a limited range of communication and will only transmit information to the neighbors inside its transmission range. Transmission range is also an important in the rendez-vous problem. Agents must agree with a meeting point and the only information available is the derived position of its neighbors. This limited range of communication means that communication between agents increase as they get close to each other, and finally, at the meeting point.

Another well known problem that requires the use of switching networks is the *gossiping* problem, motivated from the advent of sensor, wireless ad hoc and peer-to-peer networks. In this case the network does not change due to practical limitations such as transmission range, but instead follows a very specific protocol. In gossiping at a given time only two agents will communicate, or *gossip*. The order of these communication can happen in two different ways: randomized, where any two agents can communicate at a given time, or by a fixed sequence of interactions.

In this section we present results concerning this kind of network topologies. At first we present results on continuous switching networks. Secondly we focus on discrete dynamics for switching networks.

3.4.1 Continuous

One of the papers that provide not only tools for static network but also for networks with switching topology was [OSM04a]. In this work, the authors apply the disagreement functions first presented to the static case to a directed network with switching topology. This is done considering directed graphs, where the information between two agents does not necessarily flows on both directions.

One of the most general framework used for the study of networks with switching topology is based on the cut-balance assumption. This technique was first presented in [HT13] and its motivation comes from studies on social networks and opinion dynamics [CFL09, Lor07]. The idea behind this assumption is that in many systems an agent cannot influence the others without receiving back at least a part of its own influence. In other words, if a set of agents influence another set of agents the former group is influenced in a proportional amount. To clarify this statement, we present Assumption 1 in [HT13]:

Assumption 1 *There exists a constant $K \geq 1$ such that for all t, and any nonempty proper*

subset $S = \{1, ..., n\}$ *we have*

$$K^{-1} \sum_{i \in S, j \notin S} a_{ji}(t) \leq \sum_{i \in S, j \notin S} a_{ij}(t) \leq K \sum_{i \in S, j \notin S} a_{ji}(t), \tag{3.15}$$

where $a_{ij}(.)$ *is a nonnegative and measurable function.*

The main contribution of this work is to prove convergence of a multi agent system only with the cut-balance condition. Convergence to consensus is possible under some strong assumptions about persistent global connectivity. The importance of obtaining convergence of the system with no other requirement except for the cut-balance is important where evolution of the function $a_{ij}(t)$ is not previously known. This is particularly interesting in cases where the evolution of a_{ij} is dependent on the vector $x(t)$ itself, when the connectivity cannot be checked a priori. Results that apply the cut-balance condition will be presented in the next subsection.

Another work that uses the idea of cut-balance condition introduced by [HT13] is in [MG13]. The results in this work prove consensus for continuous switching network topologies using two techniques. The first one is the assumption on persistent connectivity, which can be considered as the continuous counterpart of the notion of ultimate connectivity from the discrete domain. The second on is the the slow divergence of reciprocal interaction weights, which comes from the cut-balance assumption defined before. The authors show that under these two conditions a continuous time-protocol succeeds to achieve consensus. Moreover, the proof also allows an estimate of convergence rate of the system.

3.4.2 Discrete

In this subsection we present some results on discrete dynamics of switching networks. Most of the literature in this subsection deals with the convergence of the system to an agreement value. Results that were first presented in continuous part return here as their discrete counterpart.

In [JLM03] the authors present the results on mobile agents using nearest neighbor rules. This work follows the work of Vicsek et al. [VCBJ+95] where autonomous agents (i.e., points or particles) are all moving in the plane with the same speed but with different headings. Each agent's heading is updated at a discrete time t by taking the average of its heading plus the headings of its neighbors. In [VCBJ+95] simulations of the results demonstrate that eventually all agents converge to the same direction despite the absence of a centralized control and the set of neighbors for each agent changes as time evolves. In [JLM03] a theoretical explanation for this phenomena is provided in two different ways. Firstly, it is proven that all agents converge to common a steady state heading provided that all agents remain connected with sufficient frequency as the system evolves. The second result explores the idea where an agent 0 is added

to the network, know as leader. The agents may or may not have the leader as a neighbor. The authors prove that all agents converge to the leader's heading if all agents are connected frequently enough to the leader via their neighbors.

In [RB+05], the authors investigate the consensus problem under dynamically changing topologies. In this case, both continuous and discrete dynamics are investigated considering directed graphs. In some cases, switching topologies are used because the connections are not reliable and the topology of the network changes as time passes. The authors prove that for some given conditions consensus is reached in a dynamically changing network. More specifically, consensus is achieved if the union of the collection of interaction graphs has a spanning tree frequently enough. The authors also relax the contribution from the Perron-Frobenius theorem by showing that for non negative matrices is the enough to define a spanning tree, while the Perron-Frobenius only deals with irreducible matrices.

The cut-balance condition introduced in [HT13] was also considered for discrete switching networks. In [TN11] the authors propose a modification of the cut-balance condition for discrete dynamics, which is called cut-balanced chains. In [TN14] the authors deal with the convergence properties of the products of random (row-)stochastic matrices. The authors study a class of random stochastic matrices, referred as balanced chains and show that this class contains many of the previously studied chains of random and deterministic stochastic matrices. This kind of chains have been studied in [SJ11, BM12, MG13].

While most of these works deal with the stability of linear systems, the work in [Mor05] takes into consideration a network with discrete nonlinear dynamics. The model proposed by the authors is of a network interacting via time-dependent communication links. Time-dependent communication allows for an array of possible situations in the network, such as: take into account link failure and link creation, reconfigurable networks and nearest neighbor coupling. Non bidirectional communications are also considered, which is important in a problem like flocking.

The work in [Mor05] can be seen as an expansion of the works in [JLM03, OSM04a, RB+05]. At first, considering a nonlinear discrete model the authors provide necessary and sufficient conditions on the network topology for the convergence of all agents to a single value. The analysis that leads to this results is done through a novel approach centered in the notion of convexity. For clarity, convexity can be defined as follows: consider a communication graph $\mathcal{G}$ with discrete time dynamics state $x_i(t)$ for each agent i. In order for the network to present convexity, for each agent i the updated state $x_i(t+1)$ is a strict convex combination of agent's i current state $x_i(t)$ and the current states $x_j(t)$ of its neighbors. Normally results that study the stability of networks with switching topology rely on tools of algebraic graph theory [JLM03, OSM04a, RB+05, FM01, FM04] while the work [Mor05] utilizes a blend of graph-theoretic

and system-theoretic tools.

In [AB09] the purpose of the paper is to provide several criteria to estimate quantitatively the contraction rate of a set of agents towards consensus. This is classically done by estimating the second largest Lyapunov exponent of an infinite product of matrices. Conversely, the approach proposed by the authors works by ensuring a lower bound to the matrix entries of the agents already attained by the information flow along the spanning-tree, therefore permitting to obtain tighter estimates.

3.5 Clustering of networks

This section results concerning the clustering of networks will be discussed. Clustering has two main lines of research, the first one focus on the division of network into groups or clusters of agents. The usefulness of this approach comes mainly from its ties to social networks, such as hierarchical consensus, but other fields of science may benefit from this separation. In engineering the separation of a network into smaller clusters can be used in sensoring, where a network of clusters is separated into smaller groups with a *leader* or, in a more specific term, cluster head [HGJ03, SPNW06]. As example of another application is clusters of satellites, where a group of satellites exchange information inside the group and in certain moments exchange information with other clusters. In engineering all these applications have one thing in common: keeping frequent or more important communication inside a group and less frequent information exchange inter-clusters.

The second line of research focus on consensus (often local) of these clusters. These works naturally expand from initial results such as the Krause model [Kra97] or the Deffuant-Weisbuch model [DNAW00]. In this case presenting a method for the separation of clusters is not enough, the works also provide a method to prove that consensus is reached inside each cluster (even if the consensus reached is a local one, related to each cluster).

3.5.1 Graph partitioning

Graph partitioning is an important problem in network clustering. An interesting example to motivate the clustering of networks is the problem of computer processes which need to be distributed over a number of processors. We can represent this as a graph where each process is a vertex and the communications are edges. In this case, the objective of the problem is to partitionate the processes between the computer processors in order to distribute the work

load of the system. An additional goal is to minimize the communication between computer processors, so the inter-processor communication (often more slow) is reduced.

Graph partitioning is very important in the social network analysis, specially in a technique called hierarchical clustering. The idea here is to discover the natural divisions inside a network to form groups, and this is done by two different methods: the agglomerative and divisive methods [Sco87]. These methods are separated depending on whether edges are added to or removed from network. In the agglomerative method the network starts without any edges which are added later by comparing two vertices through a given method, e.g. in a graph corresponding to film actors, two vertices are connected if these two actors appeared in the same movie [WS98, BBA75]. The agglomerative method, however, present some problems. Sometimes the agglomerative method fail to find the correct communities (if they were known a priori). Agglomerative methods can also find the cores of the clusters but leave the peripheries out.

The divisive method considers a fully populated graph and by checking the connected pairs and cutting the connections of the pairs that are least similar to each other. This is a iterative method, and by repeating this process we can reach smaller and smaller clusters.

Divisive method

In [NG04] the authors propose a slight modification of this method. Instead of looking to the nodes that are least similar, they look at the "betweenness" of two nodes. In the following we present these measures and the algorithms used.

The shortest path: A example of such a betweenness measure is that based on shortest (geodesic) paths: shortest paths between all pairs of vertices and count how many run along each edge. This measure was first introduced by Anthonisse [Ant71]. In this technical paper, such run was called "rush". However in [NG04] the term used is edge betweenness. This method for searching betweeness happens in the following steps:

1. The initial vertex s is given $distanced_s = 0$ and a weight $w_s = 1$.
2. Every vertex i adjacent to s is given distance $d_i = d_s + 1 = 1$, and weight $w_i = w_s = 1$.
3. For each vertex j adjacent to one of those vertices i one of three things are done:
   (a) If j has not yet been assigned a distance, it is assigned distance $d_j = d_i + 1$ and weight $w_j = w_i$.

(b) If j has already been assigned a distance and $d_j = d_i + 1$, then the vertex's weight is increased by w_i, that is $w_j \leftarrow w_j + w_i$.

(c) If j has already been assigned a distance and $d_j < d_i + 1$, do nothing.

4. Repeat from step 3 until no vertices remain that have assigned distances but whose neighbors do not have assigned distances.

Resistor networks: Coming from elementary circuitry theory is the resistor networks betweenness. Consider a circuit created by placing a unit resistance on each edge of the network and unit current source and sink at a particular pair of vertices. The resulting current flow in the network will travel from source to sink along a multitude of paths, those with least resistance carrying the greatest fraction of the current. The current-flow betweenness for an edge we define to be the absolute value of the current along the edge summed over all source/sink pairs. This is calculated using Kirchhoff's law. Let V_i be the voltage at vertex i, measured relative to any convenient point. Then for all i we have

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}, \tag{3.16}$$

where A_{ij} is the ij element of the adjacency matrix of the graph. The left-hand side of Equation (3.16) represents the net current flow out of vertex i along edges of the network, and the right-hand side represents the source and sink. Moreover, $\delta_{is} = 1$ if agent i is the source, $\delta_{it} = 1$ is agent i is the sink and $\delta_{is} = \delta_{it} = 0$ otherwise. Defining $k_i = \sum_j A_{ij}$ and creating a diagonal matrix $\mathbf{D}$ with $D_{ii} = k_i$. The equation can be rewritten as $(\mathbf{D} - \mathbf{A})\mathbf{V} = \mathbf{s}$, where vector $\mathbf{s}$ has the components:

$$s_i = \begin{cases} +1 & \text{for} \quad i = s \\ -1 & \text{for} \quad i = t \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

The matrix $(\mathbf{D} - \mathbf{A})$ (which is equivalent to the Laplacian matrix) cannot be inverted because is singular. By choosing a reference potential, one degree of freedom is fixed and $(n-1)$ must be found. By considering any vertex v to be the reference point, a column and a row of the matrix are removed. Therefore, the resulting equation can be written as:

$$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1}\mathbf{s}, \tag{3.18}$$

where $\mathbf{D}_v$ and $\mathbf{A}_v$ are the resulting matrices $(n-1) \times (n-1)$. By choosing a convenient vertex v the calculate of the differences between sink and sources are defined. The complete betweenness score is defined by summing over all sinks and sources.

Random walk: In terms of the shortest path betweenness we can think of signals that travel around the network in the geodesic path. Suppose that signals instead just perform a random walk about the network until they reach their destination. This gives another measure on edges, the random-walk betweenness: the expected net number of times that a random walk between a particular pair of vertices will pass down a particular edge is calculated and sum over all vertex pairs. The algorithm is fairly similar to the resistor networks algorithm. In this case, however, we consider the vector $\mathbf{s}$ with one element equal to 1, corresponding to the source element, and all the others equal to 0. Still, the equation is fairly similar, giving $\mathbf{V} = (\mathbf{D}_t - \mathbf{A}_t)^{-1}\mathbf{s}$ The betweeness of two edges v and w is characterized by the difference between V_v and V_w.

Other approaches to clustering of networks are [BA08] and [CK85], which provide a separation of the network by viewing it as a singular perturbation system. In this singular perturbation modeling the densely connected nodes in these areas synchronize in the fast time-scale, and behave as aggregate nodes that dominate the slow dynamics of the network. . In [BA08] the authors study a nonlinear network which exhibits areas of internally dense and externally sparse interconnections. The authors derive a singular perturbation model which makes this time-scale separation explicit and prove the validity of the reduced-model approximation. In [CK85] The authors reach a singular perturbation model analyzing two parameters: time-scale and weak coupling properties.

3.5.2 Consensus in clustered networks

In some cases the authors are not only interested in finding the divisions of a network, but also to reach consensus within these sub networks. This approach is specially useful for social networks, specially hierarchical networks. In hierarchical networks the overall network may have several smaller groups that are separated by an hierarchy. Therefore consensus is reached within these sub networks before the information is spread through higher ranks. In the last subsection we presented results concerning the division or "modularity" of a network. In this subsection we present results of literature where the interest here is to reach a consensus within these sub networks or clusters.

One of the first approaches to this problem was presented by Krause in [Kra97], which can also be called Hegselmann-Krause model as in [HK02]. This is interpreted as a opinion dynamics with bounded confidence, where all the agents have an opinion to a certain subject which is quantified in some way. Moreover, every agent updates its opinion averaging the opinions of its neighbors that differ from its opinion less than one. To ease the understanding, we present the discrete consensus protocol from [Kra97]. Consider a system with n agents where x_i is the

opinion of agent i. These values are updated according to:

$$x_i(t+1) = \frac{\sum_{j:|x_i(t)-x_j(t)|<1} x_j(t)}{\sum_{j:|x_i(t)-x_j(t)|<1} 1}, \tag{3.19}$$

where two agents i, j are said to be connected if $|x_i(t) - x_j(t)| < 1$ at time t. In social dynamics we can see this protocol as agent i refusing to talk with other agents unless their opinion is reasonable, that is, close enough to their own opinion. Krause's model has been well studied in the literature, to cite a few [Kra00, Lor06, Lor05]. Another model which also considers a bounded confidence is the Deffuant-Weisbuch model [DNAW00]. In that model, two randomly selected agents update their opinions at any given time step. If the agents differ by more than a certain threshold, their opinions remain unchanged; otherwise, each agent moves to a new opinion which is a weighted average of its previous opinion and that of the other agent. The main difference of this model is that it takes into consideration a pair of agents at a time, while Krause's model can potentially consider the opinion of the whole network. Other similar models can be seen here [BNKVR03, BNKR03].

In this subsection we will focus primarily in two different approaches that derive from the initial Krause's method. The first one is an study on Krause's model by Blondel et al. [BHT09]. In this paper the authors present a new convergence proof for the established model and equilibrium stability of the model concerning a perturbing agent. The second result we will present comes from Morărescu and Girard [MG11]. In this paper the authors propose an expansion to Krause's model and instead of considering a fixed bound the authors analyze the speed of convergence of the agents, making so that only agents that approach each other at a certain prescribed speed remain connected. While the network as whole may not reach consensus, each cluster reaches a local consensus. These results lead to a decentralized technique of finding clusters inside a graph.

In [BHT09], following Krause's model as defined in (3.19), the authors proposed an alternative proof for the convergence of this model. While the convergence of this particular model has already been studied in other works [Dit01, Lor05], this is a simpler method to provide stability of this model. We present the following theorem, from [BHT09]:

Theorem 15 *If $x(t)$ evolves according to (3.19), then for every i, $x_i(t)$ converges to a limit x_i^* in finite time. Moreover, for any i, j, we have either $x_i^* = x_j^*$ or $|x_i^* - x_j^*| \geq 1$.*

It remains to prove that the convergence happens in finite time. Consider a set of agents converging to a limiting value. There will be a time where none of the agents are connected to agents outside this set. Eventually they are close enough that all agents are connected to each other. All agents will compute the same average and reach agreement in the next time step

where they remain for all subsequent time. Since this happens in finite time, it can be said that the convergence happens in finite time.

Motivated by the stability proof presented in Theorem 15 the authors also developed stability conditions for protocol (3.19) considering a perturbing agent. The algorithm works by adding a perturbing agent to the network with a weight δ. This will lead to a new equilibrium point called $\bar{x}$. Once this equilibrium is reached, the perturbing agent is taken from the network and a new equilibrium point is reached called $\bar{x}'$. The equilibrium $\bar{x}$ is stable if the distances between the original and perturbed equilibria converges to 0 as the weight δ of the perturbing agent vanishes. In other words, an equilibrium is unstable if a substantial change in the equilibrium can be induced by a perturbing agent of arbitrarily small weight.

The work of [MG11] comes from an expansion of the Krause's model. The network studied is discrete-time, relevant to opinion dynamics both in the case discussed in the previous sub section (graph partitioning) as well as consensus of clusters. To reach this objective, the proposed model presents some restrictions for the connection of agents. Agents only remain connected if their values (or opinions in social dynamics) are converging with at least a priori defined convergence speed. Given this restriction, global consensus may not be reached but local consensus is reached in every cluster.

What makes this strategy important comparing to the other graph partitioning techniques is that it does not focus only in the partitioning. Since the agents divide themselves considering how fast they agree to each other local consensus is ensured by the control protocol itself. This is done by assuming that all agents converge to the equilibrium with a certain decay rate, an assumption called *fast convergence assumption*. Considering that all the agents converge to a limit value no slower than $O(\rho^t)$, then there exists $\underline{\rho} < \rho$ and a matrix $M \geq 0$ such that for all $i \in V$

$$|x_i(t) - x_i^*| \leq M\underline{\rho}^t. \tag{3.20}$$

This assumption gives finally the condition for the separation of the edges. The agents only remain connected if it approaches its neighbors faster than the a priori defined convergence rate $\underline{\rho}$, as given by the following equation

$$|x_i(t) - x_j(t)| \leq 2M\underline{\rho}^t. \tag{3.21}$$

3.6 Thesis Contributions

In this section we state the contributions of the thesis in relation to the state of art previous presented. More generally, we present a novel framework to the consensus theory based on

reset systems. The main contributions can be separated into three main topics: characterization of a consensus value, proof of stability by means of LMI and control design of the leader's network to reach an a priori defined consensus value.

At first, we will introduce this novel framework. The motivation comes from studies on social networks [WF94], where there might be a separation of a large group into smaller ones. A large group might no be able to converse and exchange information frequently, but smaller groups might be capable of doing so. While in our work we consider that the network is fixed and known a priori, concepts on modularity of clustered networks can be used to define these divisions [BHT09, MG11, Kra97].

We are now ready to introduce the system. Consider a network of n agents described by the digraph (i.e. directed graph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V}$ represents the set of agents and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents the interactions. The agent set $\mathcal{V}$ is then partitioned in m strongly connected clusters/communities $\mathcal{C}_1, \ldots, \mathcal{C}_m$ and no link between agents belonging to different communities exists. Each community possesses one particular agent called leader and denoted in the following by $l_i \in \mathcal{C}_i,\ \forall i \in \{1, \ldots, m\}$. The set of leaders will be referred to as $\mathcal{L} = \{l_1, \ldots, l_m\}$. Each agent has a scalar state denoted also by l_i for the leader l_i and f_j for the follower f_j. We also introduce the vectors $x = (l_1, f_2, \ldots, f_{m_1}, \ldots, l_m, \ldots, f_{m_m})^\top \in \mathbb{R}^n$ and $x_l = (l_1, l_2, \ldots, l_m)^\top \in \mathbb{R}^m$ that collects the states of all the agents. Finally, we present the dynamics of the system

$$\begin{cases} \dot{x}(t) = -Lx(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ x_l(t_k) = P_l x_l(t_k^-) & \forall t_k \in \mathcal{T} \\ x(0) = x_0 \end{cases} \tag{3.22}$$

where P_l is a stochastic (Perron) matrix, and L is a Laplacian matrix in the block diagonal form with each element of the diagonal L_i as the Laplacian matrix associated to the cluster i, that means:

$$L = \begin{pmatrix} L_1 & & \\ & \ddots & \\ & & L_m \end{pmatrix},\ L_i \in \mathbb{R}^{n_i} \tag{3.23}$$

3.6.1 Characterization of the consensus value

In most cases, the main interest in a new agreement framework is to prove that a consensus can be reached. Therefore in some cases the agreement value is not important, and instead the focus remains on proving stability. However, in some papers the consensus value is characterized. Our first result is the characterization of the consensus value for this type of multiagent

systems. The importance of this result comes not only from characterization itself, but as a step stone for our next results. Therefore, we devise an algebraic equation that only depends of the network topology and the initial conditions. As a matter of fact, the time between resets is not a requirement to find the consensus value. In Equation (3.26) we show that consensus can be reached without the previous knowledge of the reset times.

We start this part by recalling Lemma 11. This Lemma states that the eigenvector u defines an invariant subspace for the continuous dynamics: $u^\top x(t) = u^\top x(0), \forall t \geq 0$. What Lemma 11 implies is that between two consecutive reset instants t_k and t_{k+1}, the agents belonging to the same community try to approach a local agreement defined by $x_i^*(k) = w_i^\top x_{\mathcal{C}_i}(t_k)$ where $x_{\mathcal{C}_i}(\cdot)$ is the vector collecting the states of the agents belonging to the cluster $\mathcal{C}_i$. The following vectors must be introduced:

$$\begin{aligned} x^*(t) &= (x_1^*(t), x_2^*(t), ..., x_m^*(t))^\top \in \mathbb{R}^m \\ u &= (v_1/w_{1,l}, v_2/w_{2,l}, ..., v_m/w_{m,l})^\top \in \mathbb{R}^m \end{aligned} \tag{3.24}$$

where $x_i^*(\cdot)$ represents the local agreement of the cluster $\mathcal{C}_i$ and $v \in \mathbb{R}^m$ and $w_i \in \mathbb{R}^{n_i}$ are defined at the beginning of the section as left eigenvectors associated with the matrices describing the reset dynamics of the leaders and the continuous dynamics of each cluster, respectively. With these introductory statements, we can define the following result:

Proposition 16 *Consider the system* (3.22). *Then,*

$$u^\top x^*(t) = u^\top x^*(0), \ \forall t \in \mathbb{R}_+. \tag{3.25}$$

What Proposition 16 means is that in the same way as we have an invariant subspace for the local networks, we can find an invariant subspace for the complete network. This allows us to define the consensus value for the whole network

$$x^* = \frac{u^\top W x(0)}{\sum_{i=1}^m u_i}. \tag{3.26}$$

This result states that the consensus value of system (3.22) only depends on the network topology and the initial conditions of system (3.22).

3.6.2 Stability in term of LMI

In this subsection we present the second major result of this thesis, the stability proof of the system dynamics defined in (3.22). Due to the similarity of the system dynamics (3.22) with the dynamics of reset systems, we drew inspiration from other results of reset systems. Normally stability of a consensus problem is proved by means of algebraic connectivity theory and

matrix theory. However, taking inspiration from the results of [HDTP13] we use Linear Matrix Inequalities to prove the stability of system (3.22).

Following our first result, we define the overall consensus value as x^*. As we are able to calculate this consensus value, we define a disagreement vector $y = x - x^*\mathbf{1}_n$ and expand the stochastic matrix P_l to P_{ex}:

$$P_{ex} = T^\top \begin{bmatrix} P_l & 0 \\ 0 & I_{n-m} \end{bmatrix} T \tag{3.27}$$

where T is the permutation matrix. We are now able to rewrite Equation (3.22) using the disagreement dynamics:

$$\begin{cases} \dot{y}(t) = -Ly(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ y(t_k) = P_{ex}y(t_k^-) & \forall t_k \in \mathcal{T} \\ y(0) = y_0 \end{cases} . \tag{3.28}$$

Due to uncertainties that affect the reset instant in practice, instead of considering a periodic reset sequence, a nearly periodic one defined by $t_{k+1} - t_k = \delta + \delta'$ is considered, where $\delta \in \mathbb{R}_+$ is the fixed period and $\delta' \in \Delta$ is a jitter belonging to the compact set $\Delta \subset \mathbb{R}_+$. Thus the set of reset times $\mathcal{T}$ belongs to the set of all admissible reset sequences associated with Δ:

$$\Phi(\Delta) \triangleq \left\{ \{t_k\}_{k\in\mathbb{N}}, t_{k+1} - t_k = \delta + \delta'_k, \delta'_k \in \Delta, \forall k \in \mathbb{N} \right\} \tag{3.29}$$

where it is always considered $t_0 = 0$.

Finally we can prove stability by means of parametric LMI. By solving this LMI we are able to find a Lyapunov function that proves the stability for the whole system. This leads to main result of the stability section:

Theorem 17 *Consider the system (3.22) with $\mathcal{T}$ in the admissible reset sequences $\Phi(\Delta)$. If there exist matrices $S(\delta')$, $S(\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ continuous with respect to δ', $S(\delta') = S^\top(\delta') > 0$, $\delta' \in \Delta$ such that the LMI*

$$\begin{aligned} &\left(I_n - \mathbf{1}_n u^\top W\right)^\top S(\delta_a)\left(I_n - \mathbf{1}_n u^\top W\right) - \\ &\left(Y(\delta_a) - \mathbf{1}_n u^\top W\right)^\top S(\delta_b)\left(Y(\delta_a) - \mathbf{1}_n u^\top W\right) > 0, \\ &Y(\delta_a) \triangleq P_{ex}e^{-L(\delta+\delta_a)} \end{aligned} \tag{3.30}$$

is satisfied on $span\{\mathbf{1}_n\}^\perp$ for all $\delta_a, \delta_b \in \Delta$, then x^* is GUES for (3.22). Moreover, the stability is characterized by the quasi-quadratic Lyapunov function $V(t) = V(x(t)) \triangleq \max_{\delta'\in\Delta}(x(t) - x^*\mathbf{1}_n)^\top S(\delta')(x(t) - x^*\mathbf{1}_n)$ satisfying $V(t_k) > V(t_{k+1})$.

3.6.3 Control design

The last main result is the control design of the leader's network. The results of this subsection are threefold: design of the leader's network to reach an a priori consensus value, decay rate and a joint design of agreement value and decay rate. Equation (3.26) gives us the consensus value of entire network as a convex combination of the initial conditions. Therefore, by setting a desired value we are able to find the left eigenvalue v of the matrix P_l. Finally, by modifying the stability result from Theorem 3.30, we have the following result:

Theorem 18 *Let us consider the system (3.22) with $\mathcal{T}$ in the admissible reset sequences $\Phi(\Delta)$ and let x^* be a priori fixed by a certain choice of v. If there exist matrices $R(\delta')$, $R(\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ continuous with respect to δ', $R(\delta') = R^\top(\delta') > 0$, $\delta' \in \Delta$ and P_l stochastic such that the LMI*

$$\begin{aligned}
&\begin{bmatrix} Z(\delta_a) & \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)^\top \\ \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big) & R(\delta_b) \end{bmatrix} > 0, \\
&Y(\delta_a) \triangleq P_{ex} e^{-L(\delta+\delta_a)} \\
&Z(\delta_a) \triangleq \Big(I_n - \mathbf{1}_n u^\top W\Big)^\top + \Big(I_n - \mathbf{1}_n u^\top W\Big) - R(\delta_a)
\end{aligned} \tag{3.31}$$

with the constraint

$$v^\top P_l = v^\top$$

is satisfied on $span\{\mathbf{1}_n\}^\perp$ for all $\delta_a, \delta_b \in \Delta$, then x^* is GUES for (3.22). Moreover, the stability is characterized by the quasi-quadratic Lyapunov function $V(t) = V(x(t)) = \max_{\delta'\in\Delta}(x(t) - x^*\mathbf{1}_n)^\top R(\delta')^{-1}(x(t) - x^*\mathbf{1}_n)$ satisfying $V(t_k) > V(t_{k+1})$.

The next step is the design using a prescribed decay rate. Following the results of [MG11], we are able to modify the LMI (3.31) and add a new condition related to the conditioning of the Lyapunov matrix $S(\delta)$. Finally we reach the result below:

Proposition 19 *Assume there exist $\alpha > 0$, $\beta > 0$, $\xi \in (0, 1]$ and the matrices $S(\delta')$, $S\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ continuous with respect to δ', $S(\delta') = S^\top(\delta') > 0$, $\delta' \in \Delta$ fulfilling the following constraints*

$$\begin{aligned}
&\alpha I_n \le S(\delta') \le \beta I_n, \quad \forall \delta' \in \Delta \\
&\xi^2 \Big(I_n - \mathbf{1}_n u^\top W\Big)^\top S(\delta_a) \Big(I_n - \mathbf{1}_n u^\top W\Big) - \\
&\quad \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)^\top S(\delta_b) \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big) > 0, \\
&Y(\delta_a) \triangleq P_{ex} e^{-L(\delta+\delta_a)}.
\end{aligned} \tag{3.32}$$

on $span\{\mathbf{1}_n\}^{\perp}$ *for all* $\delta_a, \delta_b \in \Delta$*. Then, the decay rate is defined as*

$$\lambda_d = \min_{\xi \text{ satisfies } (3.32)} \xi$$

and

$$\|x(t_k) - x^*\mathbf{1}_n\| \leq \frac{\beta}{\alpha}(\lambda_d)^k \|x(0) - x^*\mathbf{1}_n\|, \quad \forall k \in \mathbb{N}.$$

In practice, since $0 < \lambda_d \leq 1$, we find the result by defining α and β and applying the bisection algorithm to search for the smaller λ_d. In the following we have the result mixing both previous results. We are able to define a priori both the consensus agreement and the decay rate for system (3.22). This leads to this final theorem:

Theorem 20 *Let us consider the system* (3.22) *with* $\mathcal{T}$ *in the admissible reset sequences* $\Phi(\Delta)$ *and let* x^* *be a priori fixed by a certain choice of* v*. If there exist matrices* $R(\delta')$*,* $R(\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ *continuous with respect to* δ'*,* $R(\delta') = R^\top(\delta') > 0$*,* $\delta' \in \Delta$ *and* P_l *stochastic such that the LMI*

$$\begin{aligned}
&\begin{bmatrix} Z(\delta_a) & \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right)^\top \\ \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right) & R(\delta_b) \end{bmatrix} > 0, \\
&Y(\delta_a) \triangleq P_{ex} e^{-L(\delta+\delta_a)} \\
&Z(\delta_a) \triangleq \left(I_n - \mathbf{1}_n u^\top W\right)^\top + \left(I_n - \mathbf{1}_n u^\top W\right) - R(\delta_a)
\end{aligned} \tag{3.33}$$

with the constraint

$$v^\top P_l = v^\top$$

is satisfied on $span\{\mathbf{1}_n\}^{\perp}$ *for all* $\delta_a, \delta_b \in \Delta$*, then* x^* *is GUES for* (3.22)*. Moreover, the stability is characterized by the quasi-quadratic Lyapunov function* $V(t) = V(x(t)) = \max_{\delta' \in \Delta}(x(t) - x^*\mathbf{1}_n)^\top R(\delta')^{-1}(x(t) - x^*\mathbf{1}_n)$ *satisfying* $V(t_k) > V(t_{k+1})$*.*

Chapter 4

Consensus on interconnected networks: stability and design

This chapter presents a novel approach for consensus that we introduced in [BMDR14]. Consensus on a large number of agents is usually desirable, although ensuring the connectivity of all agents can be unpractical. It is more common to observe a number of clusters inside a larger network, although if said clusters are completely decoupled a general consensus cannot be reached. However, if one agent of each cluster (defined as a leader of said cluster) is allowed to exchange information with the leaders of other clusters, then a global consensus can be reached. Figure 4.1 presents an example of said network, where the dashed line represents the communication between leaders and the continuous line the communication between agents inside a cluster.

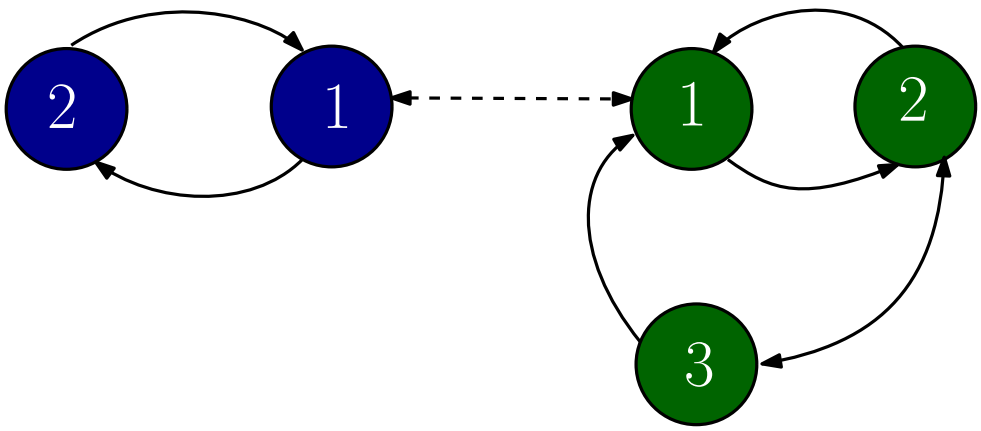

Figure 4.1: Example of a graph with interconnected networks

The initial motivation for this approach arrived from social networks. While there is a desire

to exchange information between a large group of people, it is often cumbersome to hold large meetings. Therefore, it is simpler to exchange information in smaller groups in order to reach local agreement. The leaders would then be able to bring this new information to their groups which leads to a new round of discussion that ends up with a new local agreement. Repeating these steps a consensus is eventually achieved in the overall network. This approach is not completely realistic and it corresponds to a mathematical modeling that is provided in the following. Reasonable assumptions on the network structure will be imposed in the sequel.

In this chapter,we characterize the possible consensus value, analyze their stability and design the interaction topology between leaders in order to reach an a priori fixed consensus value. A method to calculate the general consensus value of the network is also presented. The results are provided in terms of Linear Matrix Inequalities, and some numerical examples are presented in the end of the chapter, considering both a smaller and a larger example.

4.1 Problem formulation

Consider a network of n agents described by the digraph (i.e. directed graph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V}$ represents the set of agents and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents the interactions. The agent set $\mathcal{V}$ is then partitioned in m strongly connected clusters/communities $\mathcal{C}_1, \ldots, \mathcal{C}_m$ and no link between agents belonging to different communities exists. Each community possesses one particular agent called leader and denoted in the following by $l_i \in \mathcal{C}_i,\ \forall i \in \{1, \ldots, m\}$. The set of leaders will be referred to as $\mathcal{L} = \{l_1, \ldots, l_m\}$. At specific time instants $t_k,\ k \geq 1$, called reset times, the leaders interact between them following a predefined interaction map $\mathcal{E}_l \subset \mathcal{L} \times \mathcal{L}$. We also suppose that $\mathcal{G}_l = (\mathcal{L}, \mathcal{E}_l)$ is strongly connected. The remaining of the agents will be called followers and denoted by f_j. For the sake of clarity the leader is considered to be the first element of its community:

$$\mathcal{C}_i = \{l_i, f_{m_{i-1}+2}, \ldots, f_{m_i}\},\ \forall i \in \{1, \ldots, m\} \tag{4.1}$$

where $m_0 = 0,\ m_m = n$ and the cardinality of $\mathcal{C}_i$ is given by

$$|\mathcal{C}_i| \triangleq n_i = m_i - m_{i-1}, \forall i \geq 1.$$

In order to keep the presentation simple and making an abuse of notation, each agent will have a scalar state denoted also by l_i for the leader l_i and f_j for the follower f_j. We also introduce the vectors $x = (l_1, f_2, \ldots, f_{m_1}, \ldots, l_m, \ldots, f_{m_m})^\top \in \mathbb{R}^n$ and $x_l = (l_1, l_2, \ldots, l_m)^\top \in \mathbb{R}^m$ collecting all the states of the agents and all the leaders' states, respectively.

We are ready now to introduce the linear reset system describing the overall network dynamics:

$$\begin{cases} \dot{x}(t) = -Lx(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ x_l(t_k) = P_l x_l(t_k^-) & \forall t_k \in \mathcal{T} \\ x(0) = x_0 \end{cases} \tag{4.2}$$

where $\mathcal{T} = \{t_k \in \mathbb{R}_+ \mid t_k < t_{k+1},\ \forall k \in \mathbb{N}, t_k \text{ reset time}\}$, $L \in \mathbb{R}^{n\times n}$ is a generalized Laplacian matrix associated to the graph $\mathcal{G}$ and $P_l \in \mathbb{R}^{m\times m}$ is a row stochastic (Perron) matrix associated to the graph $\mathcal{G}_l = (\mathcal{L}, \mathcal{E}_l)$. Precisely, the entries of L and P_l satisfies the following relations:

$$\begin{cases} L_{(i,j)} = 0, \text{ if } (i,j) \notin \mathcal{E} \\ L_{(i,j)} < 0, \text{ if } (i,j) \in \mathcal{E},\ i \neq j \\ L_{(i,i)} = -\sum_{j\neq i} L_{i,j}, \forall i = 1, \dots, n \end{cases}, \tag{4.3}$$

$$\begin{cases} P_{l(i,j)} = 0, \text{ if } (i,j) \notin \mathcal{E}_l \\ P_{l(i,j)} > 0, \text{ if } (i,j) \in \mathcal{E}_l,\ i \neq j \\ \sum_{j=1}^{m} P_{l(i,j)} = 1, \forall i = 1, \dots, m \end{cases}. \tag{4.4}$$

The values $L_{(i,j)}$ and $P_{l(i,j)}$ represent the weight of the state of the agent j in the updating process of the state of agent i when using the continuous and discrete dynamics, respectively. In particular, L has the following block diagonal structure

$$L = \begin{pmatrix} L_1 & & \\ & \ddots & \\ & & L_m \end{pmatrix},\ L_i \in \mathbb{R}^{n_i} \tag{4.5}$$

with $L_i \mathbf{1}_{n_i} = \mathbf{0}_{n_i}$ and $P_l \mathbf{1}_m = \mathbf{1}_m$. Due to strong connectivity of $\mathcal{C}_i,\ i = 1, m$ and $\mathcal{G}_l$, 0 is simple eigenvalue of each L_i and 1 is simple eigenvalue of P_l.

4.2 Agreement behavior

In this section it is assumed that system (4.2) achieves consensus and it is possible to characterize its possible values. Firstly, it is shown that each agent tracks a local agreement function which is piecewise constant. In the second subsection it is proven that the vector of local agreements lies in a subspace defined by the system's dynamics and initial condition. Therefore, if the consensus is achieved and the corresponding consensus value is x^* then $x^*\mathbf{1}_m$ belongs to the same subspace. Moreover, this value is determined only by the initial condition of the network

and by the interconnection structure.
As we have noticed $\mathbf{1}_{n_i}$ is the right eigenvector of L_i associated with the eigenvalue 0 and $\mathbf{1}_m$ is the right eigenvector of P_l associated with the eigenvalue 1. In the sequel, consider w_i the left eigenvector of L_i associated with the eigenvalue 0 such that $w_i^\top \mathbf{1}_{n_i} = 1$. Similarly, let $v = (v_1, \ldots, v_m)^\top$ be the left eigenvector of P_l associated with the eigenvalue 1 such that $v^\top \mathbf{1}_m = 1$. Due to the structure (4.1) of the communities, we emphasize that each vector w_i can be decomposed in its first component $w_{i,l}$ and the rest of its components grouped in the vector $w_{i,f}$.

4.2.1 Local agreements

Let us first recall a well known result concerning the consensus in networks of agents with continuous time dynamics (see [OSM04b] for instance). We start this section by recalling a known result concerning the consensus in networks of agents with continuous time dynamics, stated previously in Lemma 11. Lemma 11 states that, for a strongly connected digraph with Laplacian matrix L with left eigenvector u, the agents reach a consensus and the consensus value is given by $x^* = u^\top x(0)$. The vector u defines an invariant subspace for the collective dynamics: $u^\top x(t) = u^\top x(0), \forall t \geq 0$.

Remark 1 *When dynamics (4.2) is considered, Lemma 11 implies that between two consecutive reset instants t_k and t_{k+1}, the agents belonging to the same community try to approach a local agreement defined by $x_i^*(k) = w_i^\top x_{\mathcal{C}_i}(t_k)$ where $x_{\mathcal{C}_i}(\cdot)$ is the vector collecting the states of the agents belonging to the cluster $\mathcal{C}_i$. Nevertheless, at the reset times the value of the local agreement can change. Thus,*

$$\begin{aligned} &w_i^\top x_{\mathcal{C}_i}(t) = w_i^\top x_{\mathcal{C}_i}(t_k),\ \forall t \in (t_k, t_{k+1}) \textit{ and possibly} \\ &w_i^\top x_{\mathcal{C}_i}(t) \neq w_i^\top x_{\mathcal{C}_i}(t_k),\ \textit{for } t \notin (t_k, t_{k+1}) \end{aligned}$$

Therefore, the agents whose collective dynamics is described by the hybrid system (4.2), may reach a consensus only if the local agreements converge one to each other.

4.2.2 Consensus value

Before presenting the next result, the following vectors must be introduced:

$$\begin{aligned} x^*(t) &= (x_1^*(t), x_2^*(t), ..., x_m^*(t))^\top \in \mathbb{R}^m \\ u &= (v_1/w_{1,l}, v_2/w_{2,l}, ..., v_m/w_{m,l})^\top \in \mathbb{R}^m \end{aligned} \tag{4.6}$$

where $x_i^*(\cdot)$ represents the local agreement of the cluster $\mathcal{C}_i$ and $v \in \mathbb{R}^m$ and $w_i \in \mathbb{R}^{n_i}$ are defined at the beginning of the section as left eigenvectors associated with the matrices describing the reset dynamics of the leaders and the continuous dynamics of each cluster, respectively.
It is noteworthy that $x^*(t)$ is time-varying but piecewise constant: $x^*(t) = x^*(k)\ \forall t \in (t_k, t_{k+1})$.

Proposition 21 *Consider the system* (4.2) *with* L *and* P_l *defined by* (4.3) *and* (4.4)*, respectively. Then,*

$$u^\top x^*(t) = u^\top x^*(0),\ \forall t \in \mathbb{R}_+. \tag{4.7}$$

Proof: Consider the matrix of the left eigenvectors of the communities:

$$W = \begin{bmatrix} w_1^\top & 0 & \cdots & 0 \\ 0 & w_2^\top & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_m^\top \end{bmatrix} \in \mathbb{R}^{m\times n}. \tag{4.8}$$

Then, the following relation holds:

$$x^*(t) = Wx(t) \qquad \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \tag{4.9}$$

Since $w_i = (w_{il}, w_{if})$, a permutation matrix T such that $WT^\top = U = (U_1, U_2)$ can be defined. The permutation matrix T can also rearrange the order of the agents, meaning that $(x_l(t), x_f(t)) = Tx(t)$. The matrix U_1 is a diagonal matrix corresponding to the leaders' components $w_{i,l}$, while U_2 is a block diagonal matrix corresponding to the followers' components $w_{i,f}$. In other terms

$$U_1 = \begin{bmatrix} w_{1,l} & 0 & \cdots & 0 \\ 0 & w_{2,l} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{m,l} \end{bmatrix} \in \mathbb{R}^{m\times m} \tag{4.10}$$

$$U_2 = \begin{bmatrix} w_{1,f} & 0 & \cdots & 0 \\ 0 & w_{2,f} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{m,f} \end{bmatrix} \in \mathbb{R}^{m\times (n-m)}. \tag{4.11}$$

thus, equation (4.9) can be rewritten as:

$$x^*(t) = WT^\top Tx(t) = U \cdot (x_l(t), x_f(t)). \tag{4.12}$$

Note that at the reset time t_k the vector x_f one has. $x_f(t_k) = x_f(t_k^-)$. This yields

$$\begin{aligned} x^*(t_k) - x^*(t_k^-) &= U \cdot (x_l(t_k) - x_l(t_k^-), x_f(t_k) - x_f(t_k^-)) \\ &= U \cdot (x_l(t_k) - x_l(t_k^-), 0) = U_1 \cdot (x_l(t_k) - x_l(t_k^-)) + U_2 \cdot 0 \\ &= U_1(x_l(t_k) - x_l(t_k^-)). \end{aligned}$$

Thus,

$$x^*(t_k) = x^*(t_k^-) + U_1 \cdot (P_l - I_m)x_l(t_k^-)). \tag{4.13}$$

Multiplying equation (4.13) by $u^\top$ and using $u^\top U_1 = v^\top$ one obtains

$$\begin{aligned} u^\top x^*(t_k) &= u^\top x^*(t_k^-) + u^\top U_1(P_l - I_m)x_l \\ &= u^\top x^*(t_k^-) + v^\top(P_l - I_m) \\ &= u^\top x^*(t_k^-) + v^\top - v^\top = u^\top x^*(t_k^-) \end{aligned} \tag{4.14}$$

According to Remark 1, $x^*(t)$ remains constant for all $t \in (t_k, t_{k+1})$ leading to

$$u^\top x^*(t) = u^\top x^*(0) \qquad \forall t \in \mathbb{R}_+. \tag{4.15}$$

Corollary 1 *Consider the system (4.2) with L and P_l defined by (4.3) and (4.4), respectively. Assuming the agents of this system reach a consensus, the consensus value is*

$$x^* = \frac{u^\top W x(0)}{\sum_{i=1}^m u_i}. \tag{4.16}$$

Proof: Let x^* be the consensus value reached by the system (4.2). It means that $x(t) \to x^*\mathbf{1}_n$. Thus, when t goes to ∞ in (4.15) one obtains

$$u^\top x^*\mathbf{1}_n = u^\top x^*(0) = u^\top W x(0)$$

leading to (4.16).

In order to simplify the presentation and without loss of generality, in what follows, we consider that $\sum_{i=1}^m u_i = 1$.

Remark 2 *It is important to note that the consensus value depends only on the system matrices L, P_l and does not depend on the reset sequence $\mathcal{T}$.*

A trivial result which may be seen as a consequence of Corollary 1 is the following.

Corollary 2 *If the matrices L, P_l are symmetric (i.e. i^{th} agent takes into account the state of jth agent as far as j^{th} takes into account the ith one and they give the same importance one to another) the consensus value is the average of the initial states.*

Proof: In this case $w_i = \frac{1}{n_i}\mathbf{1}_{n_i}$ and $v = \frac{1}{m}\mathbf{1}_m$ which leads to $u = (\frac{n_1}{m}, \frac{n_2}{m}, \ldots, \frac{n_m}{m})$. The result follows from (4.16).

4.3 Stability analysis

The stability analysis of the equilibrium point x^* will be given by means of some LMI conditions. Since the consensus value is computed in the previous section it is possible to define an disagreement vector $y = x - x^*\mathbf{1}_n$. The stochastic matrix P_{ex} is also introduced as follows:

$$P_{ex} = T^\top \begin{bmatrix} P_l & 0 \\ 0 & I_{n-m} \end{bmatrix} T \tag{4.17}$$

where T is the permutation matrix used in the proof of Proposition 21. It is noteworthy that $L\mathbf{1}_n = \mathbf{0}_n$ and $P_{ex}\mathbf{1}_n = \mathbf{1}_n$. Thus, the disagreement dynamics is exactly the same as the system one:

$$\begin{cases} \dot{y}(t) = -Ly(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ y(t_k) = P_{ex}y(t_k^-) & \forall t_k \in \mathcal{T} \\ y(0) = y_0 \end{cases} . \tag{4.18}$$

Due to uncertainties that affect the reset instant in practice, instead of considering a periodic reset sequence, a nearly periodic one defined by $t_{k+1} - t_k = \delta + \delta'$ is considered, where $\delta \in \mathbb{R}_+$ is the fixed period and $\delta' \in \Delta$ is a jitter belonging to the compact set $\Delta \subset \mathbb{R}_+$. Thus the set of reset times $\mathcal{T}$ belongs to the set of all admissible reset sequences associated with Δ:

$$\Phi(\Delta) \triangleq \Big\{ \{t_k\}_{k\in\mathbb{N}}, t_{k+1} - t_k = \delta + \delta'_k, \delta'_k \in \Delta, \forall k \in \mathbb{N} \Big\} \tag{4.19}$$

where it is always considered $t_0 = 0$. Recall that for any $\mathcal{T} \in \Phi(\Delta)$ and any initial condition x_0 the system (4.2) has a unique solution denoted by $\varphi_{\mathcal{T}}(t, x_0)$.

4.3.1 Parametric LMI condition

In the sequel, a quasi-quadratic Lyapunov function satisfying Theorem 6 is defined, by means of some LMI. Therefore, the following result gives sufficient conditions for the stability of the equilibrium point $y^* = \mathbf{0}_n$ for the system (4.18) or equivalently of $x^*\mathbf{1}_n$ for the system (4.2). Even if other sufficient condition for GUES can be given, the next result is presented since it will be useful in the following section.

Theorem 22 *Consider the system (4.2) with $\mathcal{T}$ in the admissible reset sequences $\Phi(\Delta)$. If there exist matrices $S(\delta')$, $S(\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ continuous with respect to δ', $S(\delta') = S^\top(\delta') > 0$, $\delta' \in \Delta$ such that the LMI*

$$\begin{aligned} &\Big(I_n - \mathbf{1}_n u^\top W\Big)^\top S(\delta_a)\Big(I_n - \mathbf{1}_n u^\top W\Big) - \\ &\Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)^\top S(\delta_b)\Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big) > 0, \\ &Y(\delta_a) \triangleq P_{ex}e^{-L(\delta+\delta_a)} \end{aligned} \tag{4.20}$$

is satisfied on $span\{\mathbf{1}_n\}^{\perp}$ *for all* $\delta_a, \delta_b \in \Delta$*, then* x^* *is GUES for* (4.2)*. Moreover, the stability is characterized by the quasi-quadratic Lyapunov function* $V(t) = V(x(t)) \triangleq \max_{\delta' \in \Delta}(x(t) - x^*\mathbf{1}_n)^\top S(\delta')(x(t) - x^*\mathbf{1}_n)$ *satisfying* $V(t_k) > V(t_{k+1})$*.*

Remark 3 *We recall that a matrix* A *satisfies* $A > 0$ *if and only if* $x^\top Ax > 0$ *for all* $x \neq 0$*. Therefore, the expression "solving an LMI on the* $span\{\mathbf{1}_n\}^{\perp}$*" means that, for any* $x \perp \mathbf{1}_n$ *the inequality* $x^\top Ax > 0$ *holds, where* A *replaces the expression at the left hand side of the LMIs. The reason why we enforce this is because* $x^\top Ax = 0$ *on the* $span\{\mathbf{1}_n\}$*. Therefore, the LMIs will never have a solution.*

Using the disagreement vector $y(t) = x(t) - x^*\mathbf{1}_n$ and supposing that there exist matrices $S(\delta')$ satisfying (4.20) for all $\delta_a, \delta_b \in \Delta$ the following Lyapunov matrix is defined

$$S_{[y]} = S(\delta^*(y)) \text{ with } \delta^*(y) = \arg\max_{\delta' \in \Delta} y^\top S(\delta')y \tag{4.21}$$

Following [HDTP13] the Lyapunov function

$$V(y) = y^\top S_{[y]}y = \max_{\delta' \in \Delta} y^\top S(\delta')y,$$

is convex and homogeneous of the second order.
The convexity of V follows from the fact that for a set of convex functions the maximum preserves the convexity property. The $S(\cdot)$ solution of (4.20) ensures that $V(\cdot)$ defined above satisfies Theorem 6.
Note that any $x(t_k) \in \mathbb{R}^n$ can be decomposed as $x(t_k) = \bar{x}(t_k) + \tilde{x}(t_k)$ with $\bar{x}(t_k) \in span\{\mathbf{1}_n\}^{\perp}$ and $\tilde{x}(t_k) \in span\{\mathbf{1}_n\}$. We recall that $W\mathbf{1}_n = \mathbf{1}_n$ and $u^\top\mathbf{1}_n = 1$. Since $\tilde{x}(t_k)$ in on the $span\{\mathbf{1}_n\}$, $W\tilde{x}(t_k) = \tilde{x}(t_k)$. Moreover, we recall that for the stochastic matrix $Y(\delta_a)$, $Y(\delta_a)\tilde{x}(t_k) = \tilde{x}(t_k)$. Therefore, on the $span\{\mathbf{1}_n\}$

$$\begin{aligned}\Big(I_n - \mathbf{1}_n u^\top W\Big)\tilde{x}(t_k) &= I_n\tilde{x}(t_k) - \mathbf{1}_n u^\top W\tilde{x}(t_k),\\ \tilde{x}(t_k) - \mathbf{1}_n u^\top \tilde{x}(t_k) &= 0\end{aligned}$$

and

$$\begin{aligned}\Big((Y(\delta_a) - \mathbf{1}_n u^\top W\Big)\tilde{x}(t_k) &= Y(\delta_a)\tilde{x}(t_k) - \mathbf{1}_n u^\top W\tilde{x}(t_k),\\ \tilde{x}(t_k) - \mathbf{1}_n u^\top \tilde{x}(t_k) &= 0\end{aligned}$$

hence

$$\begin{aligned}\Big(I_n - \mathbf{1}_n u^\top W\Big)x(t_k) &= \Big(I_n - \mathbf{1}_n u^\top W\Big)\bar{x}(t_k),\\ \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)x(t_k) &= \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)\bar{x}(t_k)\end{aligned}$$

Thus, the LMI (4.20) yields

$$\begin{aligned}
&x(t_k)^\top \Big(I_n - \mathbf{1}_n u^\top W\Big)^\top S(\delta_a)\Big(I_n - \mathbf{1}_n u^\top W\Big)x(t_k) > \\
&x(t_k)^\top \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)^\top S(\delta_b)\Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)x(t_k), \\
&\quad \forall \delta_a, \delta_b \in \Delta
\end{aligned}$$

Consequently, using $u^\top W x(t_k) = x^*$ one gets

$$\begin{aligned}
&\Big(x(t_k) - \mathbf{1}_n x^*\Big)^\top S(\delta_a)\Big(x(t_k) - \mathbf{1}_n x^*\Big) > \\
&\Big(Y(\delta_a)x(t_k) - \mathbf{1}_n x^*\Big)^\top S(\delta_b)\Big(Y(\delta_a)x(t_k) - \mathbf{1}_n x^*\Big), \\
&\quad \forall \delta_a, \delta_b \in \Delta,\ x(t_k) \in \mathbb{R}^n
\end{aligned} \tag{4.22}$$

For any $\{t_k\}_{k\in\mathbb{N}} \in \Phi(\Delta)$ we have $x(t_{k+1}) = Y(\delta'_k)x(t_k)$ with some $\delta'_k \in \Delta$. Thus, for $\delta_a = \delta'_k$, (4.22) rewrites as:

$$\begin{aligned}
&\Big(x(t_k) - x^*\mathbf{1}_n\Big)^\top S(\delta'_k)\Big(x(t_k) - x^*\mathbf{1}_n\Big) > \\
&\Big(x(t_{k+1}) - x^*\mathbf{1}_n\Big)^\top S(\delta_b)\Big(x(t_{k+1}) - x^*\mathbf{1}_n\Big) \\
&\quad \forall \delta'_k,\ \delta_b \in \Delta,\ x(t_k) \in \mathbb{R}^n
\end{aligned}$$

or equivalently

$$y(t_k)^\top S(\delta'_k)y(t_k) > y(t_{k+1})^\top S(\delta_b)y(t_{k+1}) \quad \forall \delta'_k, \delta_b \in \Delta$$

Taking $\delta_b = \delta^*(y(t_{k+1}))$, defined by (4.21) one obtains

$$V(y(t_k)) > y(t_k)^\top S(\delta'_k)y(t_k) > V(y(t_{k+1}))$$

for all $y(t_k)$, which ends the proof.

4.3.2 Decay rate analysis

Once the global uniform exponential stability of x^* ensured by Theorem 22 it is possible to compute the convergence speed of the state of system (4.2). In other words, λ in (2.18) must be evaluated. Let us recall that the decay rate of the linear difference inclusion (LDI)

$$x(t_{k+1}) \in \mathcal{F}(x(t_k)), \quad k \in \mathbb{N} \tag{4.23}$$

where

$$\mathcal{F}(x) = \Big\{P_{ex}e^{-L(\delta+\delta')},\ \delta' \in \Delta\Big\}.$$

From the proof of Theorem 1 of [HDTP13] we have that $\lambda = \dfrac{\ln \lambda_d}{\delta + \delta_{max}}$ where $\delta_{max} = \max_{\delta' \in \Delta} \delta'$ and λ_d defined as the decay rate of the linear difference inclusion in (4.23). Precisely, for the LDI (4.23) there exist $M > 0$ and $\xi \in [0, 1]$ such that

$$\|x(t_k) - x^* \mathbf{1}_n\| \leq M \xi^k \|x(0) - x^* \mathbf{1}_n\|, \quad \forall k \in \mathbb{N} \tag{4.24}$$

and λ_d is defined as the smaller ξ satisfying (4.24).
Thus, in order to quantify the convergence speed of system (4.2), it is only necessary to evaluate λ_d. Considering again $y = x - x^* \mathbf{1}_n$ and $V(y)$ defined by Theorem 22 as a norm. That implies the existence of $\alpha, \beta > 0$ such that

$$\alpha \|y\|^2 \leq V(y) \leq \beta \|y\|^2.$$

Consequently, one obtains that the decay rate λ_d coincides with the decay rate of V. Thus, the following result can be derived directly from Theorem 22.

Proposition 23 *Assume there exist $\alpha > 0$, $\beta > 0$, $\xi \in (0, 1]$ and the matrices $S(\delta')$, $S\cdot) : \Delta \mapsto \mathbb{R}^{n \times n}$ continuous with respect to δ', $S(\delta') = S^\top(\delta') > 0$, $\delta' \in \Delta$ fulfilling the following constraints*

$$\begin{aligned}
&\alpha I_n \leq S(\delta') \leq \beta I_n, \quad \forall \delta' \in \Delta \\
&\xi^2 \Big(I_n - \mathbf{1}_n u^\top W\Big)^\top S(\delta_a) \Big(I_n - \mathbf{1}_n u^\top W\Big) - \\
&\quad \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big)^\top S(\delta_b) \Big(Y(\delta_a) - \mathbf{1}_n u^\top W\Big) > 0, \\
&Y(\delta_a) \triangleq P_{ex} e^{-L(\delta + \delta_a)}.
\end{aligned} \tag{4.25}$$

on $span\{\mathbf{1}_n\}^\perp$ for all $\delta_a, \delta_b \in \Delta$. Then, the decay rate is defined as

$$\lambda_d = \min_{\xi \text{ satisfies } (4.25)} \xi$$

and

$$\|x(t_k) - x^* \mathbf{1}_n\| \leq \frac{\beta}{\alpha} (\lambda_d)^k \|x(0) - x^* \mathbf{1}_n\|, \quad \forall k \in \mathbb{N}.$$

Remark 4 *It is noteworthy that $0 < \lambda_d \leq 1$ and for a priori fixed values of α, β we can use the bisection algorithm to approach as close as we want the value of λ_d.*

Remark 5 *To complete the decay rate analysis, we can consider that P_{ex}, L and λ_d are fixed and perform a line search to find the nominal reset period δ that ensures the convergence speed constraint. In other words, we check if (4.25) has solutions for $\xi = \lambda_d$ and δ heuristically sweeping the positive real axis. Moreover, we can progressively decrease λ_d and re-iterate the line search in order to find the smaller reachable decay rate.*

4.4 Convergence toward a prescribed value

In what follows we assume that the value x^* is a priori fixed and at least a vector u satisfying (4.16) exists. Under this assumption we are wondering if there exists a matrix P_l that allows system (4.2) to reach the consensus value x^*. It is worth noting that the network topology is considered fixed and known for each cluster. Under these assumptions, a consensus value is imposed by a certain choice of v such that $v^\top \mathbf{1}_m = 1$ and v left eigenvector of P_l associated with the eigenvalue 1. In other words we arrive to a joint design of P_l and the Lyapunov function V guaranteeing the trajectory of (4.2) ends up on x^*.

Theorem 24 *Let us consider the system (4.2) with $\mathcal{T}$ in the admissible reset sequences $\Phi(\Delta)$ and let x^* be a priori fixed by a certain choice of v. If there exist matrices $R(\delta')$, $R(\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ continuous with respect to δ', $R(\delta') = R^\top(\delta') > 0$, $\delta' \in \Delta$ and P_l stochastic such that the LMI*

$$\begin{aligned}
&\begin{bmatrix} Z(\delta_a) & \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right)^\top \\ \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right) & R(\delta_b) \end{bmatrix} > 0, \\
&Y(\delta_a) \triangleq P_{ex} e^{-L(\delta+\delta_a)} \\
&Z(\delta_a) \triangleq \left(I_n - \mathbf{1}_n u^\top W\right)^\top + \left(I_n - \mathbf{1}_n u^\top W\right) - R(\delta_a)
\end{aligned} \tag{4.26}$$

with the constraint

$$v^\top P_l = v^\top$$

is satisfied on $span\{\mathbf{1}_n\}^\perp$ for all $\delta_a, \delta_b \in \Delta$, then x^* is GUES for (4.2). Moreover, the stability is characterized by the quasi-quadratic Lyapunov function $V(t) = V(x(t)) = \max_{\delta'\in\Delta}(x(t) - x^*\mathbf{1}_n)^\top R(\delta')^{-1}(x(t) - x^*\mathbf{1}_n)$ satisfying $V(t_k) > V(t_{k+1})$.

Proof: First notice that

$$\begin{aligned}
&\left((I_n - \mathbf{1}_n u^\top W)^\top S(\delta_a) - I_n\right) S(\delta_a)^{-1} \times \\
&\left(S(\delta_a)(I_n - \mathbf{1}_n u^\top W) - I_n\right) \geq 0
\end{aligned}$$

leads to

$$\begin{aligned}
&(I_n - \mathbf{1}_n u^\top W)^\top S(\delta_a)(I_n - \mathbf{1}_n u^\top W) \geq \\
&(I_n - \mathbf{1}_n u^\top W)^\top + (I_n - \mathbf{1}_n u^\top W) - S(\delta_a)^{-1}
\end{aligned}$$

Thus, once the solution to the LMI problem (4.26) is obtained we can define $S(\delta_a) = R(\delta_a)^{-1}$ and $S(\delta_b) = R(\delta_b)^{-1}$. Then:

$$\begin{bmatrix} Z(\delta_a) & \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right)^\top \\ \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right) & S(\delta_b)^{-1} \end{bmatrix} > 0$$

where

$$Z(\delta_a) = (I_n - \mathbf{1}_n u^\top W)^\top + (I_n - \mathbf{1}_n u^\top W) - S(\delta_a)^{-1}$$

and hence

$$\begin{bmatrix} \bar{Z}(\delta_a) & \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right)^\top \\ \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right) & S(\delta_b)^{-1} \end{bmatrix} > 0$$

where

$$\bar{Z}(\delta_a) = (I_n - \mathbf{1}_n u^\top W)^\top S(\delta_a)(I_n - \mathbf{1}_n u^\top W).$$

By Schur complement, the last LMI is nothing than (4.20) in Theorem 22. Moreover, the constraints $v^\top P_l = v^\top$, $P_l \mathbf{1}_m = \mathbf{1}_m$ and the coefficients of P_l positive ensure the matrix P_l is stochastic and the consensus value is exactly x^*.

4.4.1 Convergence toward a prescribed value with a prescribed decay rate

Combining the results of Proposition 23 and Theorem 24 we can design the matrix P_l that allows to reach an a priori given consensus value x^* with a decay rate inferior to an a priori fixed value. Precisely, the following result holds.

Theorem 25 *Let us consider the system (4.2) with $\mathcal{T}$ in the admissible reset sequences $\Phi(\Delta)$ and let x^* be a priori fixed by a certain choice of v. Let us also consider $\bar{\lambda} \in (0,1)$ a priori fixed. If there exist matrices $R(\delta')$, $R(\cdot) : \Delta \mapsto \mathbb{R}^{n\times n}$ continuous with respect to δ', $R(\delta') = R^\top(\delta') > 0$, $\delta' \in \Delta$ and P_l row stochastic such that the LMI*

$$\begin{aligned} &\begin{bmatrix} Z(\delta_a) & \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right)^\top \\ \left(Y(\delta_a) - \mathbf{1}_n u^\top W\right) & \bar{\lambda}^2 R(\delta_b) \end{bmatrix} > 0, \\ &Y(\delta_a) \triangleq P_{ex} e^{-L(\delta+\delta_a)} \\ &Z(\delta_a) \triangleq \left(I_n - \mathbf{1}_n u^\top W\right)^\top + \left(I_n - \mathbf{1}_n u^\top W\right) - R(\delta_a) \end{aligned} \tag{4.27}$$

with the constraint

$$v^\top P_l = v^\top$$

is satisfied on $span\{\mathbf{1}_n\}^\perp$ for all $\delta_a, \delta_b \in \Delta$, then x^* is GUES for (4.2) and (4.24) is satisfied for $\xi = \bar{\lambda}$ and $M = \beta/\alpha$ where β and α are the minimum and the maximum eigenvalue of $R(\delta')$, $\delta' \in \Delta$, respectively.

Proof: If (4.27) holds, following the proof of Theorem 24 one obtains that x^* is GUES for (4.2) and (4.25) is satisfied for $\xi = \bar{\lambda}$ and $S(\delta') = R(\delta')^{-1}$, $\delta' \in \Delta$. Thus from Proposition 23 one concludes that $\lambda_d \leq \bar{\lambda}$.

Remark 6 *It is noteworthy that LMI (4.27) implies LMI (4.25) but they are not equivalent. Therefore, the decay rate λ_d is smaller than $\bar{\lambda}$.*

4.5 Numerical implementation

In this section we consider the problem of approximation of the parametric LMI (4.20) by a finite number of conditions using polytopic embeddings.

4.5.1 Periodic resets

If $\Delta = \{0\}$ i.e. the resets take place periodically with the period δ, Theorem 22 is rephrased as follows.

Corollary 3 *If there exists a positive definite matrix S such that the LMI*

$$\begin{aligned} &\Big(I_n - \mathbf{1}_n u^\top W\Big)^\top S\Big(I_n - \mathbf{1}_n u^\top W\Big) - \\ &\Big(Y(\delta) - \mathbf{1}_n u^\top W\Big)^\top S\Big(Y(\delta) - \mathbf{1}_n u^\top W\Big) > 0, \\ &Y(\delta) \triangleq P_{ex} e^{-L(\delta)} \end{aligned} \tag{4.28}$$

is satisfied on $span\{\mathbf{1}_n\}^\perp$, then x^* is globally uniformly exponentially stable for (4.2). Moreover, the stability is characterized by the quadratic Lyapunov function $V(t) = (x(t) - x^*\mathbf{1}_n)^\top S(x(t) - x^*\mathbf{1}_n)$ satisfying $V(t_k) > V(t_{k+1})$.

In this case the parametric LMI (4.20) is simply replaced by the LMI (4.28) and no numerical difficulties arise.

4.5.2 Quasi-periodic resets.

In this case, to replace the parametric LMI (4.20) by a finite number of LMIs we have to deal with the parametric uncertainty δ'_k in the definition (4.19) of the admissible reset sequence. As in [HDTP13] the matrix exponential $e^{-L\delta_a}$ is approximated by its $h-$ order Taylor expansion

$\sum_{i=0}^{h} \frac{(-L)^i}{i!}\delta_a^i$. Thus the set $\{X \in \mathbb{R}^{n\times n} \mid X = e^{-L\delta_a},\ \delta_a \in \Delta\}$ can be embedded into the polytopic set defined by the vertices $Z_1, \ldots, Z_{h+1}$ where

$$
\begin{aligned}
Z_1 &= I_n \\
Z_i &= \sum_{l=0}^{i-1} \frac{(-L)^l}{l!}\delta_{max}^l,\ \forall i \in \{2, \ldots, h+1\}
\end{aligned}
$$

with $\delta_{max} = \max_{\delta' \in \Delta} \delta'$, $(-L)^0 = I_n$ and $0! = 1$. Then, Theorem 22 can be replaced by the following result.

Theorem 26 *Consider the system (4.2) with $\mathcal{T}$ in the admissible reset sequences $\Phi(\Delta)$. If there exist symmetric positive definite matrices $S_i,\ 1 \le i \le h+1$ such that the LMI*

$$
\begin{aligned}
&\left(I_n - \mathbf{1}_n u^\top W\right)^\top S_i \left(I_n - \mathbf{1}_n u^\top W\right) - \\
&\left(Y(\delta) Z_i - \mathbf{1}_n u^\top W\right)^\top S_j \left(Y(\delta) Z_i - \mathbf{1}_n u^\top W\right) > 0, \\
&Y(\delta) \triangleq P_{ex} e^{-L(\delta)}
\end{aligned}
\tag{4.29}
$$

is satisfied on $span\{\mathbf{1}_n\}^\perp$ for all $i, j \in \{1, \ldots, h+1\}$, then x^* is globally uniformly exponentially stable for (4.2).

Proof: Assume that the set of LMIs (4.29) is satisfied for a set of matrices $S_i,\ 1 \le i \le h+1$. Thus,

$$
\begin{aligned}
&\left(I_n - \mathbf{1}_n u^\top W\right)^\top \left(\sum_{i=1}^{h+1} \mu_i S_i\right) \left(I_n - \mathbf{1}_n u^\top W\right) - \\
&\left(Y(\delta) \sum_{i=1}^{h+1} \mu_i Z_i - \mathbf{1}_n u^\top W\right)^\top \left(\sum_{i=1}^{h+1} \mu_j S_j\right) \times \\
&\left(Y(\delta) \sum_{i=1}^{h+1} \mu_i Z_i - \mathbf{1}_n u^\top W\right) > 0,
\end{aligned}
$$

is satisfied for all $\mu_i, \mu_j \in [0, 1],\ i, j \in \{1, \ldots, h+1\}$ such that $\sum_{i=1}^{h+1} \mu_i = \sum_{j=1}^{h+1} \mu_j = 1$. It is noteworthy that the polytopic embedding provided above implies that for all $\delta_a \in [0, \delta_{max}]$ there exists the set of scalars $\mu_i \in [0, 1]$ such that $e^{-L\delta_a} = \sum_{i=1}^{h+1} \mu_i Z_i$ and $\sum_{i=1}^{h+1} \mu_i = 1$. In other words, Theorem 22 holds with $S(\delta') = \sum_{i=1}^{h+1} \mu_i(\delta') S_i$.

4.6 Illustrative examples

4.6.1 Small network analysis

An academic example consisting in a network of 5 agents partitioned in 2 clusters ($n_1 = 3, n_2 = 2$) is used in the sequel to illustrate the theoretical results. Consider the dynamics (4.2) with

$$L = \begin{bmatrix} 4 & -2 & -2 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \; P_l = \begin{bmatrix} 0.45 & 0.55 \\ 0.25 & 0.75 \end{bmatrix} \tag{4.30}$$

and the reset sequence given by $\delta = 0.5$ and δ'_k randomly chosen in $\Delta = [0, 0.2]$. The initial condition of the system is $x(0) = (8, 7, 9, 2, 3)$ and the corresponding consensus value computed by (4.16) is $x^* = 4.6757$. The convergence of the 5 agents towards x^* is illustrated in Figure 1 emphasizing that the leaders trajectories are non-smooth while the followers trajectories are.

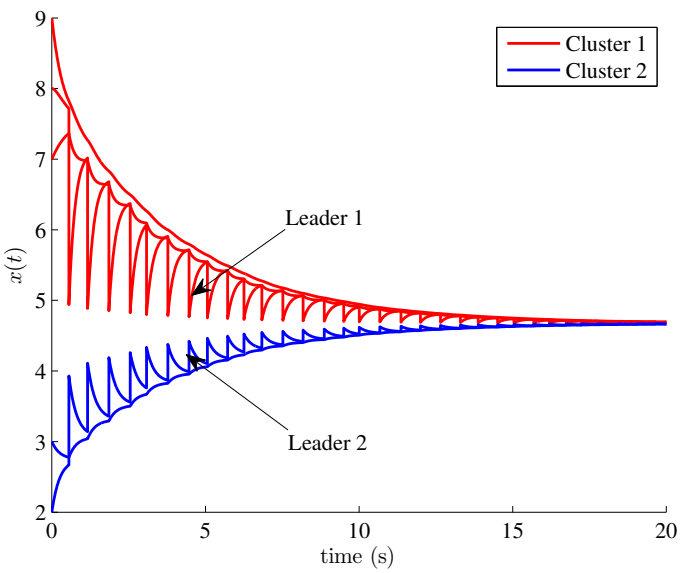

Figure 4.2: The state-trajectories of the agents converging to the calculated consensus value.

The table below collects the first 10 time intervals between consecutive reset instants. As expected, these time intervals have random lengths within $[0.5, 0.7]$ and no monotony occurs.

The jumps and decreasing of the Lyapunov function defined by Theorem 22 are pointed out in Figure 2.

$t_1 - t_0$	0.6189 s	$t_6 - t_5$	0.5979 s
$t_2 - t_1$	0.6131 s	$t_7 - t_6$	0.5372 s
$t_3 - t_2$	0.6433 s	$t_8 - t_7$	0.6401 s
$t_4 - t_3$	0.6023 s	$t_9 - t_8$	0.6965 s
$t_5 - t_4$	0.6553 s	$t_{10} - t_9$	0.6613 s

Table 4.1: The length of the first 10 time intervals between consecutive reset instants.

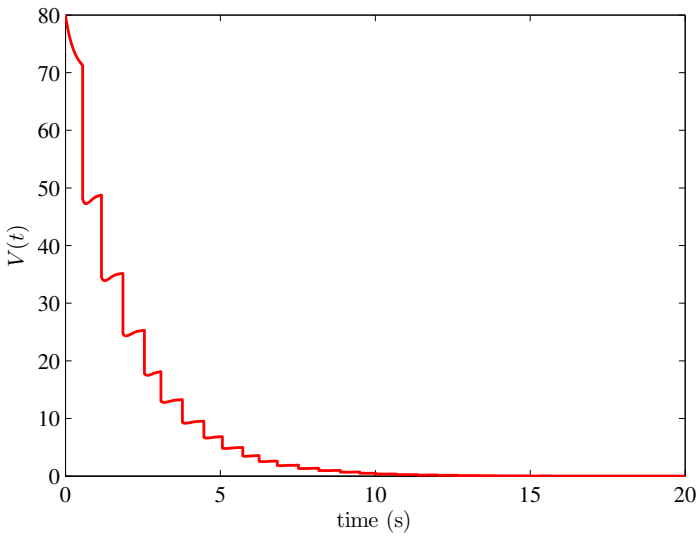

Figure 4.3: The behavior of the Lyapunov function given by Theorem 22.

In order to illustrate the independence of the consensus value on the reset sequence (see Remark 2), we also considered $\delta = 5$. In this case, as can be seen in Figure 4.4, the local agreements are reached before each reset and we better emphasize their piece-wise constant behavior (see Remark 1). As expected the consensus value remains $x^* = 4.6757$.

Coming back to $\delta = 0.5$, to find the decay rate λ_d we use the bisection algorithm as stated in Remark 4. In Figure 4.5 we demonstrate that the initial conditions of Lyapunov function may vary due to the conditioning of the matrix $S(\delta)$, but the decay rate remains the same. The value of λ_d obtained was $\lambda_d = 0.855$ and the number of iterations of the bisection algorithm is $k = 30$.

Analyzing equation (4.15) we obtain that the consensus value is always a convex combination of the initial agreement values of the clusters. In the present case, one has two clusters and the two initial agreements are 2.75 and 7.5. Thus, we can try to reach only consensus values belonging to $[2.75, 7.5]$. In Figure 4.6 the consensus value is fixed at $x^* = 6.5$ and the associated P_l matrix is

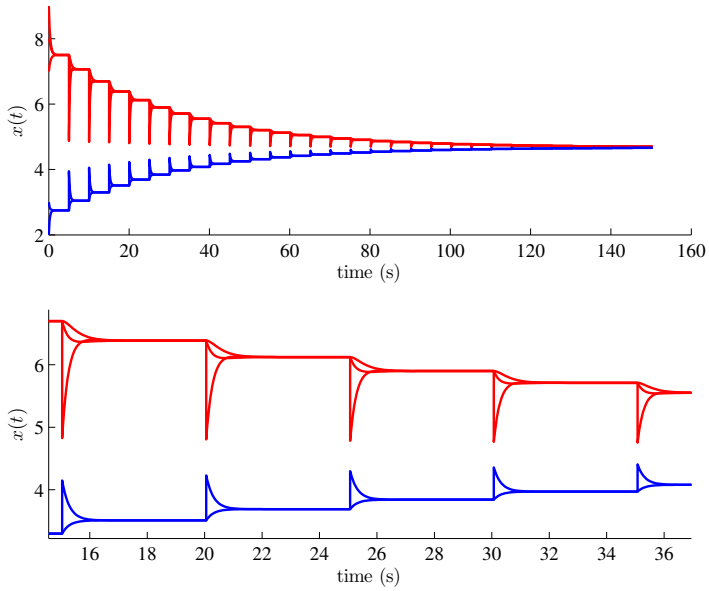

Figure 4.4: Top: State trajectories of the agents converging to the piece-wise constant local agreements. The local agreements approach one of each other at the reset times. Bottom: Zoom emphasizing the state behavior. Jumps are present only in the leaders trajectories.

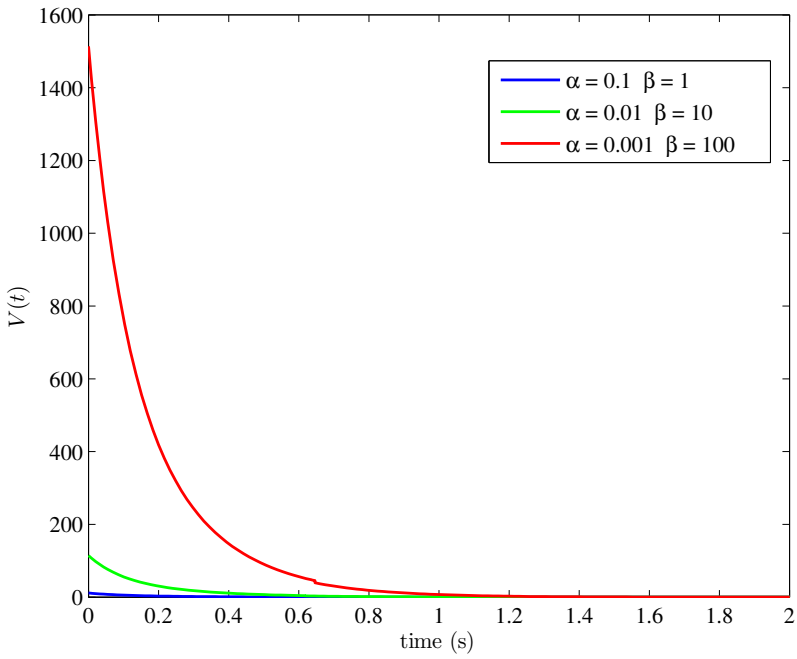

Figure 4.5: Lyapunov function for different conditionings

$$P_l = \begin{bmatrix} 0.6870 & 0.3130 \\ 0.7825 & 0.2175 \end{bmatrix}.$$

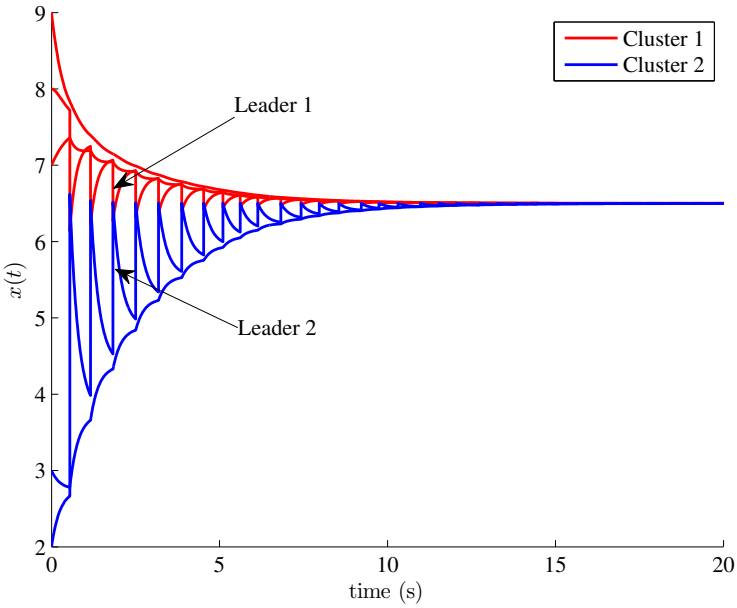

Figure 4.6: The states of a system ($x^* = 6.5$).

The decay rate associated with this P_l is $\lambda_d = 0.782$ and can be improved by using Theorem 25. Imposing $\bar{\lambda} = 0.82$ in (4.27) one gets

$$P_l = \begin{bmatrix} 0.6425 & 0.3575 \\ 0.8937 & 0.1063 \end{bmatrix}. \tag{4.31}$$

and for this P_l, the corresponding decay rate is $\lambda_d = 0.756 < \bar{\lambda}$ as noticed in Remark 6. Similar analysis has been done for $x^* = 6$. When P_l is designed without decay rate constraint one gets $\lambda_d = 0.799$ and it is improved to $\lambda_d = 0.747$ designing P_l based on Theorem 25.

In Figure 4.7 the consensus value is fixed as $x^* = 3.5$ and we get P_l matrix is

$$P_l = \begin{bmatrix} 0.3010 & 0.6990 \\ 0.0874 & 0.9126 \end{bmatrix}. \tag{4.32}$$

Finally, in Figure 4.8 the associated Lyapunov function is plotted for both consensus values.

Numerical simulations have confirmed the intuition that P_l tends to $\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ when x^* approaches the initial local agreement of the second cluster 2.75 while P_l tends to $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ when x^* approaches 7.5 the initial local agreement of the first cluster .

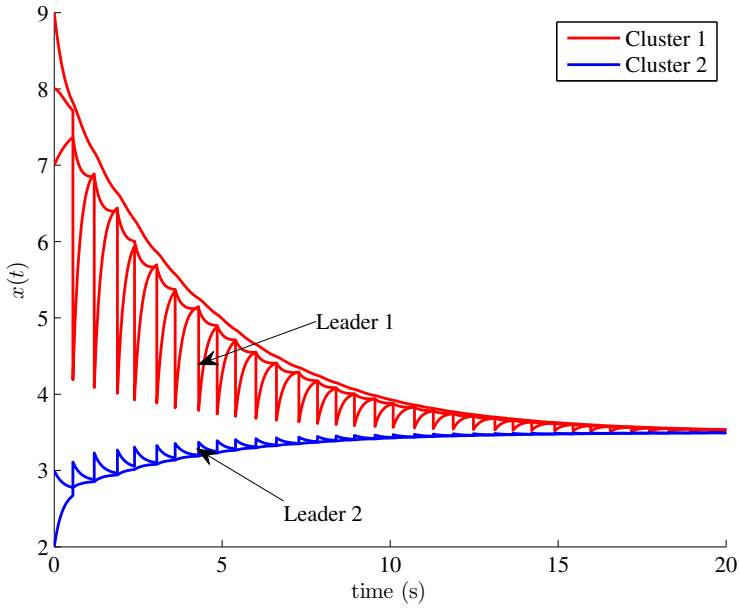

Figure 4.7: The states of a system ($x^* = 3.5$).

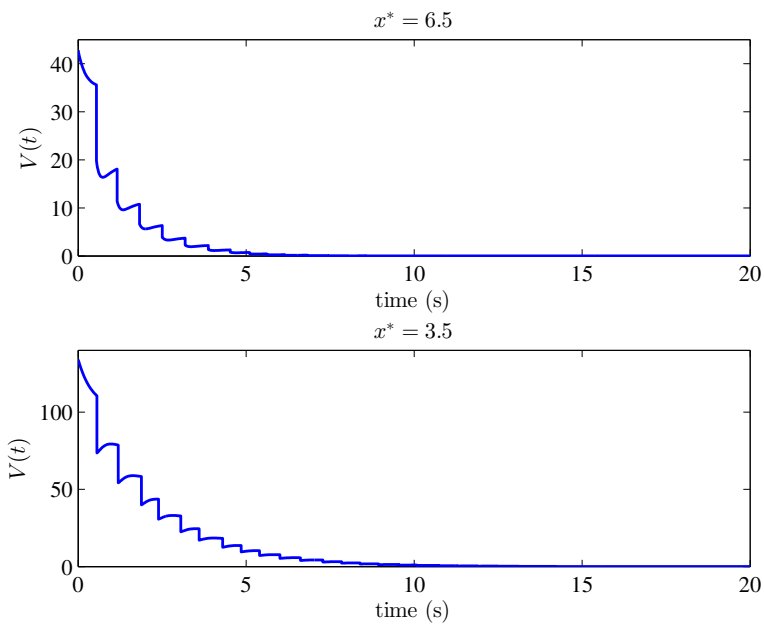

Figure 4.8: The Lyapunov function of the system

4.6.2 Larger network analysis

In order to prove that the algorithms are implementable in real networks we consider in the following a larger system. Precisely, we present an example consisting of a network of 100

agents partitioned in 3 clusters. The size of the clusters as well as the connections between agents are randomized, resulting in non-symmetric matrices L and P_l. The random initialization leads at $n_1 = 59, n_2 = 20, n_3 = 21$ and

$$P_l = \begin{bmatrix} 0.1538 & 0.8080 & 0.0382 \\ 0.4886 & 0.3876 & 0.1238 \\ 0.1266 & 0.2805 & 0.5929 \end{bmatrix}. \tag{4.33}$$

The initial condition is also randomized but, in order to guarantee a relatively large interval for the possible consensus value, for the first cluster the initial states of the agents are randomly chosen within $[0, 3]$, for the second one within $[3, 7]$ and for the third one within $[7, 10]$. The corresponding initial local agreement values are 1.2970, 5.2578 and 8.7556, respectively. We illustrate the theoretical results by using the dynamics (4.2) with the reset sequence given by $\delta = 0.5$. The corresponding consensus value computed by (4.16) is $x^* = 4.2562$. The convergence of the 100 agents towards x^* is shown in Figure 4.9.

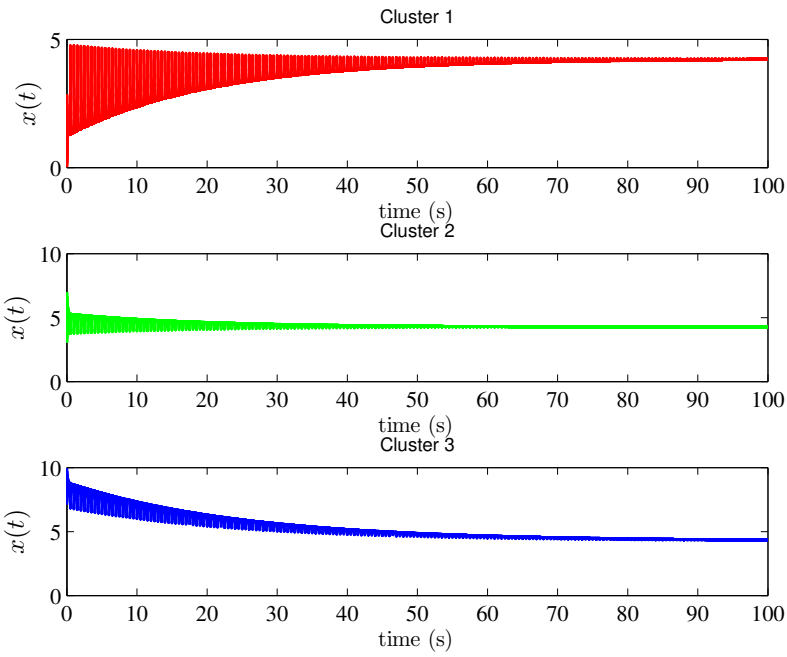

Figure 4.9: The state-trajectories of the agents converging to the calculated consensus value.

To find the decay rate λ_d we use the bisection algorithm as stated in Remark 4. In Figure 4.5 we demonstrate that the initial conditions of Lyapunov function may vary due to the conditioning of the matrix $S(\delta)$, but the decay rate remains the same. The value of λ_d obtained was $\lambda_d = 0.9712$ and the number of iterations of the bisection algorithm is $k = 10$.

As noticed before, from (4.15) we deduce that the consensus value is always a convex combination of the initial agreement values of the clusters. Therefore, for the initialization above, any consensus value can be imposed between 1.2970 and 8.7556. In Figure 4.10 the consensus value was fixed at $x^* = 6.5$ and one obtained P_l matrix is

$$P_l = \begin{bmatrix} 0.0643 & 0.3720 & 0.5637 \\ 0.3064 & 0.0358 & 0.6578 \\ 0.0360 & 0.1917 & 0.7723 \end{bmatrix}.$$

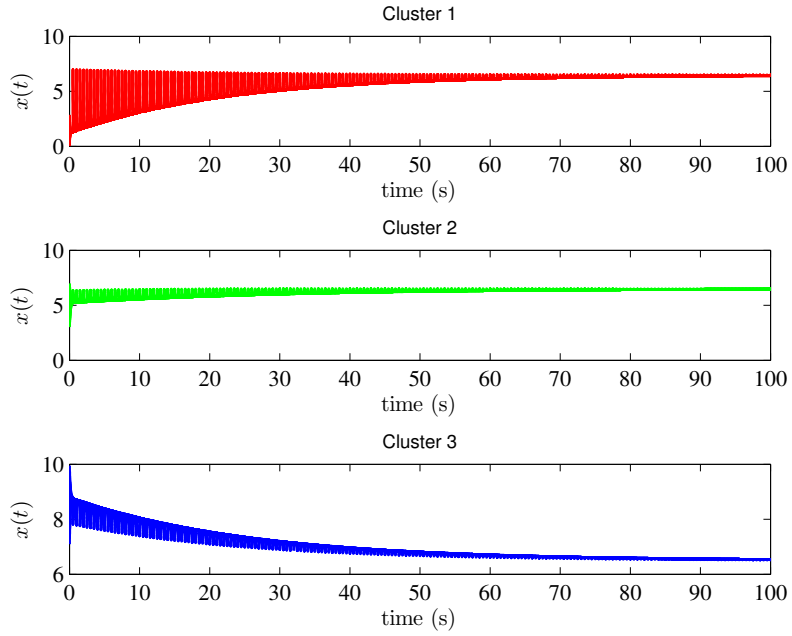

Figure 4.10: The states of a system ($x^* = 6.5$).

4.7 Conclusion

In this chapter the stability conditions and the design of the leader's network have been presented. At first, it was shown that, if a consensus can be reached, a value for the global consensus of the network can be calculated. Then, considering this calculated consensus value, the stability analysis was presented. The analysis of the decay rate was also shown. Afterwards, we presented the design of the leader's network to reach a prescribed consensus value with or without a prescribed decay rate. The chapter ends with the numerical implementation of the parametric LMIs and illustrative examples for small and large systems.

This work was the starting point for different relaxations to the case of time-varying graphs and asynchronous resets, event triggered resets and stochastic resets [MMG14, RMD15]. Notice that in all these subsequent works the consensus value is not characterized. As a consequence a control design cannot be done in order to obtain an a priori fixed consensus.

Chapter 5

Consensus on interconnected networks of nonholonomic robots

In this chapter we make use of the framework proposed in the previous chapter to address the problem of formation control of nonholonomic systems. More specifically, our objective here is to use the cluster-like structure of the network to accomplish a partial formation inside the clusters. Eventually, since the agents are capable of reaching a global consensus value, all robots converge to a final formation equal to the initially intended formation.

This chapter is organized as follows: the first section introduces the concept of a nonholonomic system, the kinematic model that will be used in this chapter and some background on control of nonholonomic robots and trajectory tracking. Then, we provide the motivation for formation control in these systems separating it in three main categories: aerial, underwater and wheeled robots. The chapter proceeds with a problem formulation, dividing the main problem into a consensus and trajectory tracking one. In this section we also recall the network topology and present the error dynamics of each robot. We present then the controller for trajectory tracking from [PLLN98]. The next section unifies the trajectory tracking controller and the consensus controller, responsible for trajectory planning. Illustrative examples are provided with small and large scale systems. The chapter ends with the conclusion.

5.1 Nonholonomic robots

Nonholonomic systems are characterized by constraint equations involving the time derivatives of the system configuration variables. These equations typically arise when the system has less controls than configuration variables. As an example, a car has three degrees of freedom: its

position in two axes and its orientation. However, there are only two controllable degrees of freedom which are acceleration (or braking) and turning angle of steering wheel. This makes it difficult for the driver to turn the car in any direction (unless the car skids or slides). As a consequence, any path in the configuration space does not necessarily correspond to a feasible path for the system.

In this chapter whenever we refer to a nonholonomic system or a nonholonomic robot/vehicle we mean a particular kind of dynamics. In fact, as we consider the classical hypothesis of "rolling without slipping", the dynamics of these robots can be expressed by a kinematic model. We present the kinematic model:

$$\begin{aligned} \dot{x}(t) &= v\cos(\theta), \\ \dot{y}(t) &= v\sin(\theta), \\ \dot{\theta}(t) &= \omega. \end{aligned} \tag{5.1}$$

where v is the linear velocity and ω is angular velocity; (x, y) are the Cartesian coordinates of the center of mass of the robot, and θ is the angle between the heading direction and the x-axis.

Control techniques for these systems have been studied for several years now. In some particular papers, the main objective is the stabilization of an equilibrium point. The robot variables should thus converge to a reference position. It has been shown that this cannot be achieved by means of a continuous feedback, see [KKMN90]. As a consequence, a number of techniques based on time-varying controllers and discontinuous feedbacks have been proposed, see [KM95, BR90, HLM99]. Instead, we are more interested in techniques in tracking of a time-varying trajectories. In [KKMN90] the authors propose a stable tracking control method for a nonholonomic vehicle by defining error dynamics, i.e. the difference between a reference trajectory and the vehicle trajectory. This result paved the path to other papers which proposed advances to this model. In [JN97] the authors use a integrator backstepping method to obtain global and semi global results in the tracking problem of a mobile robot.

The work of [JN97] served as direct inspiration to some other works, such as [PBG+15] which applied the results of [JN97] in combination with event-triggered techniques to ensure a global convergence to the reference trajectory. In [PLLN98] the authors simplify the method proposed by [JN97] which eases the implementation of the control scheme in a real system. Other techniques on trajectory tracking of nonholonomic robots can be seen in [LSLT01, AAP00, ACBB94].

5.2 Formation control and nonholonomic vehicles

The problem of formation control in multiagent systems was briefly discussed in the introduction chapter. The focus of this section is to present results on formation specifically to robots with non-holonomic dynamics thus motivating our own work with this kind of robots. Several vehicles can be modeled utilizing nonholonomic dynamics, therefore we separate this section into three subsections containing results related to ground vehicles, aerial vehicles and underwater vehicles.

5.2.1 Ground vehicles

Ground vehicles are the most used example for nonholonomic robots. Differently than the other configurations, there is no necessity of an altitude or depth control, therefore making wheeled robots the go to example for nonholonomic robots. The formation control is mainly realized, as discussed in the Introduction, in one of three main ways: leader-follower [CMPT08, DOK01], reactive behaviors [BRK99, BA98] and virtual structures [TL96, YB02].

The study of formation on ground vehicles is specially influenced by the swarm behavior of insects More specifically, ground robots are able to cooperate and perform several tasks including exploration [FBKT00], surveillance [FS01], search and rescue [JWE97], mapping of unknown or partially known environments, distributed manipulation [RDJ95, MNS95], and transportation of large objects [SB93].

5.2.2 Aerial Vehicles

The same nonholonomic model used for ground vehicles can be used for aerial vehicles. Unmanned air vehicles (UAVs) equipped with low-level altitude-hold, velocity-hold, and heading-hold autopilots can be modeled by kinematic equations similar to those of mobile robots, as in Equation (5.1). However, the inherent properties of fixed-wing UAVs impose input constraints of positive minimum velocity due to the stall conditions of the aircraft, bounded maximum velocity, and saturated heading rate. Therefore, techniques applied to ground robots cannot be directly applied to UAVs.

Formation control of aerial vehicles is most often related to military applications, specially with non tripulated aircrafts (popularly known as drones). One of the first papers on formation control of aircrafts was [SPR77] where the authors proposed a multiple spacecraft interferometer for infrared synthetic aperture imaging. Since then other applications for aerial vehicles

have been discussed. One example of application for UAVs is wide-area surveillance where low-altitude UAVs must provide coverage of a region and investigate events of interest as they manifest themselves [ESF08]. Other applications are: safety (monitoring of the airspace, urban and interurban traffic), environmental protection (measurement of air pollution and forest monitoring), intervention in hostile sites (radioactive workspace and mine clearance), management of the large infrastructures (dams, high-tension lines and pipelines). For more information on control of aircrafts we suggest the survey [SHP04].

5.2.3 Underwater Vehicles

In a similar manner to the aerial vehicles, Autonomous Underwater Vehicles (AUVs) can also be modeled by non-holonomic dynamics. The requirements to model underwater vehicles are similar to aerial vehicles. A velocity-hold controller must be used, otherwise the underwater vehicles will sink. In [BASCdW11, BASCDW09] the authors propose a circular formation to keep all vehicles in motion. By fixing a radius of the circle and having a trajectory for the center of the formation, the AUVs can keep themselves in motion.

The applications for this kind of robots are mainly mine search, a technique called Naval Countermeasure Missions [SBS06]. Other applications include distributed wide-area ocean exploration, large-scale multisensor survey, cooperatively handling of large object, and multisite inspections. Compared with an individual AUV system or a human-based operation system, multiple AUV systems are advantageous to be exploited in these complicated environments since they can reduce the costs and improve the successful rates of missions significantly.

5.3 Problem formulation

In this section we present the problem. Consider a network of mobile robots. Due to communication restrictions the network is not overall connected, instead it is separated into several clusters, or smaller networks. The connections inside each cluster happen continuously or very rapidly. Moreover, one robot of each cluster is allowed to communicate outside its cluster at discrete time instants. The goal is to realize an a priori specified formation in a decentralized manner.

For the sake of clarity we give a simple example that illustrates this type of problem. In Figure 5.1 we consider two clusters, red and blue, of two robots that have to realize a given formation (a square, in this case). The only information that is provided to each robot is the state of its neighbors. In this case each robot accesses only the position and velocity of another robot of the

same color. Due to the cluster pattern the robots would never be able to accomplish their task without communication between the clusters. Thus, at some discrete-time instants we allow a communication between one red and one blue robot (tagged in Figure 5.1 as L), that we call leaders in the sequel.

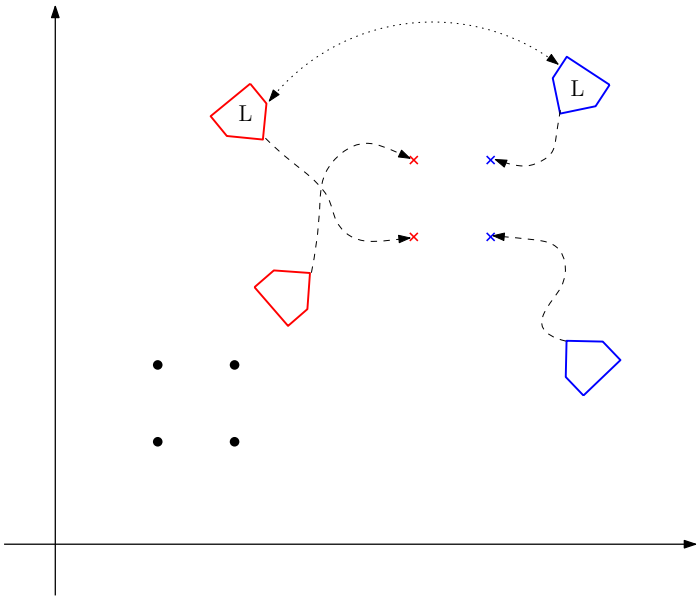

Figure 5.1: Formation problem. The bottom formation (black dots) shows the intended formation while the top formation (crosses) is the final formation. Agents that can communicate outside the cluster (leaders) are tagged by L, and communication outside the clusters is shown by the dotted line.

The approach we take separates the formation realization problem in two subproblems: the trajectory tracking of a reference under non-holonomic dynamics; and the consensus approach used to generate this reference. Before we can formally define and solve these subproblems, some preliminaries about the robot dynamics and the network topology need to be presented.

5.3.1 Robot dynamics and tracking error

The dynamics of each individual robot are described by the following differential equation

$$\begin{aligned} \dot{q}_i(t) &= v_i \cos(\theta_i), \\ \dot{y}_i(t) &= v_i \sin(\theta_i), \\ \dot{\theta}_i(t) &= \omega_i \end{aligned} \tag{5.2}$$

where v_i is the linear velocity and ω_i is angular velocity of robot i; (q_i, y_i) are the Cartesian coordinates of the center of mass of the robot, and θ_i is the angle between the heading direction and the q-axis (see Figure 5.2).

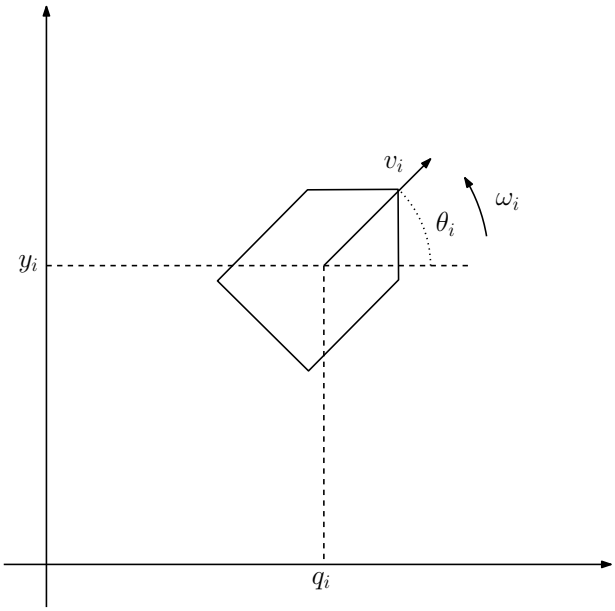

Figure 5.2: The mobile robot

In the sequel we will have to solve a trajectory tracking problem, so we wish to find control laws for v_i and ω_i such that the robot follows a reference position $p_{ri}(t) = (q_{ri}(t), y_{ri}, \theta_{ri}(t))$ with velocities $v_{ri}(t)$ and $\omega_{ri}(t)$. We now define the error coordinates from [KKMN90] (see Figure 5.3).

$$\begin{bmatrix} q_{ei} \\ y_{ei} \\ \theta_{ei} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 \\ -\sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{ri} - q_i \\ y_{ri} - y_i \\ \theta_{ri} - \theta_i \end{bmatrix} \tag{5.3}$$

and the corresponding error dynamics

$$\begin{aligned} \dot{q}_{ei}(t) &= \omega_i y_{ei} - v_i + v_{ri}\cos(\theta_{ei}), \\ \dot{y}_{ei}(t) &= -\omega_i q_{ei} + v_{ri}\sin(\theta_{ei}), \\ \dot{\theta}_{ei}(t) &= \omega_{ri} - \omega_i \end{aligned} \tag{5.4}$$

5.3.2 Network topology and agreement dynamics

As explained earlier we consider that each robot represents an agent in a network. Therefore we provide here some basic concepts related to multi-agent systems and we describe the structure of the network at hand. The communications among agents are specified by a directed graph. A link in this graph represents that an information is transmitted between the vertices situated at its ends. The direction of the link simply states which of the vertices/robots receive the information and which of them send it. The general framework is as follows. We consider a

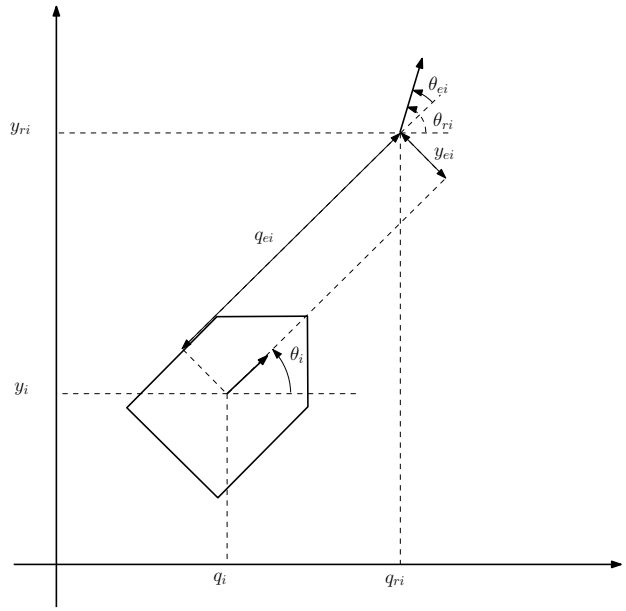

Figure 5.3: The error dynamics

network of n agents described by the digraph (i.e. directed graph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V}$ represents the set of agents and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents the interactions.

In the sequel, we consider that the agent set $\mathcal{V}$ is partitioned in m strongly connected clusters/communities $\mathcal{C}_1, \ldots, \mathcal{C}_m$ and no link between agents belonging to different communities exists. Each community possesses one particular agent called leader and denoted in the following by $l_i \in \mathcal{C}_i,\ \forall i \in \{1, \ldots, m\}$. The set of leaders will be referred to as $\mathcal{L} = \{l_1, \ldots, l_m\}$. The rest of the agents will be called followers and denoted by f_j. For the sake of clarity we consider that the leader is the first element of its cluster:

$$\mathcal{C}_i = \{l_i, f_{m_{i-1}+2}, \ldots, f_{m_i}\},\ \forall i \in \{1, \ldots, m\} \tag{5.5}$$

where $m_0 = 0,\ m_m = n$ and the cardinality of $\mathcal{C}_i$ is given by $|\mathcal{C}_i| \triangleq n_i = m_i - m_{i-1}, \forall i \geq 1$. At specific time instants $t_k,\ k \geq 1$, called reset times, the leaders interact between them following a predefined interaction map $\mathcal{E}_l \subset \mathcal{L} \times \mathcal{L}$. We also suppose that $\mathcal{G}_l = (\mathcal{L}, \mathcal{E}_l)$ is strongly connected meaning that there exists a directed path (i.e. sequence of directed edges) between any two different agents. In order to keep the presentation simple, by an abuse of notation each agent will have a scalar state denoted also by l_i for the leader l_i and f_j for the follower f_j. We also introduce the vectors $x = (l_1, f_2, \ldots, f_{m_1}, \ldots, l_m, \ldots, f_{m_m})^\top \in \mathbb{R}^n$ and $x_l = (l_1, l_2, \ldots, l_m)^\top \in \mathbb{R}^m$ collecting all the states of the agents and all the leaders' states, respectively.

As a part of control strategy we have to design the reference trajectories through a linear reset dynamics that agrees with the communications constraints introduced before. In order to ensure the collaborative control achieves global agreement, the references will be defined by the

following overall network dynamics:

$$\begin{cases} \dot{x}(t) = -Lx(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ x_l(t_k) = P_l x_l(t_k^-) & \forall t_k \in \mathcal{T} \\ x(0) = x_0 \end{cases} \tag{5.6}$$

where $\mathcal{T} = \{t_k \in \mathbb{R}_+ \mid t_k < t_{k+1}, \forall k \in \mathbb{N}, t_k \text{ reset time}\}$, $L \in \mathbb{R}^{n\times n}$ is a generalized Laplacian matrix associated to the graph $\mathcal{G}$ and $P_l \in \mathbb{R}^{m\times m}$ is a Perron matrix associated to the graph $\mathcal{G}_l = (\mathcal{L}, \mathcal{E}_l)$. To make the link between consensus and formation control, note that a simple change of coordinates allows transforming the problem of realizing a formation into a consensus problem. Let us define the desired formation as a vector of positions $p = (p_1, p_2, ..., p_n)$ where p_i is the position that corresponds to the ith robot. Define also:

$$z(t) = r^*(t) - p, \tag{5.7}$$

where z is the vector that aggregates the distances between the agents' desired positions (r^*) and the formation positions (p). With the background developed so far, we can formalize the subproblems presented in the beginning of the section.

Subproblem 1 (Linear consensus): *Consider a network of robots separated in clusters. Given the proposed change of coordinates z, prove that consensus will be reached in z.*

If consensus is achieved in z (not necessarily at 0), then all the robots' references will achieve the formation somewhere in the plane (not necessarily at the locations where the formation is defined). Then, to actually follow the reference trajectories, we need to solve the following trajectory tracking problem:

Subproblem 2 (Tracking for non-holonomic systems): *Consider a mobile robot with non-holonomic dynamics. Given a reference trajectory, find appropriate velocity control laws v and ω in the form $v = v(t, q_e, y_e, \theta_e)$ and $\omega = \omega(t, q_e, y_e, \theta_e)$ that follow this reference trajectory.*

5.4 Consensus and tracking problems

The separation of the main problem into two simpler problems means they can also be solved separately. The change of variable does not change the proof of the stability and convergence of the consensus problem and were proven in Chapter 4. In Section 5.4.1 we present the controller from [PLLN98], which will be useful in tracking the trajectories defined in the consensus phase.

5.4.1 Tracking for nonholonomic systems

In this section we present the controller from [PLLN98]. Denote the reference signals for robot i by $(q_{ri}, y_{ri}, v_{ri}, \omega_{ri}, \theta_{ri})$. Recall the error dynamics defined in (5.4):

$$\begin{aligned}\dot{q}_{ei}(t) &= \omega_i y_{ei} - v_i + v_{ri}\cos(\theta_{ei}),\\ \dot{y}_{ei}(t) &= -\omega_i q_{ei} + v_{ri}\sin(\theta_{ei}),\\ \dot{\theta}_{ei}(t) &= \omega_{ri} - \omega_i\end{aligned}$$

We can stabilize the mobile robot's orientation, via the linear system $\dot{\theta}_{ei}(t) = \omega_{ri} - \omega_i$, by using the linear controller

$$\omega_i = \omega_{ri} + c_{1i}\theta_{ei} \tag{5.8}$$

which yields GUES for θ_{ei}, provided $c_1 > 0$. If we now set $\theta_{ei} = 0$ in (5.4) we obtain

$$\begin{aligned}\dot{q}_{ei}(t) &= \omega_i y_{ei} - v_i + v_{ri}\\ \dot{y}_{ei}(t) &= -\omega_i q_{ei}\end{aligned} \tag{5.9}$$

Concerning the position of the robot, if we choose the linear controller

$$v_i = v_{ri} + c_{2i} q_{ei} \tag{5.10}$$

where $c_2 > 0$, we obtain for the closed-loop system (5.4):

$$\begin{bmatrix}\dot{q}_{ei}\\ \dot{y}_{ei}\end{bmatrix} = \begin{bmatrix}-c_{2i} & \omega_{ri}(t)\\ -\omega_{ri}(t) & 0\end{bmatrix}\begin{bmatrix}q_{ei}\\ y_{ei}\end{bmatrix}$$

which under appropriate conditions on $\omega_{ri}(t)$ is asymptotically stable.

Remark 7 *In [PLLN98] the result is made rigorous by Proposition 11. Considering the system (5.4) in closed loop with the controller (5.8,5.10), Proposition 11 states that the closed loop system is globally exponentially stable if $\omega_{ri}(t)$, $\dot{\omega}_{ri}(t)$ and $v_{ri}(t)$ are bounded.*

Remark 8 *The tracking controller presented in this section provides an exponential decrease of the error as long as the reference is continuous. This is the case for almost all the robots in the network. However, the leaders have discontinuous references (see (4.18)), so combining the tracking controller in [PLLN98] with the references design defined in [BMDR14] requires a supplementary condition that is provided in the next section.*

5.5 Overall controller design

We are ready to define the overall control strategy. As in [BM14], and as described above in Section 5.4, the main idea is to track the reference trajectories defined by the consensus strategy.

It is important to note that only local information is needed for the reference design, and once the reference is available, the tracking problem is completely decentralized since no additional information from the neighbors is needed.

Consider the directed graph defined in (5.6), with the nonlinear dynamics defined by (5.2). As we are interested in realizing a formation on the 2D plane, we will collect information only about the position (q_i, y_i) of each agent, and this will be the consensus state. As an example, for the leader of the first cluster we have $l_1 = (q_1, y_1)$ and the next agent of the same cluster has $f_2 = (q_2, y_2)$. Define also the collective state $x = (l_1^\top, f_2^\top, \ldots, f_{m_1}^\top, \ldots, l_m^\top, \ldots, f_{m_m}^\top)^\top \in \mathbb{R}^{nm}$, containing the consensus states of all agents. Consensus will be sought on the distances z between the robot states and the imposed formation. Thus, finally, the linear reset system describing the overall network dynamics is:

$$\begin{cases} \dot{z}(t) = -(L \otimes I_2)z(t), & \forall t \in \mathbb{R}_+ \setminus \mathcal{T} \\ z_l(t_k) = (P_l \otimes I_2)z_l(t_k^-) & \forall t_k \in \mathcal{T} \\ z(0) = z_0 \end{cases} \tag{5.11}$$

From $z(t)$, the references for the robots must be obtained. First, since $z(t) = r^*(t) - p$ from (5.7), the collective reference position $r^*(t)$ is $z(t) + p$, from which the reference position of robot i can be extracted: $(q_{ri}, y_{ri})^\top = r_i^*(t)$. To apply the local tracking controller, additional information is necessary including the linear velocity v_{ri}, the angle θ_{ri}, and the angular velocity ω_{ri}.

We have $\dot{z}(t) = \dot{r}^*(t)$ since the vector p is constant, so the derivative of $r_i^*(t)$ gives us the linear velocities on the x and y axes, $\dot{r}_i^*(t) = (\dot{q}_{ri}, \dot{y}_{ri})^\top$. By taking advantage of the consensus dynamics in (5.6), this derivative can be found as:

$$\dot{r}^*(t) \;\; = \dot{z}(t) \;\; = -(L \otimes I_2)z(t) \;\; , \tag{5.12}$$

Then, we obtain the linear velocity by:

$$v_{ri} = \sqrt{\dot{q}_{ri}^2 + \dot{y}_{ri}^2}. \tag{5.13}$$

By taking into account the velocities on both axes we can now find the orientation of our reference trajectory (θ_{ri}),

$$\theta_{ri} = \arctan \frac{\dot{y}_{ri}}{\dot{q}_{ri}}, \tag{5.14}$$

and by differentiating this we directly get the reference angular velocity ω_{ri}

$$\omega_{ri} = \frac{\partial \theta_{ri}}{\partial t}. \tag{5.15}$$

This allows us to complete the first part of the control scheme, to generate the correct trajectories for the robots. Furthermore, using the the trajectory tracking controller presented in Section 5.4.1, we have an overall controller to generate and follow trajectories. Figure 5.4 demonstrates this control scheme.

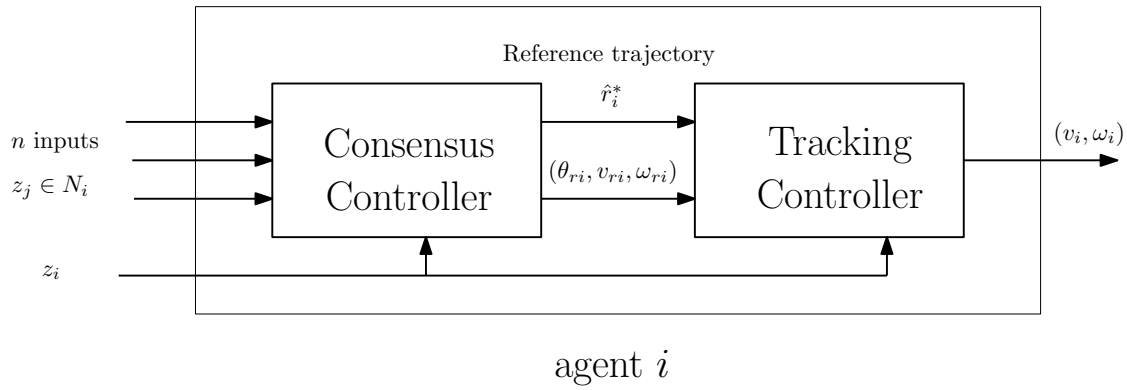

Figure 5.4: Control scheme for robot i

In practice, instead of the continuous dynamics we implement a discretization of the Laplacian matrix, which is a more realistic model. We consider a new reset time $\tau_{k+1} - \tau_k = \frac{\delta_{max}}{K}$, where K is an integer constant and δ_{max} is the maximum reset time in the admissible reset sequences $\Phi(\Delta)$. To make an abuse of notation, and considering that the new reset time will remain constant, we define $\tau = \tau_{k+1} - \tau_k$. Specifically, the stochastic counterpart $P = e^{-L\tau}$ of the Laplacian matrix is computed. The network topology remains the same. This means the dynamics (5.11) change to:

$$\begin{cases} \hat{z}(\tau_{k+1}) = (P \otimes I_2)z(\tau_k), \\ z(t_k) = (P_{ex} \otimes I_2)z(t_k^-) \quad \forall t_k \in \mathcal{T} \\ z(0) = z_0 \end{cases} \tag{5.16}$$

Here, $\hat{z}(\tau_{k+1})$ is no longer the next step in the consensus but it is instead a target consensus. By using the equation (5.7), this leads us to a target position $\hat{r}_i^*(\tau_k) = (\hat{q}_{ri}, \hat{y}_{ri})^\top$ as shown in figure 5.5.

Another practical consideration is that due to physical constraints, the robots might not be able to follow the trajectories defined by the consensus algorithm, when the speed of convergence of the Laplacian matrix leads to faster trajectories than what the robot can implement. We solve this by considering a saturation of the reference linear velocity and of the angle. With this change, the final formulas for the reference velocity, angle, and angular velocity are:

$$v_{ri}(\tau_k) = \begin{cases} \sqrt{(\hat{q}_{ri}(\tau_{k+1}) - q_{ri}(\tau_k))^2 + (\hat{y}_{ri}(\tau_{k+1}) - y_{ri}(\tau_k))^2} & \text{if} < v_{max} \\ v_{max} & \text{otherwise} \end{cases} \tag{5.17}$$

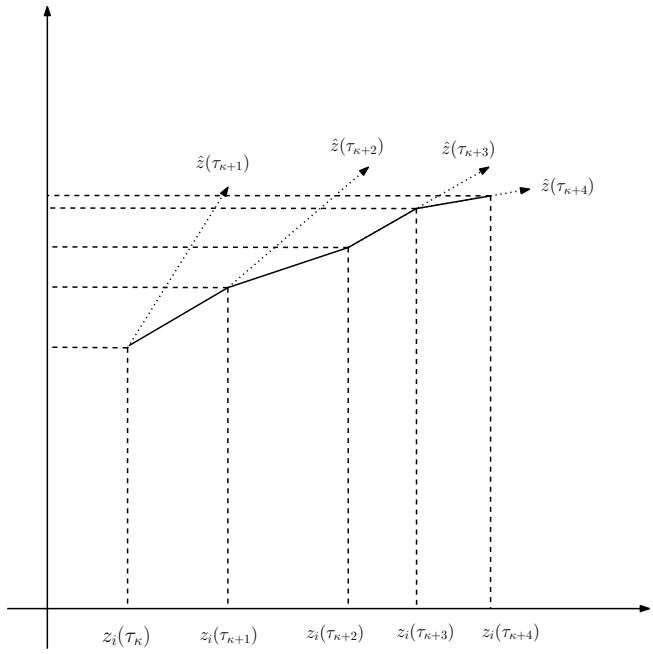

Figure 5.5: The target value and the actual trajectory

$$\theta_{ri}(t) = \begin{cases} \arctan \frac{(\hat{y}_{ri}(\tau_{k+1}) - y_{ri}(\tau_k))}{(\hat{q}_{ri}(\tau_{k+1}) - q_{ri}(\tau_k))} & \text{if} < \theta_{max} \\ \theta_{max} & \text{otherwise} \end{cases} \tag{5.18}$$

$$\omega_{ri} = \frac{\theta_{ri}(\tau_k) - \theta_{ri}(\tau_{k-1})}{\tau} \tag{5.19}$$

A crucial property of the overall algorithm is that, in-between the resets, the distances to the references trajectories will be reduced. This happens because the tracking controller ensures exponential stability and is allowed to run for a time $t_{k+1} - t_k$ which is always finite. Furthermore, the initial tracking errors will get smaller and smaller after each reset, due to the convergence properties of the consensus algorithm. Thus, we expect that the fleet of non-holonomic robots will reach any given formation with arbitrary precision.

The property of error reduction in-between the resets is similar to the controllability requirement in [BM14], which assumed that each robot can reach any reference in a given range with at most K consecutive actions. In our case, the constant K in the time discretization of the consensus dynamics can be seen as playing a similar role to K in [BM14].

5.6 Simulation results

We first validate our approach in a small-scale example with 5 robots. Afterwards, a larger-scale example with 15 robots is shown. In all these simulations we consider the reset time of the leaders to be periodic, i.e. $t_{k+1} - t_k = \delta$.

5.6.1 Small-scale example: Ellipse formation

Consider a network of 5 robots partitioned in 2 clusters ($n_1 = 3, n_2 = 2$). The topology of the clusters and the leader interconnections is described by:

$$L = \begin{bmatrix} 0.04 & -0.02 & -0.02 & 0 & 0 \\ -0.01 & 0.01 & 0 & 0 & 0 \\ 0 & -0.02 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & -0.03 \\ 0 & 0 & 0 & -0.01 & 0.01 \end{bmatrix}, \; P_l = \begin{bmatrix} 0.45 & 0.55 \\ 0.25 & 0.75 \end{bmatrix}$$

The reset sequence is given by the reset time $\delta = 5s$ with $\tau = 0.05s$. The initial positions of the robots are:

$$\begin{aligned} q = &\quad (4.0985, 4.1130, 0.2262, 3.9355, 1.2436) \\ y = &\quad (1.7976, 2.5639, 3.9691, 2.1607, 4.2717) \\ \theta = &\quad (1.0463, -2.0224, -2.3373, 3.1358, -2.0664) \end{aligned} .$$

The formation was defined as an ellipse, where the robots are equally spaced by an angle of $2\pi/n$. The positions defined in the formation are:

$$\begin{aligned} p_q = &\quad (1.0000, 1.9511, 1.5878, 0.4122, 0.0489) \\ p_y = &\quad (0.5000, 0.8455, 1.4045, 1.4045, 0.8455) \end{aligned} .$$

The controller parameters were defined as $c_{1i} = 0.7, c_{2i} = 0.5, \; \forall i$. In Figure 5.6 the trajectories of the robots are shown, with the intended formation shown as stars and the final position of the robots as circles.

Figure 5.7 shows, for a different experiment, the positions of each robot at the reset, as well as the initial conditions and the intended formation. The only change from the experiment above was made to the reset schedule, with $\delta = 100s$ and $\tau = 0.1s$. These changes were made such that the convergence to the partial formation (local consensus at each cluster) could be ensured prior to the reset, as shown by the dashed ellipses. The red ellipse shows all the robots of

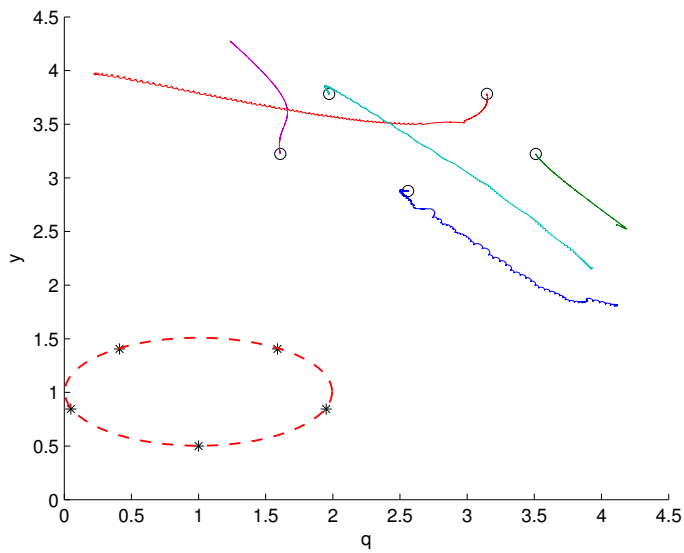

Figure 5.6: Trajectories of the robots with their final position as circles and intended formation as stars.

cluster 1 in a partial formation at the first reset while the blue dashed ellipse shows the same for cluster 2. The initial positions of the robots are represented by diamonds colored red or blue to represent cluster membership.

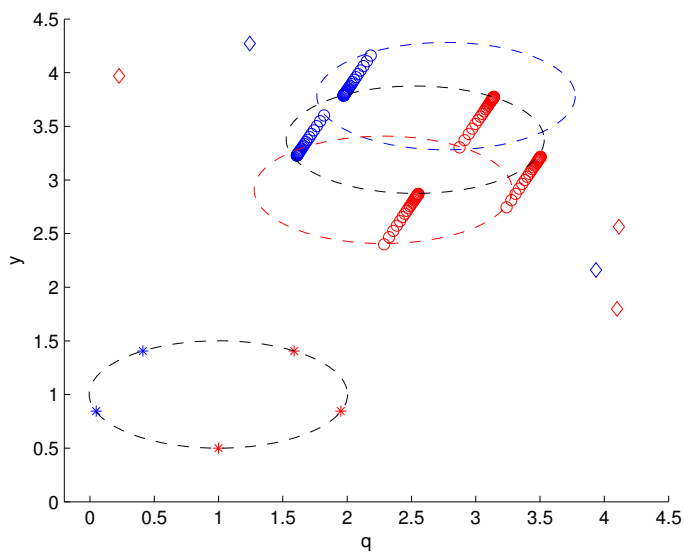

Figure 5.7: A second experiment: The circles represent the local consensus point at each reset; the diamonds are the initial positions and stars the intended formation. Each cluster is color coded with red and blue.

We conclude from these results that the algorithm is able to achieve the formation; and that partial consensus can be achieved at each reset, given that a large enough interval between resets is provided. In fact, in Figure 5.7 we can see the trajectory not as robots individually going to the formation, but as partial formations getting close to each other.

5.6.2 Large-scale example: Three-leaf clover formation

In the next example we consider a larger system of 15 robots equally separated into 3 clusters ($n_1 = 5, n_2 = 5, n_3 = 5$). The inter-leader communication is driven by:

$$P_l = \begin{bmatrix} 0.1160 & 0.4650 & 0.4190 \\ 0.1360 & 0.5540 & 0.3100 \\ 0.2850 & 0.3100 & 0.4050 \end{bmatrix}$$

while the matrix L is randomized. The reset sequence is given by $\delta = 5s$ with $\tau = 0.05s$. The initial positions and orientations of the robots are also randomized. The controller parameters c_{1i} and c_{2i} are the same as those used in the five-robot example. The formation considered this time is a three-leaf clover, represented by the the following parametric equations:

$$\begin{aligned} q &= q_c + a\cos(3t)\cos(t) \\ y &= y_c + a\cos(3t)\sin(t) \end{aligned}$$

where $(q_c, y_c) = (1, 1)$ is the center of the formation and $a = 1$ the length of the leaves, with the robots equally spaced from each other by an angle of $2\pi/n$. In Figure 5.8 the trajectories show the convergence to the three-leaved clover formation.

Similarly to the first example, we run a new experiment with more widely spaced resets, $\delta = 200s$ and $\tau = 0.1s$, so as to better present the partial formations during the resets. Figure 5.9 shows the positions of the agents during the first and the last reset. The intended three-leaf clover formation is depicted at the bottom left, while the center of the figure shows the final formation as achieved on the 2D plane. Each cluster is color coded, where the first cluster is red, second is blue and third is green.

These results showcase the fact that the algorithm works despite the larger number of robots or the complexity of the formation. In Figure 5.9, two of the three clusters have already reached their partial formation at the first reset, while one cluster has not completely reached local consensus yet.

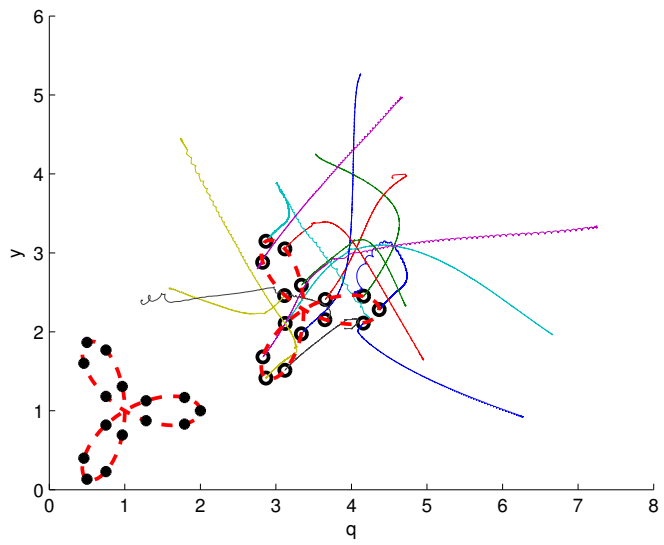

Figure 5.8: Robots in a three-leaved clover formation.

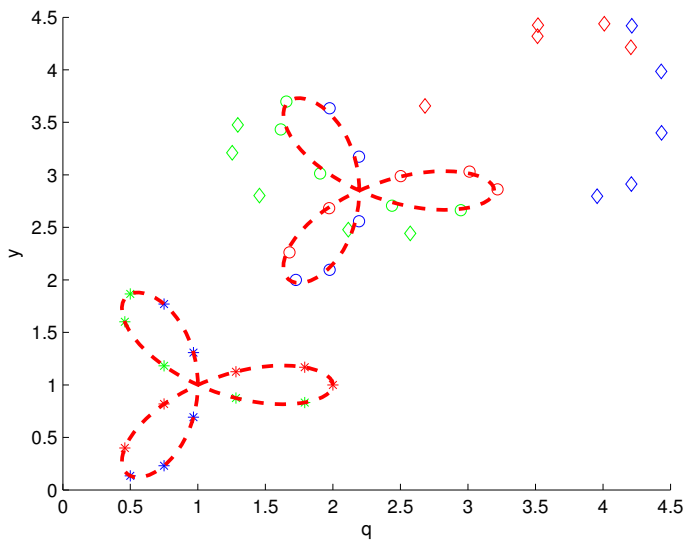

Figure 5.9: Robots in a three-leaved clover formation, difference between first and last reset.

5.7 Conclusion

In this chapter we presented an application to the results of Chapter 4. The chapter began with a short background on nonholonomic robots and trajectory tracking techniques, including the kinematic model for this class of robots. We proceed with the motivation for studying formation control on nonholonomic robots. The kinematic model that was presented can be used to model different types of vehicles (ground, air and sea). Afterward we presented the problem stating

that it can be separated into two subproblems: the trajectory tracking of a reference under nonholonomic dynamics and the consensus approach used to generate this reference. We then recalled the network topology from Chapter 4 and the dynamics of each robot. The controller for this vehicles is introduced, using a two-part method: the consensus problem to generate the trajectories and trajectory tracking controller to follow these trajectories. We then provide illustrative examples, with a small scale system with 5 robots and a larger scale system with 15 robots.

Chapter 6

Conclusion

6.1 Conclusions

We have investigated the control and analysis of cluster-patterned networks. We can summarize this dissertation in the following points:

- In Chapter 1 we introduced multiagent systems and motivated it accordingly. We gave some details into two common problems in multiagent systems: flocking and formation. We also provide motivation for an important matter during the thesis: consensus in clustered networks.

- Chapter 2 presented some preliminaries in graph theory reset systems and some concepts of matrices and Linear Matrix Inequalities (LMIs).

- The state of the art in consensus theory was presented in Chapter 3. We presented the results of the literature for fixed and switching networks, in both continuous and discrete time. We also presented the state of art in clustering of networks, with the results related to modularity of clusters and consensus in clustered networks. We related our work to the existing literature.

- Chapter 4 presented the main results in clustered networks. We presented a novel network topology where all the agents inside a network are divided into clusters and communication outside these clusters is done only at specific times in a special network called leader's network. Afterward, we provided a way to compute the consensus value given only the initial conditions of the system and the topology of the network. The stability of the system is also proven. We then provided tools to design the leader's network to accomplish some specific goals, such as reaching a priori defined consensus value and

reaching consensus quicker than some defined decay rate.

- Chapter 5 presented an application to the consensus problem introduced in Chapter 4: formation control of mobile robots. we are capable of driving a number of mobile robots to a certain formation in the space. We presented an introduction to mobile robots with nonholonomic dynamics and formation control. We then propose a solution to the formation problem by separating it two subproblems : trajectory tracking and consensus agreement between the agents.

6.2 Contributions

The contributions of the thesis are the following.

- The framework proposed for the network connections is, as the best of our knowledge, a first of its kind. Clustering in networks is often considered solely with agents with one type only of dynamics (either continuous or discrete). Hybrid consensus has been studied but in a very different approach than ours. Adding a connection outside of clusters that resets the value of only one agent allows to diminish the size of a network while keeping it connected. We can see this applied to networks that suffer from long distance communications or faulty transmissions.

- The Sections 4.2.2 and 4.3.1 provide both the consensus value and the stability proof for our system. In the literature the focus is in either one of these problems.

- We designed the reset interactions of the leaders in order to achieve a prescribed consensus value and reach the consensus faster than a certain decay rate as well as both limitations.

- We provide an application to the clustered pattern network which can be useful to solve some formation problems. The ability to move to a final formation while keeping partial formations intra-clusters can be used for both transportation and assembly.

6.3 Future research

We think that the obtained results in this thesis can be further extended in several directions. We suggest some of these directions below.

- The network proposed can be used to study hierarchical chains in social networks. In fact, the followers-leaders division can be seen as a simple hierarchical network with two layers of separation. We suggest that a problem worth investigating in the hierarchical consensus framework is to characterize the consensus value, with synchronous and asynchronous resets.
- The controller proposed in the application chapter can be expanded to work with AUVs and UAVs.
- Further works can be done in a special case of consensus called gossiping. Gossip is a special kind of agreement protocol where only two agents communicate at a given time. The paper entitled "Planning methods for the optimal control and performance certification of general nonlinear switched systems" (accepted to the 54th IEEE Conference on Decision and Control), which was worked on during the thesis, provides tools to expand on these results. By considering each communicating pair of agents as a mode of a switched system, we can develop the technique proposed in this paper to gossiping.

Bibliography

[AAP00] A. P. Aguiar, A. N. Atassi, and A. M. Pascoal. Regulation of a nonholonomic dynamic wheeled mobile robot with parametric modeling uncertainty using lyapunov functions. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 3, pages 2995–3000. IEEE, 2000.

[AB09] D. Angeli and P.-A. Bliman. Convergence speed of unsteady distributed consensus: decay estimate along the settling spanning-trees. *SIAM Journal on Control and Optimization*, 48(1):1–32, 2009.

[ACBB94] M. Aicardi, G. Casalino, A. Balestrino, and A. Bicchi. Closed loop smooth steering of unicycle-like vehicles. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 3, pages 2455–2458. IEEE, 1994.

[Alt13] C. Altafini. Consensus problems on networks with antagonistic interactions. *IEEE Transactions on Automatic Control*, 2013.

[Ant71] J. M. Anthonisse. The rush in a directed graph. *Stichting Mathematisch Centrum. Mathematische Besliskunde*, (BN 9/71):1–10, 1971.

[BA98] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions on*, 14(6):926–939, 1998.

[BA08] E. Bıyık and M. Arcak. Area aggregation and time-scale modeling for sparse nonlinear networks. *Systems & Control Letters*, 57(2):142–149, 2008.

[BASCDW09] L. Briňón-Arranz, A. Seuret, and C. Canudas De Wit. Translation control of a fleet circular formation of auvs under finite communication range. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 8345–8350. IEEE, 2009.

[BASCdW11] L. Brinón-Arranz, A. Seuret, and C. Canudas-de Wit. Collaborative estimation of gradient direction by a formation of auvs under communication constraints.

In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5583–5588. IEEE, 2011.

[BBA75] R. L. Breiger, S. A. Boorman, and P. Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of mathematical psychology*, 12(3):328–383, 1975.

[BHT09] V. D. Blondel, J. M. Hendrickx, and J. N. Tsitsiklis. On krause's multi-agent consensus model with state-dependent connectivity. *Automatic Control, IEEE Transactions on*, 54(11):2586–2597, 2009.

[BM12] S. Bolouki and R. P. Malhame. Theorems about ergodicity and class-ergodicity of chains with applications in known consensus models. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1425–1431. IEEE, 2012.

[BM14] L. Buşoniu and I.-C. Morărescu. Consensus for black-box nonlinear agents using optimistic optimization. *Automatica*, 50(4):1201–1208, http://dx.doi.org/10.1016/j.automatica.2014.02.021 2014.

[BMDR14] M. C. Bragagnolo, I.-C. Morărescu, J. Daafouz, and P. Riedinger. LMI sufficient conditions for the consensus of linear agents with nearly-periodic resets. In *American Control Conference*, pages 2575–2580, 2014.

[BNKR03] E. Ben-Naim, P. L. Krapivsky, and S. Redner. Bifurcations and patterns in compromise processes. *Physica D: Nonlinear Phenomena*, 183(3):190–204, 2003.

[BNKVR03] E. Ben-Naim, P. L. Krapivsky, F. Vazquez, and S. Redner. Unity and discord in opinion dynamics. *Physica A: Statistical Mechanics and its Applications*, 330(1):99–106, 2003.

[BR90] A. M. Bloch and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pages 1312–1314. IEEE, 1990.

[BRK99] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

[CFL09] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of modern physics*, 81(2):591, 2009.

[CK85] J. H. Chow and P. V. Kokotovic. Time scale modeling of sparse dynamic networks. *Automatic Control, IEEE Transactions on*, 30(8):714–722, 1985.

[Cle58] J. C. Clegg. A nonlinear integrator for servomechanisms. *American Institute of Electrical Engineers, Part II: Applications and Industry, Transactions of the*, 77(1):41–42, 1958.

[CMPT08] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques. Leader–follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44(5):1343–1349, 2008.

[CS07] F. Cucker and S. Smale. On the mathematics of emergence. *Japanese Journal of Mathematics*, 2007.

[DH04] A. Dimeas and N. Hatziargyriou. A multi-agent system for microgrids. In *Methods and applications of artificial intelligence*, pages 447–455. Springer, 2004.

[DH07] A. Dimeas and N. Hatziargyriou. Agent based control of virtual power plants. In *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, pages 1–6. IEEE, 2007.

[Dit01] J. C. Dittmer. Consensus formation under bounded confidence. *Nonlinear Analysis: Theory, Methods & Applications*, 47(7):4615–4621, 2001.

[DNAW00] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98, 2000.

[DOK01] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *Robotics and Automation, IEEE Transactions on*, 17(6):905–908, 2001.

[DRE09] X.C. Ding, A. Rahmani, and M. Egerstedt. Optimal multi-UAV convoy protection,. In *ROBOCOMM '09. Second International Conference on Robot Communication and Coordination*, 2009.

[ESF08] J. Enright, K. Savla, and E. Frazzoli. Coverage control for nonholonomic agents. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4250–4256. IEEE, 2008.

[FBKT00] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous robots*, 8(3):325–344, 2000.

[FM01] J. A. Fax and R. M. Murray. Graph laplacians and stabilization of vehicle formations. 2001.

[FM04] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *Automatic Control, IEEE Transactions on*, 49(9):1465–1476, 2004.

[FS01] J. T. Feddema and D. Schoenwald. Decentralized control of cooperative robotic vehicles. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 136–146. International Society for Optics and Photonics, 2001.

[FXZ14] Y. Feng, S. Xu, and B. Zhang. Group consensus control for double-integrator dynamic multiagent systems with fixed communication topology. *International Journal of Robust and Nonlinear Control*, 24(3):532–547, 2014.

[GKES89] C. M. Gray, P. König, A. K. Engel, and W. Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 1989.

[GR13] C. Godsil and G. F. Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[HBA+13] L. Hernandez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, D. Chinarro, J. J. Gomez-Sanz, and D. Cook. A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants. *Communications Magazine, IEEE*, 51(1):106–113, 2013.

[HDTP13] L. Hetel, J. Daafouz, S. Tarbouriech, and C. Prieur. Stabilization of linear impulsive systems through a nearly-periodic reset. *Nonlinear Analysis: Hybrid Systems*, 7:4–15, 2013.

[HGJ03] M. N. Halgamuge, S. M. Guru, and A. Jennings. Energy efficient cluster formation in wireless sensor networks. In *10th International Conference on Telecommunications, 2003.*, 2003.

[HK02] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[HLM99] J. P. Hespanha, D. Liberzon, and A. S. Morse. Towards the supervisory control of uncertain nonholonomic systems. In *Proceedings of the American Control Conference*, pages 3520–3524, 1999.

[HT13] J. M. Hendrickx and J. N. Tsitsiklis. Convergence of type-symmetric and cut-

balanced consensus seeking systems. *Automatic Control, IEEE Transactions on*, 58(1):214–218, 2013.

[HWK+07] M. P. F. Hommelberg, C. J. Warmer, I. G. Kamphuis, J. K. Kok, and G. J. Schaeffer. Distributed control concepts using multi-agent technology and automatic markets: An indispensable feature of smart power grids. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–7. IEEE, 2007.

[JCR08] H. Jiayi, J. Chuanwen, and X. Rong. A review on distributed energy resources and microgrid. *Renewable and Sustainable Energy Reviews*, 12(9):2472–2483, 2008.

[JLM03] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.

[JN97] Z.-P. Jiang and H. Nijmeijer. Tracking control of mobile robots: a case study in backstepping. *Automatica*, 33(7):1393–1399, 1997.

[JWE97] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *Advanced Robotics, 1997. ICAR'97. Proceedings., 8th International Conference on*, pages 193–200. IEEE, 1997.

[KKMN90] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings of 1990 IEEE International Conference on Robotics and Automation*, 1990.

[KM95] I. Kolmanovsky and N. H. McClamroch. Developments in nonholonomic control problems. *Control Systems, IEEE*, 15(6):20–36, 1995.

[Kra97] U. Krause. Soziale dynamiken mit vielen interakteuren. eine problemskizze. *Modellierung und Simulation von Dynamiken mit vielen interagierenden Akteuren*, 3751:2, 1997.

[Kra00] U. Krause. A discrete nonlinear and non-autonomous model of consensus formation. *Communications in difference equations*, pages 227–236, 2000.

[KRS04] V. Kumar, D. Rus, and S. Singh. Robot and sensor networks for first responders. *Pervasive Computing, IEEE*, 3(4):24–33, 2004.

[Lee06] S.-H. Lee. Predator's attack-induced phase-like transition in prey flock. *Physics Letters A*, 357(4):270–274, 2006.

[LFPA06] W. Li, J. Farrell, S. Pang, and R. M. Arrieta. Moth-inspired chemical plume

tracing on an autonomous underwater vehicle. *Robotics, IEEE Transactions on*, 22(2):292–307, 2006.

[Lor05] J. Lorenz. A stabilization theorem for continuous opinions dynamics. *Physica A*, 355(1):217–233, 2005.

[Lor06] J. Lorenz. Consensus strikes back in the hegselmann-krause model of continuous opinion dynamics under bounded confidence. *Journal of Artificial Societies and Social Simulation*, 9(1), 2006.

[Lor07] J. Lorenz. Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C*, 18(12):1819–1838, 2007.

[LSLT01] T.-C. Lee, K.-T. Song, C.-H. Lee, and C.-C. Teng. Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *Control Systems Technology, IEEE Transactions on*, 9(2):305–318, 2001.

[LSW08] T. Logenthiran, D. Srinivasan, and D. Wong. Multi-agent coordination for DER in microgrid. In *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on*, pages 77–82. IEEE, 2008.

[ME10] M. Mesbahi and M. Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[MG11] I.-C. Morărescu and A. Girard. Opinion dynamics with decaying confidence: Application to community detection in graphs. *Automatic Control, IEEE Transactions on*, 56(8):1862–1873, 2011.

[MG13] S. Martin and A. Girard. Continuous-time consensus under persistent connectivity and slow divergence of reciprocal interaction weights. *SIAM Journal on Control and Optimization*, 51(3):2568–2584, 2013.

[MMG14] I.-C. Morarescu, S. Martin, and A. Girard. Coordination in networks of linear impulsive agents. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 3713–3718. IEEE, 2014.

[MNS95] M. J. Matarić, M. Nilsson, and K. T. Simsarin. Cooperative multi-robot box-pushing. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 556–561. IEEE, 1995.

[Mor05] L. Moreau. Stability of multiagent systems with time-dependent communication links. *Automatic Control, IEEE Transactions on*, 50(2):169–182, 2005.

[Nas] Nasa. Planet quest - the search of another earth. http://planetquest.jpl.nasa.gov/.

[NC10] W. Ni and D. Cheng. Leader-following consensus of multi-agent systems under fixed and switching topologies. *Systems & Control Letters*, 59(3):209–217, 2010.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[OK91] K. Okuda and Y. Kuramoto. Mutual entrainment between populations of coupled oscillators. *Progress of theoretical physics*, 1991.

[OS06] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 2006.

[OSFM07] R. Olfati-Saber, A. Fax, and R. M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[OSM04a] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.

[OSM04b] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49:1520–1533, 2004.

[OT09] A. Olshevsky and J. N Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.

[PBG+15] R. Postoyan, M. C. Bragagnolo, E. Galbrun, J. Daafouz, D. Nešić, and E. B. Castelan. Event-triggered tracking control of unicycle mobile robots. *Automatica*, 52:302–308, 2015.

[PFR09] M. Pipattanasomporn, H. Feroze, and S. Rahman. Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pages 1–8. IEEE, 2009.

[PJM10] A. Papachristodoulou, A. Jadbabaie, and U. Munz. Effects of delay in multi-agent consensus and oscillator synchronization. *Automatic Control, IEEE Transactions on*, 55(6):1471–1477, 2010.

[PLLN98] E. Panteley, E. Lefeber, A. Loria, and H. Nijmeijer. Exponential tracking control of a mobile car using a cascaded approach. In *Proceedings of the IFAC workshop on motion control*, 1998.

[RB+05] W. Ren, R. W. Beard, et al. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5):655–661, 2005.

[RDJ95] D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 235–242. IEEE, 1995.

[Rey87] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM Siggraph Computer Graphics*, 1987.

[RMD15] J. B. Rejeb, I.-C. Morarescu, and J. Daafouz. Event triggering strategies for consensus in clustered networks. In *14th European Control Conference, ECC'15*, 2015.

[RS08] W. Ren and N. Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.

[SB93] D. J. Stilwell and J. S. Bay. Toward the development of a material transport system using swarms of ant-like robots. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 766–771. IEEE, 1993.

[SBS06] S. Sariel, T. Balch, and J. Stack. Distributed multi-AUV coordination in naval mine countermeasure missions. Technical report, Georgia Institute of Technology, 2006.

[Sco87] J. Scott. *Social network analysis: a Handbook*. Sage Publications, 1987.

[SHP04] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen. A survey of spacecraft formation flying guidance and control. part II: Control. In *American Control Conference, 2004. Proceedings of the 2004*, volume 4, pages 2976–2985. IEEE, 2004.

[SJ11] G. Shi and K. H. Johansson. Towards a global agreement: the persistent graph. *arXiv preprint arXiv:1112.1338*, 2011.

[SPNW06] K. Sun, P. Peng, P. Ning, and C. Wang. Secure distributed cluster formation in wireless sensor networks. In *22nd Annual Computer Security Applications Conference.*, 2006.

[SPR77] G. B. Sholomitskii, O. F. Prilutskii, and V. G. Rodin. Infrared space interferometer. In *Prague International Astronautical Federation Congress*, volume 1, 1977.

[SS02] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 475–482. ACM, 2002.

[TL96] K.-H. Tan and M. A. Lewis. Virtual structures for high-precision cooperative mobile robotic control. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 1, pages 132–139. IEEE, 1996.

[TN11] B. Touri and A. Nedić. On existence of a quadratic comparison function for random weighted averaging dynamics and its implications. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 3806–3811. IEEE, 2011.

[TN14] B. Touri and A. Nedić. Product of random stochastic matrices. *Automatic Control, IEEE Transactions on*, 59(2):437–448, 2014.

[VCBJ+95] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 1995.

[WF94] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

[XLWJ09] G. Xie, H. Liu, L. Wang, and Y. Jia. Consensus in networked multi-agent systems via sampled control: fixed topology case. In *American Control Conference, 2009. ACC'09.*, pages 3902–3907. IEEE, 2009.

[XW05] G. Xie and L. Wang. Consensus control for a class of networks of dynamic agents: fixed topology. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 96–101. IEEE, 2005.

[YB02] B. Young and R. W. Beard. A decentralized approach to elementary formation maneuvers. *IEEE Transactions on Robotics and Automation, Princeton University*, 2002.

[YW09] J. Yu and L. Wang. Group consensus of multi-agent systems with undirected communication graphs. In *Asian Control Conference, 2009. ASCC 2009. 7th*, pages 105–110. IEEE, 2009.

[ZC12] H. Zhang and J. Chen. Bipartite consensus of general linear multi-agent systems. In *American Control Conference (ACC), 2014*, 808–812.

[ZC12] Z. Zhang and M.-Y. Chow. Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid. *Power Systems, IEEE Transactions on*, 27(4):1761–1768, 2012.

[ZFBZ08] C. Zhang, D. Florêncio, D. E. Ba, and Z. Zhang. Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *Multimedia, IEEE Transactions on*, 10(3):538–548, 2008.

Résumé

Les réseaux sont présents dans plusieurs domaines scientifiques et d'ingénierie tels que la biologie, la physique, la sociologie ainsi que la robotique ou la théorie de la communication. L'étude de ces réseaux montre qu'ils sont souvent structurés en sous-groupes. Entre eux il n'y a pas ou il y a très peu d'interaction. Par conséquent, un accord local au sein de chaque groupe est naturellement atteint alors que le consensus associé à un tel réseau doit être imposé par une loi de commande spécifique. Nous proposons donc un contrôleur discret quasi-périodique pour échanger des informations entre les groupes. Un agent de chaque groupe est choisi le « leader » et, à certains moments, ces leaders communiquent entre eux à travers un nouveau réseau. Ceci permet d'obtenir le consensus dans tout le réseau mais engendre des réinitialisation/sauts dans l'état de leaders.

La première contribution de la thèse est la caractérisation de la valeur de consensus dans le cadre des systèmes linéaires impulsifs. Il est remarquable que la valeur de consensus dépende seulement des conditions initiales et des topologies des réseaux impliqués. Elle n'est donc pas sensible aux instants de réinitialisation des états de leaders. Afin d'étudier la stabilité de la valeur de consensus obtenue, nous proposons une méthode fondée sur la vérification d'une condition LMI. Cela peut être adaptée pour la conception du réseau d'interaction entre les leaders permettant d'atteindre une valeur de consensus a priori choisie. Il est aussi possible d'utiliser la condition LMI afin de garantir une vitesse de convergence désirée vers le consensus. Pour ces derniers objectifs, la topologie du réseau continue à être considérée comme fixe et connue pour chaque groupe. L'ensemble des valeurs de consensus qui peuvent être atteintes est contenu dans l'intervalle défini par le minimum et le maximum des accords locaux initiaux. Ensuite nous présentons l'étude d'un problème pratique. Des robots mobiles non-holonome, séparés dans des groupes, doivent atteindre une formation donnée. L'algorithme de consensus à pour mission de définir les trajectoires de référence pour ces robots en prenant en compte juste les informations locale. Le robot poursuit la trajectoire de référence en utilisant une commande classique pour cela.

Mots-clés: systèmes multi-agents,commande décentralisée,théorie des graphes

Abstract

Networks appear in several areas of science and engineering such as biology, physics, sociology as well as robotics and communication theory. Studying these networks it is possible to see cluster-like structures, which are disconnected or very weakly connected one to another. The presence of these clusters hampers consensus throughout the overall network. Instead, local agreement is reached within each cluster. To enforce consensus we have to design an appropriate decentralized controller that imposes interactions between clusters. While the interactions inside each cluster are continuous, we propose a quasi-periodic discrete controller to exchange information between clusters. A single agent from each cluster is chosen to be the leader, and at certain moments, the leaders communicate with each other through a new network. This allows consensus in the entire network but generates resets/jumps on the leaders' state.

The first contribution of this manuscript is related to the characterization of the consensus value in the linear impulsive dynamics framework. It is noteworthy that the consensus value depends only on the initial conditions and the topologies of the involved networks. Therefore, the consensus value does not depend on the reset sequence used for the leaders' states. To study the stability of the consensus value a LMI based condition is proposed. The main advantage of this approach is its flexibility. Indeed with some modifications to the LMI condition it is possible to analyse the convergence speed of the network or to design the leaders' network. The purpose of leaders' network design is to reach an a priori specified consensus value with a specified convergence speed. Whatever is the objective, throughout the manuscript we consider that the network topology is fixed and known for each cluster. The set of consensus values that can be reached is restricted to the interval defined by the minimum and maximum initial local agreements. A last contribution is related to the application of the proposed methodology to a practical situation. We consider a fleet of non-holonomic mobile robots separated in clusters. The communication inside each cluster are secured and cheap while between clusters it is expensive and not securely to communicate. Nevertheless the robots have to reach a given formation. In this case our consensus algorithm is in charge of providing reference trajectories to each robot by using only the available local information. The robot follows the reference by using a classical trajectory tracking control.

Keywords: multiagent systems, decentralized control, graph theory