Active perception and localization for multi-robot systems

Nicola De Carli

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *AST : Automatique, Productique et Robotique - Signal, Image, Vision - Télécommunications*

Par
Nicola DE CARLI

Active Perception and Localization for Multi-Robot Systems

Thèse présentée et soutenue à Rennes, le 18/04/2024
Unité de recherche : IRISA UMR6074

Rapporteurs avant soutenance :

Dimos V. DIMAROGONAS — Professeur, KTH Royal Institute of Technology
Eduardo MONTIJANO — Professeur associé, Universidad de Zaragoza

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Examinateurs :	Fabio MORBIDI	Maître de Conférences, MIS laboratory, Université de Picardie Jules Verne
	Isabelle FANTONI	Directrice de Recherche CNRS, LS2N Nantes
	Dimos V. DIMAROGONAS	Professeur, KTH Royal Institute of Technology
	Eduardo MONTIJANO	Professeur associé, Universidad de Zaragoza
Dir. de thèse :	Paolo ROBUFFO GIORDANO	Directeur de recherche CNRS, IRISA/INRIA Rennes
Co-dir. de thèse :	Paolo SALARIS	Professeur, associé, Centro Piaggio et Università di Pisa

ACKNOWLEDGEMENT

As this doctoral journey draws to a close, reflecting on the experience brings forth a mix of challenges and cherished moments.

First and foremost, I extend my heartfelt gratitude to my Ph.D. advisor, Dr. Paolo Robuffo Giordano, whose unwavering support and encouragement propelled me forward. I am deeply grateful for his mentorship, respect for his students, and for always been ready to share convivial moments with his students beyond academia. I also extend my thanks to Prof. Paolo Salaris for kindly hosting me at Centro Piaggio for six months and for his advices.

Moreover, I extend my gratitude to Dr. Marco Tognon for the collaborative opportunities and co-supervision of students. His arrival has not only facilitated knowledge exchange but also instilled inspiring motivation within the team. His support and engagement have significantly contributed to the growth of the drone group within the Rainbow team.

I would like to extend special thanks to Dr. Esteban Restrepo for his recent collaborations and engaging discussions on nonlinear control, which have enriched my understanding of the field.

My sincere appreciation also goes to the master students I supervised, Lorenzo Balandi and Riccardo Belletti, as well as the Ph.D. student I worked with, Francesca Pagano, for their dedication and contributions to our shared projects. Their enthusiasm and commitment have been invaluable to our collaborative efforts.

My journey was enriched by the vibrant camaraderie of my colleagues and friends at the Rainbow team (including visiting students), whose companionship extended far beyond academic pursuits. Our shared experiences, both within and outside the laboratory, were invaluable, with countless memorable moments, especially at our "unofficial headquarters", Tiffany's. While it is impossible to name everyone and, surely, I will forget someone, I want to express my appreciation to those who especially made these three years extraordinary: Anto, John, Alex, Albi, Max B., Max R., Thibault, Elodie, Joudy, Pierre, Lev, Ali, Salva, Tommà, Petit Monsieur, Marco F., Massi, Loré, Francesca, Danilo, Riccardo.

I wish to acknowledge Prof. Dimos Dimarogonas and Prof. Eduardo Montijano who

kindly accepted to review this manuscript. An acknowledgement also goes to the other members of my jury, Dr. Isabelle Fantoni and Prof. Fabio Morbidi, for their availability to attend my defence.

To my lifelong friends, whether in Genova or scattered across the globe, your enduring support has been a source of strength and joy.

Lastly, I reserve a special place in my heart for my parents, my sister Giulia and 'nonno'. Their unconditional love and unwavering support have been the cornerstone of my journey. Thank you, from the bottom of my heart.

Table of Contents

INTRODUCTION

The field of robotics, as defined in [Siciliano et al., 2009], revolves around the intelligent interplay between perception and action. The accuracy of a robot actions is profoundly influenced by the precision with which it estimates its state relative to the surrounding environment based on available measurements. Moreover, given that robots are inherently nonlinear systems in practical scenarios, the converse is also true, actions affect perception, as demonstrated in both natural beings [Gibson, 1962] and robotic systems [Bajcsy, 1988].

Perception, rather than being a passive reception of information, is an active and exploratory process characterized by probing and searching, as opposed to merely waiting for relevant information to be captured by sensors. This realization forms the basis of the concept of *active sensing*. In the realm of robotics, active sensing involves control strategies applied to the data acquisition process, which adapt based on the current state of data interpretation and the main goal or task of the system, as articulated in [Bajcsy, 1988]. Another perspective, offered in [Sontag et al., 2022], describes active sensing as the purposeful expenditure of energy, often through movement, to enhance sensing capabilities.

Building upon this notion, the theory discussed in [Sontag et al., 2022] suggests that active sensing is, at least in part, a response to the demands of nonlinear state estimation. The hypothesis is that animals, through active sensing, generate time-varying motor commands that continuously stimulate their sensory receptors. This dynamic stimulation allows the system states to be estimated with satisfactory error bounds from the sensor measurements. In essence, these movements are designed to maintain the *observability* of the system, providing valuable insights into the intricate relationship between motion, perception, and state estimation in both natural and robotic systems.

In many sensor-based robot applications, the state of a robot with respect to the environment is only partially accessible through on-board sensors. For instance, a robot may be equipped with a sensor able to measure only a certain function of the state, such as relative distance or direction (i.e., the *bearing*) with respect to known landmarks. State estimation schemes become essential in such scenarios, enabling the online recovery of 'missing information', subsequently employed by planners/motion controllers as substi-

tutes for unmeasurable states. However, in addressing non-trivial cases, state estimation must cope with nonlinear sensor mappings from the observed environment to the sensor space. The impact of these nonlinearities on estimation convergence and accuracy is significant, with the trajectory followed by the robot/sensor playing a crucial role in ensuring a sufficient level of *excitation* during motion.

We also wish to highlight the insightful observations made in [Sontag et al., 2022]. In conventional control systems engineering, a prevalent paradigm involves designing state feedback and state estimation as independent components. This separation principle, for linear plants affected by Gaussian noise, dictates that separately designing state estimation and task-level control design is not only effective but it is the optimal solution. However, as it is well known, the separation principle does not apply to general nonlinear systems, for this reason active sensing adopts the opposite approach. Here, control inputs are intentionally crafted to 'excite' the system, aligning with the needs of the state estimator. This departure from the separation principle highlights the unique and purposeful integration of perception and action, providing a novel perspective on the interplay between motion control and active sensing.

A specific scenario requiring robots to extract accurate information from partial measurements, given by onboard sensors, arises when multiple robots need to *cooperate* and, consequently, be localized relative to each other with high precision. The past decades have witnessed a surge in interest in multi-robot applications, spanning cooperative manipulation, mapping, surveillance, disaster response, exploration, border security, and patrol missions [Aggravi et al., 2021; Murphy et al., 2011; Schranz et al., 2020]. To accomplish such tasks, a crucial aspect is each robot ability to sense and communicate with a limited subset of the entire group, commonly referred to as the *neighbors* of the considered robot. Robots should localize themselves with respect to their neighbors and make decisions based solely on local information. *Distributed* solutions have a crucial role in multi-robot (and, more broadly, *multi-agent*) applications, allowing for scalable algorithms in terms of computational and communicational loads, particularly concerning the group size.

In summary, it is desirable for a group of robots to produce actions involving information about their state relative to other robots in the group rather than to a global reference frame. Examples of relative measurements include relative distances, bearings, or positions, which can be directly obtained from onboard sensors such as cameras and ultra-wideband sensors. Employing such sensors frees from the need for centralized local-

ization systems like motion capture or global positioning systems, which is particularly relevant in all those scenarios where these systems are unavailable, such as indoor, underwater, or deep-space missions.

In scenarios where robots collaboratively localize themselves within a shared (implicit) frame based on partial measurements of the relative state, the trajectory followed by the robots significantly influences estimation convergence. This impact becomes more pronounced in practical situations where *sensing limitations*, such as limited range and field of view, are present. Therefore, a key motivation driving this thesis is the formulation and development of distributed active sensing control strategies tailored to the localization of multi-robot systems. These strategies aim to address the challenges posed by real-world sensing constraints, thereby enhancing the accuracy and efficiency of the cooperative localization process while concurrently achieving primary mission objectives.

Topic of the thesis

In light of these considerations, this thesis aims at advancing the state-of-the-art in sensor based cooperative control and (active and not) cooperative localization for multi-robot systems. We address the design of fully distributed active sensing control laws, specifically tackling the challenges posed by cooperative localization based on relative bearing and distance measurements, as well as target tracking using multi-robot systems. The interplay between active perception/localization and action/control to achieve a specific task is "mediated" using a quadratic program formulation and the *control barrier functions* and *control Lyapunov functions* formalism.

Control barrier functions (CBFs) recently emerged as a valuable alternative to constraint-handling control design methods such as classical optimal control, model predictive control (MPC) [Rawlings et al., 2017], or barrier Lyapunov functions (BLFs) [Tee et al., 2009]. Despite the significant progress made in recent years on control barrier functions [Ames et al., 2019; Garg et al., 2023], with contributions extending to their application in multi-agent systems [Lindemann & Dimarogonas, 2020; Tan & Dimarogonas, 2021; Wang et al., 2017], certain limitations persist, particularly in the context of multi-agent systems. This thesis tackles one such limitation, aiming to enhance the versatility of distributed CBFs by extending their application to a broader class of constraints.

Moreover, we delve into the challenges related to the observer design for cooperative localization of multi-robot systems in a common frame, specifically focusing on relative

body-frame position measurements. This involves the development of an adaptive observer designed to extract relative orientations among the robots. Estimating the robots state in a common frame allows the robots to exchange information initially captured in their local frames and to effectively coordinate to achieve a common task.

Structure of the thesis

This thesis is organized into five main parts. Part I provides essential preliminaries on nonlinear observability, multi-robot systems, and control barrier functions, laying the foundations for a comprehensive understanding of the subsequent chapters. The original contributions of this thesis are then illustrated over Part II, Part III, and Part IV, with each Part dedicated to a specific contribution. Finally, Part V draws the conclusions and outlines potential future research directions. Below is a summary of each part.

Outline of Part I

This part comprises three chapters that delve into fundamental concepts crucial for comprehending the core of the thesis:

- **Chapter 1** introduces relevant concepts related to nonlinear observability, observability measures, and their application in active sensing control.
- **Chapter 2** covers basic concepts of multi-robot systems, spanning graph theory, static and dynamic consensus, and the challenges of cooperative localization and infinitesimal rigidity.
- **Chapter 3** provides an introduction to control barrier functions (CBFs) and control Lyapunov functions (CLFs), along with the CBF-CLF-QP formulation which is at the basis of the subsequent contributions. The chapter then explores extensions for high-order systems and distributed systems.

Outline of Part II

This part includes a single chapter (**Chapter 4**) covering works on trajectory optimization for active sensing control in cooperative localization in presence of sensing limitations.

Outline of Part III

This part comprises three chapters exploring the integration of active sensing with a desired task through QP-based formulations using CBFs and CLFs along with a contribution on the implementation of distributed CBFs:

- **Chapter 5** presents an active sensing control strategy for cooperative localization from relative bearing measurements, integrating an additional higher-priority task through CLFs as well as a connectivity maintenance constraint through CBFs.
- **Chapter 6** introduces an active sensing control strategy for the localization of multiple moving targets using flying robots with down-looking cameras. The strategy imposes a constraint on acquiring a minimum prescribed level of information (required for the convergence of the employed observer) while maximizing the achievement of another task.
- **Chapter 7** outlines an algorithm for distributed control barrier functions converging to centralized optimality. The algorithm is then applied to the constraint of connectivity maintenance in the presence of limited range and field of view.

Outline of Part IV

This part features a single chapter (**Chapter 8**) presenting an observer for cooperative localization in a common frame based on body-frame position measurements of a group of robots.

List of Publications

Published articles

1. N. De Carli, P. Salaris, and P. Robuffo Giordano. "Online Decentralized perception-aware path planning for multi-robot systems." 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). IEEE, 2021.
2. N. De Carli, P. Salaris, and P. Robuffo Giordano. "Multi-Robot Active Sensing for Bearing Formations." 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). IEEE, 2023.

3. L. Balandi, N. De Carli, and P. Robuffo Giordano. "Persistent Monitoring of Multiple Moving Targets Using High Order Control Barrier Functions." IEEE Robotics and Automation Letters (2023).

4. N. De Carli, P. Salaris, and P. Robuffo Giordano. "Distributed Control Barrier Functions for Global Connectivity Maintenance." ICRA 2024-IEEE International Conference on Robotics and Automation. IEEE, 2024.

5. N. De Carli, E. Restrepo, and P. Robuffo Giordano. "Adaptive Observer from Body-Frame Relative Position Measurements for Cooperative Localization." IEEE Control Systems Letters (2024).

Articles in preparation

6 N. De Carli, P. Salaris, and P. Robuffo Giordano. "Bearing-based multi-robot control: beyond infinitesimal rigidity." IEEE Robotics and Automation Letters (2024). The work is presented in Chapter 5.

Résumé étendu en Français

Le domaine de la robotique, tel que défini par [Siciliano et al., 2009], repose sur l'interaction intelligente entre perception et action. La précision des actions d'un robot est profondément influencée par la précision avec laquelle il estime son état par rapport à l'environnement en se basant sur les mesures disponibles. De plus, étant donné que les robots sont intrinsèquement des systèmes non linéaires dans des scénarios pratiques, l'inverse est également vrai : les actions influencent la perception, comme le montrent tant les êtres naturels [Gibson, 1962] que les systèmes robotiques [Bajcsy, 1988].

La perception, plutôt que d'être une réception passive d'informations, est un processus actif et exploratoire caractérisé par des sondages et des recherches, par opposition à une simple attente de capture d'informations pertinentes par les capteurs. Cette réalisation forme la base du concept de *sensation active*. Dans le domaine de la robotique, la sensation active implique des stratégies de contrôle appliquées au processus d'acquisition de données, qui s'adaptent en fonction de l'état actuel de l'interprétation des données et de l'objectif principal du système, comme articulé dans [Bajcsy, 1988]. Une autre perspective, offerte dans [Sontag et al., 2022], décrit la sensation active comme la dépense intentionnelle d'énergie, souvent par le mouvement, pour améliorer les capacités de détection.

S'appuyant sur cette notion, la théorie discutée dans [Sontag et al., 2022] suggère que la sensation active est, au moins en partie, une réponse aux exigences de l'estimation d'état non linéaire. L'hypothèse est que les animaux, par le biais de la sensation active, génèrent des commandes motrices variables dans le temps qui stimulent continuellement leurs récepteurs sensoriels. Cette stimulation dynamique permet d'estimer les états du système avec des marges d'erreur satisfaisantes à partir des mesures des capteurs. En essence, ces mouvements sont conçus pour maintenir l'*observabilité* du système, fournissant des informations précieuses sur la relation complexe entre mouvement, perception et estimation d'état dans les systèmes naturels et robotiques.

Dans de nombreuses applications robotiques basées sur des capteurs, l'état d'un robot par rapport à l'environnement n'est que partiellement accessible par les capteurs embarqués. Par exemple, un robot peut être équipé d'un capteur capable de mesurer uniquement une certaine fonction de l'état, telle que la distance relative ou la direction (c'est-

à-dire, le *relèvement*) par rapport à des repères connus. Les schémas d'estimation d'état deviennent essentiels dans de tels scénarios, permettant la récupération en ligne des « informations manquantes », ensuite utilisées par les planificateurs/contrôleurs de mouvement comme substituts pour les états non mesurables. Cependant, pour aborder des cas non triviaux, l'estimation d'état doit faire face à des mappages non linéaires de capteurs de l'environnement observé à l'espace des capteurs. L'impact de ces non-linéarités sur la convergence et la précision de l'estimation est significatif, la trajectoire suivie par le robot/capteur jouant un rôle crucial pour garantir un niveau suffisant d'*excitation* pendant le mouvement.

Nous souhaitons également souligner les observations perspicaces faites dans [Sontag et al., 2022]. Dans le génie des systèmes de contrôle conventionnel, un paradigme prévalent implique la conception de la rétroaction d'état et de l'estimation d'état comme des composants indépendants. Ce principe de séparation, pour les plantes linéaires affectées par le bruit gaussien, dicte que concevoir séparément l'estimation d'état et la conception de contrôle au niveau de la tâche n'est pas seulement efficace, mais aussi optimal. Cependant, comme il est bien connu, le principe de séparation ne s'applique pas aux systèmes non linéaires généraux, c'est pourquoi la sensation active adopte l'approche opposée. Ici, les entrées de contrôle sont intentionnellement conçues pour « exciter » le système, en alignement avec les besoins de l'estimateur d'état. Cet écart par rapport au principe de séparation met en évidence l'intégration unique et intentionnelle de la perception et de l'action, offrant une nouvelle perspective sur l'interaction entre le contrôle de mouvement et la sensation active.

Un scénario spécifique nécessitant que les robots extraient des informations précises à partir de mesures partielles, fournies par des capteurs embarqués, se présente lorsque plusieurs robots doivent *coopérer* et, par conséquent, être localisés les uns par rapport aux autres avec une grande précision. Les dernières décennies ont connu un intérêt croissant pour les applications multi-robots, couvrant la manipulation coopérative, la cartographie, la surveillance, la réponse aux catastrophes, l'exploration, la sécurité des frontières et les missions de patrouille [Aggravi et al., 2021; Murphy et al., 2011; Schranz et al., 2020]. Pour accomplir ces tâches, un aspect crucial est la capacité de chaque robot à détecter et à communiquer avec un sous-ensemble limité du groupe entier, communément appelé les *voisins* du robot considéré. Les robots doivent se localiser par rapport à leurs voisins et prendre des décisions basées uniquement sur des informations locales. Les solutions *distribuées* jouent un rôle crucial dans les applications multi-robots (et, plus largement,

multi-agents), permettant des algorithmes évolutifs en termes de charges computationnelles et de communication, en particulier en ce qui concerne la taille du groupe.

En résumé, il est souhaitable qu'un groupe de robots produise des actions impliquant des informations sur leur état relatif par rapport aux autres robots du groupe plutôt que par rapport à un cadre de référence global. Des exemples de mesures relatives incluent les distances relatives, les relèvements ou les positions, qui peuvent être directement obtenus à partir de capteurs embarqués tels que les caméras et les capteurs ultra large bande. L'utilisation de ces capteurs dispense de la nécessité de systèmes de localisation centralisés comme la capture de mouvement ou les systèmes de positionnement global, ce qui est particulièrement pertinent dans tous les scénarios où ces systèmes ne sont pas disponibles, tels que les missions en intérieur, sous-marines ou dans l'espace profond.

Dans des scénarios où les robots se localisent de manière collaborative dans un cadre partagé (implicite) basé sur des mesures partielles de l'état relatif, la trajectoire suivie par les robots influence significativement la convergence de l'estimation. Cet impact devient plus prononcé dans des situations pratiques où des *limitations de détection*, telles que la portée limitée et le champ de vision, sont présentes. Par conséquent, une motivation clé de cette thèse est la formulation et le développement de stratégies de contrôle de la sensation active distribuée adaptées à la localisation des systèmes multi-robots. Ces stratégies visent à relever les défis posés par les contraintes de détection réelles, améliorant ainsi la précision et l'efficacité du processus de localisation coopérative tout en atteignant simultanément les objectifs principaux de la mission.

Sujet de la thèse

À la lumière de ces considérations, cette thèse vise à faire progresser l'état de l'art dans le contrôle coopératif basé sur des capteurs et la localisation coopérative (active et non active) pour les systèmes multi-robots. Nous abordons la conception de lois de contrôle de la sensation active entièrement distribuées, en traitant spécifiquement les défis posés par la localisation coopérative basée sur des mesures de relèvement et de distance relatives, ainsi que par le suivi de cibles utilisant des systèmes multi-robots. L'interaction entre la perception/localisation active et l'action/contrôle pour accomplir une tâche spécifique est "médiée" par une formulation de programme quadratique et le formalisme des *fonctions de barrière de contrôle* et des *fonctions de Lyapunov de contrôle*.

Les fonctions de barrière de contrôle (CBF) ont récemment émergé comme une alter-

native précieuse aux méthodes de conception de contrôle gérant les contraintes telles que le contrôle optimal classique, le contrôle prédictif basé sur des modèles (MPC) [Rawlings et al., 2017], ou les fonctions de Lyapunov de barrière (BLF) [Tee et al., 2009]. Malgré les progrès significatifs réalisés ces dernières années sur les fonctions de barrière de contrôle [Ames et al., 2019; Garg et al., 2023], avec des contributions s'étendant à leur application dans les systèmes multi-agents [Lindemann & Dimarogonas, 2020; Tan & Dimarogonas, 2021; Wang et al., 2017], certaines limitations persistent, en particulier dans le contexte des systèmes multi-agents. Cette thèse s'attaque à l'une de ces limitations, visant à améliorer la polyvalence des CBF distribuées en étendant leur application à une classe plus large de contraintes.

De plus, nous nous penchons sur les défis liés à la conception d'observateurs pour la localisation coopérative des systèmes multi-robots dans un cadre commun, en nous concentrant spécifiquement sur les mesures de position relative dans le *cadre du corps*. Cela implique le développement d'un observateur adaptatif conçu pour extraire les orientations relatives entre les robots. Estimer l'état des robots dans un cadre commun permet aux robots d'échanger des informations initialement capturées dans leurs cadres locaux et de se coordonner efficacement pour accomplir une tâche commune.

Structure de la thèse

Cette thèse est organisée en cinq parties principales. La Partie I fournit des préliminaires essentiels sur l'observabilité non linéaire, les systèmes multi-robots et les fonctions de barrière de contrôle, posant les bases pour une compréhension complète des chapitres suivants. Les contributions originales de cette thèse sont ensuite illustrées dans la Partie II, la Partie III, et la Partie IV, chaque Partie étant dédiée à une contribution spécifique. Enfin, la Partie V tire les conclusions et expose les directions potentielles de recherches futures. Voici un résumé de chaque partie.

Aperçu de la Partie I

Cette partie comprend trois chapitres qui approfondissent les concepts fondamentaux cruciaux pour comprendre le cœur de la thèse :

- **Chapitre 1** introduit les concepts pertinents liés à l'observabilité non linéaire, aux mesures d'observabilité et à leur application dans le contrôle de la sensation active.

- **Chapitre 2** couvre les concepts de base des systèmes multi-robots, englobant la théorie des graphes, le consensus statique et dynamique, ainsi que les défis de la localisation coopérative et de la rigidité infinitésimale.

- **Chapitre 3** fournit une introduction aux fonctions de barrière de contrôle (CBFs) et aux fonctions de Lyapunov de contrôle (CLFs), ainsi qu'à la formulation CBF-CLF-QP qui est à la base des contributions suivantes. Le chapitre explore ensuite des extensions pour les systèmes de haut ordre et les systèmes distribués.

Aperçu de la Partie II

Cette partie comprend un seul chapitre (**Chapitre 4**) couvrant les travaux sur l'optimisation de trajectoire pour le contrôle de la sensation active dans la localisation coopérative en présence de limitations de détection.

Aperçu de la Partie III

Cette partie comprend trois chapitres explorant l'intégration de la sensation active avec une tâche souhaitée par des formulations basées sur QP utilisant des CBFs et des CLFs ainsi qu'une contribution sur la mise en œuvre des CBFs distribuées :

- **Chapitre 5** présente une stratégie de contrôle de la sensation active pour la localisation coopérative à partir de mesures de relèvement relatives, intégrant une tâche de priorité supérieure par le biais des CLFs ainsi qu'une contrainte de maintien de la connectivité par le biais des CBFs.

- **Chapitre 6** introduit une stratégie de contrôle de la sensation active pour la localisation de multiples cibles mobiles utilisant des robots volants équipés de caméras orientées vers le bas. La stratégie impose une contrainte d'acquisition d'un niveau minimum d'information prescrit (nécessaire à la convergence de l'observateur employé) tout en maximisant l'accomplissement d'une autre tâche.

- **Chapitre 7** expose un algorithme pour les fonctions de barrière de contrôle distribuées convergeant vers l'optimalité centralisée. L'algorithme est ensuite appliqué à la contrainte de maintien de la connectivité en présence de portée et de champ de vision limités.

Aperçu de la Partie IV

Cette partie comporte un seul chapitre (**Chapitre 8**) présentant un observateur pour la localisation coopérative dans un cadre commun basé sur des mesures de position dans le cadre du corps d'un groupe de robots.

Part I

Preliminaries

Chapter 1

Observability Measures

This chapter serves as a comprehensive overview of nonlinear observability, observability measures, and their role in active sensing. We start this overview by introducing two pivotal concepts: the *observability matrix* and the *observability rank condition*. The observability rank condition serves as a decisive criterion, determining the system observability with a binary outcome – a system is either observable or not. It is crucial to note that while this criterion provides a definitive answer regarding observability, it does not quantify the degree of observability, i.e., how 'close' a system is to being unobservable.

Subsequently, we delve into the interconnection between the observability matrix and the resultant estimation uncertainty derived from a least squares formulation of the state estimation problem. Additionally, we aim to establish a quantitative measure to assess the degree of observability inherent within a system.

Moving forward, we introduce two fundamental tools: the *Observability Gramian* (OG) and the *Constructability* (also known as *Determinability*) *Gramian* (CG). Both gramians are instrumental in quantifying the degree of local observability or unobservability within a system. We then explore their intrinsic relationship with the *Fisher Information Matrix* (FIM), which is widely used in Bayesian estimation and statistics.

Finally, we discuss the use of information measures, with particular emphasis on those based on the OG, for active sensing problems, and we review the existing literature.

Throughout the thesis, we will make extensive use of the OG to quantify the observability of the system. In particular, we will consider the observability of the relative positions between robots for cooperative localization as well as target tracking applications. In these tasks, the OG plays a crucial role in shaping robot trajectories to gather sufficient information for localization purposes.

In the following, we establish various connections with related concepts which, although not strictly needed for the core understanding of the thesis, can offer an interesting and comprehensive perspective to the reader. Key references that have significantly contributed to the development of these topics include [P. Bernard et al., 2022; Besançon,

2007; Hinson, 2014; Krener & Ide, 2009].

1.1 Nonlinear observability and the observability rank condition

This section considers the topic of nonlinear observability and introduces the crucial, concept of the observability rank condition.

Consider a nonlinear control-affine dynamical system:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{f}_i(\boldsymbol{x}) u_i \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \end{aligned} \tag{1.1}$$

with state $\boldsymbol{x} \in \mathbb{R}^d$, input $\boldsymbol{u} = \begin{bmatrix} u_1 & \dots & u_m \end{bmatrix}^T \in \mathbb{R}^m$ and measured output $\boldsymbol{y} \in \mathbb{R}^p$. Here, $\boldsymbol{f}_0$ is the drift of the system while the $\boldsymbol{f}_i$ with $i \in \{1, ..., m\}$ are the control vector fields. The system is observable over the interval $[0, T]$ if the mapping from the initial state $\boldsymbol{x}_0$ to the output trajectory $\boldsymbol{y}(0:T)$ is one to one, where we used the notation $\boldsymbol{y}(0:T)$ to indicate the mapping $t \to \boldsymbol{y}(t)$ for $0 \leq t < T$. It is *locally* observable over the interval $[0, T]$ if this mapping is locally one to one. We indicate the Lie derivative of $\boldsymbol{h}(\boldsymbol{x})$ along the flow of the vector field $\boldsymbol{f}_i$ as:

$$L_{\boldsymbol{f}_i} \boldsymbol{h}(\boldsymbol{x}) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \boldsymbol{f}_i(\boldsymbol{x}). \tag{1.2}$$

This is the familiar notion of the derivative of $\boldsymbol{h}$ along the trajectories of the system $\dot{\boldsymbol{x}} = \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{f}_i(\boldsymbol{x}) u_i$, i.e.:

$$\dot{\boldsymbol{y}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = L_{\boldsymbol{f}_0} \boldsymbol{h}(\boldsymbol{x}) + \sum_{i=1}^{m} L_{\boldsymbol{f}_i} \boldsymbol{h}(\boldsymbol{x}) u_i. \tag{1.3}$$

The notation when repeating the computation of the derivative with respect to the

same vector field or a new one is the following [Khalil, 2014]:

$$\begin{aligned} L_{f_i f_j}\boldsymbol{h}(\boldsymbol{x}) &= \frac{\partial L_{\boldsymbol{f}_i}\boldsymbol{h}}{\partial \boldsymbol{x}}\boldsymbol{f}_j(\boldsymbol{x}) \\ L^2_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x}) &= L_{\boldsymbol{f}_i}L_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial L_{\boldsymbol{f}_i}\boldsymbol{h}}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) \\ &\vdots \\ L^k_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x}) &= L_{\boldsymbol{f}_i}L^{k-1}_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial L^{k-1}_{\boldsymbol{f}_i}\boldsymbol{h}}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) \end{aligned} \tag{1.4}$$

The Lie derivatives of the output $\boldsymbol{y}$ provide the mapping to the observation space defined by:

$$\begin{aligned} &\boldsymbol{\mathcal{O}}(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}, ..., \boldsymbol{u}^{(n-r-1)}) \\ &= \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}) \\ L_{\boldsymbol{f}_0}\boldsymbol{h}(\boldsymbol{x}) + \sum_{i=1}^m L_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x})u_i \\ L^2_{\boldsymbol{f}_0}\boldsymbol{h}(\boldsymbol{x}) + L_{\boldsymbol{f}_0}\left(\sum_{i=1}^m L_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x})u_i\right) + \sum_{i=1}^m L_{\boldsymbol{f}_i}\left(L_{\boldsymbol{f}_0}\boldsymbol{h}(\boldsymbol{x}) + \sum_{i=1}^m L_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x})u_i\right)u_i + \sum_{i=1}^m L_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x})\dot{u}_i \\ \vdots \end{bmatrix} \end{aligned} \tag{1.5}$$

where derivatives up to order $d-1$ are considered and r is the relative degree of the output. Let us define the $(k+1)p \times d$ Jacobian:

$$\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \\ \frac{\partial\left(L_{\boldsymbol{f}_0}\boldsymbol{h}(\boldsymbol{x}) + \sum_{i=1}^m L_{\boldsymbol{f}_i}\boldsymbol{h}(\boldsymbol{x})u_i\right)}{\partial \boldsymbol{x}} \\ \vdots \end{bmatrix} \tag{1.6}$$

which will be referred to as the *observability matrix*. We now give some definitions which are instrumental to the statement of a fundamental theorem for nonlinear observability [Besançon, 2007].

Definition 1.1.1 (Indistinguishability)**.** *A pair* $(\boldsymbol{x}_a, \boldsymbol{x}_b) \in \mathbb{R}^d \times \mathbb{R}^d$ *is indistinguishable for a system* (1.1) *if*

$$\forall \boldsymbol{u} \in \mathbb{R}^m, \quad \forall t \geq 0, \quad \boldsymbol{h}(\boldsymbol{\phi}(t, \boldsymbol{x}_a)) = \boldsymbol{h}(\boldsymbol{\phi}(t, \boldsymbol{x}_b)) \tag{1.7}$$

where we denote as $\boldsymbol{\phi}(t, \boldsymbol{x}_0)$ *the solution of the system dynamics when starting from initial condition* $\boldsymbol{x}_0$*.*

Definition 1.1.2 (Weak observability)**.** *System* (1.1) *is locally weakly observable at* $\boldsymbol{x}_0$ *if there exists a neighborhood* U *of* $\boldsymbol{x}_0$ *such that for any neighborhood* V *of* $\boldsymbol{x}_0$ *contained in*

U, there is no indistinguishable state from $\boldsymbol{x}_0$ in V when considering time intervals for which trajectories remain in V.

Definition 1.1.3 (Observability rank condition)**.** *A system* (1.1) *is said to satisfy the observability rank condition at $\boldsymbol{x}_0$ if*

$$rank\left(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}_0}(\boldsymbol{x})\right) = d. \tag{1.8}$$

Theorem 1. *A system satisfying the observability rank condition at $\boldsymbol{x}_0$ is locally weakly differentially observable.*

This means that locally the mapping from V to $\boldsymbol{\mathcal{O}}$ is injective, i.e. from output and inputs along with their derivatives we can locally uniquely retrieve the state.

In the case of a linear system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$, (1.8) becomes the classical observability condition:

$$\text{rank}\left(\begin{bmatrix}\boldsymbol{C} & \boldsymbol{C}\boldsymbol{A} & \dots & \boldsymbol{C}\boldsymbol{A}^{d-1}\end{bmatrix}\right) = d. \tag{1.9}$$

Notice that, in this case, observability of a system is independent from the state trajectory and inputs applied. However, the same is not true in the nonlinear case, where based on the current state and inputs, condition (1.8) may or may not be satisfied and certain inputs may destroy observability. This can also be seen by the state dependence of the control vector fields in (1.1). The benefit of the observability rank condition is that it explicitly accounts for this coupling and provides information about what control actuation, if any, is required to obtain system observability.

The dependency of the observability of a state on the system trajectory and applied inputs is a main motivation for the field of *active sensing*, which aims at actively selecting the inputs of a nonlinear system to follow informative/observable trajectories which maximize the system observability.

1.1.1 Relationship between nonlinear observability and estimation uncertainty

This section, adapted primarily from [Hinson, 2014], explores the relation between the observability rank condition (1.8) and least squares estimation techniques.

Assuming the observability rank condition holds, with access to the output and its derivatives, state estimation can be framed as a nonlinear least squares problem. Consider

the vector containing the output and its derivatives

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y} \\ \dot{\boldsymbol{y}} \\ \vdots \\ \boldsymbol{y}^{(d-1)} \end{bmatrix} = \boldsymbol{\mathcal{O}}(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}, ..., \boldsymbol{u}^{(n-r-1)}). \tag{1.10}$$

The residual $\boldsymbol{r}$ between $\boldsymbol{Y}$ and the observation vector obtained using an estimated state $\hat{\boldsymbol{x}}$ is

$$\boldsymbol{r} = \boldsymbol{Y} - \boldsymbol{\mathcal{O}}(\hat{\boldsymbol{x}}, \boldsymbol{u}, \dot{\boldsymbol{u}}, ..., \boldsymbol{u}^{(n-r-1)}) \tag{1.11}$$

with $\hat{\boldsymbol{x}}$ being the state estimate. Consider a first-order Taylor series expansion of the output derivatives around the point $\hat{\boldsymbol{x}}$:

$$\boldsymbol{Y} \approx \boldsymbol{\mathcal{O}}(\hat{\boldsymbol{x}}, \boldsymbol{u}, \dot{\boldsymbol{u}}, ..., \boldsymbol{u}^{(n-r-1)}) + \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}(\hat{\boldsymbol{x}})\Delta \boldsymbol{x} \tag{1.12}$$

Considering measurements $\boldsymbol{Y}$ with covariance $\boldsymbol{R}$, one can formulate a weighted least squares problem as

$$\min_{\Delta \boldsymbol{x}} . \quad \frac{1}{2} \left\| \boldsymbol{r} - \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}(\hat{\boldsymbol{x}})\Delta \boldsymbol{x} \right\|_2^2, \tag{1.13}$$

whose solution is given by

$$\Delta \boldsymbol{x} = \left(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}^T \boldsymbol{R}^{-1} \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}} \right)^{-1} \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}^T \boldsymbol{R}^{-1} \boldsymbol{r}. \tag{1.14}$$

Under the assumption of independent and identically distributed noise (i.i.d.), the output covariance simplifies to $\boldsymbol{R} = \sigma \boldsymbol{I}$ leading to the estimation covariance matrix $\boldsymbol{P}$

$$\boldsymbol{P} = \left(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}^T \boldsymbol{R}^{-1} \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}} \right)^{-1} = \sigma \left(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}} \right)^{-1}. \tag{1.15}$$

This highlights how the nonlinear observability matrix directly influences the estimation uncertainty. Consequently, leveraging this matrix allows for strategic selection of the control and state trajectories and, thus, for the possibility of optimizing the state estimation performance.

1.1.2 Observability measures using the nonlinear observability matrix

The interplay between the nonlinear observability matrix and estimation uncertainty in a least squares estimator motivates the use of this matrix for obtaining quantitative measures of the system observability.

The singular values of the observability matrix (eigenvalues of $\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}$) serve as effective indicators of observability due to their connection with the estimation covariance. This relationship mirrors the principles commonly employed in *optimal experimental design* (OED) literature [Boyd & Vandenberghe, 2004; Pukelsheim, 2006; Ucinski, 2004]. Specifically:

- The minimum singular value, $\sigma_{\min}\left(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}\right)$, measures proximity to a singular matrix (indicative of unobservability), inversely correlating with maximum estimation covariance.

- The maximum singular value, $\sigma_{\max}\left(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}\right)$, measures the energy of the most observable mode, inversely linked to minimum estimation covariance.

- The condition number, $\kappa(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}) = \sigma_{\max}(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}})/\sigma_{\min}(\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}})$, represents the energy ratio between the most and least observable modes, shaping the estimation uncertainty ellipsoid. A large condition number results in an elongated uncertainty ellipsoid, while unity denotes uniform estimation uncertainty.

- The determinant of $\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}$ yields the volume of the observation ellipsoid, $\mathcal{B} = \{\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}\boldsymbol{\xi} |\, \|\boldsymbol{\xi}\| \leq 1\}$, inversely related to the volume of the estimation uncertainty ellipsoid.

While the least squares estimate presents a coherent link between estimation uncertainty and the nonlinear observability matrix, it assumes the simultaneous measurement of all output derivatives in a single step. In practice, this is avoided by relying on an iterative estimator that does not require availability of all the derivatives. However, this poses additional challenges for establishing a clear relationship between the uncertainty of the employed state estimator and the observability matrix.

1.2 Observability Gramian

While the observability matrix provides insights into the observability of a system at a specific time t by considering the measured output, inputs, and all their higher-order derivatives, it lacks consideration of the system *past information history*. Two additional observability analysis tools, closely interlinked, are instead better suited for this purpose and will be extensively used in this thesis. These tools are the *Observability Gramian* (OG) and *Constructability Gramian* (CG), which are classical instruments for studying the observability of linear time-varying systems. They are also frequently employed to examine the local observability of nonlinear systems by considering a linear approximating system around a nominal trajectory $\boldsymbol{x}_0(t)$ [P. Bernard et al., 2020; Krener & Ide, 2009]. This approximation can be expressed as:

$$\begin{aligned} \delta\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(\boldsymbol{x}_0(t), \boldsymbol{u}(t))\delta\boldsymbol{x}(t) \\ \delta\boldsymbol{y}(t) &= \boldsymbol{C}(\boldsymbol{x}_0(t))\delta\boldsymbol{x}(t) \end{aligned} \tag{1.16}$$

with $\delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_0(t)$ and $\delta\boldsymbol{y}(t) = \boldsymbol{y}(t) - \boldsymbol{y}_0(t)$ being small variations while

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{x}_0(t), \boldsymbol{u}(t)) &= \left.\frac{\partial(\boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{f}_i(\boldsymbol{x})u_i)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_0(t)} \\ \boldsymbol{C}(\boldsymbol{x}_0(t)) &= \left.\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_0(t)} \end{aligned}. \tag{1.17}$$

Definition 1.2.1 (Observability Gramian)**.** *For a given measured signal $t \to \boldsymbol{y}(t)$, the observability gramian associated to* (1.1) *evolves according to the dynamics*

$$\dot{\boldsymbol{G}}_o(t) = -\boldsymbol{A}(t)^T\boldsymbol{G}_o(t) - \boldsymbol{G}_o(t)\boldsymbol{A}(t) + \boldsymbol{C}^T(t)\boldsymbol{C}(t) \tag{1.18}$$

and its solution on an interval $[t_0, t_1] \subset [t_0, +\infty)$ is the positive semidefinite matrix defined by

$$\boldsymbol{G}_o(t_0, t_f) = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau, t_0)\boldsymbol{C}^T(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau, t_0)\, d\tau \tag{1.19}$$

where $\boldsymbol{\Phi}(\tau, t_0)$ is the *state transition* (or *state sensitivity*) matrix associated to the linear dynamics in (1.16), namely the unique solution of

$$\begin{aligned} \dot{\boldsymbol{\Phi}}(\tau, t) &= \boldsymbol{A}(\boldsymbol{x}_0(\tau), \boldsymbol{u}(\tau))\boldsymbol{\Phi}(\tau, t) \\ \boldsymbol{\Phi}(t, t) &= \boldsymbol{I} \end{aligned}. \tag{1.20}$$

Some useful properties of the state transition matrix

- It is **never singular**
- $\boldsymbol{\Phi}^{-1}(t_1, t_2) = \boldsymbol{\Phi}(t_2, t_1)$
- $\boldsymbol{\Phi}(t_2, t_1)\boldsymbol{\Phi}(t_1, t_0) = \boldsymbol{\Phi}(t_2, t_0) \quad \forall \quad t_0 \leq t_1 \leq t_2$
- $\boldsymbol{\Phi}(t_1, t_0) = \frac{\partial \boldsymbol{x}(t_1)}{\partial \boldsymbol{x}(t_0)}$
- For system (1.16), we have $\delta\boldsymbol{x}(t_2) = \boldsymbol{\Phi}(t_2, t_1)\delta\boldsymbol{x}(t_1)$

See also [Antsaklis & Michel, 1997].

Before proceeding, we give a fundamental definition:

Definition 1.2.2 (Persistency of excitation)**.** *A positive semidefinite matrix-valued function $\boldsymbol{\Sigma}(t) \in \mathbb{R}^{d\times d}$, is called persistently exciting (PE), if there exist $T, \mu > 0$ such that for all $t \geq 0$*

$$\int_t^{t+T} \boldsymbol{\Sigma}(\tau)\, d\tau \succeq \mu \boldsymbol{I} \tag{1.21}$$

Here, the symbol '$\succeq$' indicates the *Loewner* partial ordering of positive semidefinite matrices [Horn & Johnson, 2012, p.493], i.e. $\boldsymbol{A} \succeq \boldsymbol{B} \implies \boldsymbol{A} - \boldsymbol{B} \succeq \boldsymbol{0}$, that is $\boldsymbol{A} - \boldsymbol{B}$ is positive semidefinite.

Given a pair of sufficiently close solutions $\boldsymbol{x}_a(t)$ and $\boldsymbol{x}_b(t)$ of system (1.16) having the same output $\boldsymbol{y}(t)$ for all $t \in [0, T]$ for some $T > 0$, their difference $\delta\boldsymbol{x}(t) = \boldsymbol{x}_a(t) - \boldsymbol{x}_b(t)$ satisfies

$$\begin{aligned} &\delta\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(\boldsymbol{x}_a(t), \boldsymbol{u}(t))\delta\boldsymbol{x}(t) \\ &\boldsymbol{C}(\boldsymbol{x}_a(t))\delta\boldsymbol{x}(t) = 0 \\ &\forall t \in [0, T] \end{aligned} \tag{1.22}$$

which implies

$$\begin{aligned} &\|\boldsymbol{C}(\boldsymbol{x}_a(t))\delta\boldsymbol{x}(t)\|^2 = \delta\boldsymbol{x}(t)^T\boldsymbol{C}(\boldsymbol{x}_a(t))^T\boldsymbol{C}(\boldsymbol{x}_a(t))\delta\boldsymbol{x}(t) = 0 \\ &\forall t \in [0, T] \end{aligned} . \tag{1.23}$$

Furthermore, notice that

$$\begin{aligned}\delta\boldsymbol{x}(t) &= \boldsymbol{\Phi}(t,0)\delta\boldsymbol{x}(0) = \boldsymbol{\Phi}(t,0)(\boldsymbol{x}_a(0) - \boldsymbol{x}_b(0)) \\ \delta\boldsymbol{y}(t) &= \boldsymbol{C}(\boldsymbol{x}_a(t))\delta\boldsymbol{x}(t) = \boldsymbol{C}(\boldsymbol{x}_a(t))\boldsymbol{\Phi}(t,0)\delta\boldsymbol{x}(0).\end{aligned} \tag{1.24}$$

From (1.23) and (1.24), it then follows

$$(\boldsymbol{x}_a(0) - \boldsymbol{x}_b(0))^T \boldsymbol{G}_o(0,T)(\boldsymbol{x}_a(0) - \boldsymbol{x}_b(0)) = 0 \tag{1.25}$$

implying that $\boldsymbol{x}_a(0) = \boldsymbol{x}_b(0)$ and thus $\boldsymbol{x}_a(t) = \boldsymbol{x}_b(t)\ \forall t \in [0,T]$, if and only if $\boldsymbol{G}_o(0,T)$ is full-rank. We point out that this is again a local notion of observability, as it was the case for the observability rank condition.

Assuming that the gramian is invertibile along any trajectory and assuming that $\boldsymbol{A}$ is bounded, one has that Kalman's well-known *uniform complete observability* (UCO) [B. Anderson & Moore, 1969] is satisfied if there exist $\alpha_1, \alpha_2, T > 0$ such that

$$\alpha_1 \boldsymbol{I} \leq \boldsymbol{G}_o \leq \alpha_2 \boldsymbol{I}. \tag{1.26}$$

In (1.26) the existence of a finite α_2 is guaranteed as long as $\boldsymbol{A}$ and $\boldsymbol{C}$ are uniformly bounded, while most importantly the existence of a positive α_1 depends on the invertibility of the OG and, therefore, constitutes a PE condition (see Definition 1.2.2). Furthermore, assume that the design of a linear Kalman Filter (or more generally a Riccati observer [Hamel & Samson, 2016]) estimating the state of the system is possible, then the scalar α_1 is also related to the worst case rate of convergence of the Kalman filter [Hamel & Samson, 2016]. A similar result is also presented in [P. Bernard et al., 2020] for an Extended Kalman-like observer. This link between the worst case rate of convergence of the observer and the eigenvalues of the OG prompts the use of active sensing applied to state estimation in robotics [Salaris et al., 2019].

When the time dependency of $\boldsymbol{A}$ and $\boldsymbol{C}$ is due to the linearization along the nominal trajectory then the related observability property may depend on the input $\boldsymbol{u}$ of the system. In this case, as already mentioned when discussing the observability matrix, it becomes interesting to use the input for *actively steering* the system along observable/informative trajectories.

We note that results analogous to the OG can be derived for the strictly related CG.

This is defined as

$$\boldsymbol{G}_c(t_0, t_f) = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau, t_f)\boldsymbol{C}^T(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau, t_f)\, d\tau \tag{1.27}$$

where, compared to the OG, the time arguments of the state transition matrix are different, since this matrix now corresponds to an integration *backwards in time*, which can be computed using the properties of the state transition matrix. In this case, an equivalent condition to (1.25) is given by

$$(\boldsymbol{x}_a(t_f) - \boldsymbol{x}_b(t_f))^T \boldsymbol{G}_c(t_0, t_f)(\boldsymbol{x}_a(t_f) - \boldsymbol{x}_b(t_f)) = 0, \tag{1.28}$$

from which, again, we can characterize local observability of (1.1) from the invertibility of the CG. Using the properties of the state-transition matrix, it can be shown that [Salaris et al., 2019]

$$\boldsymbol{G}_c(t_0, t_f) = \boldsymbol{\Phi}^T(t_0, t_f)\boldsymbol{G}_o(t_0, t_f)\boldsymbol{\Phi}(t_0, t_f). \tag{1.29}$$

Since $\boldsymbol{\Phi}^T(t_0, t_f)$ is invertible, the OG and the CG are *congruent*. Hence, the *Sylvester's law of inertia* [Meyer & Stewart, 2023, p.563] establishes the fact that the two matrices have the same inertia, i.e. same number of positive, negative and zero eigenvalues. Indeed, in terms of observability analysis, these two matrices provide properties which are qualitatively equivalent. However, when precise quantification of observability is necessary, one matrix might be more suitable than the other one depending on the specific context. For example, it has been shown in [P. Bernard, 2019; Salaris et al., 2019] that, when interested in the reconstruction of the *current* state of the plant rather than of its initial condition, the CG may be more suited than the OG. This will be made clearer later on in Sect. 1.2.1.

Regarding the notation used for the OG or CG, we sometimes denote their dynamics with a single time dependency, along with the associated initial conditions. In other cases, we may explicit the time dependence using two time instances, such as t_0 and t_f, particularly when focusing solely on the information acquired within the specific time interval $[t_0, t_f]$.

Furthermore, we note that the initial conditions of the OG at time t_0 can be intended as the *prior information* about the state of the system. For example, in some cases, when computing the information acquired over a future horizon, if the employed state estimator is an Extended Kalman Filter, one may use the inverse of the solution of the Riccati

equation to represent the prior information about the system, i.e., using the initialization

$$\begin{aligned}\dot{\boldsymbol{G}}_o(t) &= -\boldsymbol{A}(t)^T\boldsymbol{G}_o(t) - \boldsymbol{G}_o(t)\boldsymbol{A}(t) + \boldsymbol{C}^T(t)\boldsymbol{C}(t)\\ \boldsymbol{G}_o(t_0) &= \boldsymbol{P}(t_0)^{-1}.\end{aligned} \tag{1.30}$$

In what follows, we will show how the OG can provide valuable measures of information about the system.

1.2.1 Observability Gramian as a measure of information

This section, primarily derived from [Krener & Ide, 2009], mainly discuss the Observability Gramian (OG) as a tool to quantify observability. Although the following discussion focuses on the OG, equivalent conclusions also hold for the Constructability Gramian (CG).

The linear approximating system defines a linear mapping from small changes in the initial condition $\delta\boldsymbol{x}(0)$ to changes in the output $\delta\boldsymbol{y}(0:T)$, and this linear mapping is tangent at $\boldsymbol{x}_0$ to the corresponding nonlinear mapping from $\boldsymbol{x}(0)$ to $\boldsymbol{y}(0:T)$ defined by (1.1). The local singular values at $\boldsymbol{x}_0$ of the nonlinear mapping—defined by (1.1)—are equivalent to the singular values of its tangent linear mapping—defined by (1.16). If these local singular values at $\boldsymbol{x}_0$ are relatively large, it becomes easier to invert the mapping from $\boldsymbol{x}_0$ to $\boldsymbol{y}(0:T)$ near $\boldsymbol{x}_0$. Essentially, it becomes easier to distinguish initial states around $\boldsymbol{x}_0$ from their respective output trajectories.

Considering an estimation scheme that is exact in the absence of observation noise — precisely inverting the map from $\boldsymbol{x}_0$ to $\boldsymbol{y}(0:T)$ — the smallest singular value of this mapping becomes a measure of the difficulty in estimating the initial state $\boldsymbol{x}_0$ from the output trajectory $\boldsymbol{y}(0:T)$. A larger reciprocal of this smallest singular value indicates that observation noise could significantly impact the estimation error. In [Krener & Ide, 2009], this smallest singular value is termed the *local unobservability index* of the nonlinear system (1.1).

Another metric proposed in [Krener & Ide, 2009] is the ratio between the largest and smallest singular values, termed the *local estimation condition number*. A higher value of this metric suggests that changes in output caused by small perturbations in one direction could dominate changes from another direction. This scenario indicates that the estimation problem is ill-conditioned in state space directions exhibiting a high local estimation condition number.

Some remarks on the observability Gramian

- The integrand $\boldsymbol{\Phi}^T(\tau,0)\boldsymbol{C}^T(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau,0)$ within the OG is a Gram matrix, ensuring positive semidefiniteness of the OG. Consequently, the local observability Gramian (and its eigenvalues) increases monotonically over the interval $[0,T]$.
- Properly scaling the state coordinates $\boldsymbol{x}$ is crucial for meaningful numerical interpretations of the OG eigenvalues
- The total local observability Gramian $\boldsymbol{G}_o(0,T)$ for two measured output vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is the sum of separate local observability Gramians $\boldsymbol{G}_{o1}(0,T)$ and $\boldsymbol{G}_{o2}(0,T)$. Consequently, the changes in the local unobservability index and the local estimation condition number illustrate the additional observability $\boldsymbol{y}_2$ brings to $\boldsymbol{y}_1$
- The computation of the entire local observability Gramian might not be necessary; in some cases, one can only focus on significant state directions.
- It is possible to extend the local observability Gramian to measure the effect of parameter changes on the output. Indeed, parameters can always be viewed as additional states with time derivative zero.

More information about some properties of the OG are given in [Krener & Ide, 2009].

It is noteworthy, as highlighted in [Krener & Ide, 2009], that the local singular values pertaining to the mapping from the initial state $\boldsymbol{x}_0$ to the output trajectory $\boldsymbol{y}(0:T)$ correspond to the square roots of the eigenvalues of the OG (1.19).

Another perspective on the relationship between the OG and the ease of inverting the mapping from the initial state to the output trajectory is as follows. Imagine a perturbation of the initial state $\delta\boldsymbol{x}(0)$, which propagates to the output along the trajectory, causing a perturbation $\delta\boldsymbol{y}(t)$. The energy of this output perturbation along the trajectory can be expressed as:

$$\|\delta\boldsymbol{y}(t)\|_{\mathcal{L}_2}^2 = \int_0^T \delta\boldsymbol{y}(\tau)^T\delta\boldsymbol{y}(\tau)\,d\tau = \delta\boldsymbol{x}^T(0)\boldsymbol{G}_o(0,T)\delta\boldsymbol{x}(0), \tag{1.31}$$

where $\|\cdot\|_{\mathcal{L}_2}$ indicates the $\mathcal{L}_2$ norm of a signal. Equation (1.31) highlights that when a small perturbation in the initial state leads to a substantial energy change in the output, it becomes easier to estimate the initial state perturbation from the output trajectory. Using a unit norm perturbation $\delta\boldsymbol{x}(0)$, it is possible to define an observability ellipsoid associated with the eigenpairs of the OG. Specifically, key observability indices such as the minimum eigenvalue, condition number, and determinant of the OG exhibit a positive correlation with distinct attributes of this ellipsoid: they respectively correspond to the length of its minimum axis, its overall shape, and the enclosed volume

It is important to note that just as the OG measures how easy it is to invert the mapping from the initial state to the output trajectory, the CG similarly quantifies the ease of inverting the mapping, but this time from the *final state* to the output trajectory:

$$\|\delta\boldsymbol{y}(t)\|_{\mathcal{L}_2}^2 = \int_0^T \delta\boldsymbol{y}(\tau)^T\delta\boldsymbol{y}(\tau)\,d\tau = \delta\boldsymbol{x}^T(T)\boldsymbol{G}_c(0,T)\delta\boldsymbol{x}(T). \tag{1.32}$$

This distinction between the OG and CG provides two different perspectives on how easily one can invert the mapping between the system states and the output trajectory, offering insights for both initial and final state perturbations. This clarification emphasizes the energy perspective of the OG and CG and highlights their dual role in measuring the invertibility of mappings from initial and final states to the output trajectory.

Furthermore, it is possible to encode the confidence in the different measured output as well as sensing constraints by suitably weighting the various outputs in the OG, i.e.

$$\boldsymbol{G}_o(t_0,t_f) = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau,t_0)\boldsymbol{C}^T(\tau)\boldsymbol{W}(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau,t_0)\,d\tau, \tag{1.33}$$

where $\boldsymbol{W}(\tau)$ is a suitably designed block diagonal positive semidefinite matrix, in which each block corresponds to a measured output. Throughout the thesis, this approach will be frequently adopted for considering presence of sensing constraints.

1.2.2 The connection to the Fisher information matrix

The Fisher information measures the amount of information carried by a random variable, e.g. $\boldsymbol{y}$, regarding an unknown parameter—in our case, the system state $\boldsymbol{x}$, on which the probability of $\boldsymbol{y}$ depends [Kay, 1993]. Consider $f(\boldsymbol{y};\boldsymbol{x})$, the probability density function of $\boldsymbol{y}$ given a known $\boldsymbol{x}$. If this function is sharply peaked with respect to changes in $\boldsymbol{x}$, it is easier to discern the "correct" $\boldsymbol{x}$ from the data. Conversely, if f is flat and dispersed,

estimating the true $\boldsymbol{x}$ from samples of $\boldsymbol{y}$ becomes more challenging, implying the need of studying some kind of variance with respect to $\boldsymbol{x}$.

Let us first start by the scenario where $\boldsymbol{x}$ is a scalar (although we will represent it in bold for generality) before extending to the vector case. The derivative of the logarithm of the likelihood function with respect to $\boldsymbol{x}$ is termed the *score*. Under certain regularity conditions, when $\boldsymbol{x}$ is the true state, the expected value of the score, evaluated at the true $\boldsymbol{x}$, equals 0:

$$E\left[\frac{\partial}{\partial \boldsymbol{x}} \log f(\boldsymbol{y};\boldsymbol{x}) \middle| \boldsymbol{x}\right] = 0. \tag{1.34}$$

This property results from the fact that the logarithm, being a monotonic function, implies that given the true $\boldsymbol{x}$, $\boldsymbol{y}$ maximizes the probability density function $f(\boldsymbol{y};\boldsymbol{x})$.

The *Fisher information* represents the variance of the score:

$$\mathcal{F}(\boldsymbol{x}) = E\left[\left(\frac{\partial}{\partial \boldsymbol{x}} \log f(\boldsymbol{y};\boldsymbol{x})\right)^2 \middle| \boldsymbol{x}\right]. \tag{1.35}$$

For twice-differentiable $f(\boldsymbol{y};\boldsymbol{x})$, the Fisher information is also represented as:

$$\mathcal{F}(\boldsymbol{x}) = -E\left[\frac{\partial^2}{\partial \boldsymbol{x}^2} \log f(\boldsymbol{y};\boldsymbol{x}) \middle| \boldsymbol{x}\right]. \tag{1.36}$$

Thus, the Fisher information characterizes the curvature of the log-likelihood. Lower Fisher information around the maximum likelihood estimate indicates a "blunt" maximum, where nearby values exhibit similar log-likelihoods. Conversely, higher Fisher information signifies a "sharper" maximum.

When $\boldsymbol{x}$ becomes a vector, the Fisher information becomes a matrix known as the *Fisher information matrix* (FIM). The FIM is symmetric and positive semidefinite and each of its elements is given by:

$$\begin{aligned}\left[\boldsymbol{F}(\boldsymbol{x})\right]_{ij} &= E\left[\left(\frac{\partial}{\partial x_i} \log f(\boldsymbol{y};\boldsymbol{x})\right)\left(\frac{\partial}{\partial x_j} \log f(\boldsymbol{y};\boldsymbol{x})\right) \middle| \boldsymbol{x}\right] \\ &= E\left[\frac{\partial^2}{\partial x_i \partial x_j} \log f(\boldsymbol{y};\boldsymbol{x}) \middle| \boldsymbol{x}\right]\end{aligned}. \tag{1.37}$$

For the dynamical system (1.1), assuming a multivariate normal distribution, one obtains the following important result: the FIM takes the form of a weighted OG (see

Some interesting facts about the FIM

- **Hessian of Kullback-Leibler Divergence**: The FIM can be interpreted as the Hessian matrix of the Kullback-Leibler divergence, which is widely used in statistics and machine learning to measure the "distance" among two probability distributions.
- **Riemannian Metric & Fisher-Rao Geometry**: In cases where the FIM is positive-definite, it can serve as a Riemannian metric, contributing to the Fisher-Rao geometry.

The implications of these two aspects in learning are quite remarkable. The FIM is commonly employed as a Riemannian metric in the *natural gradient algorithm*, renowned for significantly accelerating convergence compared to traditional gradient descent methods [Amari, 1998].
The application of natural gradient descent has recently captured attention in the control community, particularly in the domain of adaptive control, for estimating physically consistent parameters [Boffi & Slotine, 2021; T. Lee et al., 2018; Wensing & Slotine, 2020].

(1.33))

$$\boldsymbol{F} = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau, t_0)\boldsymbol{C}^T(\tau)\boldsymbol{R}^{-1}(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau, t_0)\, d\tau, \tag{1.38}$$

where $\boldsymbol{R}$ weights the measured outputs based on the correponding noise covariance matrix.

In absence of process noise, the classical formulation of the Kalman Filter uses as gain matrix the covariance matrix $\boldsymbol{R}$ and the FIM $\boldsymbol{F}$ corresponds to the inverse of the solution of the Riccati equation

$$\dot{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^T - \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}, \tag{1.39}$$

with $\boldsymbol{K}$ being the Kalman gain $\boldsymbol{K} = \boldsymbol{P}\boldsymbol{C}^T\boldsymbol{R}^{-1}$, further strengthening the connection between the OG and the convergence properties of the Kalman Filter, as well as, the covariance of the estimation error.

Furthermore, a very important fact about the FIM is that it provides a lower bound on the variance which can be achieved by any unbiased estimate $\hat{\boldsymbol{x}}$ from measurements $\boldsymbol{y}$ [Barfoot, 2017; Kay, 1993]

$$\mathrm{cov}[\hat{\boldsymbol{x}}] \succeq \boldsymbol{F}^{-1}. \tag{1.40}$$

This fundamental property motivates the widespread use of the FIM in optimal experimental design, as it will be discussed in the next section, as it provides valuable information on the error distribution.

1.3 Active sensing and optimal experimental design

Maximizing the information collected through sensor measurements is the subject of the fields of *active sensing* and *optimal experimental design*. The term active sensing has been typically used (mainly in the robotics community) for the problem of maximizing the information collected from measurements about the state of the system by shaping the system trajectory through a proper selection of the inputs. Instead, the term optimal experimental design has been used across very diverse fields, commonly for the problem of identifying the parameters of a system of interest, either by optimal sensor placement or by optimally selecting the experiments to carry out or again by performing informative trajectories.

To shape these informative trajectories or solve sensor placement problems, the general strategy involves solving an optimization problem. The objective function to minimize usually considers one of the following *optimality criteria*:

- A-criterion: $J_A = \text{trace}(\boldsymbol{\mathcal{I}}^{-1}(\boldsymbol{x}))$
- D-criterion: $J_D = -\log\left(\det(\boldsymbol{\mathcal{I}}(\boldsymbol{x}))\right)$
- E-criterion: $J_E = -\lambda_{\min}(\boldsymbol{\mathcal{I}}(\boldsymbol{x}))$

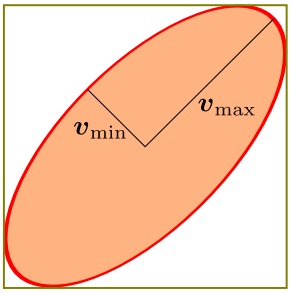
where $\boldsymbol{\mathcal{I}}(\boldsymbol{x})$ is an information matrix (e.g. FIM, OG or CG) and $\lambda_{\min}(\cdot)$ is the minimum eigenvalue function. The same criteria could be redefined analogously for a *dispersion* or covariance matrix $\boldsymbol{\mathcal{P}}$:

- A-criterion: $J_A = \text{trace}(\boldsymbol{\mathcal{P}}(\boldsymbol{x}))$
- D-criterion: $J_D = \log\left(\det(\boldsymbol{\mathcal{P}}(\boldsymbol{x}))\right)$
- E-criterion: $J_E = \lambda_{\max}(\boldsymbol{\mathcal{P}}(\boldsymbol{x}))$

However, it is important to note that while minimizing the trace of the inverse information matrix is beneficial from an information perspective, the same does not always hold for

the trace of the information matrix itself—more details on this subject in [Pukelsheim, 2006, p.138] and [Salaris et al., 2019].

These criteria closely relate to the eigenvalue interpretations discussed in Sections 1.1.2 and 1.2.1, considering functions of observability matrix and gramian eigenvalues. Figure 1.1 shows the interpretation of the different criteria in terms of the level sets of a dispersion ellipsoid.



A-criterion : $\propto (\boldsymbol{v}_{\text{min}} + \boldsymbol{v}_{\text{max}})$
D-criterion : $\propto (\boldsymbol{v}_{\text{min}} \cdot \boldsymbol{v}_{\text{max}})$
E-criterion : $\propto \boldsymbol{v}_{\text{max}}$

Figure 1.1: Illustration of the geometric meaning of the different OED-criteria for a two dimensional vector $\boldsymbol{x}$. Minimizing the sum of the two semi-axis is the A-criterion, minimizing the volume/area is the D-criterion, and minimizing the largest uncertainty axis is the E-criterion.

In the field of robotics, various studies leverage optimization using either the observability or constructability gramian, the Fisher information matrix, or the covariance matrix resulting from the solution to the Riccati equation. These have found applications in, e.g., enhancing single robot localization [Böhm et al., 2022; Lorussi et al., 2001; Napolitano et al., 2021, 2022; Salaris et al., 2019; Salaris et al., 2017], self-calibration [Böhm et al., 2020; Hausman et al., 2017; Preiss et al., 2018], source seeking/spatial field reconstruction and sampling [Calkins et al., 2020; DeVries et al., 2013; Hinson et al., 2013; Khodayi-mehr et al., 2019; Notomista et al., 2022; T. Zhang et al., 2022], localization of one or multiple targets [Balandi et al., 2023; Chung et al., 2006; Freundlich et al., 2017; Jacquet et al., 2022; Martinez & Bullo, 2006; Morbidi & Mariottini, 2012; Olfati-Saber, 2007; Zhou & Roumeliotis, 2011] and optimal cooperative localization [Cano & Le Ny, 2023; Cossette et al., 2022; Cristofaro & Martinelli, 2010; De Carli et al., 2021, 2023; S. Li et al., 2022; Riz et al., 2023; Spica & Robuffo Giordano, 2016; W. Zhang et al., 2022].

Some other relevant works in the field of active sensing and optimal experimental design are the following: [Spica & Giordano, 2013; Spica et al., 2014; Spica et al., 2017]

used active sensing for structure from motion in visual servoing tasks, [Besançon et al., 2013] studies the problem of inputs in observer design for systems which are not uniformly observable, [Hinson & Morgansen, 2013] studies active sensing solution for the control of non-holonomic systems, [Houska et al., 2015; Telen et al., 2013; Telen et al., 2014] propose sequential semidefinite programming (SDP) based solutions for optimal experimental design with application to nonlinear dynamic (bio)chemical processes, [Qian et al., 2017] studies an on-line model parameter identification approach for multivariable systems with application to an unstable rolling delta wing, [Babar & Baglietto, 2021] proposes an optimal feedback input design method for active parameter identification of dynamic nonlinear systems through an NMPC approach, [Napolitano et al., 2023] proposes a method to improve the quality of datasets employed in model learning by optimizing metrics based on the combination of exploration and active sensing measure and [Freundlich et al., 2017] develops a distributed optimization based algorithm to optimally estimate a certain hidden state using a sensor network.

In the following, we contextualize the contributions of this thesis within the existing literature.

Contributions in active sensing for multi-robot systems

Coordinated teams of robots are finding more and more applications in critical domains, including tasks such as mapping, surveillance, disaster response, exploration, border security, and patrol missions [Aggravi et al., 2021; Murphy et al., 2011; Schranz et al., 2020]. Effective autonomous coordination in multi-robot systems relies on the possibility to exchange information across the network and on the availability of accurate relative localization in a shared coordinate frame. However, centralized infrastructures like the Global Positioning System (GPS) may not always be available, motivating research on the topic of sensor-based cooperative localization in a common frame using, e.g., onboard cameras or Ultra Wideband (UWB) sensors [Cano & Le Ny, 2023; De Carli et al., 2021; Schiano et al., 2016]. Particularly in scenarios where only partial information regarding the relative state of robots is available, such as relative bearing (i.e., scale-free directional information) or distances, and/or when sensing limitations, such as limited range and field of view, are present, leveraging active sensing becomes crucial for resolving cooperative localization challenges. This holds true even in cases where one or a group of robots is assigned the task of localizing a potentially moving target.

Among the previous works on active sensing, only few consider a multi-robot scenario

for cooperative localization, e.g. [Cano & Le Ny, 2023; Cossette et al., 2022; S. Li et al., 2022; Morbidi & Mariottini, 2012; Spica & Robuffo Giordano, 2016], and even fewer explore distributed solutions for this purpose, e.g. [Cano & Le Ny, 2023; Spica & Robuffo Giordano, 2016]. Much of the work presented in this thesis is devoted to achieving **active sensing for optimal cooperative localization in a distributed way**. In comparison to [Cano & Le Ny, 2023] and related works from the same group, e.g. [Cano & Le Ny, 2021; Cossette et al., 2022], our proposed solutions are designed to scale well with the number of robots. Additionally, relying on the OG, we consider the information collected in the past. In contrast, [Cano & Le Ny, 2023] only takes into account the information acquired through current measurements, thereby optimizing solely the geometry of the system and not the entire trajectory. Instead, in [Spica & Robuffo Giordano, 2016], a specialized solution was presented to maximize the convergence rate of nonlinear adaptive observers for separately estimating relative distances associated to each edge in a formation of robots measuring relative bearings. This algorithm maximizes the information gained only considering the current measurements and not the history of past measurements. The solutions proposed in this thesis aim to provide algorithms which consider a network perspective rather than separately considering each edge, meaning that information coming from different measurements is fused and, furthermore, the past history of collected information is fully considered.

Furthermore, only a handful of works, e.g. [Cano & Le Ny, 2023; Napolitano et al., 2022; Qian et al., 2017], explores the inclusion of other tasks than active sensing. In particular, in [Cano & Le Ny, 2023], some robots only perform their desired task while others minimize a cost function which is a trade off between optimizing the localization performance and maintaining connectivity. In [Spica et al., 2017], a *large projector operator*-based approach was proposed to optimize the trajectory in the null-space of the Jacobian of the total error norm. In [Qian et al., 2017], since the considered system is open loop unstable, a *control Lyapunov function* (CLF) is defined to stabilize the system, and the cost function is used for balancing between minimizing the Lyapunov function and minimizing the condition number of the inverse of the FIM. In [Napolitano et al., 2022], a CLF is used to encode the task achievement and its decrease is posed as a constraint in a Model Predictive Control (MPC) formulation. In this thesis, we also explore the integration of a primary task that entails treating a CLF as a constraint. However, we face the challenges of a distributed architecture with limited information availability, thus motivating the design of a reactive control strategy as a viable solution. Moreover,

we also explore the possibility of considering the information acquisition as a constraint in a target tracking application. Many works have investigated the problem of active sensing for target tracking in multi-robots systems [Balandi et al., 2023; Chung et al., 2006; Freundlich et al., 2017; Jacquet et al., 2022; Martinez & Bullo, 2006; Morbidi & Mariottini, 2012; Olfati-Saber, 2007; Zhou & Roumeliotis, 2011], usually with the objective of maximizing the collected information about the position of the target. Conversely, in our work [Balandi et al., 2023], the robots are tasked with maintaining a minimum level of persistency of excitation while utilizing the redundancy within the multi-robot system to accomplish additional tasks whenever possible.

Chapter 2

Multi-Robot Systems: Consensus, Formation Control and Localization

This chapter serves as a foundation for understanding key concepts in multi-agent systems. Initially, we introduce the typical notation commonly used in the field of multi-agent systems. Subsequently, we delve into fundamental concepts and definitions revolving around graphs, algebraic graph theory and *frameworks*. Following this, we give a very basic introduction of the *consensus* algorithm, which is central to any distributed algorithm in multi-agent systems.

Towards the chapter conclusion, we provide an introduction to the frameworks of distance and bearing rigidity. These notions are key in establishing conditions to uniquely determine the geometric pattern of a group of not necessarily moving robots from relative distance and bearing measurements. Additionally, we delve into how infinitesimal rigidity connects with nonlinear observability and the Fisher information matrix.

Key references that have significantly contributed to the development of these sections include [Bullo, 2020; de Marina Peinado, 2016; Mesbahi & Egerstedt, 2010; Restrepo, 2021].

2.1 Notation

This section introduces specialized notation frequently used in multi-agent systems. Throughout this chapter, we may interchangeably use the terms *agent* and *robot* to encapsulate the broader scope beyond robotics.

- **Kronecker product**: the Kronecker product, denoted by $\otimes$, of $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and

$\boldsymbol{B} \in \mathbb{R}^{q\times r}$ is the $nq \times mr$ matrix $\boldsymbol{A} \otimes \boldsymbol{B}$ defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1m}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\boldsymbol{B} & \dots & a_{nm}\boldsymbol{B} \end{bmatrix} \tag{2.1}$$

For a matrix $\boldsymbol{A} \in \mathbb{R}^{N\times p}$, we employ $\boldsymbol{A}_d \coloneqq \boldsymbol{A} \otimes \boldsymbol{I}_d \in \mathbb{R}^{Nd\times pd}$.

- **Set cardinality**: $|\mathcal{P}|$ denotes the cardinality of set $\mathcal{P}$.
- **Block-diagonal matrix**: $\text{diag}(\boldsymbol{A}_i)$ represents a block diagonal matrix comprising blocks $\boldsymbol{A}_i$, $i = 1, 2, ...$, along the diagonal.
- **All-ones and all-zeros**: $\mathbf{1}_N$ and $\mathbf{0}_N$ denote vectors of all ones and all zeros, respectively of dimension N.
- **Stacked variables** When addressing local variables $\boldsymbol{x}_i \in \mathbb{R}^d$, we often refer to the concatenated vector of local variables as $\boldsymbol{x} \coloneqq \begin{bmatrix}\boldsymbol{x}_1^T & \dots & \boldsymbol{x}_N\end{bmatrix}^T \in \mathbb{R}^{Nd}$.

2.2 Graphs and frameworks

In analyzing multi-agent systems, one typically considers that each agent accesses only partial system information via local interactions among neighboring agents, and these interactions are often dictated by the constraints on the robot sensing and communication. In all cases, graph theory serves as a natural framework for modeling how these interactions affect the flow of information across the group.

Graph Representation

- A *graph*, denoted as $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, comprises:
  - $\mathcal{V} = \{v_1, ..., v_N\}$, the set of *vertices* or *nodes*.
  - $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, the set of *edges* ($|\mathcal{E}| = M$).
- *Subgraph*: $\mathcal{H} = (\mathcal{V}_\mathcal{H}, \mathcal{E}_\mathcal{H}) \subset \mathcal{G}$, a subset with $\mathcal{V}_\mathcal{H} \subset \mathcal{V}$ and $\mathcal{E}_\mathcal{H} \subset \mathcal{E}$.
- *Spanning Subgraph*: Contains all nodes ($\mathcal{V}$).

Interaction Representation

- An *edge* $e_k := (i, j) \in \mathcal{E}$ is an ordered pair representing a connection starting from node i and ending at node j. The existence of an edge $e_k = (i, j)$ indicates an interaction—either directed or undirected—between node i and node j. For instance, node i might transmit information to node j, or node i might measure the state of node j relative to its own. With a slight abuse of notation, we may refer to variables related to the k-th edge as $e_k := (i, j)$ with a subscript k or ij.

- *Neighbors*: For node i, $\mathcal{N}_i := \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$ defines its set of neighbors.

- Weights: each edge can be associated with a *weight* w_k (in some cases even a matrix weight $\boldsymbol{W}_k$), increasing the representation power of graphs. In this thesis, we only consider nonnegative weights $w_k \geq 0$ ($\boldsymbol{W}_k \succeq \mathbf{0}$). In this case, the graph is said to be *weighted*.

- Directionality: In addition to denoting nodes interacting with each other, the edges within a graph also encode the flow direction of interactions among the nodes. Based on the type of interaction a graph could be

  - *Undirected*: Bidirectional interactions $(i, j) \in \mathcal{E}$ imply $(j, i) \in \mathcal{E}$; edges are unordered pairs ($\{i, j\}$).
  - *Directed*: Indicates a single-directional interaction flow.

  As it will be clearer later, for an undirected graph it is sometimes useful to define an arbitrary *orientation*, consisting in the (arbitrary) assignment of directions to its edges.

Graph Properties

- **Completeness**: A graph is *complete* if an undirected edge joins every agent pair in the system. We denote the complete graph $\mathcal{K}_N$

- **Paths and Connectivity**:

  - *Directed Path*: A sequence of edges in a directed graph.
  - *Undirected Path*: A path in an undirected graph.

  - *Strong Connectivity*: Directed graph with a directed path between every node pair.
  - *Connectivity*: Undirected graph with a path between every pair of distinct nodes.

- **Trees, Spanning Trees, and Cycles**:
  - (Directed) *Tree*: A (directed) graph in which every node has one *parent* except for the *root*.
  - (Directed) *Spanning Tree*: A (directed) tree containing all nodes.
  - (Directed) *Cycle*: A (directed) path where the initial and final nodes are the same.

As mentioned above, graphs are a useful tool for illustrating relationships among entities within a network. Besides their visual depiction using vertices and edges, graphs also benefit from an alternative representation via matrices, thus increasing the versatility of graph theory across various domains and applications. Some of these matrices will be now introduced.

Some preliminaries on algebraic graph theory

- **Adjacency matrix**:
  - The *binary adjacency matrix* $\boldsymbol{A} \in \{0,1\}^{N\times N}$ is defined as

$$[\boldsymbol{A}]_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} . \tag{2.2}$$

  - The *weighted adjacency matrix* is defined as

$$[\boldsymbol{A}]_{ij} = \begin{cases} w_{ij} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} . \tag{2.3}$$

- **Degree** of a given vertex, d_i, is the cardinality of the neighboring set $\mathcal{N}_i$. For a weighted undirected graph the weighted degree is given by $d_i = \sum_{j\in\mathcal{N}_i} w_{ij}$.
- **Degree matrix**:

- The unweighted *degree matrix* is given by $\boldsymbol{D} = \text{diag}(|\mathcal{N}_i|)$.
- The *weighted degree matrix* is given by $\boldsymbol{D} = \text{diag}(\boldsymbol{A}\boldsymbol{1}_N) = \text{diag}(d_i)$ where $\boldsymbol{A}$ is the corresponding weighted adjacency matrix.

- **Incidence matrix**: Given a graph with oriented edges, the *incidence matrix* $\boldsymbol{E}$ associated to it has elements

$$[\boldsymbol{E}]_{ik} = \begin{cases} -1 & \text{if } v_i \text{ is the tail of } e_k \\ 1 & \text{if } v_i \text{ is the head of } e_k \\ 0 & \text{otherwise} \end{cases} . \tag{2.4}$$

  The incidence matrix has zero column-sum, i.e. $\boldsymbol{1}_N^T \boldsymbol{E} = \boldsymbol{0}_N$. The incidence matrix $\boldsymbol{E}$ can be regarded as a map from edge-based variables in $\mathbb{R}^M$ to node-based variables in $\mathbb{R}^N$. Specifically, given an edge-based variable $\boldsymbol{z} \in \mathbb{R}^M$ and a node v_i,

$$[\boldsymbol{E}\boldsymbol{z}]_i = \sum_{k:i \text{ is the head of } e_k} z_k - \sum_{k:i \text{ is the tail of } e_k} z_k. \tag{2.5}$$

  When the edge-based variables $\boldsymbol{z}$ are flows along edges, then $[\boldsymbol{E}\boldsymbol{z}]_i$ is the algebraic sum of the flows outgoing from node i. Similarly, the transpose of the incidence matrix $\boldsymbol{E}^T$ maps node-based variables in $\mathbb{R}^N$ to edge-based variables in $\mathbb{R}^M$. Specifically, given a node-based variable $\boldsymbol{x} \in \mathbb{R}^N$ and an edge e_k of the form $e_k = (i, j)$,

$$[\boldsymbol{E}^T \boldsymbol{x}]_k = x_j - x_i. \tag{2.6}$$

- **Laplacian matrix** (of an *undirected* graph): This matrix plays a fundamental role in the analysis of continuous-time consensus algorithms. By definition

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \quad \in \mathbb{R}^{N \times N}, \tag{2.7}$$

  but it can also be written as:

$$\boldsymbol{L} = \boldsymbol{E}\boldsymbol{E}^T. \tag{2.8}$$

  The Laplacian is a positive semidefinite matrix with zero row-sum (and zero column-sum for an undirected graph), i.e. $\boldsymbol{L}\boldsymbol{1}_N = \boldsymbol{1}_N^T \boldsymbol{L} = \boldsymbol{0}_N$. Its eigenvalues are $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$. The zero eigenvalue λ_1 is associated to the eigenvector $\boldsymbol{v}_1 = \boldsymbol{1}_N$. The second smallest eigenvalue λ_2 is called *connectivity* (or *Fiedler*) *eigenvalue* and it is

strictly greater than zero $\lambda_2 > 0$ if and only if the undirected graph is connected. The associated eigenvector $\boldsymbol{v}_2$ is known as *Fiedler vector*. For a weighted graph, the Laplacian can be written as

$$\boldsymbol{L} = \boldsymbol{E}\boldsymbol{W}\boldsymbol{E}^T, \tag{2.9}$$

where $\boldsymbol{W} = \text{diag}(w_k)$ is a matrix of weights associated to each edge.

Frameworks

Let $\boldsymbol{q}_i \in \mathcal{M}$ denote the configuration of a robot in the considered space $\mathcal{M}$, e.g. for a robot evolving in $\mathbb{R}^3$ its configuration is simply given by its position $\boldsymbol{p}_i$. We also denote $\boldsymbol{q} \coloneqq \begin{bmatrix}\boldsymbol{q}_1^T & \dots & \boldsymbol{q}_N\end{bmatrix}^T$ as the stacked vector of the robots configuration. A framework is then defined by the pair $(\mathcal{G}, \boldsymbol{q})$, where $\boldsymbol{q} : \mathcal{V} \to \mathcal{M}$ maps each vertex to a point in $\mathcal{M}$.

Different assumptions can be made on the robot interaction and the corresponding graphs. They could be constant or time-varying (e.g. depending on sensing constraints such as limited range or field of view). The interaction graph can be either directed or undirected and distinct graphs can exist, for instance, for representing separately the sensing and the communication graphs. Indeed, in many cases of interest this distinction is needed since the communication graph is typically taken as *undirected*, while the sensing graph can either be directed or undirected. In the following chapters the specific assumptions considered for the sensing/communication graphs will be made clear.

2.3 Consensus

Consensus algorithms form the core of cooperative interactions in multi-agent systems. At its essence, the consensus problem involves devising distributed dynamics that enable all agents to converge to a shared value for a specific variable, using information from only a subset of (neighboring) agents.

In continuous-time consensus, the pivotal component is the Laplacian matrix. In its simplest formulation, assuming the graph is connected, the consensus problem is represented by the following *Laplacian flow* on $\mathbb{R}^N$:

$$\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x} \tag{2.10}$$

or, equivalently for each agent component,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i). \tag{2.11}$$

Here, a_{ij} is the ij-element of the adjacency matrix. Let us assume the graph is undirected. One way to assess the stability of the algorithm is by decomposing the state $\boldsymbol{x}$ in its *average* $\bar{x} = \frac{1}{N}\mathbf{1}_N^T\boldsymbol{x}$ and the *disagreement vector* $\boldsymbol{x}_\perp = \boldsymbol{x} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{x} = \boldsymbol{\Pi}_{1_N}\boldsymbol{x}$, where $\boldsymbol{\Pi}_{1_N} \coloneqq \boldsymbol{I} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ is an orthogonal projector (i.e. $\boldsymbol{\Pi}_{1_N} = \boldsymbol{\Pi}_{1_N}^T$ and $\boldsymbol{\Pi}_{1_N}\boldsymbol{\Pi}_{1_N} = \boldsymbol{\Pi}_{1_N}$). The disagreement vector corresponds to $\boldsymbol{x}$ after subtracting the average $\bar{x}$ from all its components. It is easy to show that the average is an invariant of system (2.10):

$$\dot{\bar{x}} = \frac{1}{N}\mathbf{1}_N^T\dot{\boldsymbol{x}} = -\frac{1}{N}\mathbf{1}_N^T\boldsymbol{L}\boldsymbol{x} = \mathbf{0}_N, \tag{2.12}$$

where we used the fact that for an undirected graph $\mathbf{1}_N^T\boldsymbol{L} = \mathbf{0}_N$ (i.e. the Laplacian has zero column sum). The dynamics of the disagreement vector are

$$\dot{\boldsymbol{x}}_\perp = -\boldsymbol{\Pi}_{1_N}\boldsymbol{L}\boldsymbol{x} = -\boldsymbol{L}\boldsymbol{\Pi}_{1_N}\boldsymbol{x} = -\boldsymbol{L}\boldsymbol{x}_\perp, \tag{2.13}$$

where we used the fact that $\boldsymbol{\Pi}_{1_N}\boldsymbol{L} = \boldsymbol{L}\boldsymbol{\Pi}_{1_N}$ [Bullo, 2020]. For this system, one can take a simple quadratic Lyapunov function $V = \frac{1}{2}\boldsymbol{x}_\perp^T\boldsymbol{x}_\perp$, whose derivative along the system trajectories is

$$\dot{V} = -\boldsymbol{x}_\perp^T\boldsymbol{L}\boldsymbol{x}_\perp \leq -\lambda_2\boldsymbol{x}_\perp^T\boldsymbol{\Pi}_{1_N}\boldsymbol{x}_\perp = -\lambda_2\boldsymbol{x}_\perp^T\boldsymbol{x}_\perp = -\lambda_2 V, \tag{2.14}$$

where we used the idempotence of the orthogonal projector i.e. $\boldsymbol{\Pi}_{1_N}\boldsymbol{x}_\perp = \boldsymbol{\Pi}_{1_N}\boldsymbol{\Pi}_{1_N}\boldsymbol{x} = \boldsymbol{\Pi}_{1_N}\boldsymbol{x} = \boldsymbol{x}_\perp$ and the fact that for a connected graph $\boldsymbol{L} \succeq \lambda_2\boldsymbol{\Pi}_{1_N}$ [Bullo, 2020]. This implies exponential stability of the *consensus manifold* $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^N : x_1 = x_2 = ... = x_N\}$. This proof relies on the graph being undirected; for directed graphs the proof is slightly more involved, see [Moreau, 2004] and Sect. 3.2 in [Mesbahi & Egerstedt, 2010]. In this thesis, the *sensing graphs* may be considered directed due to field of view limitations, which naturally induce asymmetry. However, it is important to note that the *communication graph* is always taken as undirected, thus ensuring the application of consensus algorithms and similar Laplacian flows on an undirected graph.

In Fig.2.1, we show a simulation example depicting a Laplacian flow (2.10) involving 15 nodes which perform consensus over their initial states, which are sampled from a

random distribution. All the states asymptotically converge to the average of the initial states.

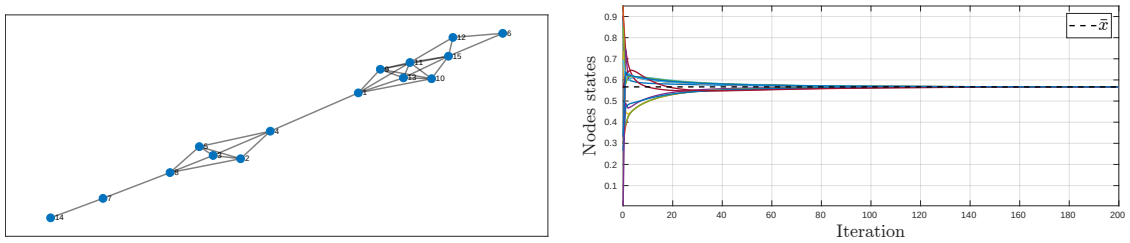


Figure 2.1: Simulation of consensus over a graph with 15 nodes. On the left, we show the graph $\mathcal{G}$, on the right, the evolution of the nodes states. As explained the state of all the nodes converge to their initial average value.

2.3.1 Dynamic average consensus

Consensus algorithms can be used to compute the average of local quantities. For example, for an undirected graph, (2.10) could be used to compute the average of the initial vector $\boldsymbol{x}(0)$. The algorithm in (2.10) falls under the category of static algorithms [Kia et al., 2019]. In these algorithms, the reference signal is introduced as the initial condition for the integrator state (see Fig. 2.2). However, because the reference signals act as initial conditions, static average consensus algorithms like (2.10) cannot track time-varying signals.

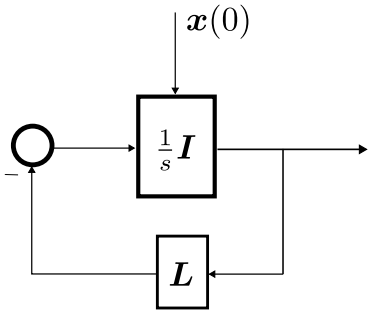


Figure 2.2: Block scheme of the static consensus algorithm.

Consider a scenario where each agent possesses a local scalar reference signal $u_i(t)$, the average of which needs to be tracked. This signal could represent an agent sensor output or the output of another algorithm the agent is executing. The dynamic average consensus problem involves designing an algorithm that enables individual agents to track

the time-varying average of these reference signals:

$$\bar{u}(t) = \frac{1}{N}\sum_{i=1}^{N} u_i(t) \tag{2.15}$$

Here, the reference inputs are introduced as external inputs into the system. Several algorithms exist for this purpose [F. Chen et al., 2012; Kia et al., 2019; Montijano, Montijano, et al., 2014; Zhu & Martınez, 2010], which are beyond the scope of this thesis. For an extensive discussion on this subject we refer the reader to [Kia et al., 2019]. Throughout the thesis, the dynamic average consensus used will primarily refer to the discrete-time *accelerated robust dynamic average consensus* proposed in [Van Scoy et al., 2015] and further detailed in [Kia et al., 2019]. We report below the corresponding equations:

$$\begin{cases} q^i_{k+1} = 2\rho q^i_k - \rho^2 q^i_{k-1} + k_p \sum_{j=1}^{N} a_{ij}((x^i_k - x^j_k) + (p^i_k - p^j_k)) \\ p^i_{k+1} = (1+\rho^2)p^i_k - \rho^2 p^i_{k-1} + k_I \sum_{j=1}^{N} a_{ij}(x^i_k - x^j_k) \\ x^i_k = u^i_k - q^i_k, \quad p^i_0, q^i_0 \in \mathbb{R}, \quad i \in \{1, ..., N\} \end{cases} \tag{2.16}$$

where k_p and k_I are, respectively, proportional and integral gains and ρ is the convergence rate to the average of the input signals $\bar{u}$ [Kia et al., 2019]. Moreover, $q^i_k, q^i_{k-1}, p^i_k, p^i_{k-1}$ are internal state variables.

2.4 Distance and bearing rigidity

In this section, we explore the fundamental concepts of *graph rigidity*—a fundamental property for multi-robot localization and formation control [Krick et al., 2009; Mao et al., 2007; Zhao & Zelazo, 2019].

Traditional cooperative control assumes the agents being able to access relative positions of their neighbors, often obtained through either GPS or a Motion Capture (Mocap) system and wireless communication. However, this approach fails in unstructured GPS-denied environments or when high precision is required. Instead, onboard sensors like cameras or Ultra Wideband Sensors (UWB) become crucial for agents to sense their neighbors [Montijano et al., 2016].

Yet, sensors like cameras or UWB provide only partial information about robot rela-

tive poses, typically bearing-only[1] or distance-only measurements. This prompts a critical question: under which conditions is it possible to reconstruct the spatial geometric arrangement of a network from the partial measurements of the onboard sensors? This is where distance and bearing rigidity theories find a useful application in a multi-robot context—they can be used to analyze when *fixed* edge distances or bearings (*locally*) uniquely determine a network geometric arrangement.

In the multi-robot field, the notion of rigidity has originally been applied to the case of relative distance measurements [B. D. Anderson et al., 2008; Krick et al., 2009; Oh & Ahn, 2014]. However, rigidity concepts have now been extended to include robots measuring relative bearings [Schiano et al., 2016; Zhao & Zelazo, 2019]. The goal of this Section is to present the main ideas of (infinitesimal) distance and bearing rigidity in a unified manner. The interested reader is referred to [Zhao & Zelazo, 2019] for delving deeper into this topic.

Consider the framework $(\mathcal{G}, \boldsymbol{q})$ (see Sect. 2.2) where inter-neighbor distances and bearings are respectively expressed as:

$$d_k = d_{ij} = \|\boldsymbol{p}_j - \boldsymbol{p}_i\| \quad k \in \{1, ..., M\} \tag{2.17}$$

and (see also Fig. 2.3)

$$\boldsymbol{\beta}_k = \boldsymbol{\beta}_{ij} = \boldsymbol{R}_i^T \frac{\boldsymbol{p}_{ij}}{d_{ij}} = \boldsymbol{R}_i^T \frac{\boldsymbol{p}_j - \boldsymbol{p}_i}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|} \quad k \in \{1, ..., M\}. \tag{2.18}$$

In this thesis, we examine two cases: $\boldsymbol{q} \in \mathbb{R}^3$ (i.e. the confiuration of the robot is

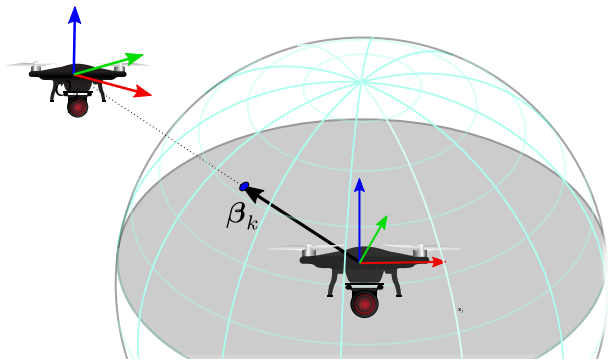


Figure 2.3: Relative bearing from the frame of robot i to the one of robot j.

1. In this Thesis the term bearing refers to a unit vector from the sensor to the target point.

represented by its position) and $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$ (i.e. the confiuration of the robot is represented by both its position and yaw orientation). The former is relevant when all bearings are expressed in a common frame, aligned with the world frame orientation, resulting in $\boldsymbol{R}_i = \boldsymbol{I}_3$. In this scenario, the underlying assumption is that the robots have access to their yaw rotations in a common frame, such as through a Mocap system or using the magnetometer, and no field of view constraints are considered. In contrast, the latter case $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$ assumes that the bearings are measured in the individual robot body frame $\mathcal{R}_i$ with $\boldsymbol{R}_i = \boldsymbol{R}_z(\psi_i) \in SO(3)$ representing a yaw rotation about a vertical axis. This case becomes relevant when robots either lack access to their yaw in a common frame, for example when the magnetometer becomes unreliable (e.g., indoors), or when the employed sensor has field of view limitations. The decision to not consider the pitch and roll angles is justified by the typical reliability of roll/pitch estimation from onboard Inertial Measurement Unit (IMU) measurements [Mahony et al., 2008]. Therefore, one can 'derotate' the measured bearings so that they are expressed in a 'flat' frame, having the xy-plane parallel to the world reference frame [Franchi et al., 2012]:

$$\boldsymbol{\beta}_{ij} = \boldsymbol{R}_y(\theta_i)\boldsymbol{R}_x(\phi_i)^{\mathcal{B}_i}\boldsymbol{\beta}_{ij}, \tag{2.19}$$

where $\mathcal{B}_i$ refers to the robot i body frame, including the roll and pitch angles, denoted as ϕ_i and θ_i, respectively and $\boldsymbol{R}_x(\cdot), \boldsymbol{R}_y(\cdot) \in SO(3)$ are the canonical rotation matrices about the x and y axis. Moreover, considering the relative bearing as a function of only position and yaw is especially interesting for quadrotors, as these correspond to the flat outputs of the systems. Consequently, quadrotors can track sufficiently smooth trajectories in position and yaw. Also note that, when working within $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$, there are instances where expressing bearings in the world frame becomes beneficial; in such cases, we denote them as $\bar{\boldsymbol{\beta}}_k$.

Define the *distance* $\boldsymbol{f}_d : \mathcal{V} \to \mathbb{R}^M$ and *bearing* $\boldsymbol{f}_b : \mathcal{V} \to \mathbb{R}^{3M}$ *functions* respectively as:

$$\boldsymbol{f}_d = \begin{bmatrix} d_1^2 \dots d_M^2 \end{bmatrix}^T \tag{2.20}$$

$$\boldsymbol{f}_b = \begin{bmatrix} \boldsymbol{\beta}_1^T \dots \boldsymbol{\beta}_M^T \end{bmatrix}^T \tag{2.21}$$

The *distance* and *bearing rigidity matrix* are defined respectively as the Jacobian of the distance and bearing functions

$$\boldsymbol{R}_d = \frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{q}} \tag{2.22}$$

$$\boldsymbol{R}_b = \frac{\partial \boldsymbol{f}_b}{\partial \boldsymbol{q}} \tag{2.23}$$

We will now derive the expression of the rigidity matrix for the case $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$. To reduce from $\mathbb{R}^3 \times \mathbb{S}^1$ to $\mathbb{R}^3$, one can simply remove the last column of the rigidity matrix and consider $\boldsymbol{R}_i = \boldsymbol{I}$.

It can be easily shown [Krick et al., 2009] that the distance rigidity matrix can be expressed as

$$\boldsymbol{R}_d = \begin{bmatrix} \text{diag}(\boldsymbol{p}_{ij}^T)\boldsymbol{E}_3^T & | & \boldsymbol{0}_{M\times N} \end{bmatrix} \in \mathbb{R}^{M\times 4N} \tag{2.24}$$

while the bearing rigidity matrix is

$$\boldsymbol{R}_b = \begin{bmatrix} \text{diag}\left(\frac{\boldsymbol{\Pi}_{\beta_{ij}}}{d_{ij}}\boldsymbol{R}_i^T\right)\boldsymbol{E}_3^T & | & -(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\text{diag}(\boldsymbol{\beta}_k)\boldsymbol{E}_{\otimes}^T \end{bmatrix} \in \mathbb{R}^{3M\times 4N}, \tag{2.25}$$

where $\boldsymbol{E}_3 = \boldsymbol{E} \otimes \boldsymbol{I}_3$, while $\boldsymbol{E}_{\otimes}$ is the *out-incidence* matrix, i.e.

$$[\boldsymbol{E}_{\otimes}]_{ik} = \begin{cases} 1 & \text{if node } i \text{ is the tail of edge } k \\ 0 & \text{otherwise} \end{cases}. \tag{2.26}$$

Furthermore, $\boldsymbol{S}_{e_3} = [\boldsymbol{e}_3]_\times$, with $\boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and $\boldsymbol{\Pi}_{\beta_{ij}} = \boldsymbol{I} - \boldsymbol{\beta}_{ij}\boldsymbol{\beta}_{ij}^T$ is an orthogonal projection matrix that geometrically projects any vector onto the orthogonal complement of $\boldsymbol{\beta}_{ij}$ (see Fig. 2.4). Interestingly, using the fact that

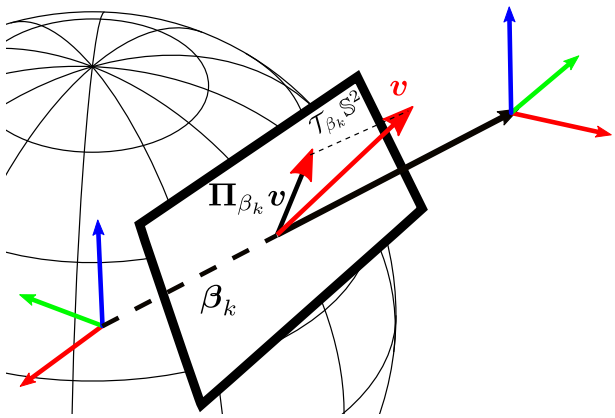


Figure 2.4: The orthogonal projector $\boldsymbol{\Pi}_{\beta_{ij}}$ projects any vector $\boldsymbol{v}$ onto the plane orthogonal to the the bearing β_{ij}.

$$\boldsymbol{S}_{e_3}\boldsymbol{\beta}_k = \boldsymbol{S}_{e_3}\boldsymbol{R}_i^T\bar{\boldsymbol{\beta}}_k = \boldsymbol{R}_i^T\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_k \tag{2.27}$$

Some useful properties of the orthogonal projector $\Pi_{\beta_{ij}}$

The orthogonal projection matrix plays an important role in the bearing rigidity theory and its applications. Here is a non-exaustive list of its properties, see [Zhao & Zelazo, 2019] for further details.

- $\boldsymbol{\Pi}_{\beta_{ij}}^T = \boldsymbol{\Pi}_{\beta_{ij}}$ (symmetric)
- $\boldsymbol{\Pi}_{\beta_{ij}}\boldsymbol{\Pi}_{\beta_{ij}} = \boldsymbol{\Pi}_{\beta_{ij}}$ (idempotent)
- $\ker(\boldsymbol{\Pi}_{\beta_{ij}}) = \text{span}(\boldsymbol{\beta}_{ij})$
- $\boldsymbol{\Pi}_{\beta_{ij}} \succeq 0$ (positive semidefinite)
- $\boldsymbol{\Pi}_{\beta_{ij}}$ has one eigenvalue equal to zero associated to the eigenvector $\boldsymbol{\beta}_{ij}$
- Any two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ satisfy $\boldsymbol{x}^T\boldsymbol{\Pi}_y\boldsymbol{x} = \boldsymbol{y}^T\boldsymbol{\Pi}_x\boldsymbol{y}$
- For any nonzero vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_M \in \mathbb{R}^d$ with $d \geq 2$, the matrix $\sum_{k=1}^M \boldsymbol{\Pi}_{x_k}$ is nonsingular if and only if at least two of $\boldsymbol{x}_1, ..., \boldsymbol{x}_M$ are not collinear

and

$$\boldsymbol{\Pi}_{\beta_k}\boldsymbol{R}_i^T = \boldsymbol{R}_i^T\boldsymbol{\Pi}_{\bar{\beta}_k}, \tag{2.28}$$

one can show that $\boldsymbol{R}_b$ can also be written as:

$$\boldsymbol{R}_b = \text{diag}(\boldsymbol{R}_i)\left[\text{diag}\left(\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}}\right)\boldsymbol{E}_3^T \quad | \quad -(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\text{diag}(\bar{\boldsymbol{\beta}}_k)\boldsymbol{E}_\otimes^T\right] \in \mathbb{R}^{3M\times 4N}. \tag{2.29}$$

This fact will be used later introducing the *symmetric rigidity matrix*.

We now introduce the concept of *infinitesimal rigidity*. Let $\delta\boldsymbol{q} \in \mathbb{R}^d$, where $d = 3$ if $\boldsymbol{q} \in \mathbb{R}^3$ and $d = 4$ if $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$, be a variation of the configuration $\boldsymbol{q}$. If $\boldsymbol{R}_*(\boldsymbol{q})\delta\boldsymbol{q} = \boldsymbol{0}$, with $* \in \{d, b\}$, then $\delta\boldsymbol{q}$ is called an *infinitesimal motion* of $(\mathcal{G}, \boldsymbol{q})$. An infinitesimal motion is said to be *trivial* if it would keep existing (as an infinitesimal motion) also in case of a complete graph. For a distance formation, an infinitesimal distance motion is *trivial* if it only corresponds to a translation and a rotation of the whole formation. While, for a bearing formation, an infinitesimal bearing motion is *trivial* if it only corresponds to a translation and a scaling of the whole formation. Furthermore, if $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$, then, a coordinated rotation involving an angular rotation of each agent about its own body axis

with a rigid-body rotation of the formation is also a trivial motion of the framework. We highlight the fact that, if the robots do not move, although the formation is infinitesimally bearing rigid, the scale of the formation (implying the inter robot distances as well) cannot be retrieved (i.e. it is not observable).

Definition 2.4.1 (Infinitesimal rigidity)**.** *A network is infinitesimally distance (resp. bearing) rigid at a configuration $\boldsymbol{q}_0$ if all the infinitesimal distance (resp. bearing) motions are trivial when $\boldsymbol{q} = \boldsymbol{q}_0$.*

Infinitesimal rigidity of a framework can be characterized by the rank of the bearing rigidity matrix.

Lemma 1. *A framework $(\mathcal{G}, \boldsymbol{q})$ is infinitesimally distance (resp. bearing) rigid if and only if rank$(\boldsymbol{R}_d) = 3N - \tau$ (resp. rank$(\boldsymbol{R}_b) = 3N - \tau$), where τ is the number of independent trivial motions.*

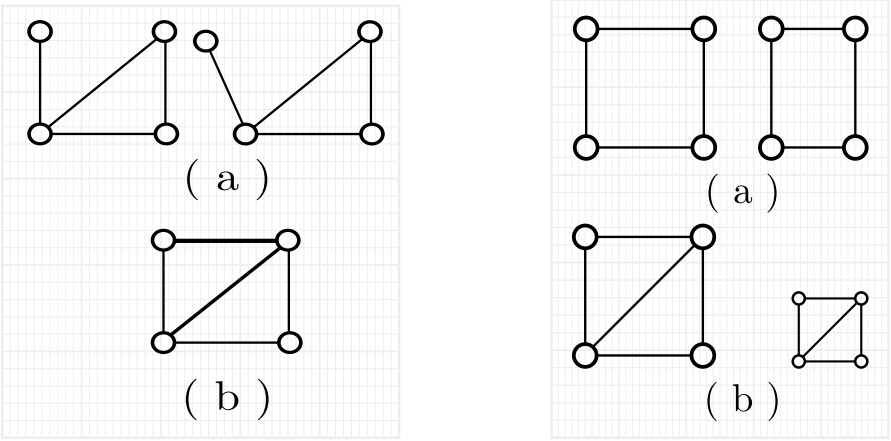


Figure 2.5: The two figures show two formations (a) and (b). On the left, we consider distance rigidity, while on the right we consider bearing rigidity. In both cases, formation (a) is not infinitesimally rigid while formation (b) is. On the left, formation (a) is not infinitesimally distance rigid because the node in the upper left corner is free to rotate around the node in the lower left corner without changing any relative distance. While for formation (b) no motion of the nodes is allowed without changing any relative distance except for the trivial motions. On the right, formation (a) can be stretched, e.g. horizontally as in the figure, without changing any relative bearing. Formation (b) instead being bearing rigid can only translate or expand/contract.

Both the bearing rigidity matrix and the distance rigidity matrix are not, in general, square matrices. Therefore, it is more convenient to express the rank condition in Lemma 1

in terms of the eigenvalues of the so-called *symmetric distance (resp. bearing) rigidity matrix* [Zelazo et al., 2015], which is the following symmetric and semipositive definite matrix

$$\boldsymbol{S}_d = \boldsymbol{R}_d^T \boldsymbol{R}_d \tag{2.30}$$

for distance measurements and

$$\boldsymbol{S}_b = \boldsymbol{R}_b^T \boldsymbol{R}_b \tag{2.31}$$

for bearing measurements. Notice that $\text{rank}(\boldsymbol{S}_d) = \text{rank}(\boldsymbol{R}_d)$ and analogously for bearings $\text{rank}(\boldsymbol{S}_b) = \text{rank}(\boldsymbol{R}_b)$. Hence, the rank condition of Lemma 1 can be equivalently stated in terms of the eigenvalues of $\boldsymbol{S}_d$ and $\boldsymbol{S}_b$. Denoting the eigenvalues of $\boldsymbol{S}_d$ and $\boldsymbol{S}_b$ as $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{dN}$, we note that, infinitesimal rigidity is equivalent to $\lambda_i = 0$ for $i \in \{1, 2, ..., \tau\}$ and $\lambda_{\tau+1} > 0$. Consequently, $\lambda_{\tau+1}$ is denoted as the *Rigidity Eigenvalue*. This fact was used in [Schiano & Giordano, 2017; Zelazo et al., 2015] along with perception weights encoding sensing constraints to maintain infinitesimal rigidity of the graph.

An important remark is that, for both $\boldsymbol{S}_d$ and $\boldsymbol{S}_b$, the block corresponding to the positions (i.e. the upper-left block) is a matrix-weighted Laplacian. In fact, we can decompose

$$\boldsymbol{S}_d = \boldsymbol{R}_d^T \boldsymbol{R}_d = \begin{bmatrix} \boldsymbol{E}_3 \text{diag}(\boldsymbol{p}_{ij}) \\ - \\ \boldsymbol{0}_{M\times N} \end{bmatrix} \begin{bmatrix} \text{diag}(\boldsymbol{p}_{ij}^T)\boldsymbol{E}_3^T & | & \boldsymbol{0}_{M\times N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}_3 \text{diag}(\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T)\boldsymbol{E}_3^T & \boldsymbol{0}_{3N\times N} \\ \boldsymbol{0}_{N\times 3N} & \boldsymbol{0}_{N\times N} \end{bmatrix} \tag{2.32}$$

where the upper left block is associated to the position part of the configuration vector $\boldsymbol{q}$, while the lower right block is associated to the yaw. Notice that $\boldsymbol{E}_3 \text{diag}(\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T)\boldsymbol{E}_3^T$ is a matrix-weighted Laplacian as in (2.9), with positive semidefinite weight matrix $\boldsymbol{W} = \text{diag}(\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T)$, this fact will be used in the following chapter to study the unobservable motions of distance formation.

Concerning the symmetric bearing rigidity matrix, instead, from (2.29) one has

$$\begin{aligned} \boldsymbol{S}_b &= \boldsymbol{R}_b^T \boldsymbol{R}_b \\ &= \begin{bmatrix} \boldsymbol{E}_3 \text{diag}\left(\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}}\right) \\ - \\ -\boldsymbol{E}_{\otimes} \text{diag}(\bar{\boldsymbol{\beta}}_k^T)(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}^T) \end{bmatrix} \text{diag}(\boldsymbol{R}_i^T)\text{diag}(\boldsymbol{R}_i) \begin{bmatrix} \text{diag}\left(\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}}\right)\boldsymbol{E}_3^T & | & -(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\text{diag}(\bar{\boldsymbol{\beta}}_k)\boldsymbol{E}_{\otimes}^T \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{E}_3 \text{diag}\left(\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\right)\boldsymbol{E}_3^T & -\boldsymbol{E}_3 \text{diag}\left(\frac{\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right)\boldsymbol{E}_{\otimes}^T \\ -\boldsymbol{E}_{\otimes} \text{diag}\left(\frac{\bar{\boldsymbol{\beta}}_{ij}^T \boldsymbol{S}_{e_3}^T}{d_{ij}}\right)\boldsymbol{E}_3^T & \text{diag}\left(\sum_{j\in\mathcal{N}_i^{out}} \bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) \end{bmatrix} \end{aligned} \tag{2.33}$$

where we used the idempotence of the orthogonal projector, the fact that, since $\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_k \perp \bar{\boldsymbol{\beta}}_k$, then $\boldsymbol{\Pi}_{\bar{\beta}_k}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_k = \boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_k$ and $\bar{\boldsymbol{\beta}}_k^T\boldsymbol{S}_{e_3}^T\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_k = \bar{\boldsymbol{\beta}}_k^T(\boldsymbol{I}_3 - \boldsymbol{e}_3\boldsymbol{e}_3^T)\bar{\boldsymbol{\beta}}_k = \bar{\beta}_{kx}^2 + \bar{\beta}_{ky}^2$, with $\bar{\beta}_{kx}$ (resp. $\bar{\beta}_{ky}$) being the x (resp.y) component of the bearing $\bar{\boldsymbol{\beta}}_k$. Also, notice that $\boldsymbol{E}_{\otimes}\boldsymbol{E}_{\otimes}^T$ is the *out-degree* matrix [Bullo, 2020].

Notice that, the matrix in the upper-left block $\boldsymbol{E}_3\text{diag}\left(\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\right)\boldsymbol{E}_3^T$ is a matrix-weighted Laplacian with positive semidefinite weights associated to each edge $\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}$. This matrix-weighted Laplacian will appear again in the subsequent chapters. An interesting (and surprising) fact is that $\boldsymbol{S}_b$ *does not depend* on the rotation matrices $\boldsymbol{R}_i$, which indeed appears nowhere. In the following chapters, the orientation of the robots will become important from an observability point of view only when considering field of view limitations.

The following lemma for matrix-weighted Laplacians provides insights on the trivial motions of the formation and it will be used in the following chapters.

Lemma 2. *[Trinh et al., 2018] A matrix-weighted Laplacian $\boldsymbol{L} \in \mathbb{R}^{Nd\times Nd}$ with positive semidefinite weights $\boldsymbol{W}_{ij}$ is symmetric, positive semidefinite and has the nullspace* $\mathcal{N}(\boldsymbol{L}) = \text{span}\{(\boldsymbol{1}_N \otimes \boldsymbol{I}_d), \{\boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_1^T & \dots & \boldsymbol{v}_N^T\end{bmatrix}^T \in \mathbb{R}^{Nd} | (\boldsymbol{v}_j - \boldsymbol{v}_i) \in \mathcal{N}(\boldsymbol{W}_{ij}), \quad \forall (i,j) \in \mathcal{E}\}\}$.

For example, consider a framework with robot configuration $\boldsymbol{q}$ in $\mathbb{R}^3$. In this case, $\boldsymbol{S}_d = \boldsymbol{E}_3\text{diag}(\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T)\boldsymbol{E}_3^T$ is a matrix-weighted Laplacian. The nullspace of $\boldsymbol{S}_d$ is given by $\boldsymbol{1}_N \otimes \boldsymbol{I}_3$, spanning a common translation, and by $\boldsymbol{p}_\perp = (\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_k})\,\boldsymbol{p}$, spanning a common rotation around the unit vector $\boldsymbol{e}_k$, with $k \in \{1,2,3\}$. In fact, one can verify this by checking the condition in Lemma 2

$$(\boldsymbol{p}_{\perp j} - \boldsymbol{p}_{\perp i}) = (\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_k})\,(\boldsymbol{p}_j - \boldsymbol{p}_i) = (\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_k})\,\boldsymbol{p}_{ij} \tag{2.34}$$

and, since $(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_k})$ is skew symmetric,

$$\boldsymbol{W}_{ij}(\boldsymbol{p}_{\perp j} - \boldsymbol{p}_{\perp i}) = (\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T)\,(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_k})\,\boldsymbol{p}_{ij} = \boldsymbol{0}, \tag{2.35}$$

i.e. it corresponds to the null space of $\boldsymbol{W}_{ij} = \boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T$, which is the matrix weight in $\boldsymbol{S}_d$. One can repeat a similar procedure for the block of the symmetric bearing rigidity matrix corresponding to the positions of the robots to show that a scale expansion/contraction is a trivial motion for a bearing formation.

2.4.1 Exploring the connection between rigidity and observability

Both distance and bearing rigidity theories tackle the common problem of determining the unique geometric arrangement of a network. However, they differ in their focus: bearing rigidity examines inter-neighbor bearings, while distance rigidity deals with inter-neighbor distances. In bearing rigidity theory, the term 'unique arrangement' implies that the network location can be established up to a translational and scaling factor (along with a coordinated rotation if $\boldsymbol{q} \in \mathbb{R}^3 \times \mathbb{S}^1$). Conversely, in distance rigidity theory, 'unique arrangement' denotes determining the network up to translational and rotational factors.

Relation with the observability matrix

Infinitesimal rigidity can be viewed as an observability attribute of the network. Indeed, the connection between infinitesimal bearing rigidity in $\mathbb{R}^3 \times \mathbb{S}^1$ and system observability was explored in [Schiano & Tron, 2018].

As discussed in the previous chapter, when studying observability in a nonlinear system, a common approach involves assessing the rank of the observability matrix, $\frac{\partial \boldsymbol{\mathcal{O}}}{\partial \boldsymbol{x}}$, which maps state, input, and input derivatives to the output and its derivatives. However, in the context of infinitesimal rigidity, we solely consider the mapping from the state to the output without derivatives. The rank condition in Lemma 1 helps ascertain if the robot poses are observable solely from current measurements, except for the subspaces spanned by trivial motions. This stringent requirement enables the formation shape to remain observable even when the robots are static, which is a more stringent criterion than mere observability (which, in general, requires presence of a suitable robot motion for allowing a successful estimation of the state). If the robots in a connected sensing graph move so as to fulfill the observability rank condition (1.8), rigidity might no longer be necessary to retrieve the formation shape from available measurements. Moreover, additional information about the state such as the formation scale could also be retrieved [Schiano & Tron, 2018; Spica & Robuffo Giordano, 2016].

Relation with the observability gramian

Another connection among infinitesimal rigidity and observability is related to the OG. In fact, the symmetric rigidity matrices are the input terms in the OG (as well as in the CG) dynamics (1.18). To illustrate this point, consider a scenario where $\boldsymbol{q} \in \mathbb{R}^3$. Here,

the robots are modeled as single integrators, leading to $\boldsymbol{A}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \boldsymbol{0}$, while, for distance measurements, $\boldsymbol{C}(t) = \boldsymbol{R}_d(t)$. Consequently, the OG dynamics becomes:

$$\dot{\boldsymbol{G}}_o(t) = -\boldsymbol{A}(t)^T \boldsymbol{G}_o(t) - \boldsymbol{G}_o(t)\boldsymbol{A}(t) + \boldsymbol{C}^T(t)\boldsymbol{C}(t) = \boldsymbol{R}_d(t)^T \boldsymbol{R}_d(t) \tag{2.36}$$

This implies that, in this case, the OG essentially is the integral of the symmetric rigidity matrix over time.

Relation with the Fisher information matrix

Furthermore, as it was shown in [Cano & Le Ny, 2023], the FIM corresponding to the information collected at time t is a weighted symmetric rigidity matrix, e.g. for distance measurements one has

$$\boldsymbol{F}_d = \boldsymbol{R}_d^T \boldsymbol{R}^{-1} \boldsymbol{R}_d \tag{2.37}$$

where $\boldsymbol{R}$ is a diagonal matrix in which each element on the diagonal corresponds to the covariance of the corresponding measurement. This connection accentuates the strong relationship between the information obtained from measurements and the underlying geometric structure encoded within the symmetric rigidity matrix.

Conservativeness of infinitesimal rigidity

Infinitesimal rigidity is mainly a condition on the sensing graph topology rather than on the robots configuration, albeit with potential isolated singularities [Briot & Robuffo Giordano, 2019; Erskine et al., 2023]. It is worth noting that different robot configurations, representing various geometries of the formation, can impact the robustness to perturbations in different estimation and control algorithms [Cano & Le Ny, 2023; Le Ny & Chauvière, 2018]. While rigidity provides a useful framework to design simple control and localization algorithms with guaranteed convergence, in practical scenarios, ensuring infinitesimal rigidity might require a highly connected sensing graph. This can represent a strong constraint for the robot motion, especially when sensors have limitations such as field of view (FoV) and/or restricted range (see Fig. 2.6), and even more when considering robots navigating in environments with obstacles. A part of this thesis will explore the use of active sensing strategies that allow to localize the formation while greatly relaxing the (more common) rigidity assumption. The sensing limitations will be included in the active sensing control through suitable weight functions which are described below.

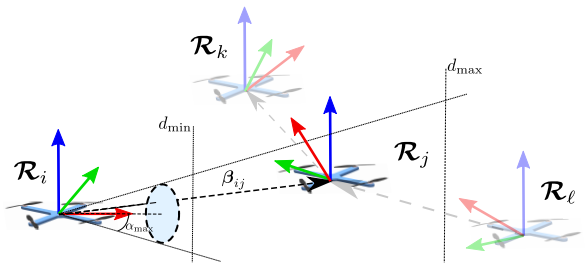


Figure 2.6: The figure shows a drone with maximum field of view angle $\alpha_{\max}$ and range limits $d_{\min}$ and $d_{\max}$

Sensing constraint and perception weights

In thesis, we will consider range constraints, expressed as $d_{\min} \leq d_{ij} \leq d_{\max}$, where $d_{ij} = \|\boldsymbol{p}_j - \boldsymbol{p}_i\|$ and field of view constraints $c_{ij} = \boldsymbol{\beta}_{ij}^T \boldsymbol{e}_i \geq c_{\min}$ where $\boldsymbol{\beta}_{ij}$ is the relative body-frame bearing and $\boldsymbol{e}_i$ is the unit vector representing the robot axis to which the camera is assumed to be aligned with. Notice that c_{ij} is the cosine of the angle α_{ij} between the bearing and the camera axis. We consider weights $w_{ij} = w_{d_{ij}} w_{b_{ij}}$, where $w_{d_{ij}}$ (resp. $w_{b_{ij}}$) is a continuously differentiable weight which smoothly vary from 1 to 0 as the distance (resp. FoV) limit of the sensor is being approached.

Throughout the thesis, these weights will be used in two cases: (i) to weight the information acquired in the OG (see (1.33)) or (ii) weighting the graph Laplacian for connectivity maintenance purposes (see (2.9)).

When weighting the acquired information, we consider weights that do not exactly vanish at the sensing limit. This choice allows for a small but non-zero gradient out of the sensing limits, which may be beneficial in some cases. Since we consider information acquired over time, we do not require the robots to be within their relative sensing limits at all times. Examples of such weight functions are

$$w_{d_{ij}} = \begin{cases} e^{-\frac{(d_{ij}-d_{\min})^2}{2\sigma_{d_{\min}}}}, & \text{if } d_{ij} \leq d_{\min} \\ 1, & \text{if } d_{\min} \leq d_{ij} \leq d_{\max} \\ e^{-\frac{(d_{ij}-d_{\max})^2}{2\sigma_{d_{\max}}}}, & \text{if } d_{ij} \geq d_{\max} \end{cases} \tag{2.38}$$

$$w_{b_{ij}} = \begin{cases} e^{-\frac{(\cos(\alpha_{\max})-\boldsymbol{\beta}_{ij}^T \boldsymbol{e}_i)^2}{2\sigma_b}}, & \text{if } \boldsymbol{\beta}_{ij}^T \boldsymbol{e}_i \leq \cos(\alpha_{\max}) \\ 1, & \text{if } \boldsymbol{\beta}_{ij}^T \boldsymbol{e}_i \geq \cos(\alpha_{\max}) \end{cases}$$

are two bell-shaped functions, as shown in Fig. 2.7, and the σ_* are standard deviations.

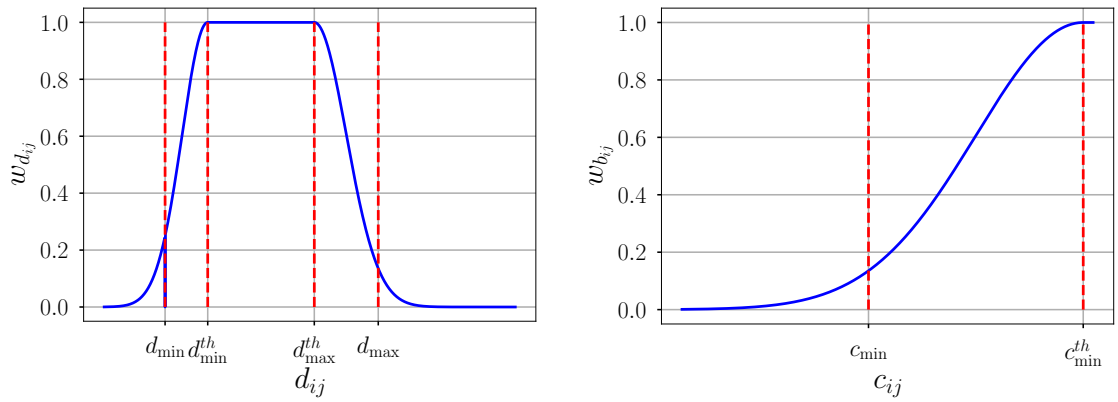


Figure 2.7: Weight functions for sensing constraints. On the left distance weight and on the right FoV weight (notice that $c_{ij} \leq 1$). This functions does not exactly vanish at the sensing limit.

Conversely, for connectivity maintenance, as we require at least a spanning tree of robots within their relative sensing limits at all times, we consider weights that exactly vanish as the limit is approached. In particular, we consider for $w_{d_{ij}}$ and $w_{b_{ij}}$ the following smooth step functions. Consider the following two functions $\zeta_l(x, x_{\min}, x^{th}_{\min}) = (x - x_{\min})/(x^{th}_{\min} - x_{\min})$ and $\zeta_u(x, x_{\max}, x^{th}_{\max}) = (x - x^{th}_{\max})/(x_{\max} - x^{th}_{\max})$, the weights are given by (see Fig. 2.8):

$$w\left(\zeta_l, \zeta_u\right) = \begin{cases} w = 0 & \text{if} \quad \zeta_l < 0 \\ w = 6\zeta_l^5 - 15\zeta_l^4 + 10\zeta_l^3 & \text{if} \quad 0 < \zeta_l < 1 \\ w = 1 & \text{if} \quad \zeta_l > 1 \quad \text{and} \quad \zeta_u < 0 \\ w = 1 - 6\zeta_u^5 + 15\zeta_u^4 - 10\zeta_u^3 & \text{if} \quad 0 < \zeta_u < 1 \\ w = 0 & \text{if} \quad \zeta_u > 1 \end{cases} . \tag{2.39}$$

These weights make the robots aware of the potential loss of connectivity with neighboring robots. In this way, when it is necessary, they can prevent loosing the connection employing, for example, *control barrier functions*, which will be discussed in the next chapter and which allow to enforce satisfaction of state constraints.

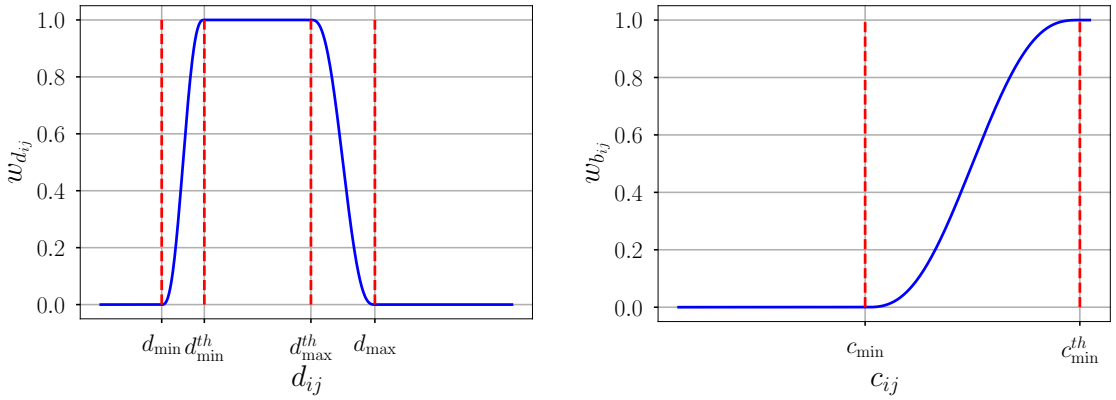


Figure 2.8: Weight functions for sensing constraints. On the left distance weight and on the right FoV weight (notice that $c_{ij} \leq 1$).

Chapter 3

Control Lyapunov Functions and Control Barrier Functions

This chapter serves as an introduction to Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs), two increasingly popular tools which allow to optimize system performance while ensuring specific constraints [Ames et al., 2019; Garg et al., 2023; Xiao et al., 2023]. While CLFs focus on enforcing stability of an equilibrium through the decrease of an energy-like function encoding the task objective, CBFs are dedicated to ensuring that the system state remains constrained within a prescribed set termed the *safe set*.

We begin by exploring barrier functions and reciprocal barrier functions, which guarantee constraint satisfaction by employing functions that approach infinity near the boundaries of a safe set and smoothly decrease within it. Originally utilized in optimization to penalize constraint violation [Nocedal & Wright, 1999], these functions have found widespread use in control and verification literature due to their natural alignment with Lyapunov-like functions [Tee et al., 2009; Wieland & Allgöwer, 2007], their ability to establish set invariance, and their relevance in multi-objective control scenarios [Panagou et al., 2013; Restrepo et al., 2022; Robuffo Giordano et al., 2013].

Next, we introduce CBFs, more precisely termed *zeroing control barrier functions* (ZCBFs), which have the property of also being well-defined *outside* the safe set. Notably, ZCBFs not only enforce forward invariance but also demonstrate robustness against model perturbations by ensuring the asymptotic stability of the safe set as well as input-to-state stability [Jankovic, 2018; Kolathaya & Ames, 2018; Xu et al., 2015]. For systems which are affine in the control, an input satisfying the CBF condition can be efficiently found by solving a Quadratic Program (QP), thus facilitating consideration of multiple control objectives.

Following this, we present CLFs leveraging Lyapunov functions with inequality constraints on their derivatives to devise classes of controllers stabilizing a system. Then,

we show how one can also consider CBFs and CLFs in a unified QP, optimizing system performance while ensuring the CBF inequality and respecting as much as possible the prescribed CLF decrease. Lastly, we explore CBFs tailored to high-order and distributed systems, extending their applicability in complex systems scenarios.

Extensive use of the concepts of CBFs and CLFs will be made throughout the thesis. For example, they will be used to achieve a certain task while maximizing as much as possible the information about the state of the system, or viceversa to maintain the collected information above a certain threshold. Also, they will be used to preserve connectivity of the graph as well as for collision avoidance.

Key references that have contributed to the general development of these sections include [Ames et al., 2019; Xiao et al., 2023]. Works more specifically related to the specific sections will be pointed out where relevant.

3.1 Barrier functions in constrained optimization

In the field of mathematics known as constrained optimization, a *barrier function* is a continuous function whose value increases to infinity at the boundary of the feasible region (i.e., the set of solutions that satisfy the constraints) in a given optimization problem [Nocedal & Wright, 1999]. Consider the constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}} . \quad & J(\boldsymbol{x}) \\ \text{s.t.} \quad & h(\boldsymbol{x}) \leq 0 \end{aligned} \tag{3.1}$$

The inequality constraint in (3.1) makes the solution of the problem non differentiable, because the constraint gets suddenly activated when $h(\boldsymbol{x}) = 0$. We can *approximate* the original optimization problem (3.1) as a continuously differentiable unconstrained problem

$$\min_{\boldsymbol{x}} . \quad J(\boldsymbol{x}) + \mu c(h(\boldsymbol{x})) \tag{3.2}$$

where the function $c(\cdot)$ is the barrier function. Several choices of $c(\boldsymbol{x})$ are possible. A very common one is the *logarithmic barrier function* $c(\boldsymbol{x}) = -\log(-h(\boldsymbol{x}))$. The parameter $\mu > 0$ is a tuning parameter; smaller values of μ result in a closer approximation of the original problem.

3.2 Barrier functions for dynamical systems

Consider a nonlinear system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{3.3}$$

with $\boldsymbol{x} \in \mathbb{R}^d$ and a safe set $\mathcal{C}$ defined by the superlevel set of a continuously differentiable function $h : \mathbb{R}^d \to \mathbb{R}$, i.e. (see also Fig. 3.1)

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^d | h(\boldsymbol{x}) \geq 0\} \tag{3.4}$$

$$\partial\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^d | h(\boldsymbol{x}) = 0\} \tag{3.5}$$

$$\text{Int}(\mathcal{C}) = \{\boldsymbol{x} \in \mathbb{R}^d | h(\boldsymbol{x}) > 0\} \tag{3.6}$$

Here, $\partial\mathcal{C}$ denotes the boundary and $\text{Int}(\mathcal{C})$ the interior of $\mathcal{C}$. We assume that $\frac{\partial h}{\partial \boldsymbol{x}} \neq \boldsymbol{0}$ for all $\boldsymbol{x} \in \partial\mathcal{C}$.

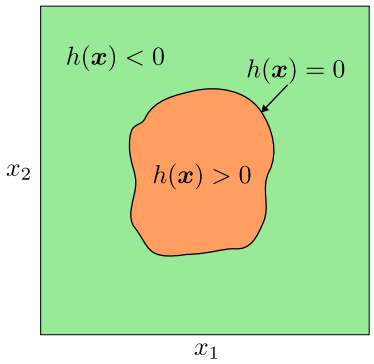


Figure 3.1: The figure shows an example of a compact safe set for a planar system.

The safety of system (3.3) is defined as:

Definition 3.2.1 (Forward invariance & safety)**.** *[Ames et al., 2019] The set $\mathcal{C}$ is forward invariant if for every $\boldsymbol{x}_0 \in \mathcal{C}$, $\boldsymbol{x}(t) \in \mathcal{C}$ for $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and all $t \geq 0$. The system (3.3) is safe with respect to the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant (see also Fig. 3.2).*

At this point, we introduce a fundamental result known as Nagumo's theorem. It states that if system (3.3) is initially safe, i.e., $h(\boldsymbol{x}(0)) \geq 0$, then it will always be safe, i.e., $h(\boldsymbol{x}(t)) \geq 0$, $\forall t \geq 0$, if and only if

$$h(\boldsymbol{x}) = 0 \quad \Longrightarrow \quad \dot{h}(\boldsymbol{x}) \geq 0 \tag{3.7}$$

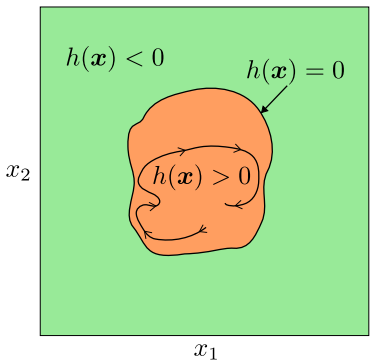


Figure 3.2: The figure shows a safe planar system. The line inside the set represents the system trajectory and the arrows along the line represent the time direction.

that is, the safe set $\mathcal{C}$ is forward invariant. If $h(\boldsymbol{x})$ satisfies this condition, then we call $h(\boldsymbol{x})$ a barrier function for system (3.3).

However, this safety condition is only defined at the boundary of the safe set, potentially creating discontinuities similar to those seen in the barrier function of optimization problems. The question then is: how to explicitly find a safe controller satisfying the conditions of Nagumo's theorem? Approaches to address this include those relying on (*i*) a function going to infinity (also known as *reciprocal control barrier functions* (RCBF)) [Bechlioulis & Rovithakis, 2008; Tee et al., 2009] or (*ii*) Control Barrier Function (CBF) (also known as *zeroing control barrier functions*) [Ames et al., 2014]. Both offer safety conditions that introduce some conservativeness, potentially limiting the system performance while allowing the use of locally Lipschitz continuous controllers. The fundamental concept behind reciprocal barrier functions is analogous to what is employed in optimization. It entails the definition of an *energy-like* barrier function, which remains bounded within the set and continuously increases towards infinity as it approaches the boundary of the safe set. Constraint satisfaction follows by ensuring the boundedness of the barrier function. For CBFs, given a constraint function whose zero super-level sets define the constrained set, the core concept involves limiting the rate of change of the barrier function along system trajectories. This is achieved by employing a class $\mathcal{K}$ function associated with the barrier function.

As a concluding remark, RCBF represent a more conservative approach in which the system is actively repelled from the boundary, as opposed to being just slowed down in its approach to the boundary [Krstic, 2023].

3.2.1 Reciprocal barrier functions

Motivated by the barrier method in optimization, a similar approach has been adopted in the control literature [Tee et al., 2009]. Consider, for example, the logarithmic candidate barrier function shown in Fig. 3.3

$$B(\boldsymbol{x}) = -\log\left(\frac{h(\boldsymbol{x})}{1+h(\boldsymbol{x})}\right). \tag{3.8}$$

This function satisfies the important properties

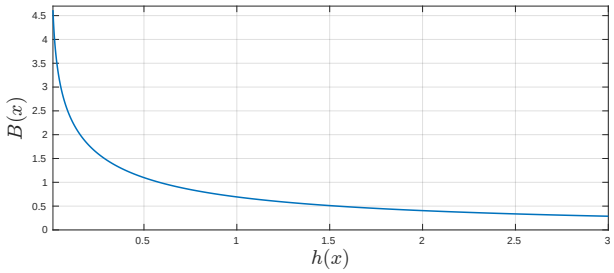


Figure 3.3: Logarithmic barrier function.

$$\inf_{\boldsymbol{x}\in\mathrm{Int}(\mathcal{C})} B(\boldsymbol{x}) \geq 0 \qquad \lim_{\boldsymbol{x}\to\partial\mathcal{C}} B(\boldsymbol{x}) = +\infty \tag{3.9}$$

By ensuring boundedness of B, Nagumo's condition is met. Trajectories starting inside $\mathcal{C}$ will remain within, ensuring the system safety. To achieve the desired task, a viable strategy might involve combining such a barrier function with a candidate Lyapunov function such that their combination is positive definite around the desired equilibrium (see also *recentered barrier functions* [Feller & Ebenbauer, 2015; Restrepo et al., 2022]). Ensuring the negativity of the derivative of the composite Lyapunov function ensures system stability while maintaining safety.

Limitations of such barrier functions include the fact that (i) they require large control actions near the safety-set boundary, making them sensitive to noise, (ii) they are not defined outside the set, thus preventing the possibility of approaching the safe set from external regions.

3.2.2 Control barrier functions

We now introduce CBFs. We begin with the following definition

Definition 3.2.2 ((Extended) Class $\mathcal{K}$ function)**.** *A continuous function $\alpha : [0, a) \to [0, \infty)$, with $a > 0$, is a class $\mathcal{K}$ function if it is strictly increasing and $\alpha(0) = 0$. Moreover, $\alpha : (-b, a) \to (-\infty, \infty)$, with $a, b > 0$ is an extended class $\mathcal{K}$ function if it is strictly increasing and $\alpha(0) = 0$.*

We now limit our attention to control-affine systems:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}, \tag{3.10}$$

with $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{u} \in \mathbb{R}^m$, while $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$ and $\boldsymbol{G} : \mathbb{R}^d \to \mathbb{R}^{d \times m}$ are locally Lipschitz continuous functions.

Definition 3.2.3 (Control barrier function)**.** *[Ames et al., 2019] Let $\mathcal{C}$ be defined by* (3.4)*. $h(\boldsymbol{x})$ is a control barrier function (CBF) for the system* (3.10) *if there exists a locally Lipschitz extended class $\mathcal{K}$ function α such that:*

$$\sup_{\boldsymbol{u} \in \mathbb{R}^m} [L_f h(\boldsymbol{x}) + L_G h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x}))] \geq 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n \tag{3.11}$$

where L_f, L_G denote the Lie derivatives along $\boldsymbol{f}$ and $\boldsymbol{G}$, respectively. It is assumed that $L_G h(\boldsymbol{x}) \neq \boldsymbol{0}$ when $h(\boldsymbol{x}) = 0$.

The assumption $L_G h(\boldsymbol{x}) \neq \boldsymbol{0}$ ensures that the CBF is not subject to a singularity (i.e., $L_G h(\boldsymbol{x}) = \boldsymbol{0}$) at the boundary of the safe set. As long as this assumption holds, a control $\boldsymbol{u}$ can be determined through (3.11) so as to preserve forward invariance in terms of $\mathcal{C}$.

Theorem 2. *[Ames et al., 2019] Given a CBF $h(\boldsymbol{x})$ with the associated set $\mathcal{C}$ from* (3.4)*, any Lipschitz continuous controller $\boldsymbol{u}(t)$, $\forall t \geq 0$, that satisfies* (3.11) *renders the set $\mathcal{C}$ forward invariant for* (3.10)*.*

Remark 1. *[Ames et al., 2019] It is important to stress that this result not only guarantees that the safe set $\mathcal{C}$ is invariant, but makes the set $\mathcal{C}$ asymptotically stable. This has very important consequences for what concerns practical implementations. Indeed, while a system will not formally leave the safe set $\mathcal{C}$, noise and modeling errors may make the system leave this set. As a result of the main CBF theorem, controllers satisfying* (3.11) *will then be able to drive the system back to the set $\mathcal{C}$, thus adding a layer of* robustness *to the CBF machinery.*

Having established that control barrier functions give (necessary and sufficient) conditions on safety, the question becomes: how does one synthesize controllers which achieve

the desired task while preserving safety ? Importantly, the aim is to do so in a minimally invasive fashion, i.e., by modifying an existing controller in a minimal way so as to guarantee safety. This naturally leads to optimization-based control design. It is very important to notice that the inequality (3.11) is linear in $\boldsymbol{u}$. This is crucial, as it leads to computationally efficient methods to find an input $\boldsymbol{u}$ satisfying the inequality (3.11) as the solution of a QP.

Consider the scenario in which a Lipschitz continuous feedback controller $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ for the control system (3.10) is given, but $\boldsymbol{k}(\boldsymbol{x})$ does not meet the safety requirements. In order to minimally modify this controller for ensuring safety, one can consider the following QP-based optimization problem meant to find the smallest change on $\boldsymbol{u}$ that can ensure safety

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathbb{R}^m} \; & \frac{1}{2}\left\|\boldsymbol{u}-\boldsymbol{k}(\boldsymbol{x})\right\|_2^2 \\ \text{s.t.} \quad & L_G h(\boldsymbol{x})\boldsymbol{u} + L_f h(\boldsymbol{x}) \geq -\alpha(h(\boldsymbol{x})) \end{aligned} . \tag{3.12}$$

The QP (3.12) is feasible (i.e. it admits a solution) as long as $L_f h(\boldsymbol{x}) + \alpha(h(\boldsymbol{x})) > 0$ whenever $L_G h(\boldsymbol{x}) = \boldsymbol{0}$. Hence, assuming the QP is feasible, the KKT optimality conditions (3.15) imply that (see the Box in the next page for a basic introduction to QPs)

$$\begin{aligned} &\boldsymbol{u}^* = \boldsymbol{k}(\boldsymbol{x}) + \lambda^* L_G h(\boldsymbol{x})^T \\ &L_G h(\boldsymbol{x})\boldsymbol{u}^* + L_f h(\boldsymbol{x}) \geq -\alpha(h(\boldsymbol{x})) \\ &\lambda = 0 \quad \text{if} \quad L_G h(\boldsymbol{x})\boldsymbol{u}^* + L_f h(\boldsymbol{x}) > -\alpha(h(\boldsymbol{x})) \\ &\lambda^* \geq 0 \end{aligned} \tag{3.16}$$

where $(\boldsymbol{u}^*, \lambda^*)$ denotes the optimal solution. Also, we used the fact that the least squares cost $\frac{1}{2}\left\|\boldsymbol{u}-\boldsymbol{k}(\boldsymbol{x})\right\|_2^2$ corresponds to a quadratic function with $\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{c} = -\boldsymbol{k}(\boldsymbol{x})$.

If the constraint is not active, we have $\lambda^* = 0$ and, hence, $\boldsymbol{u}^* = \boldsymbol{k}(\boldsymbol{x})$ which is the nominal feedback. Otherwise, if the constraint is active, substituting the first equation of the KKT into the second on (3.16) leads to

$$L_G h(\boldsymbol{x})(\boldsymbol{k}(\boldsymbol{x}) + \lambda^* L_G h(\boldsymbol{x})^T) + L_f h(\boldsymbol{x}) = -\alpha(h(\boldsymbol{x})) \tag{3.17}$$

from which

$$\lambda^* = -\frac{(L_G h(\boldsymbol{x})\boldsymbol{k}(\boldsymbol{x}) + L_f h(\boldsymbol{x}) + \alpha(h(\boldsymbol{x})))}{L_G h(\boldsymbol{x}) L_G h(\boldsymbol{x})^T}. \tag{3.18}$$

Notice that, by convention, $L_G h(\boldsymbol{x})$ is assumed to be a row vector, hence, counterin-

Basic concepts in QP optimization

Consider a QP:

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathbb{R}^m} \ . \quad & \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \end{aligned} \tag{3.13}$$

with $\boldsymbol{Q} \succ \boldsymbol{0} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{a}_1 & \dots & \boldsymbol{a}_m\end{bmatrix}^T \in \mathbb{R}^{m\times n}$. The lagrangian is given by:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \tag{3.14}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The *Karush–Kuhn–Tucker* (KKT) optimality conditions are first order necessary conditions for optimality of nonlinear programs taking into account both equality and inequality constraints [Nocedal & Wright, 1999]. For a QP, the KKT optimality conditions are:

$$\begin{aligned} &\nabla_x\mathcal{L} = \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda} = \boldsymbol{0} \\ &\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ &\boldsymbol{\lambda} \geq \boldsymbol{0} \\ &\boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = 0 \end{aligned} \tag{3.15}$$

where the last constraint is a *complementary slackness* condition, which states that if one of the constraints is inactive (i.e. at the optimal solution $\boldsymbol{x}^*$ strict inequality holds $\boldsymbol{a}_i^T\boldsymbol{x}^* < b_i$) then the corresponding multiplier $\lambda_i = 0$. Notice that, since $\boldsymbol{Q} \succ \boldsymbol{0}$, the QP is strictly convex and hence the optimal solution $\boldsymbol{x}^*$ is unique and the KKT conditions are a necessary and *sufficient* condition for optimality.

tuitively, $L_G h(\boldsymbol{x}) L_G h(\boldsymbol{x})^T$ is a scalar. Using (3.18) in the first equation of the KKT conditions (3.16) gives

$$\boldsymbol{u}^* = \boldsymbol{k}(\boldsymbol{x}) - \frac{L_G h(\boldsymbol{x})\boldsymbol{k}(\boldsymbol{x}) + L_f h(\boldsymbol{x}) + \alpha(h(\boldsymbol{x}))}{L_G h(\boldsymbol{x}) L_G h(\boldsymbol{x})^T} L_G h(\boldsymbol{x})^T. \tag{3.19}$$

The denominator in (3.19) is always positive (except for singular points), while the numerator is positive when the constraint is satisfied by the nominal controller and it is negative when the nominal controller violates the constraint. Thus, the complete analytical solution of the QP can be expressed as

$$\boldsymbol{u}^* = \boldsymbol{k}(\boldsymbol{x}) - \min\left(0, \frac{L_G h(\boldsymbol{x})\boldsymbol{k}(\boldsymbol{x}) + L_f h(\boldsymbol{x}) + \alpha(h(\boldsymbol{x}))}{L_G h(\boldsymbol{x}) L_G h(\boldsymbol{x})^T}\right) L_G h(\boldsymbol{x})^T \tag{3.20}$$

The resulting controller is locally Lipschitz continuous [Xu et al., 2015].

3.3 Control Lyapunov functions

A Control Barrier Function (CBF), as defined in Definition 3.2.3, is an extension of Control Lyapunov Functions (CLFs), which enforce stability or state convergence requirements.

Definition 3.3.1 (Control Lyapunov function)**.** *[Khalil, 2014] A continuously differentiable positive definite function $V : \mathbb{R}^d \to \mathbb{R}$ is a control Lyapunov function (CLF) for system* (3.10) *if it satisfies:*

$$\inf_{\boldsymbol{u}\in\mathbb{R}^m} \{L_f V(\boldsymbol{x}) + L_G V(\boldsymbol{x})\boldsymbol{u}\} \leq -\gamma(V(\boldsymbol{x})) \tag{3.21}$$

with γ of class $\mathcal{K}$.

An intuitive idea about the difference between the conditions posed by CBFs and CLFs is shown in Fig. 3.4, considering as an example a linear extended class $\mathcal{K}$ function. In the case of CBFs, the derivative of the function h along the system trajectories must be higher than a certain threshold dictated by an exponential decrease (due to the consideration of a linear extended class $\mathcal{K}$ function). Conversely, for CLFs, the function V must decrease faster than a specified exponential.

Theorem 3. *[Sontag, 1989] Given a CLF V with the associated set $\mathcal{C}$ from* (3.4)*, any Lipschitz continuous controller $\boldsymbol{u}(t)$, $\forall t \geq 0$, that satisfies* (3.21) *asymptotically stabilizes the system* (3.10) *to the origin.*

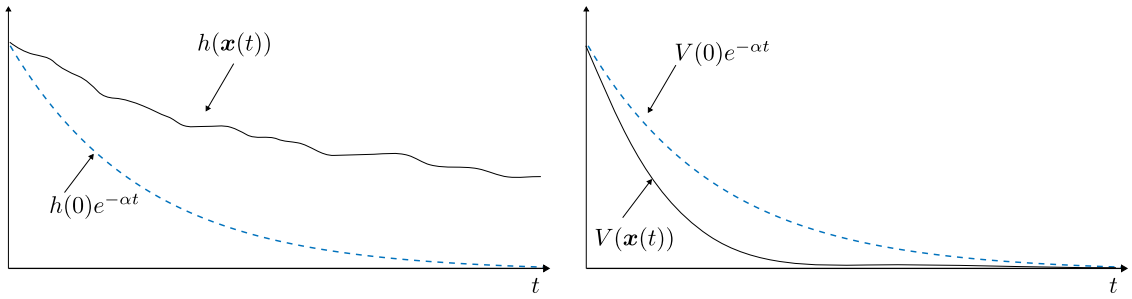


Figure 3.4: This figure is meant to give an intuitive idea about the difference in the conditions posed by CBF (on the left) and CLF (on the right) for the case of linear class $\mathcal{K}$ functions (i.e. exponential stability). Although we only plot in dashed the exponential corresponding to the given initial conditions, we remind the reader that the constraint on the slope is pointwise.

Similar to CBFs, the affine inequality in $\boldsymbol{u}$ allows for computationally efficient QP-based controllers.

The formulation of safety-critical controllers via QP suggests a way to unify safety and stability. In fact, optimization-based controllers were initially utilized in the context of CLFs to achieve multi-objective nonlinear control [Ames & Powell, 2013], such as combining stability with input constraints [Galloway et al., 2015]. One such formulation involves the following CLF-CBF-QP-based controller:

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathbb{R}^m} . \quad & \frac{1}{2}\boldsymbol{u}^T\boldsymbol{Q}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{c}^T\boldsymbol{u} + \mu\delta^2 \\ \text{s.t.} \quad & L_G h(\boldsymbol{x})\boldsymbol{u} + L_f h(\boldsymbol{x}) \geq -\alpha(h(\boldsymbol{x})) \\ & L_f V(\boldsymbol{x}) + L_G V(\boldsymbol{x})\boldsymbol{u} \leq -\gamma(V(\boldsymbol{x})) + \delta \end{aligned} \tag{3.22}$$

Here, $\boldsymbol{Q}(\boldsymbol{x})$ represents the positive definite Hessian of the cost function, and δ serves as a relaxation/slack variable ensuring QP feasibility, penalized by a large scalar $\mu > 0$ (to guarantee the QP has a solution by relaxing the stability condition to ensure safety). This formulation has been proven to yield a Lipschitz continuous controller [Ames et al., 2016].

3.4 High-order control barrier functions

CBFs define constraints on the state, which represent outputs of the system. Hence, when considering the input-output relationship, they exhibit relative degrees. For instance, a position constraint, like the relative distance between cars on a road, possesses a relative degree of two concerning an idealized acceleration input for a car.

The safety constraints have previously been assumed to be of relative degree of one, meaning that the first time-derivative of the CBF must depend on the control input. In fact, in cases where the system has a relative degree higher than one, meeting the condition in Definition 3.2.3, which requires $L_G h(\boldsymbol{x}) \neq 0$, at least on the border of the set, becomes unattainable. However, this assumption is often too restrictive for many safety constraints in robotic systems. Consequently, there is a need to establish a method for enforcing safety constraints with arbitrarily high relative degrees.

Various works have studied the conditions on the higher order derivatives of constraint functions to guarantee set invariance for systems of high relative degree [Nguyen & Sreenath, 2016; Tan et al., 2021; Xiao & Belta, 2019, 2021]. This section provides a basic summary of some of the ideas in [Tan et al., 2021].

Let us start by defining the concept of relative degree for a given output function.

Definition 3.4.1 (Relative degree)**.** *[Khalil, 2014] The relative degree ϱ of a (sufficiently) differentiable function $h : \mathbb{R}^d \to \mathbb{R}$ with respect to system* (3.10) *is the number of times one needs to differentiate it along the dynamics of* (3.10) *until any component of the control $\boldsymbol{u}$ explicitly appears. That is*

$$L_G L_f^{i-1} h(\boldsymbol{x}) = \mathbf{0}, \quad \text{for } i = 1, 2, ..., \varrho - 1; \quad L_G L_f^{\varrho-1} h(\boldsymbol{x}) \neq \mathbf{0}. \tag{3.23}$$

Given a sufficiently differentiable function $h : \mathbb{R}^d \to \mathbb{R}$ and sufficiently smooth extended class $\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot), ..., \alpha_\rho(\cdot)$, we define a series of functions as

$$\psi_0(\boldsymbol{x}) = h(\boldsymbol{x}), \quad \psi_k(\boldsymbol{x}) = \left(\frac{d}{dt} + \alpha_k\right) \psi_{k-1}, \quad 1 \leq k \leq \varrho \tag{3.24}$$

with the corresponding sets $\mathcal{C}_{k-1} = \{\boldsymbol{x} : \psi_{k-1}(\boldsymbol{x}) \geq 0\}$.

Definition 3.4.2 (High order control barrier function)**.** *[Tan et al., 2021] Function $h(\boldsymbol{x})$ is a high order control barrier function (HOCBF) of order ϱ for* (3.10) *if there exist ϱ extended class $\mathcal{K}$ functions $\alpha_1, \ldots, \alpha_\varrho$ and an open set $\mathcal{D}$ with $\mathcal{C} := \cap_{k=1}^{\varrho} \mathcal{C}_{k-1} \subset \mathcal{D} \subset \mathbb{R}^d$ such that*

$$\sup_{\boldsymbol{u} \in \mathcal{U}} \left[L_f \psi_{\varrho-1}(\boldsymbol{x}) + L_g \psi_{\varrho-1}(\boldsymbol{x}) \boldsymbol{u} + \alpha_\varrho\left(\psi_{\varrho-1}(\boldsymbol{x})\right)\right] \geq 0. \tag{3.25}$$

This definition generalizes to higher relative degrees the classical CBF definition [Ames et al., 2019] that only applies to constraints of relative degree $\varrho = 1$. Then, as usual, given a Lipschitz continuous desired input $\boldsymbol{u}^d$ (given by the nominal state feedback $\boldsymbol{k}(\boldsymbol{x})$), one

can define a minimally invasive controller by solving the following Quadratic Program (QP):

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathcal{U}} . \quad & \frac{1}{2}\left\|\boldsymbol{u}-\boldsymbol{u}^{d}\right\|_{2}^{2} \\ \text{s.t.} \quad & L_f\psi_{\varrho-1}(\boldsymbol{x})+L_G\psi_{\varrho-1}(\boldsymbol{x})\boldsymbol{u}+\alpha_{\varrho}(\psi_{\varrho-1}(\boldsymbol{x}))\geq 0. \end{aligned} \tag{3.26}$$

In the upcoming chapters, we will employ HOCBFs, since, as demonstrated later, any information measure obtained from the OG exhibits a relative degree of 2 for a velocity-controlled robot.

3.5 Distributed control barrier functions

The extension of the CBF framework to multi-agent systems has been explored in various works [Balandi et al., 2023; Y. Chen et al., 2020; Fernandez-Ayala et al., 2023; Lindemann & Dimarogonas, 2020; Tan & Dimarogonas, 2021, 2022; Wang et al., 2017]. Some of these works solve the CBF-induced quadratic program in a centralized manner [Capelli & Sabattini, 2020], utilizing a central module with access to all agents states. This approach yields optimal solutions at the price of high computational and communication costs as well as computational delays.

Extensions of the CBF framework to a decentralized or distributed implementation fall into three main strategies: (*i*) schemes pre-allocating the constraint [Balandi et al., 2023; Lindemann & Dimarogonas, 2020; Wang et al., 2017], (*ii*) schemes asymptotically tracking the centralized optimal solution [Tan & Dimarogonas, 2022], and (*iii*) distributed optimization-based schemes [Fernandez-Ayala et al., 2023; Tan & Dimarogonas, 2021].

Strategy (*i*) typically sacrifices optimality, potentially resulting in overly conservative solutions and leading to infeasibility. Strategy (*ii*) aims to closely track the centralized optimal solution but without guaranteeing constraint satisfaction. Strategy (*iii*) allows the recovery of the optimal solution while guaranteeing constraint satisfaction at the cost of higher computation and communication costs.

In this section, we introduce all three methods. They will each be employed in the subsequent chapters. Specifically, in Chapter 7, as a contribution of this thesis, we extend the method based on distributed optimization presented in [Tan & Dimarogonas, 2021], addressing a particular shortcoming of the approach.

Before delving into the distributed formulations, let us introduce the centralized prob-

lem we aim to solve. Consider a multi-agent system in control-affine form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \tag{3.27}$$

with state $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_1^T & \dots & \boldsymbol{x}_N^T\end{bmatrix}^T \in \mathbb{R}^n$ and input $\boldsymbol{u} = \begin{bmatrix}\boldsymbol{u}_1^T & \dots & \boldsymbol{u}_N^T\end{bmatrix}^T \in \mathbb{R}^m$. The function $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{f}_1(\boldsymbol{x}_1)^T & \dots & \boldsymbol{f}(\boldsymbol{x}_N)^T\end{bmatrix}^T$ and $\boldsymbol{G}(\boldsymbol{x}) := \text{diag}(\boldsymbol{G}_i(\boldsymbol{x}_i))$ is block-diagonal. The system must be confined within a safe set

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^d | h(\boldsymbol{x}) \geq 0\} \tag{3.28}$$

which could represent, e.g. a collision free region $h(\boldsymbol{x}) = d_{\text{obs}} - d_{\min} \geq 0$, where d_{obs} is the distance from an obstacle and $d_{\min}$ the minimum allowed distance from the obstacle, or it could represent the space in which the robot needs to move $h(\boldsymbol{x}) = R_{\max} - \|\boldsymbol{x}\| \geq 0$ where $R_{\max}$ is the maximum allowed distance from the origin. Similarly to the previous sections, for h to be a CBF it must satisfy the condition:

$$\sup_{\boldsymbol{u}\in\mathbb{R}^m} [L_f h(\boldsymbol{x}) + L_G h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x}))] \geq 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \tag{3.29}$$

which, for system (3.27), can be rewritten as a summation of local terms:

$$\sup_{\boldsymbol{u}\in\mathbb{R}^m} \left[\sum_{i=1}^{N} (L_{f_i} h(\boldsymbol{x}) + L_{G_i} h(\boldsymbol{x})\boldsymbol{u}_i) + \alpha(h(\boldsymbol{x}))\right] \geq 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{3.30}$$

In the following subsections, we discuss the three possible solutions to design distributed CBFs.

3.5.1 Pre-allocation based method

In this strategy, each agent individually addresses its decentralized control barrier function condition, ensuring their collective conjunction satisfies (3.29). A straightforward idea is to let each agent solve a QP with the following constraint

$$\frac{\partial h}{\partial \boldsymbol{x}_i} (\boldsymbol{f}_i(\boldsymbol{x}_i) + \boldsymbol{G}_i(\boldsymbol{x}_i)\boldsymbol{u}_i) \geq -\gamma_i \alpha(h(\boldsymbol{x})) \qquad \forall i \in \{1, 2, ..., N\} \tag{3.31}$$

with $\gamma_i = \frac{1}{N}$, which distributes (3.29) equally among the agents. To see why this approach provides a feasible solution, consider that if $\sum_{i=1}^{N} \gamma_i = 1$ is verified then, adding up all the

inequalities (3.31), one obtains

$$\sum_{i=1}^{N} \left(L_{f_i} h(\boldsymbol{x}) + L_{G_i} h(\boldsymbol{x}) \boldsymbol{u}_i\right) \geq -\sum_{i=1}^{N} \left(\gamma_i\right) \alpha(h(\boldsymbol{x})) = -\alpha(h(\boldsymbol{x})) \tag{3.32}$$

which is the CBF inequality in (3.30). However, this approach may encounter situations where for a specific agent i, $\frac{\partial h}{\partial \boldsymbol{x}_i} = \mathbf{0}$, while for another agent j, $\frac{\partial h}{\partial \boldsymbol{x}_j} \neq \mathbf{0}$. Consequently, although the centralized problem remains feasible, the local QP to be solved by agent i may become infeasible.

For instance, the inequality $0 \geq -\gamma_i \alpha(h(\boldsymbol{x}))$ cannot be satisfied if the system at time t is outside the safe set ($-\gamma_i \alpha(h(\boldsymbol{x})) > 0$), rendering the constraint unsatisfiable. This critical situation might lead to $\|\boldsymbol{u}_i\| \to \infty$ as $\frac{\partial h}{\partial \boldsymbol{x}_i} \to \mathbf{0}$.

To address this issue, alternative choices for γ_i exist as long as $\sum_{i=1}^{N} \gamma_i = 1$. One suitable choice for γ_i is:

$$\gamma_i = \frac{\left\| \frac{\partial h}{\partial \boldsymbol{x}_i} \right\|_p^p}{\sum_{i=1}^{N} \left\| \frac{\partial h}{\partial \boldsymbol{x}_i} \right\|_p^p} \tag{3.33}$$

with $p = \{1, 2\}$. This formulation ensures that as $\frac{\partial h}{\partial \boldsymbol{x}_i} \to \mathbf{0}$, $\gamma_i \to 0$, preserving $\sum_{i=1}^{N} \gamma_i = 1$. Moreover, if the number of agents N is known or estimated through consensus [Garin & Schenato, 2010], γ_i can be computed in a distributed manner by tracking the average of $\sum_{i=1}^{N} \left\| \frac{\partial h}{\partial \boldsymbol{x}_i} \right\|_p^p$ and multiplying it by N.

This method prevents constraint violations but may yield suboptimal solutions, potentially overconstraining the system and hindering performance in practical scenarios.

3.5.2 Tracking the centralized optimal solution

This approach, detailed in [Tan & Dimarogonas, 2022], emphasizes tracking a centralized optimal solution, a solution that may not ensure constraint satisfaction at all times but allows for distributed solvability.

For the problem to be solvable in a distributed way, we make the following assumption concerning h.

Assumption 1. *The parameters in the CBF condition* (3.29) *are locally obtainable, i.e., the condition can be written in the form*

$$\sum_{i=1}^{N} \left(\boldsymbol{a}_i^T(\boldsymbol{x}_i^{\mathcal{N}_i}) \boldsymbol{u}_i + b_i(\boldsymbol{x}_i^{\mathcal{N}_i})\right) \leq 0 \tag{3.34}$$

where $\boldsymbol{x}_i^{\mathcal{N}_i} = \begin{bmatrix} \boldsymbol{x}_i^T & \{\boldsymbol{x}_j^T\}_{j\in\mathcal{N}_i} \end{bmatrix}^T$, hence, the functions $\boldsymbol{a}_i(\boldsymbol{x}_i^{\mathcal{N}_i})$, $b_i(\boldsymbol{x}_i^{\mathcal{N}_i})$ can be locally evaluated. Notice that, relating (3.34) to (3.29), one has $\boldsymbol{a}_i^T = -L_{gi}h(\boldsymbol{x})$ while for b_i different choices are possible. One has to distribute the constraint as done for the pre-allocation based method, but this time without discontinuity problems. Hence, the most reasonable choice seems to use $b_i = -\frac{1}{N}(\alpha(h(\boldsymbol{x})) + L_{f_i}h(\boldsymbol{x}))$.

Consider the analytical solution provided in (3.20). For each agent i, it can be written as

$$\boldsymbol{u}_i^* = \boldsymbol{k}_i(\boldsymbol{x}) - \min\left(0, \frac{\sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{k}_i(\boldsymbol{x}) + b_i}{\sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{a}_i}\right)\boldsymbol{a}_i. \tag{3.35}$$

In this expression, the only terms which are not locally available are the two summations $\sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{k}_i(\boldsymbol{x})+b_i$ and $\sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{a}_i$. Employing a dynamic average consensus algorithm [Kia et al., 2019] enables tracking their averages, denoted μ_n and μ_d. The ratio $\frac{\mu_n}{\mu_d}$ allows the recovery of the term:

$$\frac{\mu_n}{\mu_d} = \frac{\sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{k}_i(\boldsymbol{x}) + b_i}{\sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{a}_i}. \tag{3.36}$$

In this way, we are able to asymptotically track the centralized optimal solution, but at the cost of not ensuring constraint satisfaction at all times. The original algorithm [Tan & Dimarogonas, 2022] involves employing a finite-time dynamic consensus algorithm [F. Chen et al., 2012] to track the averages μ_n and μ_d.

3.5.3 Distributed optimization based method

Here, we summarize the solution presented in [Tan & Dimarogonas, 2021]. Although other methods based on distributed optimization were introduced in [Y. Chen et al., 2020], they do not ensure constraint satisfaction at all times.

Assumption 1 is considered to hold. The QP to be solved is as follows

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathbb{R}^m} \quad & \frac{1}{2}\sum_{i=1}^N \left\|\boldsymbol{u}_i - \boldsymbol{u}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \sum_{i=1}^N \boldsymbol{a}_i^T\boldsymbol{u}_i + \sum_{i=1}^N b_i \leq 0, \end{aligned} \tag{3.37}$$

where we omitted the state dependency of $\boldsymbol{a}_i$ and b_i for the sake of readability. However, we remind that these terms change over time as the state evolves along the system trajectories.

The QP depicted in (3.37) is centralized. The objective is to develop a distributed algorithm that achieves asymptotic convergence to the time-varying centralized optimal

solution of the QP while always enforcing the safety constraint. To achieve this objective, we introduce the following equivalent QP

$$\begin{aligned} \min_{(\boldsymbol{u},\boldsymbol{y})\in\mathbb{R}^{m+N}} \quad & \frac{1}{2}\sum_{i=1}^{N}\left\|\boldsymbol{u}_i-\boldsymbol{u}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{u}_i+\sum_{j\in\mathcal{N}_i}(y_i-y_j)+b_i\leq 0, \quad \forall i\in\mathcal{V} \end{aligned} \tag{3.38}$$

where $\boldsymbol{y}=\begin{bmatrix}y_1,\dots y_N\end{bmatrix}^T\in\mathbb{R}^N$ is an auxiliary variable, with the element y_i associated to robot i. The two QPs, (3.38) and the original QP, are equivalent (see the Box in the next page). In particular, this implies that each solution $(\boldsymbol{u}',\boldsymbol{y}')$ satisfying the constraint in (3.38), also satisfies the constraint in the original QP (3.37) and viceversa.

The constraints in the previous QP (3.38) can be equivalently written in matrix form as:

$$\bar{\boldsymbol{A}}\boldsymbol{u}+\boldsymbol{L}\boldsymbol{y}+\boldsymbol{b}\leq 0 \tag{3.39}$$

with $\bar{\boldsymbol{A}}=\text{diag}(\boldsymbol{a}_i^T)$, $\boldsymbol{L}$ is the unweighted Laplacian matrix of the undirected graph and $\boldsymbol{b}=\begin{bmatrix}b_1 & \dots & b_N\end{bmatrix}^T$.

One may then formulate, for each agent i, the following QP

$$\begin{aligned} \min_{(\boldsymbol{u}_i,\boldsymbol{y})\in\mathbb{R}^{m_i+N}} \quad & \frac{1}{2}\left\|\boldsymbol{u}_i-\boldsymbol{u}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{u}_i+\sum_{j\in\mathcal{N}_i}(y_i-y_j)+b_i\leq 0 \end{aligned}. \tag{3.44}$$

The challenge here is that consistency of $\boldsymbol{y}$ needs to be preserved across the agents, i.e. the local copy y_i of agent i needs to be equal to the local copy y_j of agent j. For addressing this point, one approach involves exploring the KKT optimality conditions of (3.38) to design a distributed algorithm that updates $\boldsymbol{y}$, aiming for convergence towards the optimal $\boldsymbol{y}^*$ over time. Consequently, each robot tackles (3.44) using solely $\boldsymbol{u}_i$ as a decision variable, while maintaining $\boldsymbol{y}$ fixed within the QP. The optimal solution $\boldsymbol{y}^*$, derived from the KKT conditions, must meet the following criterion $\forall i\in\mathcal{V}$:

$$\begin{cases} \boldsymbol{a}_i^T\boldsymbol{u}_i^d+\boldsymbol{l}_i^T\boldsymbol{y}^*+b_i\leq 0, & \text{if} \quad \bar{\boldsymbol{a}}^T\boldsymbol{u}^d+\bar{b}\leq 0 \\ \boldsymbol{a}_i^T\boldsymbol{u}_i^d+\boldsymbol{l}_i^T\boldsymbol{y}^*+b_i=k\boldsymbol{a}_i^T\boldsymbol{a}_i, & \text{if} \quad \bar{\boldsymbol{a}}^T\boldsymbol{u}^d+\bar{b}>0 \end{cases} \tag{3.45}$$

where $\boldsymbol{l}_i$ is the i-th row of $\boldsymbol{L}$ and

$$k=(\bar{\boldsymbol{a}}^T\boldsymbol{u}^d+\bar{b})/\left\|\bar{\boldsymbol{a}}\right\|_2^2. \tag{3.46}$$

Showing equivalence of the QPs

The equivalence between (3.37) and (3.38) is based on the fact that $\mathbf{1}^T \boldsymbol{L} = \mathbf{0}$. In fact, if we sum all the constraints we obtain

$$\begin{aligned} 0 = \mathbf{1}_N^T \mathbf{0} \geq \mathbf{1}_N^T \left(\bar{\boldsymbol{A}} \boldsymbol{u} + \boldsymbol{L} \boldsymbol{y} + \boldsymbol{b} \right) = \sum_{i=1}^{N} \boldsymbol{a}_i^T \boldsymbol{u}_i + \mathbf{1}_N^T \boldsymbol{L} \boldsymbol{y} + \sum_{i=1}^{N} b_i \\ = \sum_{i=1}^{N} \left(\boldsymbol{a}_i^T \boldsymbol{u}_i + b_i \right) \end{aligned} \tag{3.40}$$

which is the expression in the original constraint. This implies that given a tuple $(\boldsymbol{u}', \boldsymbol{y}')$ satisfying (3.39), it also satisfies the original constraint. Viceversa, given an input $\boldsymbol{u}'$ satisfying the original constraint in (3.37), define $\boldsymbol{v} = \boldsymbol{A}\boldsymbol{u}' + \boldsymbol{b}$ and $\bar{v} = \frac{\mathbf{1}_N^T \boldsymbol{v}}{N}$ as its average. Since $\boldsymbol{u}'$ is feasible

$$0 \geq \sum_{i=1}^{N} \left(\boldsymbol{a}_i^T \boldsymbol{u}_i' + b_i \right) = \mathbf{1}_N^T \left(\bar{\boldsymbol{A}} \boldsymbol{u}' + \boldsymbol{b} \right) = \mathbf{1}_N^T \boldsymbol{v} = \sum_{i=1}^{N} v_i = N \bar{v} \tag{3.41}$$

which shows that $\bar{v} \leq 0$. Furthermore, since $\text{Range}(\boldsymbol{L}) = \{\boldsymbol{z} \in \mathbb{R}^N | \mathbf{1}_N^T \boldsymbol{z} = 0\}$, there exists $\boldsymbol{y}'$ such that

$$\boldsymbol{L}\boldsymbol{y}' + \boldsymbol{v} = \bar{v} \mathbf{1}_N. \tag{3.42}$$

Hence, we have

$$\bar{\boldsymbol{A}}\boldsymbol{u}' + \boldsymbol{L}\boldsymbol{y}' + \boldsymbol{b} = \boldsymbol{v} + \boldsymbol{L}\boldsymbol{y}' = \bar{v} \mathbf{1}_N \leq 0. \tag{3.43}$$

This implies that $(\boldsymbol{u}', \boldsymbol{y}')$ is feasible for (3.38). Noting that they also share the same cost function, we conclude that the two QPs are equivalent.

Since $k \leq 0$ when $\bar{\boldsymbol{a}}^T \boldsymbol{u}^d + \bar{b} \leq 0$, a sufficient condition for $\boldsymbol{y}^*$ is simply to satisfy (3.45) as

$$\boldsymbol{a}_i^T \boldsymbol{u}_i^d + \boldsymbol{l}_i^T \boldsymbol{y}^* + b_i = k \boldsymbol{a}_i^T \boldsymbol{a}_i \quad \forall i \in \mathcal{V} \tag{3.47}$$

which can be rewritten in matrix form as

$$\boldsymbol{L}\boldsymbol{y}^* = k \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_N^T \boldsymbol{a}_N \end{bmatrix} - \bar{\boldsymbol{A}} \boldsymbol{u}^d - \boldsymbol{b}. \tag{3.48}$$

Hereafter, the distinction arises in the method proposed in Chapter 7. In [Tan & Dimarogonas, 2021], local variables are defined as

$$k_i = \frac{1}{\boldsymbol{a}_i^T \boldsymbol{a}_i} (\boldsymbol{a}_i^T \boldsymbol{u}_i^d + \boldsymbol{l}_i^T \boldsymbol{y}^* + b_i) \tag{3.49}$$

and it was shown that the optimal $\boldsymbol{y}^*$ are such that $k_i = k_j \ \forall (i,j) \in \mathcal{E}$ then, finding such vector $\boldsymbol{y}^*$ condition (3.45) is satisfied. Consequently, it was proposed to update the variables $\boldsymbol{y}$ according to the following finite-time modified dynamic consensus:

$$\dot{\boldsymbol{y}} = -k_0 \text{sign}(\boldsymbol{L}\boldsymbol{k}) \tag{3.50}$$

with $\boldsymbol{k} = \begin{bmatrix} k_1 & \dots & k_N \end{bmatrix}^T$. A crucial point is that, due to (3.49), this approach requires $\|\boldsymbol{a}_i(t)\| > 0 \ \forall t, \forall i$ which, as also acknowledged in [Tan & Dimarogonas, 2021], can be quite restrictive. Even small values of $\|\boldsymbol{a}_i(t)\|$ could pose conditioning challenges. Furthermore, this assumption is not verified in various applications of interest, such as connectivity maintenance, as it will be extensively discussed in Chapter 7.

For this algorithm, convergence to the optimal solution in finite-time was shown in [Tan & Dimarogonas, 2021]. Furthermore, since any solution $(\boldsymbol{u}', \boldsymbol{y}')$ which is feasible for (3.38) is also feasible for the original problem (3.37) independently from $\boldsymbol{y}'$, it follows that constraint satisfaction is always guaranteed also during transitories.

The solution presented here addresses the case in which a single CBF constraint is considered. However, multiple CBFs can be handled by employing a smooth approximation of the minimum function [Fernandez-Ayala et al., 2023; Molnar & Ames, 2023].

Part II

Trajectory Optimization for Active Sensing

Chapter 4

Active Sensing for Cooperative Localization

The previous chapters introduced the fundamental theoretical concepts that are necessary for the understanding of the rest of this thesis. From this chapter on, we will instead focus on the contributions of this thesis in the field of active sensing, control barrier functions and observer design for multi-robot systems.

This chapter outlines the research conducted during the initial phase of the Ph.D., where we explored the adaptation of an active sensing methodology akin to the one detailed in [Salaris et al., 2019], originally designed for single-robot scenarios, to address the challenge of cooperative localization in multi-robot systems with sensing constraints. The objective is to enable a fleet of quadrotors to localize their relative poses in a common shared frame without imposing a rigidity requirement (which, as explained in Sect. 2.4, would simplify the localization problem but also overconstrain the group spatial arrangement and motion).

The problem of cooperative localization from relative bearing or relative distance measurements is non uniformly observable, meaning that for some input trajectories the system is not observable. Since we do not assume formation rigidity, it is necessary to *actively* guarantee the observability of the system in a common frame, which w.l.o.g. we consider to be the one of the robot identified as robot 1, denoted as *anchor* from now on. The distributed Extended Kalman Filter presented in [Luft et al., 2016] is used to localize the robots.

In order to guarantee the observability of the system, we extend an approach similar to [Salaris et al., 2019], aiming at maximizing the information acquired over a future horizon about the relative state of the robots, generating input trajectories which aim at enforcing system observability and possibly improving the state estimation accuracy. This approach involves formulating an optimal control problem, the objective of which is to maximize the information acquired during the trajectory considering the minimum

eigenvalue of the OG (E-criterion), which is related to the worst case uncertainty. This problem cannot be directly solved in a distributed way, since it would require each robot to know the future state trajectory of all the other robots in the group. It is however possible to derive an alternative cost function which can be decomposed in a way that is suitable for distributed optimization algorithms.

To generate trajectories that are aware of the sensing constraints, we weight the acquired information by suitable weight functions. We exploit the fact that, for common classes of models used for mobile robots, such as quadrotors in $\mathbb{R}^3 \times \mathbb{S}^1$, unicycles and of course simpler linear models as single or double integrators, the state sensitivity matrix possesses a closed-form solution that avoids the need forward integration any differential equation. The OG is taken as the sum of the prior information, given by the EKF, and the information acquired over the future horizon. Other constraints such as collision avoidance and connectivity maintenance are also taken into account separately from the optimal control problem (OCP) in order to limit the complexity of the problem. In particular, collision avoidance is discouraged using weights and safety can be guaranteed at a lower level using Control Barrier Functions (CBFs) [Wang et al., 2017]. Connectivity maintenance is enforced at low-level by employing the global connectivity maintenance algorithm presented in [Robuffo Giordano et al., 2013].

The contributions of this work can then be summarized as follows: we analyze the properties of the OG for the case of cooperative localization from either relative distance or relative bearing measurements. We propose a framework for the solution of the active cooperative localization problem of non-uniformly observable formations which is *distributed* and *scalable*. We do not rely on any particular assumption (e.g. external beacons, multiple tags per robot, or formation rigidity), we assume availability of only body-frame quantities. The main properties upon which we build the distributed optimization algorithm are the following:

- the eigenvalues of the OG being indipendent of the reference frame, one can optimize the optimality criteria in the most convenient frame which does not need to be the same considered for the EKF;
- in a static frame, for the considered robot model, the state sensitivity matrix has a closed-form solution, which avoids forward integration;
- the OG corresponding to a multi-robot system with relative measurements has a sparse structure which is fully exploited for proposing an alternative cost function

that makes the problem amenable to a partition-based distributed optimization algorithm.

Part of this work was first published in [De Carli et al., 2021], focusing on scenarios where relative distances are measured. This initial exploration covered a subset of the framework detailed in this chapter, specifically delving into the structure of the OG and demonstrating the potential for a distributed solution through appropriate coordinate transformations, which will be elaborated upon later in this chapter. In this work, each robot independently solved its optimization problem, with the current future prediction of the neighbors trajectory held constant during optimization.

4.1 Related works

Few works have proposed a fully distributed solution to the active cooperative localization problem. In [Cristofaro & Martinelli, 2010; Morbidi & Mariottini, 2012] algorithms based on the covariance matrix solution of the Riccati differential equation have been proposed but without considering the problem of arriving at a distributed implementation. Similarly, in [S. Li et al., 2022] a centralized *Nonlinear Model Predictive Control* (NMPC) is used to perform some multi-robot tasks (e.g. formation control) and at the same time maximizing the determinant of the observability matrix.

In [Cano & Le Ny, 2023], the authors consider a group of robots in which some robots are defined as anchors (considered as absolutely localized) and others are defined as tags (which need to be localized). The robots measure relative distances among themselves and they move in the gradient direction to maximize the usual optimality criteria considering the *constrained* FIM. They are able to find an estimate of the centralized gradient direction for the A- and D-criteria by computing the inverse of the FIM using the *Richardson iteration*. The inverse estimation, although distributed, it is not scalable as it requires each robot to estimate the corresponding full block of rows of the inverse FIM, which has as many block of columns as the number of robots in the group. Furthermore, they consider scalable optimization of the E-criterion using the well-known distributed *power iteration method*. Differently from our work the authors consider only the information acquired by the current measure in the FIM, thus disregarding the past history of the system. In this way, they can only optimize the geometry of the formation with the intent of increasing robustness to noise. This is possible because they assume that there are enough exactly localized robots to be able to unambiguously solve the localization

problem. Also, differently from our work, the authors do not consider sensing limitations and, hence, the corresponding sensing graph is constant.

In [Cossette et al., 2022], again, an algorithm for active cooperative distance-based localization using UWB is proposed. Here, the authors detail a framework that does not require absolutely localized robots, but instead each robot is required to possess two tags: this is equivalent to the rigidity assumption and makes the problem uniformly observable. The authors do not consider the problem of computing the gradient in a distributed way as they assume the graph to be always fully-connected and, again, they aim at optimizing the geometry of the formation more than achieving PE, which is not necessary in their case.

The present work differs from the previously mentioned ones as we propose a fully distributed and scalable solution to the active cooperative localization problem and we do not make any ad-hoc assumptions about the uniform observability of the system (e.g. using absolutely localized robots, multiple tags on a single robot, or formation rigidity). As a consequence, the robot trajectories are required to satisfy a suitable 'excitation condition' for ensuring observability of their relative poses. Moreover, we consider robots with only body-frame commands and measurements, so that the robots do not share an exactly localized frame (just an estimated one) in which to express the exchanged quantities. Finally, the interactions among the robots are dictated by a time-varying graph which depends on realistic sensor limitations.

4.2 Multi-robot system model

In this section, we describe the modeling assumptions, beginning with the single robot model. We consider quadrotors modeled in $\mathbb{R}^3 \times \mathbb{S}^1$, whose dynamics are expressed in a common static frame when simulating the dynamics over a future horizon. However, during state estimation using a distributed EKF, the dynamics are expressed in the frame of a specific robot, as will be elucidated in the subsequent section. Then, we describe the measurement model and we conclude describing the whole group model.

4.2.1 Robot model

We consider a group of N robots with a particular focus on quadrotor UAVs because of their popularity and widespread use in the community. However, the proposed method-

ology would also apply to other kinds of mobile robots with minor modifications.

In line with prior works on formation control/localization for multiple quadrotors (e.g., [Franchi et al., 2012; Montijano, Zhou, et al., 2014; Schiano et al., 2016]), we simplify the quadrotor state to a 3D position and a yaw angle $\boldsymbol{q}_i = (\boldsymbol{p}_i, \psi_i) \in \mathbb{R}^3 \times \mathbb{S}^1$ which, as well-known, are the quadrotor flat outputs [Fliess et al., 1995; Mellinger & Kumar, 2011]. As a consequence, any smooth enough trajectory $\boldsymbol{q}_i(t)$ can then be tracked by existing flight controllers for quadrotors (e.g., [T. Lee et al., 2010]).

We model the quadrotors as first-order kinematic systems with *body-frame* velocity commands

$$\begin{bmatrix} \dot{\boldsymbol{p}}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ w_i \end{bmatrix} \tag{4.1}$$

where $\boldsymbol{v}_i$ and w_i are respectively the commanded body-frame linear velocity and yaw rate, and $\boldsymbol{R}_i = \boldsymbol{R}_z(\psi_i) \in SO(3)$ denotes the rotation matrix about the z-axis of an angle ψ_i. We also let $\boldsymbol{u}_i = (\boldsymbol{v}_i, w_i)$ collect the i-th robot commands. Notice that we skipped the time dependency to make the notation lighter.

No global common frame is assumed directly available to the robots since they can only obtain *relative measurements* w.r.t. each other and their velocity commands are in the robot body frames. A common frame must, however, be implicitly or explicitly selected for expressing the estimated robot poses. This could be an arbitrary 'fixed' frame (e.g., the initial pose of one particular robot in the group), or a 'moving' frame attached to the formation. In this latter case, two natural choices seem possible: the robot poses are expressed in the moving body frame of one particular robot (e.g., which we refer to as *anchor*), or in a moving frame attached to the group barycenter. The barycentral frame would seem a better choice since it does not require the selection of a 'special robot' in the group. However, being the barycentral frame linearly dependent on the frames of the robots in the formation, the associated change of coordinates would lead to a rank deficient OG. Since an Extended Kalman Filter (EKF) is employed as observer, the consequence of a rank deficient OG is that the covariance matrix of the EKF would grow unbounded along certain state space directions.

Therefore, in this work we assume presence of an anchor robot, w.l.o.g. taken as robot 1, whose body frame, which we indicate as $\mathcal{R}_1$, is taken as the common frame for expressing the estimated poses of the other robots in the group. In view of the following developments, it is then convenient to express the robot states and model in terms of quantities relative to $\mathcal{R}_1$. Let ${}^1\boldsymbol{R}_i = \boldsymbol{R}_1^T \boldsymbol{R}_i$, ${}^1\boldsymbol{p}_i = {}^1\boldsymbol{R}_i(\boldsymbol{p}_i - \boldsymbol{p}_1)$ and ${}^1\psi_i = \psi_i - \psi_1$ be the

position and orientation of robot i in $\mathcal{R}_1$, and $\boldsymbol{x}_i = ({}^1\boldsymbol{p}_i, {}^1\psi_i)$ the i-th robot configuration in $\mathcal{R}_1$. From (4.1) and standard kinematics one has

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} {}^1\dot{\boldsymbol{p}}_i \\ {}^1\dot{\psi}_i \end{bmatrix} = \begin{bmatrix} {}^1\boldsymbol{R}_i & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ w_i \end{bmatrix} - \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_{e_3} {}^1\boldsymbol{p}_i \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ w_1 \end{bmatrix} \tag{4.2}$$

where $\boldsymbol{S}_{e_3} = [\boldsymbol{e}_3]_\times$ and $\boldsymbol{v}_1$ and w_1 are the control inputs of the anchor. Clearly, ${}^1\boldsymbol{p}_1 = {}^1\dot{\boldsymbol{p}}_1 = \boldsymbol{0}$ and ${}^1\psi_1 = {}^1\dot{\psi}_1 = 0$. The dynamics (4.2) are the model considered by the employed state estimator which, in our case, is the distributed EKF (dEKF) proposed in [Luft et al., 2016, 2018].

In this thesis, we do not delve into details about the dEKF algorithm as it is not part of the contribution of this thesis and, from an input-output point of view, it can be considered as a traditional EKF. The only important details concerning the algorithm within the scope of this thesis are the following. This dEKF only requires robot i to communicate with a neighboring robot j when either i or j acquire a new measurement. To achieve such a distributed solution, approximations in the update steps of the covariances are made. In practice, the performances obtained are not much far from a centralized implementation of the EKF.

We point out that, since the employed dEKF, providing the estimated state $\hat{\boldsymbol{x}}(t)$ and pseudo-covariance matrix $\boldsymbol{P}(t)$, is built on the 'anchor-frame' dynamics (4.2), the implementation of the dEKF requires that each robot has access to the *current* anchor velocity input $\boldsymbol{u}_1(t)$. One possible way to address this issue is to assume presence of a consensus-like algorithm, as the one used in [Franchi & Robuffo Giordano, 2018], i.e. each robot maintain an estimate of the anchor velocity $\hat{\boldsymbol{u}}_1^i$ as:

$$\begin{aligned} \dot{\hat{\boldsymbol{u}}}_1^i &= -k_u \sum_{j \in \mathcal{N}_i} (\hat{\boldsymbol{u}}_1^i - \hat{\boldsymbol{u}}_1^j) \\ \hat{\boldsymbol{u}}_1^1 &= \boldsymbol{u}_1 \end{aligned} \tag{4.3}$$

This obviously introduces a delay in the estimated velocity, which particularly affects robots which are far (in terms of graph distance) from the anchor. However, by suitably weighing the uncertainties on the robots dynamics, the effect on the relative pose estimation is limited as neighboring robots are expected to have a similar estimate of the anchor velocity.

4.2.2 Sensing model

We considered both the case in which only relative distances are measured and the case in which only relative bearings are measured. In the following we will treat them separately.

Relative distance measurements

The UAVs are assumed equipped with a sensor able to provide relative range measurements corrupted by an additive normally distributed noise $\nu \sim \mathcal{N}(0,\, r)$ with zero mean and variance r

$$d_{ij}^n = \|\boldsymbol{p}_j - \boldsymbol{p}_i\|_2 + \nu = h_{ij}(\boldsymbol{p}_j - \boldsymbol{p}_i) + \nu. \tag{4.4}$$

A robot pair $(i,\, j)$ is assumed able to measure d_{ij}^n when $\|\boldsymbol{p}_j - \boldsymbol{p}_i\|_2 \leq d_{\max}$, with $d_{\max}$ being a maximum sensing range. Similarly, a robot pair $(i,\, j)$ is also assumed able to exchange information over a radio channel when within the sensing range $d_{\max}$.

Because sensing constraints are considered the induced sensing graph is time-varying. Moreover, since only the maximum range sensing constraint is considered, the induced sensing graph is undirected.

Relative bearing measurements

The bearing from robot i to robot j expressed in the body frame $\mathcal{R}_i$ is defined as the 3D unit vector:

$$\boldsymbol{\beta}_{ij} = \boldsymbol{R}_i^T \frac{\boldsymbol{p}_{ij}}{d_{ij}} \tag{4.5}$$

where $\boldsymbol{p}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i$ and $d_{ij} = ||\boldsymbol{p}_{ij}||_2$. Bearings are commonly used because they can be quite accurately retrieved using inexpensive sensors such as cameras. A simplified model for the noise affecting the measurements is the following:

$$\boldsymbol{\beta}_{ij}^n = \boldsymbol{R}_i^T \frac{\bar{\boldsymbol{p}}_{ij} + \boldsymbol{R}_i \bar{\boldsymbol{\eta}}_{ij}}{||\bar{\boldsymbol{p}}_{ij} + \boldsymbol{R}_i \bar{\boldsymbol{\eta}}_{ij}||_2} \tag{4.6}$$

where $\bar{\boldsymbol{p}}_{ij} \in \mathbb{R}^3$ is the projection of $\boldsymbol{p}_{ij}$ onto the calibrated perspective plane of the camera of the i-th robot and $\bar{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} 0 & \boldsymbol{\eta}_{ij}^T \end{bmatrix}^T$ with $\boldsymbol{\eta}_{ij} \sim \mathcal{N}(\boldsymbol{0},\, \boldsymbol{Q}_{ij})$ models image noise. Notice that the resulting measurement noise, being subject to a nonlinear transformation, is not Gaussian. As in other works employing bearing measurements [Hamel & Samson, 2016], the dEKF from [Luft et al., 2016] is then here considered as a deterministic observer and the gain matrix representing the measurement confidence (usually the inverse of the

covariance matrix) is taken to be a constant positive definite matrix $\boldsymbol{R} = r\boldsymbol{I}$ with $r > 0$.

The i-th robot is assumed able to measure the relative bearing w.r.t. the j-th robot $\boldsymbol{\beta}_{ij}^n$ when $d_{\min} \leq d_{ij} \leq d_{\max}$, with $d_{\min}$ and $d_{\max}$ being minimum and maximum sensing range **and** the j-th robot is in the FoV of the i-th robot. This latter constraint is modeled as the condition $c_{ij} > c_{\min}$, where $c_{ij} = \boldsymbol{e}_1^T \boldsymbol{\beta}_{ij}$ is the cosine among the i-th robot x body axis (which the camera is assumed to be aligned with) and the bearing (see Fig. 4.1), and $c_{\min}$ is the minimum cosine allowed by the limited camera FoV (corresponding to the maximum angle $\alpha_{\max}$). Because sensing constraints are considered the induced sensing graph is

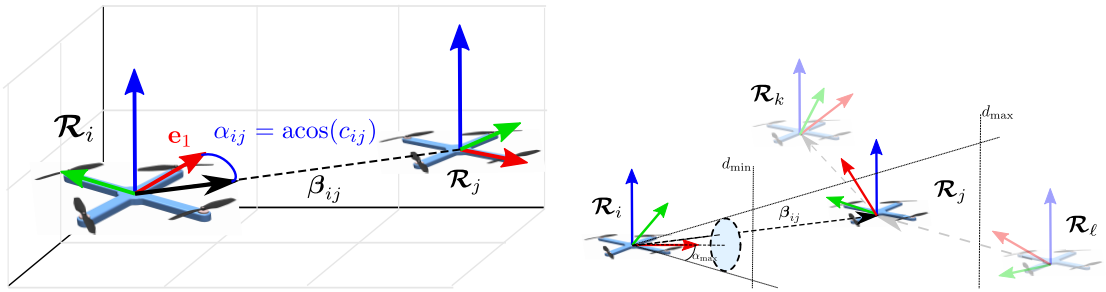


Figure 4.1: Illustrations of the quantities involved in the sensing constraints.

time-varying. Moreover, since also FoV sensing constraints are considered, the induced sensing graph is *directed*, while the communication graph corresponds to its *undirected* counterpart.

4.2.3 Group model

Let d be the state dimension of one robot ($d = 4$ in our scenario). In view of the following developments we let

$$\boldsymbol{x} = \begin{bmatrix} {}^1\boldsymbol{p}_1^T & \dots & {}^1\boldsymbol{p}_N^T & {}^1\psi_1 & \dots & {}^1\psi_N \end{bmatrix}^T \in (\mathbb{R}^3 \times \mathbb{S}^1)^N, \tag{4.7}$$

Vector $\boldsymbol{x}$ represents the stack of N robot poses $\boldsymbol{x}_i$ in $\mathcal{R}_1$, with a reordering for collecting the positions and yaw angles into two partitions. Similarly, we also introduce the vector

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{p}_1^T & \dots & \boldsymbol{p}_N^T & \psi_1 & \dots & \psi_N \end{bmatrix}^T \in (\mathbb{R}^3 \times \mathbb{S}^1)^N$$

as the collection of reordered configurations $\boldsymbol{q}_i$ for the robots in the global frame. The anchor frame state $\boldsymbol{x}$ is the vector of poses which needs to be estimated (anchor excluded) and represents the EKF state. We remind the reader that ${}^1\boldsymbol{p}_1(t) = \boldsymbol{0}$ and ${}^1\psi_1(t) = 0$ for

all $t \geq 0$, thus they are not included in the state of the dEKF, ensuring a positive definite gramian and, hence, boundedness of the covariance matrix [Pengov et al., 2001]. Vector $\boldsymbol{u} = [\boldsymbol{u}_1^T, \dots, \boldsymbol{u}_N^T]^T \in \mathbb{R}^{dN}$ is the stack of all robot inputs, while

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{|\mathcal{E}|} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1^q(\boldsymbol{q}) \\ \vdots \\ \boldsymbol{h}_{|\mathcal{E}|}^q(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1^x(\boldsymbol{x}) \\ \vdots \\ \boldsymbol{h}_{|\mathcal{E}|}^x(\boldsymbol{x}) \end{bmatrix} \tag{4.8}$$

is the stack of all the $|\mathcal{E}|$ measurements among robot pairs (either relative distances or relative bearings).

Regrouping (4.2, 4.8) for all the N robots, the group dynamics in the anchor frame is compactly expressed as

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(t) &= \boldsymbol{h}^x(\boldsymbol{x}(t)) \end{aligned} \tag{4.9}$$

Regrouping (4.1), we denote the complete group dynamics in the global frame as

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{g}(\boldsymbol{q}(t), \boldsymbol{u}(t)) \tag{4.10}$$

This distinction between vectors $\boldsymbol{x}$ and $\boldsymbol{q}$ (and respective dynamics) may appear redundant but it will be important in the next developments: indeed $\boldsymbol{x}$ represents the 'anchor-frame' robot poses that will be actually estimated by the employed filter (an EKF in our case), whereas $\boldsymbol{q}$ represents the 'fixed-frame' robot poses that are forward propagated by the robots at each optimization step for maximizing the future information gain.

4.3 The multi-robot observability gramian

In this section, we will directly dive into the structure of the OG for the specific case at hand. For more details about the properties of the OG refer to the chapters of Preliminaries.

We consider a weighted OG given as

$$\boldsymbol{G}_o(t_0, t_f) = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau, t_0) \boldsymbol{C}^T(\tau) \boldsymbol{W}(\tau) \boldsymbol{C}(\tau) \boldsymbol{\Phi}(\tau, t_0)\, d\tau \tag{4.11}$$

In the following subsections, we start by showing that the OGs associated to the dynam-

ics (4.9) and (4.10) are equivalent from the point of view of optimizing one of the the optimality criteria (see Sect. 1.3). This allows to consider the dynamics (4.10) along the future horizon, which do not require the knowledge of the anchor input. Then, we show that, for the robot model (4.10), the state sensitivity matrix can be computed in closed form, making it much cheaper to compute. Finally, we define the weighting function $\boldsymbol{W}$, so that it takes into account the robots sensing limits.

4.3.1 Change of coordinates

An important property of the OG is that, given a differential change of bases

$$\delta \boldsymbol{q}(t) = \boldsymbol{T}(\boldsymbol{x}(t))\delta \boldsymbol{x}(t) \tag{4.12}$$

and referring the OG in the two bases as $\boldsymbol{G}_o^{(x)}(t_0, t_f)$ and $\boldsymbol{G}_o^{(q)}(t_0, t_f)$, these are related as follows:

$$\boldsymbol{G}_o^{(q)}(t_0, t_f) = \boldsymbol{T}(\boldsymbol{x}(t_0))\boldsymbol{G}_o^{(x)}(t_0, t_f)\boldsymbol{T}^T(\boldsymbol{x}(t_0)). \tag{4.13}$$

Therefore, if the change of bases matrix $\boldsymbol{T}(\boldsymbol{x}(t))$ is orthogonal, this is a similarity transformation and, since the optimality criteria can be expressed as eigenvalue functions, optimizing some optimality criteria considering $\boldsymbol{G}_o^{(q)}$ is equivalent to optimizing the corresponding optimality criteria for $\boldsymbol{G}_o^{(x)}$. Consider the change of coordinates from the anchor-frame coordinates to some static common frame. The differential change of coordinates is given as:

$$\delta \boldsymbol{q}(t) = \boldsymbol{T}(\psi_1(t))\delta \boldsymbol{x}(t) \tag{4.14}$$

with

$$\boldsymbol{T}(\psi_1(t)) = \boldsymbol{I}_N \otimes \boldsymbol{T}_1(\psi_1(t)), \quad \boldsymbol{T}_1(\psi_1(t)) = \begin{bmatrix} \boldsymbol{R}_z(\psi_1(t)) & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}. \tag{4.15}$$

which is an orthogonal matrix. Hence, in light of (4.13), it follows that the OGs relative to the states $\boldsymbol{x}$ (as in the EKF case) and $\boldsymbol{q}$ are similar matrices. The OG can then be computed from the states $\boldsymbol{q}$ with the important advantage of not requiring each robot to know the trajectory of the *anchor* along the predicted trajectory.

4.3.2 Multi quadrotor state sensitivity matrix

We now consider the state sensitivity matrix for the special case of a network of quadrotors modeled as in (4.1) which, as we will show, has a *closed-form* expression.

The state sensitivity matrix for a single robot $\boldsymbol{\Phi}_i(t, t_0) = \frac{\partial \boldsymbol{q}_i(t)}{\partial \boldsymbol{q}_i(t_0)}$ has the expression

$$\boldsymbol{\Phi}_i(t, t_0) = \begin{bmatrix} 1 & 0 & 0 & -y_i(t) + y_i(t_0) \\ 0 & 1 & 0 & x_i(t) - x_i(t_0) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.16}$$

where x_i and y_i represent, respectively, the x- and y-component of the position of the i-th robot expressed in a static frame. This can be seen by expliciting each component of the state equations for the i-th robot

$$\begin{cases} x_i(t) = x_i(t_0) + \int_{t_0}^{t} (c_{\psi_i}(t) v_{xi}(t) - s_{\psi_i}(t) v_{yi}(t))\, d\tau \\ y_i(t) = y_i(t_0) + \int_{t_0}^{t} (s_{\psi_i}(t) v_{xi}(t) + c_{\psi_i}(t) v_{yi}(t))\, d\tau \\ z_i(t) = z_i(t_0) + \int_{t_0}^{t} v_{zi}(t)\, d\tau \\ \psi_i(t) = \psi_i(t_0) + \int_{t_0}^{t} \omega_i(t)\, d\tau \end{cases} \tag{4.17}$$

where we used the notation $\cos(\psi_i) := c_{\psi_i}$, $\sin(\psi_i) := s_{\psi_i}$ and $\boldsymbol{v}_i := [v_{xi}, v_{yi}, v_{zi}]$, and then, computing the jacobian and noticing that $\frac{\partial x_i(t)}{\partial \psi_i(t_0)} = -y_i(t) + y_i(t_0)$ and similarly for $\frac{\partial y_i(t)}{\partial \psi_i(t_0)}$. Because of this result, when we will formulate an optimal control problem it will not be necessary to forward integrate the dynamics of the state sensitivity matrix for obtaining $\boldsymbol{\Phi}(t, t_0)$, which could otherwise be computationally heavy, particularly since each robot also need to know the the one of its neighbors.

The resulting state sensitivity matrix $\boldsymbol{\Phi}(t_0, t) = \frac{\partial \boldsymbol{q}(t)}{\partial \boldsymbol{q}(t_0)}$, because of the permutation partitioning it in position and yaw components, has the following structure:

$$\boldsymbol{\Phi}(t, t_0) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Phi}_{p\times\psi}(t, t_0) \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{4.18}$$

where $\boldsymbol{\Phi}_{p\times\psi}$ is a block diagonal matrix

$$\boldsymbol{\Phi}_{p\times\psi}(t, t_0) = \operatorname{diag}\left(\begin{bmatrix} -y_i(t) + y_i(t_0) \\ x_i(t) - x_i(t_0) \\ 0 \end{bmatrix} \right) = (\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}) \operatorname{diag}\left(\boldsymbol{p}_i(t) - \boldsymbol{p}_i(t_0)\right) \tag{4.19}$$

4.3.3 Weights definition

As previously mentioned, we assume that the onboard sensors have a limited minimum/maximum range and, for the case of bearing measurements, limited field of view. These constraints can be taken into account by suitably weighting the information acquired by the sensors along the predicted trajectory via some scalar differentiable state-dependent weight function $w_{ij}(\boldsymbol{q}_i(t), \boldsymbol{q}_j(t))$ associated to each edge. In particular, we consider the weights defined in (2.38), which vary in the range [0, 1] and are designed so that when the i-th and j-th robots are approaching disconnection w_{ij} get close to zero, meaning the associated information will get close to zero, but they do not exactly vanish at the limit.

These weights enter the OG inside the weight matrix $\boldsymbol{W} = \text{diag}(r^{-1} w_{ij} \boldsymbol{I}_p)$, where r is the information gain in the EKF, as discussed in Section 4.2.2, and $p = 1$ for distance measurements and $p = 3$ for bearing measurements. This can also be interpreted as having a state-dependent output covariance $(w_{ij} r^{-1} \boldsymbol{I}_p)^{-1}$ which tends to infinity when the robots are approaching disconnection.

4.3.4 The multi-robot observability gramian: structure and properties

In this section, we analyze the structure of the OG, showing that, it has the same sparsity structure as a Laplacian matrix, which is beneficial for distributed computations. We remind that, the state sensitivity matrix up to the partitioning into position and yaw has a block-diagonal structure (4.18), while the term $\boldsymbol{C}^T \boldsymbol{C}$ corresponds to the symmetric rigidity matrix (see Section 2.4) which has a similar sparsity pattern as a Laplacian matrix (i.e. every block (i, k), such that $k \notin \mathcal{N}_i$ is equal to $\boldsymbol{0}$). Below, we recap the expression of the symmetric rigidity matrix both for the case of distance and bearing measurements as in Section 2.4, but also considering the weight matrix $\boldsymbol{W}$.

$$\boldsymbol{S}_d = \begin{bmatrix} \boldsymbol{E}_3 \,\text{diag}(r^{-1} w_{ij} \boldsymbol{p}_{ij} \boldsymbol{p}_{ij}^T) \boldsymbol{E}_3^T & \boldsymbol{0}_{3N \times N} \\ \boldsymbol{0}_{N \times 3N} & \boldsymbol{0}_{N \times N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{p_d} & \boldsymbol{0}_{3N \times N} \\ \boldsymbol{0}_{N \times 3N} & \boldsymbol{0}_{N \times N} \end{bmatrix} \tag{4.20}$$

$$\begin{aligned} \boldsymbol{S}_b &= \begin{bmatrix} \boldsymbol{E}_3 \,\text{diag}\left(r^{-1} w_{ij} \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\right) \boldsymbol{E}_3^T & -\boldsymbol{E}_3 \,\text{diag}\left(r^{-1} w_{ij} \frac{\boldsymbol{S}_{e_3} \bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right) \boldsymbol{E}_{\otimes}^T \\ -\boldsymbol{E}_{\otimes} \,\text{diag}\left(\frac{r^{-1} w_{ij} \bar{\boldsymbol{\beta}}_{ij}^T \boldsymbol{S}_{e_3}^T}{d_{ij}}\right) \boldsymbol{E}_3^T & \text{diag}\left(r^{-1} w_{ij} \sum_{j \in \mathcal{N}_i^{out}} \bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{L}_{p_b} & \boldsymbol{L}_{S\otimes}^T \\ \boldsymbol{L}_{S\otimes} & \boldsymbol{D}_{\otimes} \end{bmatrix} \end{aligned} \tag{4.21}$$

where $\boldsymbol{L}_{p_*}$, with $* = \{'d', 'p'\}$ are weighted Laplacian matrices, $\boldsymbol{L}_{S\otimes}$ is a directed weighted Laplacian matrix asymmetrical in the dimensions and with sign indefinite rectangular weight matrices and, finally, $\boldsymbol{D}_{\otimes}$ is a weighted out-degree matrix. As a consequence, for distance measurements

$$\begin{aligned}\boldsymbol{G}_o^d(t_0,t_f) &= \begin{bmatrix} \boldsymbol{G}_{op}^d & \boldsymbol{G}_{op\times\psi}^{dT} \\ \boldsymbol{G}_{op\times\psi}^d & \boldsymbol{G}_{o\psi}^d \end{bmatrix} = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau,t_0)\boldsymbol{C}^T(\tau)\boldsymbol{W}(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau,t_0)\,d\tau \\ &= \int_{t_0}^{t_f} \begin{bmatrix} \boldsymbol{L}_{p_d} & * \\ \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_d} & \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_d}\boldsymbol{\Phi}_{p\times\psi} \end{bmatrix} d\tau\end{aligned} \tag{4.22}$$

and for bearing measurements

$$\begin{aligned}\boldsymbol{G}_o^b(t_0,t_f) &= \begin{bmatrix} \boldsymbol{G}_{op}^b & \boldsymbol{G}_{op\times\psi}^{bT} \\ \boldsymbol{G}_{op\times\psi}^b & \boldsymbol{G}_{o\psi}^b \end{bmatrix} = \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau,t_0)\boldsymbol{C}^T(\tau)\boldsymbol{W}(\tau)\boldsymbol{C}(\tau)\boldsymbol{\Phi}(\tau,t_0)\,d\tau \\ &= \int_{t_0}^{t_f} \begin{bmatrix} \boldsymbol{L}_{p_b} & * \\ \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_b} + \boldsymbol{L}_{S\otimes} & \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_b}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{S\otimes}^T + \boldsymbol{L}_{S\otimes}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{D}_{\otimes} \end{bmatrix} d\tau\end{aligned} \tag{4.23}$$

where the '$*$' symbol is used to indicate the symmetric term. Notice that the lower right term preserves the sparsity structure of a Laplacian-like matrix, this is due to the block diagonal structure of $\boldsymbol{\Phi}_{p\times\psi}$. Moreover, the sparsity structure of the matrices involved is preserved under the integral.

Next, we list some properties related to the observability and the OG for the considered multi-robot system. We point out that this study is an original contribution of this thesis.

- **The positions partition is a matrix-weighted Laplacian**

  The integral of the upper left term $\boldsymbol{G}_{op}^* = \int_{t_0}^{t_f} \boldsymbol{L}_{p_*}\,d\tau$ both for distance and bearing measurements remains a matrix weighted Laplacian matrix, i.e. it is still true that

$$\begin{aligned}[\boldsymbol{G}_{op}^*]_{(ii)} &= \sum_{j\in\mathcal{N}_i} [\boldsymbol{G}_{op}^*]_{(ij)} \\ [\boldsymbol{G}_{op}^*]_{(ij)} &\preceq \boldsymbol{0}\end{aligned} \tag{4.24}$$

  These two properties are straightforward to verify by expanding the integral in the two cases (distance and bearing measurements) and considering the fact that the sum of Laplacians is a Laplacian itself.

Furthermore, we point out that:

$$\boldsymbol{G}_{op}^{d} = \int_{t_0}^{t_f} \boldsymbol{E}_3 \operatorname{diag}(\int_{t_0}^{t_f} r^{-1} w_{ij} \boldsymbol{p}_{ij} \boldsymbol{p}_{ij}^T) \boldsymbol{E}_3^T \, d\tau = \boldsymbol{E}_3 \operatorname{diag}\left(\int_{t_0}^{t_f} r^{-1} w_{ij} \boldsymbol{p}_{ij} \boldsymbol{p}_{ij}^T \, d\tau\right) \boldsymbol{E}_3^T \tag{4.25}$$

and

$$\boldsymbol{G}_{op}^{b} = \int_{t_0}^{t_f} \boldsymbol{E}_3 \operatorname{diag}\left(r^{-1} w_{ij} \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\right) \boldsymbol{E}_3^T \, d\tau = \boldsymbol{E}_3 \operatorname{diag}\left(\int_{t_0}^{t_f} r^{-1} w_{ij} \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2} \, d\tau\right) \boldsymbol{E}_3^T \tag{4.26}$$

These two equations are fundamental as they show that, if an edge is PE then the corresponding matrix weight in the Laplacian becomes positive definite. Many classical results for traditional Laplacians straightforwardly extended to matrix-weighted Laplacians with positive definite weights [Trinh et al., 2018]. For instance, from Lemma 2 it follows that, if there exists a path of positive definite edges then, the null space of the OG is spanned by $\mathcal{N}(\boldsymbol{G}_{op}^{*}) = \mathbf{1}_N \otimes \boldsymbol{I}_d$.

- **Observability condition for a formation in $\mathbb{R}^3 \times \mathbb{S}^1$**

  Let us first introduce an observability condition based on the OG for the case in $\mathbb{R}^3$. We are concerned with the observability of the positions of the robots in a common frame. Consider an ideal case in which each robot directly measures the relative positions with respect to its neighbors with the graph being connected. The measurement Jacobian in this case is nothing else than the incidence matrix $\boldsymbol{E}_3$ and, analogously to $\boldsymbol{S}_d$ and $\boldsymbol{S}_b$, we define the symmetric matrix $\boldsymbol{S}_p = \boldsymbol{E}_3 \boldsymbol{E}_3^T = \boldsymbol{L}_3$. In this case, it is well-known that $\mathcal{N}(\boldsymbol{L}_3) = \mathbf{1}_N \otimes \boldsymbol{I}_3$, implying that the only unobservable motions are common translations in $\mathbb{R}^3$. Then, one can state the observability conditions from distance and bearing measurements as:

  $$\boldsymbol{G}_{op}^{*} \succeq \mu \boldsymbol{L}_3 \tag{4.27}$$

  with $\mu > 0$, which shows that the robot positions are observable up to a common translation.

  Let us now repeat a similar reasoning for the case in $\mathbb{R}^3 \times \mathbb{S}^1$. This time, the baseline is represented by each robot measuring the relative *body-frame* position and the relative yaw with respect to its neighbors, which allows to localize the robots up to a common translation and a coordinated rotation about the z-axis involving an

angular rotation. Let us define the measurement vector

$$\boldsymbol{h}_q = \begin{bmatrix} \mathrm{diag}(\boldsymbol{R}_i^T)\boldsymbol{E}^T\boldsymbol{p} \\ \boldsymbol{E}^T\boldsymbol{\psi} \end{bmatrix}, \tag{4.28}$$

the measurement Jacobian can be shown to be

$$\begin{aligned} \boldsymbol{R}_q &= \begin{bmatrix} \mathrm{diag}(\boldsymbol{R}_i^T)\boldsymbol{E}_3 & -\mathrm{diag}(\boldsymbol{R}_i^T)(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\,\mathrm{diag}(\boldsymbol{p}_{ij})\boldsymbol{E}_\otimes^T \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{diag}(\boldsymbol{R}_i^T) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_3 & -(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\,\mathrm{diag}(\boldsymbol{p}_{ij})\boldsymbol{E}_\otimes^T \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \end{aligned} \tag{4.29}$$

and the corresponding symmetric matrix is

$$\begin{aligned} \boldsymbol{S}_q &= \begin{bmatrix} \boldsymbol{E}_3 & \boldsymbol{0} \\ -\boldsymbol{E}_\otimes\,\mathrm{diag}(\boldsymbol{p}_{ij})(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}^T) & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_3 & -(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\,\mathrm{diag}(\boldsymbol{p}_{ij})\boldsymbol{E}_\otimes^T \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{L}_3 & -\boldsymbol{E}_3(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\,\mathrm{diag}(\boldsymbol{p}_{ij})\boldsymbol{E}_\otimes^T \\ -\boldsymbol{E}_\otimes\,\mathrm{diag}(\boldsymbol{p}_{ij})(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}^T)\boldsymbol{E}_3^T & \mathrm{diag}\left(\sum_{j\in\mathcal{N}_i^{out}} \bar{p}_{ijx}^2 + \bar{p}_{ijy}^2\right) + \boldsymbol{L}. \end{bmatrix} \end{aligned} \tag{4.30}$$

Similarly to what done before, we can then state observability of the relative configurations in $\mathbb{R}^3 \times \mathbb{S}^1$ as

$$\boldsymbol{G}_o^* \succeq \mu \boldsymbol{S}_q(t_0). \tag{4.31}$$

It is worth noting that in this case, the LMI condition depends on the initial state of the formation $\boldsymbol{q}(t_0)$. In fact, we can expect that, at best, the formation could be localized up to a common translation and an *initial* coordinated rotation about the z-axis involving an angular rotation.

- **A rigid translation is an unobservable motion**

A rigid translation motion is spanned by $\boldsymbol{1}_N \otimes \boldsymbol{I}_3$. From the fact that $\boldsymbol{E}_3^T(\boldsymbol{1}_N \otimes \boldsymbol{I}_3) = \boldsymbol{0}$, it follows that

$$\begin{aligned} &\boldsymbol{L}_{p_*}(\boldsymbol{1}_N \otimes \boldsymbol{I}_3) = \boldsymbol{0} \\ &\boldsymbol{L}_{S\otimes}(\boldsymbol{1}_N \otimes \boldsymbol{I}_3) = -\boldsymbol{E}_\otimes\,\mathrm{diag}\left(\frac{r^{-1}w_{ij}\bar{\boldsymbol{\beta}}_k^T\boldsymbol{S}_{e_3}^T}{d_{ij}}\right)\boldsymbol{E}_3^T(\boldsymbol{1}_N \otimes \boldsymbol{I}_3) = \boldsymbol{0} \end{aligned} \tag{4.32}$$

This implies that a rigid translation is unobservable, i.e.,

$$\boldsymbol{G}_{op}^{*}(\mathbf{1}_N \otimes \boldsymbol{I}_3) = \int_{t_0}^{t_f} \boldsymbol{L}_{p_*}(\mathbf{1}_N \otimes \boldsymbol{I}_3)\, d\tau = \mathbf{0} \tag{4.33}$$

$$\boldsymbol{G}_{op\times\psi}^{*} = \int_{t_0}^{t_f} (\boldsymbol{\Phi}_{p\times\psi}^{T}\boldsymbol{L}_{p_b} + \boldsymbol{L}_{S\otimes})(\mathbf{1}_N \otimes \boldsymbol{I}_3)\, d\tau = \mathbf{0} \tag{4.34}$$

Altogether, this shows that

$$\boldsymbol{G}_{o}^{*}(\mathbf{1}_N \otimes \boldsymbol{I}_3) = \mathbf{0} \tag{4.35}$$

i.e. $\begin{bmatrix}(\mathbf{1}_N \otimes \boldsymbol{I}_3)^T & \mathbf{0}^T\end{bmatrix}^T$, which is a bases for a rigid translation motion, is in the null-space of the OG.

- **A coordinated rotation about the z-axis involving an angular rotation is unobservable for a distance formation**

  Define now the vectors $\boldsymbol{p}_\perp = (\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3})\,\boldsymbol{p}$. A coordinated rotation about the z-axis in $\mathbb{R}^3 \times \mathbb{S}^1$ involving an angular rotation at time t_0 is spanned by $\begin{bmatrix}\boldsymbol{p}_\perp(t_0)^T & \mathbf{1}_N^T\end{bmatrix}^T$. We now verify that this vector is in the null space of the OG.

$$\begin{aligned}\boldsymbol{G}_{o}^{d}(t_0,t_f)\begin{bmatrix}\boldsymbol{p}_\perp(t_0)\\ \mathbf{1}_N\end{bmatrix} &= \int_{t_0}^{t_f}\begin{bmatrix}\boldsymbol{L}_{p_d} & *\\ \boldsymbol{\Phi}_{p\times\psi}^{T}\boldsymbol{L}_{p_d} & \boldsymbol{\Phi}_{p\times\psi}^{T}\boldsymbol{L}_{p_d}\boldsymbol{\Phi}_{p\times\psi}\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_\perp(t_0)\\ \mathbf{1}_N\end{bmatrix} d\tau\\ &= \int_{t_0}^{t_f}\begin{bmatrix}\boldsymbol{L}_{p_d}\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_d}\boldsymbol{\Phi}_{p\times\psi}\mathbf{1}_N\\ \boldsymbol{\Phi}_{p\times\psi}^{T}\left(\boldsymbol{L}_{p_d}\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_d}\boldsymbol{\Phi}_{p\times\psi}\mathbf{1}_N\right)\end{bmatrix} d\tau.\end{aligned} \tag{4.36}$$

  Notice that, using (4.19),

$$\begin{aligned}&\boldsymbol{L}_{p_d}(t)\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_d}(t)\boldsymbol{\Phi}_{p\times\psi}(t,t_0)\mathbf{1}_N\\ &= \boldsymbol{L}_{p_d}(t)\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_d}(t)\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\mathrm{diag}\left(\boldsymbol{p}_i(t) - \boldsymbol{p}_i(0)\right)\mathbf{1}_N\\ &= \boldsymbol{L}_{p_d}(t)\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_d}(t)\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\left(\boldsymbol{p}(t) - \boldsymbol{p}(0)\right)\\ &= \boldsymbol{L}_{p_d}(t)\boldsymbol{p}_\perp(t)\end{aligned} \tag{4.37}$$

  where we used $\mathrm{diag}\left(\boldsymbol{p}_i(t) - \boldsymbol{p}_i(t_0)\right)\mathbf{1}_N = \boldsymbol{p}(t) - \boldsymbol{p}(t_0)$. Moreover,

$$\begin{aligned}\boldsymbol{L}_{p_d}\boldsymbol{p}_\perp &= \boldsymbol{E}_3\,\mathrm{diag}(r^{-1}w_{ij}\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^{T})\boldsymbol{E}_3^{T}\boldsymbol{p}_\perp\\ &= \boldsymbol{E}_3\,\mathrm{diag}(r^{-1}w_{ij}\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^{T})\left(\boldsymbol{I}_M \otimes \boldsymbol{S}_{e_3}\right)\mathrm{stack}(\boldsymbol{p}_j - \boldsymbol{p}_i)\\ &= \boldsymbol{E}_3\,\mathrm{diag}(r^{-1}w_{ij}\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^{T})\,\mathrm{stack}(\boldsymbol{S}_{e_3}\boldsymbol{p}_{ij}) = \mathbf{0}.\end{aligned} \tag{4.38}$$

This follows from the fact that, since $\boldsymbol{S}_{e_3}\boldsymbol{p}_{ij} = \boldsymbol{e}_3 \times \boldsymbol{p}_{ij}$ is orthogonal to $\boldsymbol{p}_{ij}$, it is verified that $\boldsymbol{p}_{ij}^T\boldsymbol{S}_{e_3}\boldsymbol{p}_{ij} = 0$. This could have also be stated directly from Lemma 2, as done in Sect. 2.4.

As a consequence, $\begin{bmatrix}\boldsymbol{p}_\perp^T(t_0) & \mathbf{1}_N^T\end{bmatrix}^T$ is in the null space of the OG and, hence, the initial coordinated rotation of the group is unobservable for a distance formation.

Remark 2. *An interesting difference emerges with respect to an infinitesimally rigid static formation, for which any rigid rotation about any axis and not necessarily involving an angular rotation is unobservable. Instead, if the trajectory is exciting, only a coordinated rotation about the z-axis and necessarily involving an angular rotation is unobservable.*

- **A coordinated rotation about the z-axis involving an angular rotation is unobservable for a bearing formation in $\mathbb{R}^3 \times \mathbb{S}^1$**

A coordinated rotation in $\mathbb{R}^3 \times \mathbb{S}^1$ involving an angular rotation is spanned by $\begin{bmatrix}\boldsymbol{p}_\perp(t_0)^T & \mathbf{1}_N^T\end{bmatrix}^T$. We now verify that this vector is in the null space of the OG.

$$
\begin{aligned}
&\boldsymbol{G}_o^b(t_0,t_f)\begin{bmatrix}\boldsymbol{p}_\perp(t_0)\\ \mathbf{1}_N\end{bmatrix}\\
&= \int_{t_0}^{t_f}\begin{bmatrix}\boldsymbol{L}_{p_b} & *\\ \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_b} + \boldsymbol{L}_{S\otimes} & \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_b}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{S\otimes}^T + \boldsymbol{L}_{S\otimes}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{D}_\otimes\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_\perp(t_0)\\ \mathbf{1}_N\end{bmatrix} d\tau\\
&= \int_{t_0}^{t_f}\begin{bmatrix}\boldsymbol{L}_{p_b}\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_b}\boldsymbol{\Phi}_{p\times\psi}\mathbf{1}_N + \boldsymbol{L}_{S\otimes}^T\mathbf{1}_N\\ (\boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_b} + \boldsymbol{L}_{S\otimes})\boldsymbol{p}_\perp(t_0) + (\boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{p_b}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{L}_{S\otimes}^T + \boldsymbol{L}_{S\otimes}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{D}_\otimes)\mathbf{1}_N\end{bmatrix} d\tau\\
&= \int_{t_0}^{t_f}\begin{bmatrix}\boldsymbol{\xi}\\ \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{\xi} + \boldsymbol{L}_{S\otimes}\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{S\otimes}\boldsymbol{\Phi}_{p\times\psi}\mathbf{1}_N + \boldsymbol{D}_\otimes\mathbf{1}_N\end{bmatrix} d\tau
\end{aligned}
\tag{4.39}
$$

where we defined $\boldsymbol{\xi} = \boldsymbol{L}_{p_b}\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{p_b}\boldsymbol{\Phi}_{p\times\psi}\mathbf{1}_N + \boldsymbol{L}_{S\otimes}^T\mathbf{1}_N$.

We consider now the two terms of the vector (resp. corresponding to position and

orientation) separately. Expanding the first term and using (4.19), we have

$$\begin{aligned}
\boldsymbol{\xi} &= \boldsymbol{L}_{p_b}\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\boldsymbol{p}(t_0) + \boldsymbol{L}_{p_b}\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\operatorname{diag}\left(\boldsymbol{p}_i(t) - \boldsymbol{p}_i(t_0)\right)\mathbf{1}_N + \boldsymbol{L}_{S\otimes}^T\mathbf{1}_N \\
&= \boldsymbol{L}_{p_b}\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\boldsymbol{p}(t) - \boldsymbol{E}_3\operatorname{diag}\left(\frac{r^{-1}w_{ij}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right)\boldsymbol{E}_{\otimes}^T\mathbf{1}_N \\
&= \boldsymbol{E}_3\operatorname{diag}\left(r^{-1}w_{ij}\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\right)\boldsymbol{E}_3^T\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\boldsymbol{p}(t) - \boldsymbol{E}_3\operatorname{stack}\left(\frac{r^{-1}w_{ij}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right) \\
&= \boldsymbol{E}_3\operatorname{diag}\left(r^{-1}w_{ij}\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\right)\left(\boldsymbol{I}_M \otimes \boldsymbol{S}_{e_3}\right)\operatorname{stack}\left(\boldsymbol{p}_j - \boldsymbol{p}_i\right) - \boldsymbol{E}_3\operatorname{stack}\left(\frac{r^{-1}w_{ij}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right) \\
&= \boldsymbol{E}_3\operatorname{diag}\left(r^{-1}w_{ij}\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\boldsymbol{S}_{e_3}\right)\operatorname{stack}\left(d_{ij}\bar{\boldsymbol{\beta}}_{ij}\right) - \boldsymbol{E}_3\operatorname{stack}\left(\frac{r^{-1}w_{ij}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right) \\
&= \boldsymbol{E}_3\operatorname{stack}\left(r^{-1}w_{ij}\frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}\right) - \boldsymbol{E}_3\operatorname{stack}\left(\frac{r^{-1}w_{ij}\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_{ij}}{d_{ij}}\right) \\
&= \mathbf{0}
\end{aligned} \tag{4.40}$$

where we used the following facts: $\boldsymbol{E}_{\otimes}^T\mathbf{1}_N = \mathbf{1}_M$, with M being the cardinality of the edge set $|\mathcal{E}|$, $\operatorname{diag}\left(\boldsymbol{p}_i(t) - \boldsymbol{p}_i(t_0)\right)\mathbf{1}_N = \boldsymbol{p}(t) - \boldsymbol{p}(t_0)$, $\boldsymbol{E}_3^T\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right) = \left(\boldsymbol{I}_M \otimes \boldsymbol{S}_{e_3}\right)\boldsymbol{E}_3^T$ and $\boldsymbol{p}_{ij} = d_{ij}\bar{\boldsymbol{\beta}}_{ij}$.

Expanding the second term, we have:

$$\begin{aligned}
&\boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{\xi} + \boldsymbol{L}_{S\otimes}\boldsymbol{p}_\perp(t_0) + \boldsymbol{L}_{S\otimes}\boldsymbol{\Phi}_{p\times\psi}\mathbf{1}_N + \boldsymbol{D}_{\otimes}\mathbf{1}_N \\
&= \boldsymbol{L}_{S\otimes}\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\boldsymbol{p}(t_0) + \boldsymbol{L}_{S\otimes}\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\left(\boldsymbol{p}(t) - \boldsymbol{p}(0)\right) + \operatorname{stack}\left(r^{-1}w_{ij}\sum_{j\in\mathcal{N}_i^{out}}\bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) \\
&= -\boldsymbol{E}_{\otimes}\operatorname{diag}\left(\frac{r^{-1}w_{ij}\bar{\boldsymbol{\beta}}_{ij}^T\boldsymbol{S}_{e_3}^T}{d_{ij}}\right)\boldsymbol{E}_3^T\left(\boldsymbol{I}_N \otimes \boldsymbol{S}_{e_3}\right)\boldsymbol{p}(t) + \operatorname{stack}\left(r^{-1}w_{ij}\sum_{j\in\mathcal{N}_i^{out}}\bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) \\
&= -\boldsymbol{E}_{\otimes}\operatorname{diag}\left(\frac{r^{-1}w_{ij}\bar{\boldsymbol{\beta}}_{ij}^T\boldsymbol{S}_{e_3}^T}{d_{ij}}\right)\left(\boldsymbol{I}_M \otimes \boldsymbol{S}_{e_3}\right)\operatorname{stack}(\boldsymbol{p}_j(t) - \boldsymbol{p}_i(t)) + \operatorname{stack}\left(r^{-1}w_{ij}\sum_{j\in\mathcal{N}_i^{out}}\bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) \\
&= -\boldsymbol{E}_{\otimes}\operatorname{diag}\left(\frac{r^{-1}w_{ij}\bar{\boldsymbol{\beta}}_{ij}^T\boldsymbol{S}_{e_3}^T\boldsymbol{S}_{e_3}}{d_{ij}}\right)\operatorname{stack}(d_{ij}\boldsymbol{\beta}_{ij}) + \operatorname{stack}\left(r^{-1}w_{ij}\sum_{j\in\mathcal{N}_i^{out}}\bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) \\
&= -\boldsymbol{E}_{\otimes}\operatorname{stack}\left(r^{-1}w_{ij}\bar{\boldsymbol{\beta}}_{ij}^T\boldsymbol{S}_{e_3}^T\boldsymbol{S}_{e_3}\boldsymbol{\beta}_{ij}\right) + \operatorname{stack}\left(r^{-1}w_{ij}\sum_{j\in\mathcal{N}_i^{out}}\bar{\beta}_{ijx}^2 + \bar{\beta}_{ijy}^2\right) = \mathbf{0}
\end{aligned} \tag{4.41}$$

where we used $\bar{\boldsymbol{\beta}}_k^T\boldsymbol{S}_{e_3}^T\boldsymbol{S}_{e_3}\bar{\boldsymbol{\beta}}_k = \bar{\boldsymbol{\beta}}_k^T(\boldsymbol{I}_3 - \boldsymbol{e}_3\boldsymbol{e}_3^T)\bar{\boldsymbol{\beta}}_k = \bar{\beta}_{kx}^2 + \bar{\beta}_{ky}^2$. This concludes the proof that $\begin{bmatrix}\boldsymbol{p}_\perp(t_0)^T & \mathbf{1}_N^T\end{bmatrix}^T$ is in the null-space of the OG. Showing that, the initial coordinated rotation of the group of robots is not observable for a bearing formation.

- **A scale variation could be observable for a bearing formation**

  A scale expansion or contraction is spanned by the vector $\begin{bmatrix}\boldsymbol{p}(t_0)^T & \mathbf{0}_N^T\end{bmatrix}^T$. Hence, it is only necessary to check that, under persistency of excitation conditions, $\boldsymbol{p}(t_0)$ is not in the null space of

$$\boldsymbol{G}_{op}^b = \int_{t_0}^{t_f} \boldsymbol{L}_{p_b}\, d\tau = \boldsymbol{E}_3 \operatorname{diag}\left(\int_{t_0}^{t_f} r^{-1} w_{ij} \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}}{d_{ij}^2}\, d\tau\right) \boldsymbol{E}_3^T. \tag{4.42}$$

  Definition 4.3.1 (Persistently exciting direction)**.** *[Z. Tang et al., 2022] A direction $\boldsymbol{y}(t) \in \mathbb{S}^{d-1}$, is called PE if the matrix $\boldsymbol{\Pi}_{\boldsymbol{y}}(t) = \boldsymbol{I}_d - \boldsymbol{y}(t)\boldsymbol{y}(t)^T$ satisfies the PE condition from Definition 1.2.2.*

  Lemma 3. *[Z. Tang et al., 2022] Consider $\boldsymbol{y}(t) \in \mathbb{S}^{d-1}$ and assume $\dot{\boldsymbol{y}}(t)$ is uniformly continuous, then $\boldsymbol{\Pi}_{\boldsymbol{y}}(t)$ being PE is equivalent to: $\forall t \geq 0$, there exist $T, \epsilon > 0$ and $\tau \in [t, t+T]$ such that $\|\dot{\boldsymbol{y}}(t)\| \geq \epsilon$.*

  Since $\boldsymbol{G}_{op}^b$ is a matrix-weighted Laplacian (4.24), then, we just need to use Lemma 2 to check that, under PE conditions, the scale expansion bases is not in the null space of $\boldsymbol{G}_{op}^b$.

  Assume there is at least an edge $e_k = (i, j)$, such that the corresponding bearing $\boldsymbol{\beta}_{ij}$ is PE. Then, for that edge, assuming that $r^{-1} w_{ij}(t) \neq 0\ \forall t$, we have

$$\boldsymbol{W}_{ij} \coloneqq \int_{t_0}^{t_f} r^{-1} w_{ij}(\tau) \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}(\tau)}}{d_{ij}^2(\tau)}\, d\tau \succeq \mu \boldsymbol{I}_3. \tag{4.43}$$

  Hence, $\mathcal{N}(\boldsymbol{W}_{ij}) = \emptyset$, i.e. there is no vector belonging to the null space of $\boldsymbol{W}_{ij}$, then, the vector $\begin{bmatrix}\boldsymbol{p}(t_0)^T & \mathbf{0}_N^T\end{bmatrix}^T$ does not satisfies the condition of Lemma 2. In fact one can show that:

$$\int_{t_0}^{t_f} r^{-1} w_{ij}(\tau) \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}(\tau)}{d_{ij}^2(\tau)}\, d\tau (\boldsymbol{p}_j(t_0) - \boldsymbol{p}_i(t_0)) = d_{ij}(t_0) \int_{t_0}^{t_f} r^{-1} w_{ij}(\tau) \frac{\boldsymbol{\Pi}_{\bar{\beta}_{ij}}(\tau)}{d_{ij}^2(\tau)} \boldsymbol{\beta}_{ij}(t_0)\, d\tau \neq \mathbf{0}. \tag{4.44}$$

  Since the bearing $\boldsymbol{\beta}_{ij}$ is PE, then, $\boldsymbol{\beta}_{ij}(\tau) \nparallel \boldsymbol{\beta}_{ij}(t_0)$ at least over some non-zero measure interval $\tau \in [t_1, t_2]$. This conclude the proof that, under certain PE conditions, the scale of the formation is observable.

- **If one robot has private measurements of the full position w.r.t. the world frame the common translation could become observable**

We assume the formation is observable in a common frame, i.e. condition (4.27) is satisfied and that one robot in the group (w.l.o.g. robot 1) is acquiring a full-state measurement with respect to the world frame.

A private full-state measurement enters $\boldsymbol{S}_*$ as follows:

$$\boldsymbol{S}_* = \boldsymbol{L}_{p_*} + \boldsymbol{D}_p \tag{4.45}$$

where $\boldsymbol{D}_p = \boldsymbol{e}_1\boldsymbol{e}_1^T \otimes \boldsymbol{I}_p$.We point out that the following analysis could be identically repeated if PE partial measurements where collected w.r.t. the world frame instead of direct state measurements.

Notice that

$$\boldsymbol{G}_{op}^* = \int_{t_0}^{t_f} \boldsymbol{L}_{p_*}\, d\tau + (t_f - t_0)\boldsymbol{D}_p \tag{4.46}$$

can be seen as a *grounded Laplacian* [Bullo, 2020; Dörfler et al., 2018] and by assumption $\int_{t_0}^{t_f} \boldsymbol{L}_{p_*}\, d\tau \succeq \mu \boldsymbol{L}_3$ corresponds to a connected matrix-weighted graph, therefore, $\boldsymbol{G}_{op}^d \succeq \mu_1 \boldsymbol{I}$, for some $\mu_1 > 0$. This shows that, for a PE formation in $\mathbb{R}^3$ with full-position measurements w.r.t. the world frame, the full system state is observable (common translation included).

- **If one robot has private measurements of the full state w.r.t. the world frame the common translation and orientation could become observable**

  We assume the formation is observable in a common frame, i.e. condition (4.31) is satisfied and that one robot in the group (w.l.o.g. robot 1) is acquiring a full-state measurement with respect to the world frame. Since, as explained in Sect. 1.2.1, the total OG for two measured vectors is the sum of the two corresponding OGs, we can study the observability of this case from

$$\boldsymbol{G}_o^* + (t_f - t_0)\begin{bmatrix} \boldsymbol{D}_p & \boldsymbol{D}_p\boldsymbol{\Phi}_{p\times\psi} \\ \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{D}_p & \boldsymbol{D}_\psi \end{bmatrix} \succeq \mu\boldsymbol{S}_q + (t_f - t_0)\begin{bmatrix} \boldsymbol{D}_p & \boldsymbol{D}_p\boldsymbol{\Phi}_{p\times\psi} \\ \boldsymbol{\Phi}_{p\times\psi}^T\boldsymbol{D}_p & \boldsymbol{D}_\psi \end{bmatrix} \tag{4.47}$$

  where $\boldsymbol{D}_p = \boldsymbol{e}_1\boldsymbol{e}_1^T \otimes \boldsymbol{I}_p$ and $\boldsymbol{D}_\psi = \boldsymbol{e}_1\boldsymbol{e}_1^T$. We point out that the following analysis could be identically repeated if PE partial measurements were collected w.r.t. the world frame instead of direct state measurements.

  Let us denote $\bar{\boldsymbol{L}}_{p_d} = \boldsymbol{L}_{p_d} + \boldsymbol{D}_p$, the OG can then be written as

$$\boldsymbol{G}_o^d(t_0, t_f) = \int_{t_0}^{t_f} \begin{bmatrix} \bar{\boldsymbol{L}}_{p_d} & * \\ \boldsymbol{\Phi}_{p\times\psi}^T\bar{\boldsymbol{L}}_{p_d} & \boldsymbol{\Phi}_{p\times\psi}^T\bar{\boldsymbol{L}}_{p_d}\boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{D}_\psi \end{bmatrix} d\tau. \tag{4.48}$$

Notice that

$$\boldsymbol{G}_{op}^{d} = \int_{t_0}^{t_f} \boldsymbol{L}_{p_d}\, d\tau + (t_f - t_0)\boldsymbol{D}_p \tag{4.49}$$

can be seen as a *ground Laplacian* [Bullo, 2020; Dörfler et al., 2018] and by assumption $\int_{t_0}^{t_f} \boldsymbol{L}_{p_d}\, d\tau \succeq \mu \boldsymbol{L}_3$ corresponds to a connected matrix-weighted graph, therefore, $\boldsymbol{G}_{op}^{d} \succeq \mu_1 \boldsymbol{I}$, for some $\mu_1 > 0$. This passage applies identically in the case of bearing measurements and it shows that, for a PE formation in $\mathbb{R}^3$ with full-position measurements w.r.t. the world frame, the full system state is observable (common translation included). Then, we use the Schur complement to show that, the full state in $\mathbb{R}^3 \times \mathbb{S}^1$ is observable. We have shown that $\boldsymbol{G}_{op}^{d} \succeq \mu_1 \boldsymbol{I}$ and now we show that also its complement $\boldsymbol{G}_{o}^{d}/\boldsymbol{G}_{op}^{d} \geq \mu_2 \boldsymbol{I}$. In fact, one has:

$$\boldsymbol{G}_{o}^{d}/\boldsymbol{G}_{op}^{d} = \int_{t_0}^{t_f} (\boldsymbol{\Phi}_{p\times\psi}^{T} \bar{\boldsymbol{L}}_{p_d} \boldsymbol{\Phi}_{p\times\psi} + \boldsymbol{D}_{\psi})\, d\tau - \int_{t_0}^{t_f} \boldsymbol{L}_{p_d}\, d\tau \left(\int_{t_0}^{t_f} \boldsymbol{L}_{p_d}\, d\tau \right) \int_{t_0}^{t_f} \boldsymbol{L}_{p_d}\, d\tau \tag{4.50}$$

As mentioned in Sect.(4.2.1), we solve the estimation problem in the frame $\mathcal{R}_1$. Consequently the state of robot 1 is treated as exactly known (i.e. $\boldsymbol{p}_1 = \boldsymbol{0}$ and $\psi_1 = 0$). To ensure convergence of the dEKF, it is imperative to analyze the observability of the system in the anchor frame. Therefore, within the OG, we remove the block of rows and columns corresponding to the anchor, analogously as how the constrained FIM is defined in [Cano & Le Ny, 2023]. In fact, this is equivalent to consider a group of $N-1$ robots, with measurements from and w.r.t. a beacon of known position and orientation, similarly to what discussed in the last described property of the OG. Because of that, under PE conditions, the resulting OG $\bar{\boldsymbol{G}}_o$ is full rank.

4.4 Distributed optimal control problem

In this section, we formulate an optimal control problem (OCP), which needs to be solved in a distributed way. We let each robot maximize the information collected at N_{wp} waypoints equally spaced in time. The trajectory among two waypoints is locally interpolated using *Bezier curves* [Biagiotti & Melchiorri, 2008], taking into account the required smoothness of the curve.

The optimality criteria discussed in Sect.1.3 are functions of the eigenvalues of the OG. Hence, they do not lend themselves to distributed computations over a future horizon, as this would require knowledge of global quantities. For this reason, we find surrogates of

the classical optimality criteria which are computable in a distributed way, which we use as cost function in our OCP.

This cost function is expressed as the summation of locally computable costs, which makes it for distributed optimization algorithms.

4.4.1 The cost function

As a reminder, common cost functions to be considered in active sensing are the following [Pukelsheim, 2006]:

- **A-criterion** : $\text{trace}(\boldsymbol{G}_o^{-1})$
- **D-criterion** : $-\log\left(\det(\boldsymbol{G}_o)\right)$
- **E-criterion** : $-\lambda_{\min}(\boldsymbol{G}_o)$

These functions are not computable in a distributed way over a future horizon since they would require knowledge of the state and future trajectory of each robot.

In the following we will take as reference the E-criterion, which is linked to the least observable mode and to the worst case rate of convergence of the EKF. From Sect. 4.3.4, we know that, both for distance and bearing measurements, the null space of the OG is at best of dimension 4 and it is spanned by a common translation and a coordinated rotation about the z-axis. Hence, the E-criterion can be considered related to the first nonzero eigenvalue of the OG, i.e. λ_5. Using the *Rayleigh quotient* [Meyer & Stewart, 2023, p.530], λ_5 can be expressed as

$$\lambda_5 = \boldsymbol{v}_5^T \boldsymbol{G}_o \boldsymbol{v}_5 = \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{v}_{5\mathcal{N}_i}^T \boldsymbol{G}_{o\mathcal{N}_i} \boldsymbol{v}_{5\mathcal{N}_i} \tag{4.51}$$

where $\boldsymbol{v}_5$ is the unit norm eigenvector associated to the eigenvalue λ_5. Moreover, $\boldsymbol{G}_{o\mathcal{N}_i} \in \mathbb{R}^{3|\mathcal{N}_i^+|\times 3|\mathcal{N}_i^+|}$, with $\mathcal{N}_i^+ = \{i, j | j \in \mathcal{N}_i\}$, contains the blocks of the OG associated to robot i and its neighbors and $\boldsymbol{v}_{5\mathcal{N}_i} \in \mathbb{R}^{3|\mathcal{N}_i^+|}$ contains the corresponding components of the eigenvector $\boldsymbol{v}_{5\mathcal{N}_i}$. We also point out that $\boldsymbol{G}_{o\mathcal{N}_i}$ can be seen as the OG corresponding to the star subgraph $\mathcal{G}_i$ with center of the star given by the vertex i, hence, all the properties of the OG discussed in Sect. 4.3.4 apply.

Notice that, $\lambda_5 > 0$ if and only if there is at least one robot i such that

$$\boldsymbol{v}_{5\mathcal{N}_i}^T \boldsymbol{G}_{o\mathcal{N}_i} \boldsymbol{v}_{5\mathcal{N}_i} > 0 \tag{4.52}$$

and this is the case, if and only if

$$\boldsymbol{v}_{5\mathcal{N}_i}^T \boldsymbol{G}_{o\mathcal{N}_i} \boldsymbol{v}_{5\mathcal{N}_i} \geq \|\boldsymbol{v}_{5\mathcal{N}_i}\|^2 \lambda_{5\mathcal{N}_i} > 0, \tag{4.53}$$

where $\lambda_{5\mathcal{N}_i}$ is the 5th eigenvalue of $\boldsymbol{G}_{o\mathcal{N}_i}$, which only depends on the trajectories of robot i and its neighbors. Moreover, if the graph is connected and $\lambda_{5\mathcal{N}_i} > 0 \ \forall i$ then $\lambda_5 > 0$. Therefore, the active sensing term in the cost function is given as

$$J_{as} = \sum_{i=1}^{N} J_{as_i} = \sum_{i=1}^{N} -\log(\lambda_{5\mathcal{N}_i}) \tag{4.54}$$

where each robot i can compute its corresponding term. Notice that, the cost function goes to $+\infty$ as one of the $\lambda_{5\mathcal{N}_i} \to 0$. Hence, if the cost function remains bounded, then all the $\lambda_{5i} > 0$ and, as a consequence, $\lambda_5 > 0$.

Furthermore, important things to consider are that, as stated in [Krener & Ide, 2009] and discussed in Sect. 1.2.1,

- computation of the entire local observability gramian might not be necessary; one could focus on significant state directions
- properly scaling the state coordinates $\boldsymbol{q}$ is crucial for meaningful numerical interpretations of the OG eigenvalues

These two considerations are important in light of the fact that, as shown in Sect.4.3.4, in the OG, the upper left block is related to the collected information about the position, while the lower right block is related to the information about the yaw orientations and the off-diagonal blocks represent cross information. The problem is that, numerically, positions and orientations can have very different scales and, as a consequence, hinder the active sensing task. For this reason, we only consider the part of the OG relative to the positions $\boldsymbol{G}_{op}^*$.

Remark 3. *One could include in the cost function a term which aims at generating PE trajectories also directly aimed at improving the yaw estimation convergence by considering the PE conditions for the convergence of an observer scheme similar to the one which will be presented in Chapter 8.*

Remark 4. *As in previous works [Spica & Robuffo Giordano, 2016], we do not directly consider the minimum eigenvalue, as the* $\min(\cdot)$ *function is not differentiable when the*

order of the two minimum eigenvalues is swapped. Instead, we consider a smooth approximation of the minimum function, e.g. consider a vector $\boldsymbol{w} = \begin{bmatrix} w_1 & \dots & w_d \end{bmatrix}^T$ *and* $\eta > 0$*, the log-sum-exp function is defined as [Boyd & Vandenberghe, 2004, p.72]*

$$\min(\boldsymbol{w}) \approx -\frac{1}{\eta} \ln \left(\sum_{k=1}^{d} e^{-\eta w_k} \right) \tag{4.55}$$

where as η *increases the approximation gets more accurate and for* $\eta \to +\infty$ *the approximation tends to the minimum function.*

4.4.2 Problem formulation

In this section, we formulate the optimal control problem to be solved.

$$\begin{aligned}
&\min_{\bar{\boldsymbol{x}}} . \sum_{i=1}^{N} J_{as_i}(\bar{\boldsymbol{x}}_i, \{\bar{\boldsymbol{x}}_j\}_{j \in \mathcal{N}_i^+}) \\
&\text{s.t.} \quad \forall k \in \{0, 1, \dots, N_{wp} - 1\}, \quad \forall i \in \{1, \dots, N\} \\
&\qquad \boldsymbol{q}(k=0) = \begin{bmatrix} \boldsymbol{0}_d \\ \hat{\boldsymbol{x}}(t) \end{bmatrix} \\
&\qquad \boldsymbol{G}_{o\mathcal{N}_i}(k=0) = \boldsymbol{P}_{\mathcal{N}_i}(t)^{-1} \\
&\qquad \|\boldsymbol{p}_i(k) - \boldsymbol{p}_i(k-1)\|_2 \leq v_{\max} \Delta T \\
&\qquad |\psi_i(k) \ominus \psi_i(k-1)| \leq \omega_{\max} \Delta T
\end{aligned} \tag{4.56}$$

where $\bar{\boldsymbol{x}}$ is used to indicate the stack of the waypoints along the future horizon. The first two constraints concern the initial conditions, which are obtained from the EKF that estimates the state of the group in the anchor frame $\hat{\boldsymbol{x}}$ (hence the zeros stacked on top which are associated to the state of the anchor at the start of the prediction), as well as $\boldsymbol{P}_{\mathcal{N}_i}$ which represents the blocks of the covariance matrix from the dEKF corresponding to robot i and its neighbors. The other constraints are respectively the constraints on the Cartesian velocity and the angular velocity, where $\ominus$ indicate a difference between two angles.

Collision avoidance is discouraged by the perception weights (due to the minimum sensing range) but only at the waypoints. Additional safety guarantees are added at a lower level using CBFs [Wang et al., 2017]. Also, connectivity maintenance is required for the convergence of the EKF and of the distributed optimization algorithm, but it is not

guaranteed by our optimal control algorithm which discourages losing edges but it does not prevent local disconnections and possible loss of global connectivity. Connectivity maintenance is then added at a lower level analogously to previous works on this topic, e.g., [Robuffo Giordano et al., 2013].We also point out that, due to sensing limitations, the graph is time-varying but future new connections cannot be foreseen. Hence, the graph is considered fixed during the future horizon.

In this work, we did not consider other tasks than active sensing, although normally robots do not need to localize themselves for its own sake. Integrating another task could be achieved through the use of control Lyapunov functions similarly to [Napolitano et al., 2022]. This is particularly common in scenarios where an economic (performance) objective is considered rather than a tracking objective. The resulting MPC is defined *Economic* MPC and a control Lyapunov function-based condition is often included to ensure stability of the control scheme. We plan to add such condition in future works.

Remark 5. *One may wonder if the fact of using the estimated poses to compute the OG in order to control the system to improve the estimation accuracy may represent an undesirable loop, only necessary in absence of the true trajectory. Actually, it was shown in [P. Bernard et al., 2020] that, the convergence of an EKF-like filter depends on the Gramian along the estimated trajectory and not along the true trajectory. It is expected a similar result to hold for the regular EKF since the lower-bound of the Riccati solution is related to the same Gramian.*

4.5 Distributed partition-based optimization

The problem formulated in (4.56), is in the following generic form:

$$\begin{aligned} \min_{\boldsymbol{x}_o} . & \sum_{i=1}^{N} J_i(\{\boldsymbol{x}_{oj}\}_{j\in\mathcal{N}_i^+}) \\ \text{s.t.} \quad & a_i(\{\boldsymbol{x}_{oj}\}_{j\in\mathcal{N}_i^+}) \leq 0 \\ & b_i(\boldsymbol{x}_{oi}) = 0 \end{aligned} \tag{4.57}$$

which can to be addressed with distributed optimization algorithms.

A problem with this kind of sparsity is common in robotics and usually referred to in the literature as *distributed partitioned optimization* or *separable variable optimization* [Chezhegov et al., 2022; Halsted et al., 2021]. For such problems, it is shown in [Chezhegov

et al., 2022] that common distributed optimization algorithms can be easily modified in order to take into account the fact that, in the consensus steps of the algorithm, a single robot does not need to know the whole optimization vector, so that the algorithm can be performed more efficiently [Chezhegov et al., 2022]. To obtain this result, the consensus matrix $\boldsymbol{W}_p$ is specifically designed. In our case, given the N star subgraphs $\mathcal{G}_i$ associated to each robot, the consensus matrix can be written as

$$\boldsymbol{W}_p = \left(\sum_{i=1}^{N} \boldsymbol{W}_c(\mathcal{G}_i) \otimes (\boldsymbol{e}_i \boldsymbol{e}_i^T) \right) \otimes \boldsymbol{I}_d \tag{4.58}$$

where $\boldsymbol{e}_i$ is the unit vector with entry 1 in the i-th position. We consider the consensus matrix $\boldsymbol{W}_c(\mathcal{G}_i)$ obtained using the *Metropolis-Hastings* weights (see [Nedić et al., 2018] and [Bullo, 2020, p.74].

Remark 6. *Some of the optimization variables are angles $\theta \in \mathbb{S}^1$, for these quantities the usual consensus is modified as [Bullo, 2020]:*

$$\theta_i[n+1] = \theta_i[n] - \epsilon \sum_{j \in \mathcal{N}_i^+} \sin(\theta_i[n] - \theta_j[n]). \tag{4.59}$$

As many other optimization problems arising in multi-robot control, our problem is non-convex. Contrarily to centralizated optimization algorithms for which convergence to a local minima of the problem is, in general, guaranteed, the same does not hold for distributed optimization algorithms. Hence, the employed algorithm needs to be chosen with care.

As distributed optimization algorithm, we use **NEXT** [Di Lorenzo & Scutari, 2016], which handles a cost function given by the sum of smooth possibly non-convex functions, and it allows considering a time-varying graph. NEXT is composed mainly of three steps. 1) A local minimization problem, formulated using a convex surrogate function, is solved and a temporary minimizer is obtained. 2) A partial step in the direction of this temporary minimizer is taken. 3) The consensus on the optimization vector is performed along with a dynamic consensus on the gradient of the total cost function. A surrogate of the local cost function $\tilde{J}_i(\boldsymbol{x}_o, \boldsymbol{x}_o[n])$ is considered, where $\boldsymbol{x}_o[n]$ is the value of $\boldsymbol{x}_o$ at iteration n. $\tilde{J}_i$ has to be a convex, local, approximation of J_i. We take $\tilde{J}_i$ to be a linearization of J_i around the value of the local copy $\bar{\boldsymbol{x}}_{oi}[n]$ of the optimization variables $\{\boldsymbol{x}_{oj}\}_{j \in \mathcal{N}_i^+}$ with an additional regularization term:

$$\begin{aligned}\tilde{J}_i(\bar{\boldsymbol{x}}_{oi}, \bar{\boldsymbol{x}}_{oi}[n]) = J_i(\bar{\boldsymbol{x}}_{oi}[n]) + \nabla J_i(\bar{\boldsymbol{x}}_{oi})^T(\bar{\boldsymbol{x}}_{oi} - \bar{\boldsymbol{x}}_{oi}[n]) + \\ + \frac{\tau_{ip}}{2} \left\| \boldsymbol{E}_{\boldsymbol{p}}(\bar{\boldsymbol{x}}_{oi} - \bar{\boldsymbol{x}}_{oi}[n]) \right\|_2^2 + \frac{\tau_{i\psi}}{2} \left\| \boldsymbol{E}_{\boldsymbol{\psi}}(\bar{\boldsymbol{x}}_{oi} - \bar{\boldsymbol{x}}_{oi}[n]) \right\|_2^2\end{aligned} \tag{4.60}$$

where we regularized differently a position variation w.r.t. a yaw variation, $\boldsymbol{E}_p$ (resp. $\boldsymbol{E}_{\psi}$) is used to select the position (resp. yaw) components.

The local problem that each robot needs to solve is the following

$$\tilde{\boldsymbol{x}}_{oi}(\bar{\boldsymbol{x}}_{oi}[n], \tilde{\pi}_i[n]) \triangleq \underset{\bar{\boldsymbol{x}}_{oi} \in \mathcal{X}_i}{\operatorname{argmin}} \tilde{J}_i(\bar{\boldsymbol{x}}_{oi}, \bar{\boldsymbol{x}}_{oi}[n]) + \tilde{\pi}_i[n]^T(\bar{\boldsymbol{x}}_{oi} - \bar{\boldsymbol{x}}_{oi}[n]) \tag{4.61}$$

where $\tilde{\pi}_i[n]$ is an estimation of $\pi_i(\bar{\boldsymbol{x}}_{oi}[n]) \triangleq \sum_{j \neq i} \nabla_{\bar{\boldsymbol{x}}_{oi}} J_j(\bar{\boldsymbol{x}}_{oj}[n])$, which is used to consider a linearization of the rest of the cost function not including the local cost J_i, i.e. $\sum_{j \neq i} J_j(\boldsymbol{x}_o[n])$. $\tilde{\pi}_i[n]$ is obtained using *dynamic average consensus* to estimate the total gradient and then subtracting the gradient of the local cost $\nabla_{\bar{\boldsymbol{x}}_{oi}} J_i$. Then, a partial step is taken

$$\boldsymbol{z}_i[n] = \boldsymbol{x}_i[n] + \alpha[n](\tilde{\boldsymbol{x}}_{oi}[n] - \bar{\boldsymbol{x}}_{oi}[n]) \tag{4.62}$$

with $\alpha[n]$ being the step-size. After this, a step of consensus is taken on the variables $\boldsymbol{z}_i[n]$ to obtain $\bar{\boldsymbol{x}}_{oi}[n+1]$, as well as a step of dynamic consensus on the total gradient. We notice that (4.61) admits an analytical solution as

$$\begin{aligned}\tilde{\boldsymbol{x}}_{oi}[n] = \Pi_{\mathcal{X}} \Bigg(\boldsymbol{x}_{oi}[n] - \\ - \left(\frac{1}{\tau_{ip}} \boldsymbol{E}_p^T \boldsymbol{E}_p + \frac{1}{\tau_{i\psi}} \boldsymbol{E}_{\psi}^T \boldsymbol{E}_{\psi} \right) (\nabla J_i(\boldsymbol{x}_{oi}[n]) + \tilde{\pi}_i[n]) \Bigg)\end{aligned} \tag{4.63}$$

where $\Pi_{\mathcal{X}}$ projects the solution onto the feasible set. We also perform line search based on *Armijo's condition* [Nocedal & Wright, 1999] by checking that actual improvements across iterations take place by re-evaluating the local cost function.

For the online optimization, it is assumed that, the first time two robots become neighbors, they will communicate the required current information to initialize their respective optimization variables.

4.6 Results

In this chapter, we have unified our discussion to encompass both scenarios involving distance measurements and bearing measurements. However, it is important to note that our previous work, where active sensing was conducted solely from relative distance measurements [De Carli et al., 2021], covered only a portion of the algorithm outlined

here. Specifically, in that work, we did not employ distributed optimization techniques and the cost function was not exactly the same.

Regarding the case of relative distance measurements, the results presented in this chapter are the ones originally outlined in [De Carli et al., 2021], hence not including the complete algorithm.

4.6.1 Relative distance measurements

For illustration purposes, we first consider a group of $N = 3$ robots that need to localize themselves using relative distance measurements, and we implement the distributed EKF algorithm from [Luft et al., 2016] for estimating the robot states in a distributed way. The measured distances are affected by Gaussian noise with covariance $r_{ij} = 0.3 \cdot T_s m^2$ where T_s is the sampling period of the EKF. The initial state estimation $\hat{\boldsymbol{x}}(t_0)$ is affected by a Gaussian noise with covariance $\sigma^2_{xyz0} = 0.4m^2$ on the position and $\sigma^2_{\psi 0} = 0.3rad^2$ on the yaw orientation. The thresholds defining the weights w_{ij} (see Fig. 2.7) are $D_l = 5m$ and $D_u = 7m$. The inputs are parameterized using B-Splines [Biagiotti & Melchiorri, 2008] with $N = 6$ control points and their limits are set to $\boldsymbol{u}_l = -1$ and $\boldsymbol{u}_u = 1$. Thanks to the B-spline parameterization, the robots can simulate the neighbors future trajectories by exchanging only a limited amount of information (the B-Spline control points), and they can also reduce the input constraints dimension considering the convex hull of the control points.

Starting from 4 different fixed configurations, such that the initial sensing graph is connected, we generated 10 random paths of equal duration t_f taken as initial guesses for the OCP. In this work, each robot was solving independently a local optimization problem maximizing the minimum eigenvalue of the gramian block corresponding to its own state $\boldsymbol{\mathcal{G}}_{o(ii)}$, indicated as SAME in the plots. Results were also compared to the case in which the minimum nonzero eigenvalue of the full OG $\tilde{\lambda}_{\min}$ was optimized in a centralized way. Each trajectory (initial guess or optimized one) is executed 10 times with a different initial condition for the EKF drawn from a Gaussian centered at the true $\boldsymbol{x}(t_0)$ with covariance $\boldsymbol{P}_0$. When executing the initial non-optimized trajectories, the sensing graph is considered fully connected (all relative measurements always available to the EKF) disregarding the sensing constraints, which are instead fully considered in the optimized cases (thus, resulting in a time-varying sensing graph). The final configuration $\boldsymbol{x}_f$ is not fixed while the final time is fixed at $t_f = 15s$.

The average of the maximum eigenvalue (corresponding to the maximum uncertainty)

and of the trace (corresponding to the average uncertainty) of $\boldsymbol{P}(t_f)$, as well as the Root Mean Square Error (RMSE) at t_f obtained optimizing the different metrics are compared and results are shown in Table 4.1. One can note how, in average, the optimized cases perform better than the initial guesses.

One advantage of the proposed approach over a strategy based on instantaneous observability (like rigidity-based localization) is that the robot group has much more freedom in terms of mobility, which can be particularly useful when navigating in cluttered environments. In this respect Table 4.2 shows the results of a single illustrative run during which, for some optimized trajectories, the group does not remain rigid at all times because of the sensing constraints (the rigidity eigenvalue, a common measure of rigidity [Zelazo et al., 2015], is reported in Fig. 4.2). Despite the loss of rigidity, the group is still able to successfully localize and achieve good performance in terms of estimation accuracy (Table 4.2).

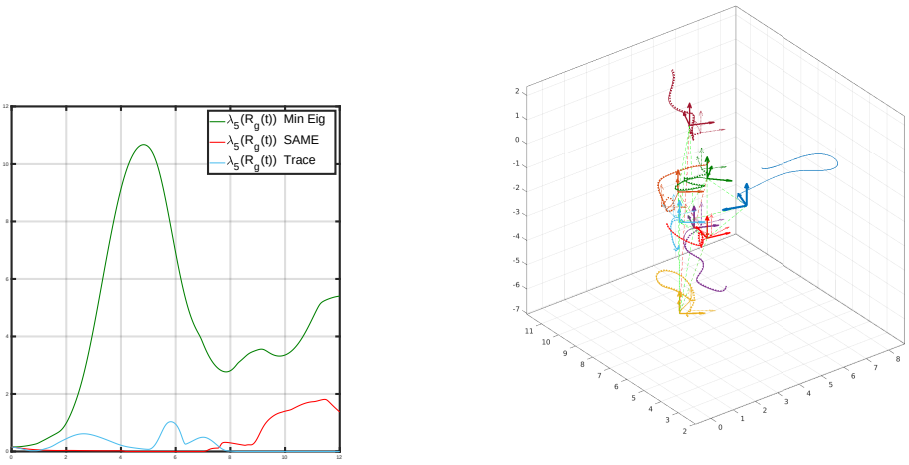


Figure 4.2: An optimal trajectory which does not maintain the formation rigidity. On the left, the rigidity eigenvalue is plotted. On the right the robot trajectories are shown, with the frames in the initial positions (in blue is robot 1) and in dotted lines the estimated trajectories.

The same simulations have also been performed by considering $N = 8$ robots and by using $r_{ij} = 0.1 \cdot T_s m^2$ and $N = 8$. The results are shown in Table 4.3 and show the good performance also in this case. The attached video allows to better appreciate the robot trajectories and behavior of the estimation errors for this case study.

Metric	$\lambda_{max}(\boldsymbol{P}_{t_f})$	trace$(\boldsymbol{P}_{t_f})$	$RMSE$
Random	0.091548	0.12742	0.1286
$\tilde{\lambda}_{min}$	0.011747	0.017749	0.0362
SAME	0.019641	0.023787	0.0466

Table 4.1: Average of the maximum and average uncertainty and RMSE at t_f for 3 robots

Metric	$\lambda_{max}(\boldsymbol{P}_{t_f})$	trace$(\boldsymbol{P}_{t_f})$	$RMSE$
Random	0.0198620	0.0270547	2.1273e-04
$\tilde{\lambda}_{min}$	0.0002825	0.0014602	7.0901e-06
SAME	0.0024085	0.0073659	1.3127e-04
Trace	0.0512124	0.0550008	1.7742e-04

Table 4.2: Maximum and average uncertainty and RMSE at t_f with SAME not preserving rigidity

4.6.2 Relative bearing measurements

In case of relative bearing measurements, we validate the effectiveness of the approach through a simulation campaign. We consider the results obtained using groups composed by $N = 8$ quadrotors modeled as in (4.1). An initial random connected configuration is generated along with the initial non-optimized waypoints. The initial waypoints are generated such that the robots have the same relative positions as the initial configuration simply translated in space in a direction randomly extracted by a conic region plus an additional perturbation which makes the system observable along the trajectory, and which is checked to preserve the connectivity of the group. This initial guess trajectory is near to be unobservable along certain state directions. The interest of this initial guess trajectory is that it represents a typical trajectory that would be performed in a formation flight without considering additional state observability requirements.

The following sensing limitations are considered for the minimum distance $d_{\min} = 0.6m$, maximum distance $d_{\max} = 5m$ and FoV $\alpha_{\max} = 70°$. Measurements are acquired on the image plane which are affected by an additive Gaussian noise η_{ij} with variance $\boldsymbol{Q}_{ij} = 0.04\boldsymbol{I}_2$. Also, we consider the system to be affected by actuation noise $\boldsymbol{\nu}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{N}_{ij})$ with $\boldsymbol{N}_{ij} = \text{diag}([0.01, 0.01, 0.008, 0.002])$. The robots can fly up to $6m/s$ and rotate at a

Metric	$\lambda_{max}(\boldsymbol{P}_{t_f})$	trace$(\boldsymbol{P}_{t_f})$	$RMSE$
Random	0.028253	0.041103	0.0621
$\tilde{\lambda}_{min}$	0.00082939	0.0027855	0.0052
SAME	0.0087442	0.015267	0.0249
Trace	0.023022	0.029117	0.0513

Table 4.3: Average of the maximum and average uncertainty and RMSE at t_f for 8 robots

speed up to $0.4 rad/s$. The state is estimated using the distributed EKF [Luft et al., 2016], which also provides the covariance matrices $\boldsymbol{P}_i(t)$ used by the OCP. The measurement confidence for the EKF has been tuned by trial and error as $v = 3.3$. The initial state estimate is obtained adding Gaussian noise with standard deviation $\sigma_p = 0.4$ along each axis for the position and $\sigma_\psi = 0.03$ for the yaw. In the OCP we consider the trajectory to be optimized over $N_{wp} = 10$ waypoints equally spaced in time at $\Delta T = 0.5s$.

Starting from 10 different initial configurations, we generated 7 trajectories for each configuration, each with different noise realizations (applied to the initial estimate as well). We assess the performance of localization obtained along the following trajectories:

1. The initial guess trajectory.
2. The trajectory optimized with the proposed algorithm but only offline, with $N_{iter} = 500$ iterations.
3. The trajectory previously optimized offline and then further optimized online.
4. The trajectory optimized in a centralized manner, with $N_{iter} = 150$ iterations.

The results corresponding to these simulations are reported in Fig. 4.3, which shows on the left the average and standard deviation across simulations of the maximum eigenvalue of the matrix $\boldsymbol{P}$ given by the EKF. Notice that, for a deterministic EKF, this has not anymore exactly the meaning of a covariance, but it is anyway representative of the effect of the information acquired along the path. In the figure on the right, we show the average and standard deviation across simulations of the Root Mean Square Error (RMSE) achieved for the position and yaw estimation. The obtained results at the final time are also summarized in Table.4.4.

The results show the improvement obtained by refining the trajectory online as newly formed edges can be exploited to improve the information. Also, the results make it evident how the estimation deteriorates in time for the case of a formation-like flight. The plots also show quite surprising results when comparing the trajectories optimized in a distributed way or in a centralized way. In fact, concerning the maximum eigenvalue of $\boldsymbol{P}(t)$, the centralized approach obtains the best results (as expected). But, these results do not correspond to better performance in RMSE compared to the distributed approach. This can be explained by the fact that the matrix $\boldsymbol{P}$ does not capture the true confidence in the measurements. Furthermore, $\lambda_{\max}(\boldsymbol{P})$ is linked to worst case convergence speed, so no guarantees are given that a lower $\lambda_{\max}$ will obtain faster convergence. The number

of iterations performed respectively by centralized and distributed optimization is chosen s.t. the computational times for the offline optimization are similar, with the distributed optimization running sequentially i.e. on a single laptop, while in the real world it would run in parallel (but communication overhead would need to be considered). In this way, the distributed approach is also penalized in this sense.

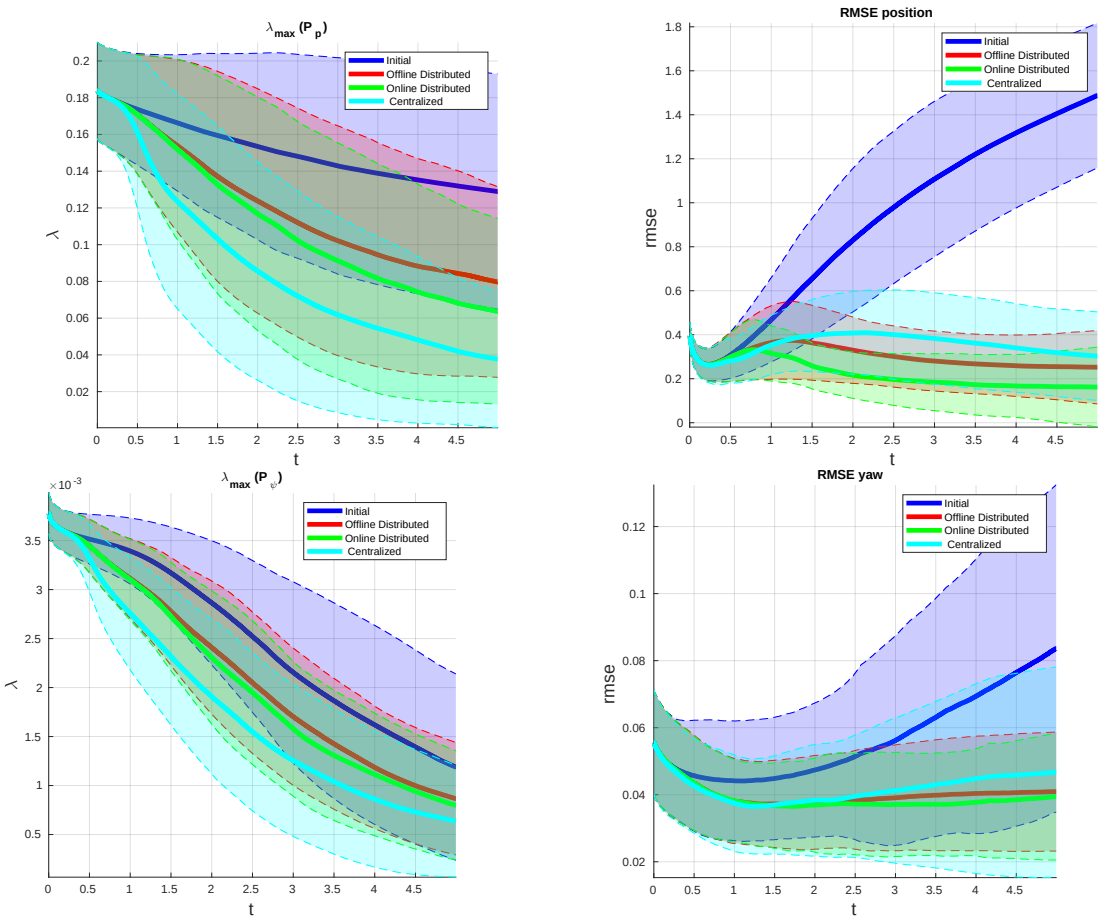


Figure 4.3: Comparison of the average estimation results with plot of the standard deviation. The thick lines represent the average result, while the dashed lines are used for the standard deviations. The upper (resp.lower) left figure is the maximum eigenvalue of of the sub-block of $\boldsymbol{P}$ corresponding to the position (resp. yaw). On the right the corresponding RMSEs are plotted.

Finally, we also analyze the results of the distributed optimization across iterations. In Fig. 4.4, we show the average and standard deviation of the optimization results obtained from 15 different initial configurations and initial guess trajectories. This time we consider more iterations, $N_{iter} = 1000$, of offline optimization using NEXT, and one can check that

Trajectory	$\lambda_{max}(\boldsymbol{P}_{t_f})$	trace$(\boldsymbol{P}_{t_f})$	MSE$_p$	MSE$_\psi$
Initial guess	0.12058	0.38001	1.6175	0.09792
Dist. Opt. Offline	0.073581	0.19767	0.1985	0.044637
Dist. Opt. Online	0.069562	0.18355	0.1894	0.043781
Centralized	0.032128	0.096087	0.2168	0.056383

Table 4.4: Comparison of average estimation results achieved at the end of the trajectory

at some point the objective stops growing also due to the term $\alpha[n]$ which is decreasing during the iterations. On the left the objective function that we are actually maximizing is reported, in the center image we report the individual objective functions associated to each robot, and finally the plot on the right shows the corresponding evolution of the minimum eigenvalue of the position sub-block of the full gramian, which is the objective we are actually maximizing. This is increasing, showing that the chosen heuristic is actually achieving the desired result. In the plot on the right we also compare the evolution of the distributed optimization w.r.t. the result obtained by the centralized optimization using $N_{iter} = 300$ iterations, which has similar computational times to $N_{iter} = 1000$ of distributed optimization when performed on a single computer (not taking advantage of the parallelization of distributed optimization). The significantly higher value obtained by the 8-th robot seems to be related to the way in which we generate the initial random formation. We sequentially add robots such that the generated robot is in the field of view and in the range of the preceding robot (with the 8-th robot being the last added), hence, this affect the average connectivity degree of each robot.

Concerning the computational times, each iteration of distributed optimization takes in average $8ms$ with non-optimized code running on MATLAB on a Dell Latitude 7400. It is important to consider that this result does not consider overhead due to communication and the need for synchronization or packet losses, these remain challenging problems in applying distributed optimization algorithms to real-world scenarios [Halsted et al., 2021].

4.7 Discussions

While the concept of incorporating a future horizon is undoubtedly enticing for active sensing applications, it introduces several challenges, particularly concerning the optimization of non-standard cost functions associated with the eigenvalues of nonlinear matrix functions. Optimizing such functions is notoriously difficult, a challenge further compounded by the necessity for distributed implementation of the optimization algorithm. Further-

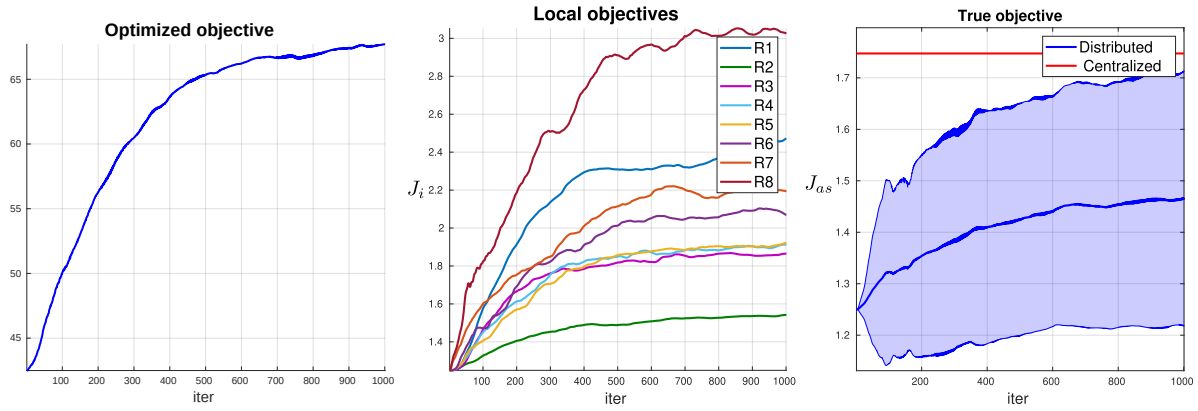


Figure 4.4: In the figure on the left, the total objective function which is distributedly optimized, in the center the corresponding individual non-weighted objectives, on the right a comparison among the optimal value got from centralized optimization and the corresponding centralized objective obtained by distributed optimization. The plots are obtained as average and, where patches are used, they represent the standard deviation. Results are obtained from 15 different initial configurations.

more, a distributed optimization based approach to MPC comes with the requirement for high communication loads among the robots. While recent advancements in distributed optimization have made distributed MPC algorithms increasingly popular, as evidenced by recent works [Lyu et al., 2019; Shorinwa & Schwager, 2023; W. Tang & Daoutidis, 2019], real-world implementations of such algorithms remain relatively scarce. For these reasons, in the next chapters, we will always consider reactive control laws with no future horizon.

4.8 Conclusions

In this chapter, we have shown that, for a multi-robot system even when only partial measures of the relative states are available, such as relative distances or bearings (but not limited to), and when the formation is not infinitesimally rigid, it is still possible to achieve good localization performance by generating, in a distributed and scalable way, optimized trajectories that satisfy certain PE conditions.

Our proposed framework addresses the active cooperative localization problem, leveraging a surrogate for the E-optimality criterion. While our primary objective was to enhance localization performance, it is crucial to recognize that effective localization is a necessary requirement for the successful and safe execution of any desired task but it is

not (usually) the task itself. Looking ahead, future extensions of this work should explore incorporating additional tasks into the framework. A potential approach could be based on the formulation of a Lyapunov-based EMPC as in [Napolitano et al., 2022]. While, in the next chapter, active sensing is performed with a reactive strategy and an additional task is incorporated using a QP-based formulation relying on CLFs.

Part III

Embedding Active Perception in Reactive Control

Chapter 5

Bearing-based multi-robot collaboration: beyond infinitesimal rigidity

In this chapter, similarly to the previous one, we consider a group of quadrotors equipped with sensors capable of measuring relative bearing. The objective is to enable the fleet to localize their relative poses in a common shared frame without imposing a rigidity requirement, all while executing a designated task reliant on inter-robot localization. To attain this goal, we introduce a novel distributed active sensing control strategy, which maximize the information acquisition without compromising task fulfillment.

In contrast to the previous chapter, this chapter introduces a new approach to distributed active sensing control, which is reactive and based on a gradient-based strategy. This method not only simplifies computational and communication tasks but also integrates more seamlessly with additional tasks. We achieve this integration using Control Lyapunov Functions (CLFs) to ensure the successful completion of tasks. Specifically, we are addressing a position-based formation control task concurrently, employing Quadratic Program-based control with a CLF linear constraint.

The results show that the inclusion of active sensing in the formation control law enhances the localization accuracy and, as a consequence, the precision of reaching the desired formation. The improvement is especially important when the underlying graphs are not Infinitesimally Bearing Rigid (IBR), as it can be expected.

As already discussed in the previous chapters, much of the previous work in the domain of cooperative localization from bearing measurements has been built around the notion of infinitesimal bearing rigidity [Schiano et al., 2016; Zhao & Zelazo, 2019]. IBR characterizes the conditions, primarily related to graph topology, under which a group of drones measuring constant relative bearings can localize themselves up to a global translation and scaling factor. In case of constant bearings, to resolve the scale ambiguity, a

distance measurement among a pair of robots is necessary.

Recently, in [Z. Tang et al., 2022; Z. Tang & Loria, 2022] the notion of *Bearing Persistently Exciting* (BPE) formations has been introduced, where at least a subset of the relative bearings is time-varying. By leveraging the concept of PE, this allows resolving the scale ambiguity and to relax the IBR assumption. Additionally, a distributed observer has been proposed to estimate relative positions in $\mathbb{R}^3$ from world frame relative bearing measurements, demonstrating Uniform Exponential (UE) convergence under the BPE formation condition [Z. Tang et al., 2022]. In this work, we use this observer to estimate the positions of robots in a shared frame.

However, there remains the problem of how to ensure the Bearing Persistent Excitation of the formation. Although a coordinated rotation was proposed as a reference trajectory satisfying the BPE condition [Z. Tang et al., 2022], this strategy may not be desirable in practical applications where more generic motions are needed. In such cases, active sensing can be exploited for producing a group motion that can (i) actively satisfy the necessary BPE conditions and (ii) be generic enough for fulfilling at the same time other tasks of interest related to the particular mission.

A preliminary version of this work was previously published in [De Carli et al., 2023]. In this work, our focus was on a scenario with a fixed graph, without considering sensing constraints, and where the robots states were represented solely by their positions in $\mathbb{R}^3$; orientation was not yet taken into account. Subsequently, we expanded upon this initial work by incorporating sensing constraints, specifically limited range and field of view, which led to the graph becoming time-varying. While the core active sensing control law remained largely unchanged, we made adjustments, such as introducing a gradient-based active sensing action to control the robots yaw, addressing the limited field of view of the cameras. Additionally, with the graph becoming state-dependent, the implementation of a connectivity maintenance algorithm became essential. In this regard, we leveraged the connectivity maintenance algorithm which is presented in Chapter 7 [De Carli, Salaris, & Robuffo Giordano, 2024], which uses Control Barrier Functions (CBFs) to make minimal adjustments to the desired input, derived, in this case, from the combination of active sensing and other concurrent tasks.

In this chapter, we first describe the work presented in [De Carli et al., 2023] and then we discuss its extension, which, at the moment of writing, is still under development. Among the missing parts is the observer from *body-frame* bearing measurements extending the one presented in [Z. Tang et al., 2022] to $\mathbb{R}^3 \times \mathbb{S}^1$. We are working towards the

formulation of such an observer, starting from ideas in [Z. Tang et al., 2022] and the position-based observer presented in Chapter 8, but the work is still ongoing. In the first part of the chapter, we first introduce the robot and sensing model in $\mathbb{R}^3$, then, we describe the localization algorithm, discussing the conditions under which the algorithm converges. Subsequently, we design a gradient-based active sensing control law which drives the robots along informative trajectories. We add the task achievement to the active sensing action in the form of a linear inequality constraint using the control Lyapunov function formalism and we conclude the first part of the chapter showing the results. In the second part of the chapter, after introducing the robot and sensing model in $\mathbb{R}^3 \times \mathbb{S}^1$, we describe the extension of the active sensing control law actively considering sensing limitations.

5.1 Formation model

As outlined in the introduction to this chapter, we address both scenarios: one utilizing a simple single integrator model in $\mathbb{R}^3$ for the robots, described by:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i, \tag{5.1}$$

and the other employing the robots model (4.1) in $\mathbb{R}^3 \times \mathbb{S}^1$. The first part of the chapter deals with the single integrator model, only later we delve into the extended case.

In this case, each drone is able to measure the world-frame relative bearing with respect to its fixed neighbors, i.e.:

$$\boldsymbol{\beta}_{ij} := \frac{\boldsymbol{p}_{ij}}{d_{ij}} \in \mathbb{S}^2. \tag{5.2}$$

In the next section, we present the observer from [Z. Tang et al., 2022], along with the observability conditions under which it converges.

5.2 Localization of bearing formations

The matrix-valued Laplacian associated with classical position-based formation control, as defined in [Pan et al., 2020], is denoted as $\boldsymbol{L}_3 := \boldsymbol{E}_3 \boldsymbol{E}_3^T$ [Cortes & Egerstedt, 2017]. If the graph is connected, then $\text{rank}(\boldsymbol{L}_3) = 3N - 3$, and the null-space is $\mathcal{N}(\boldsymbol{L}_3) = \boldsymbol{U}$, where $\boldsymbol{U} = \mathbf{1}_N \otimes \boldsymbol{I}_3$ represents a basis for a common 3D translation of the formation.

For bearing formations, the bearing Laplacian matrix is defined as $\boldsymbol{L}_\beta := \boldsymbol{E}_3 \operatorname{diag}(\boldsymbol{\Pi}_{\beta_k})\boldsymbol{E}_3^T$. As discussed in previous chapters, a bearing formation in $\mathbb{R}^3 \times \mathbb{S}^1$ is *infinitesimally bearing rigid* (IBR) if $\operatorname{rank}(\boldsymbol{L}_\beta) = 3N - 5$ (see also Sect. 2.4). This IBR condition is both necessary and sufficient for localizing the relative positions of the robots up to a scale factor from constant bearing measurements. Moreover, the IBR condition can be relaxed if the bearings are time-varying, i.e., certain PE conditions are satisfied, and the scale ambiguity can be resolved (see Sect. 4.3.4).

We now give some necessary definitions before proceeding.

Definition 5.2.1 (Bearing persistently exciting)**.** *[Z. Tang et al., 2022] A formation in $\mathbb{R}^3$ is defined bearing persistently exciting (BPE) if the graph $\mathcal{G}$ is connected and given $T > 0$ and $\gamma > 0$ the following PE condition holds:*

$$\int_t^{t+T} \boldsymbol{L}_\beta(\tau)\, d\tau = \boldsymbol{E}_3 \operatorname{diag}\left(\int_t^{t+T} \boldsymbol{\Pi}_{\beta_k}(\tau)\, d\tau\right) \boldsymbol{E}_3^T \succeq \gamma \boldsymbol{L}_3. \tag{5.3}$$

Notice that, since $\mathcal{N}(\boldsymbol{L}_3) = \boldsymbol{U}$, this is a weaker condition than classical PE, i.e.

$$\int_t^{t+T} \boldsymbol{L}_\beta(\tau)\, d\tau \not\succeq \gamma \boldsymbol{I}_3. \tag{5.4}$$

We point out that the previous integral in (5.3) represents a weighted OG for the position part of the state of the system. In fact, if we do not consider the perception weight, $\boldsymbol{G}_{op}^b$ from the previous chapter is written as

$$\boldsymbol{G}_{op}^b = \boldsymbol{E}_3 \operatorname{diag}\left(\int_{t_0}^{t_f} \frac{\boldsymbol{\Pi}_{\bar{\beta}_k(\tau)}}{d_k^2(\tau)}\, d\tau\right) \boldsymbol{E}_3^T \tag{5.5}$$

One can see that the only difference is that, in OG, the information associated to each edge is weighted by the corresponding squared distance along the trajectory. Since the scaling by the positive distances does not affect the rank of the matrix, it follows that the formation being BPE is a local observability condition of the subspace orthogonal to the common translation.

We now introduce the observer presented in [Z. Tang et al., 2022], which we use for localizing the drones in a bearing formation. In [Z. Tang et al., 2022], it is shown that, if the formation is BPE, the bearings are always well-defined (i.e. no collisions occur) and the velocities are bounded, then the following distributed observer achieves UE convergence

to the true state up to a common translation:

$$\dot{\hat{\boldsymbol{p}}}(t) = \boldsymbol{u}(t) - k_e \boldsymbol{L}_\beta(\boldsymbol{p}(t))\hat{\boldsymbol{p}}(t) \tag{5.6}$$

where $\hat{\boldsymbol{p}}$ is the estimated position vector and $k_e > 0$.

The convergence rate of this observer depends on the BPE parameter γ [Z. Tang et al., 2022]. The aim of the following section is to then provide a control law that aims at indirectly increasing the value of γ for ensuring a satisfactory observer performance even for non-rigid bearing formations.

5.3 Active sensing control

In this section, we propose a gradient-based control law for actively collecting information about the drone relative positions. Unlike the approach in the previous chapter, we do not incorporate a prediction horizon here. Therefore, we do not project the OG into the future; instead, we continuously integrate it over time to assimilate information gathered in the past. In fact, to quantify the amount of collected information, we adopt the minimum nonzero eigenvalue, specifically the fourth eigenvalue denoted as λ_4, of the weighted OG (5.3) but with an additional forgetting factor, that is, an OG with dynamics

$$\dot{\boldsymbol{G}}_\beta(t) = -\rho \boldsymbol{G}_\beta(t) + \boldsymbol{L}_\beta(t). \tag{5.7}$$

The forgetting factor $\rho > 0$ ensures that the OG remains bounded, since otherwise the term $\boldsymbol{L}_\beta(t) \succeq \boldsymbol{0}$, $\forall t \geq 0$ would make the OG grow unbounded over time. Notice that if $\boldsymbol{G}_\beta(t_0)$ is a matrix-weighted Laplacian [Pan et al., 2020] associated to the graph $\mathcal{G}$, then this also holds for $\boldsymbol{G}_\beta(t)$, $\forall t \geq t_0$. This can be simply shown by integrating the linear ordinary differential equation (5.7) and verifying the following facts: (i) the required definiteness pattern of the matrix weights (negative semidefinite off-diagonal terms) is preserved:

$$\begin{aligned}[\boldsymbol{G}_\beta(t)]_{(ij)} &= e^{-\rho(t-t_0)}[\boldsymbol{G}_\beta(t_0)]_{(ij)} + \int_{t_0}^{t} e^{-\rho(t-\tau)}[\boldsymbol{L}_\beta(\tau)]_{(ij)}\, d\tau \\ &= e^{-\rho(t-t_0)}[\boldsymbol{G}_\beta(t_0)]_{(ij)} - \int_{t_0}^{t} e^{-\rho(t-\tau)}\boldsymbol{\Pi}_{\beta_{ij}}(\tau)\, d\tau \preceq \boldsymbol{0}\end{aligned} \tag{5.8}$$

where $[\boldsymbol{G}_\beta(t_0)]_{(ij)} \preceq \boldsymbol{0}$ is the ij-th block of $\boldsymbol{G}_\beta(t_0)$ and $\boldsymbol{\Pi}_{\beta_{ij}} \succeq \boldsymbol{0}$; (ii) $\boldsymbol{U} \subseteq \mathcal{N}(\boldsymbol{G}_\beta(t))$:

$$\boldsymbol{G}_\beta(t)\boldsymbol{U} = e^{-\rho(t-t_0)}\boldsymbol{G}_\beta(t_0)\boldsymbol{U} + \int_{t_0}^{t} e^{-\rho(t-\tau)}\boldsymbol{L}_\beta(\tau)\boldsymbol{U}\, d\tau = \boldsymbol{0}_{3N\times 3N} \tag{5.9}$$

and; (*iii*) if $ij \notin \mathcal{E}$ then $[\boldsymbol{G}_\beta(t)]_{(ij)} = \mathbf{0}\ \forall t \geq 0$. These properties are used in the next derivations. In particular, the preservation of the sparsity pattern allows for a distributed implementation of the proposed control law.

Consider the state of the full system (i.e. including the current OG) $\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{p}^T & \text{vec}\,(\boldsymbol{G}_\beta)^T \end{bmatrix}^T$, where vec$(\cdot)$ is the vectorization operator. Then, the system dynamics can be written as:

$$\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{f}(\boldsymbol{\zeta}) + \boldsymbol{G}\,\boldsymbol{u}, \tag{5.10}$$

where the system is control affine and has a cascade structure with $\boldsymbol{f}(\boldsymbol{\zeta}) = \begin{bmatrix} \mathbf{0}_{3N}^T & \text{vec}\,(\dot{\boldsymbol{G}}_\beta)^T \end{bmatrix}^T$ and $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_{3N} & \mathbf{0}_{3N\times(3N)^2} \end{bmatrix}^T$. Note that the first derivative of λ_4 (our 'information metric') depends on the positions of the drones, so that λ_4 has *relative degree* 2 w.r.t. the drone velocities (the available control inputs). This poses a challenge for direct control since the drone positions are solely involved in the dynamics of λ_4 within the orthogonal projector $\boldsymbol{\Pi}_\beta$. However, the derivative of λ_4 can be more easily controlled. This has expression

$$\begin{aligned}\dot{\lambda}_4 &= L_{\boldsymbol{f}}\lambda_4 = (\mathbf{v}_4 \otimes \mathbf{v}_4)^T \,\text{vec}(\dot{\boldsymbol{G}}_\beta) = \mathbf{v}_4^T \dot{\boldsymbol{G}}_\beta \mathbf{v}_4 \\ &= -\sum_{(i,j)\in\mathcal{E}} (\boldsymbol{v}_{4i} - \boldsymbol{v}_{4j})^T [\dot{\boldsymbol{G}}_\beta(t)]_{(ij)} (\boldsymbol{v}_{4i} - \boldsymbol{v}_{4j}) = -\sum_{(i,j)\in\mathcal{E}} [(\boldsymbol{v}_{4i} - \boldsymbol{v}_{4j}) \otimes (\boldsymbol{v}_{4i} - \boldsymbol{v}_{4j})]^T \,\text{vec}\left([\dot{\boldsymbol{G}}_\beta(t)]_{(ij)}\right)\end{aligned} \tag{5.11}$$

where $L_{\boldsymbol{f}}\lambda_4 = \frac{\partial \lambda_4}{\partial \boldsymbol{\zeta}} \boldsymbol{f}(\boldsymbol{\zeta})$ is the Lie derivative of λ_4 along $\boldsymbol{f}$ (analogously we also define $\boldsymbol{L}_{\boldsymbol{g}_i}\lambda_4$) and $\mathbf{v}_4 = \begin{bmatrix} \boldsymbol{v}_{41} & \dots & \boldsymbol{v}_{4N} \end{bmatrix}^T$ is the eigenvector associated to λ_4. In (5.11), we used the mixed Kronecker matrix-vector product, i.e.

$$(\boldsymbol{A} \otimes \boldsymbol{B})\,\text{vec}(\boldsymbol{V}) = \text{vec}(\boldsymbol{B}\boldsymbol{V}\boldsymbol{A}^T), \tag{5.12}$$

with $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{V}$ being matrices of appropriate size and, in the 4th equality, we used the classical expression for a *Laplacian potential function* [Bullo, 2020, p.88]. For convenience, let us define $\boldsymbol{v}_{4ij} := \boldsymbol{v}_{4i} - \boldsymbol{v}_{4j}$. The second order derivative of λ_4 is then

$$\begin{aligned}\ddot{\lambda}_4 &= \sum_{i=1}^{N} L_{\boldsymbol{g}_i} L_{\boldsymbol{f}} \lambda_4 \boldsymbol{u}_i + L_{\boldsymbol{f}}^2 \lambda_4 \\ &= -\frac{1}{2}\sum_{i=1}^{N} \left(\sum_{j\in\mathcal{N}_i} [\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]^T \frac{\partial\, \text{vec}([\dot{\boldsymbol{G}}_\beta]_{ij})}{\partial \boldsymbol{p}_i} \right) \boldsymbol{u}_i \\ &\quad + \text{vec}(\dot{\boldsymbol{G}}_\beta)^T \boldsymbol{H}_\lambda \,\text{vec}(\dot{\boldsymbol{G}}_\beta) - \rho(\mathbf{v}_4 \otimes \mathbf{v}_4)^T \,\text{vec}(\dot{\boldsymbol{G}}_\beta)\end{aligned} \tag{5.13}$$

where $\boldsymbol{H}_\lambda := \frac{\partial^2 \lambda_4}{\partial\, \text{vec}(\boldsymbol{G}_\beta)\, \text{vec}(\boldsymbol{G}_\beta)^T}$ is the hessian of λ_4 with respect to the matrix entries [Magnus, 1985]. The important point to note here is that (5.13) has a direct (affine)

dependence on the control inputs $\boldsymbol{u}_i$, which can then be exploited for controlling the quantity $\dot{\lambda}_4$, for instance by applying the gradient-based control law

$$\boldsymbol{u}_i = k\left(L_{\boldsymbol{g}_i} L_{\boldsymbol{f}} \lambda_4\right)^T = -k \sum_{j \in \mathcal{N}_i} [\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]^T \frac{\partial \operatorname{vec}([\dot{\boldsymbol{G}}_\beta]_{ij})}{\partial \boldsymbol{p}_i}^T . \tag{5.14}$$

The control action (5.14) implements the sought active sensing since it aims at maximizing $\dot{\lambda}_4$ that, indirectly, maximizes the eigenvalue λ_4 itself (which is the metric we care about, but which is less directly controllable by acting on the control inputs $\boldsymbol{u}_i$).

We note that the convergence rate of the observer (5.6) is directly influenced by the behavior of the integral $\int_t^{t+T} \boldsymbol{L}_\beta(\tau) d\tau$, rather than by λ_4. However, there exists a correlation between the two. The distinction lies in the fact that in $\boldsymbol{G}_\beta$, older information is accorded less weight compared to more recent data. This weighting scheme is logical considering that system noise affects the model in practice. Consequently, older information integrated based on prior knowledge of system dynamics is deemed less reliable than newer data. For similar reasons, the incorporation of a forgetting factor is a common practice in adaptive control, particularly when dealing with slowly varying parameters [Slotine, Li, et al., 1991, p.374].

We now discuss some properties of the control law (5.14).

Proposition 1. *The gradient of $\dot{\lambda}_4$ associated to an edge e_{ij} is orthogonal to the corresponding bearing $\boldsymbol{\beta}_{ij}$, i.e.:*

$$[\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]^T \frac{\partial \operatorname{vec}([\dot{\boldsymbol{G}}_\beta]_{ij})}{\partial \boldsymbol{p}_i} \boldsymbol{\beta}_{ij} = 0 \tag{5.15}$$

Proof. First of all, notice that $[\dot{\boldsymbol{G}}_\beta]_{(ij)} = -\rho[\boldsymbol{G}_\beta]_{(ij)} - \boldsymbol{\Pi}_{\beta_{ij}}$ and the first term does not depend on $\boldsymbol{p}_i$. Furthermore, using the properties of the Kronecker product, we have $\operatorname{vec}(\boldsymbol{\beta}_{ij}\boldsymbol{\beta}_{ij}^T) = \boldsymbol{\beta}_{ij} \otimes \boldsymbol{\beta}_{ij}$. It follows that:

$$\frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ij}})}{\partial \boldsymbol{p}_i} = \frac{\partial \operatorname{vec}(\boldsymbol{I} - \boldsymbol{\beta}_{ij}\boldsymbol{\beta}_{ij}^T)}{\partial \boldsymbol{p}_i} = -\frac{\partial(\boldsymbol{\beta}_{ij} \otimes \boldsymbol{\beta}_{ij})}{\partial \boldsymbol{p}_i} = \frac{1}{d_{ij}} \left(\boldsymbol{\Pi}_{\beta_{ij}} \otimes \boldsymbol{\beta}_{ij} + \boldsymbol{\beta}_{ij} \otimes \boldsymbol{\Pi}_{\beta_{ij}}\right) \tag{5.16}$$

One can then use the Kronecker product properties, such as the the mixed product property [Bullo, 2020, p.122], and show that:

$$\begin{aligned} &\left(\boldsymbol{\Pi}_{\beta_{ij}} \otimes \boldsymbol{\beta}_{ij} + \boldsymbol{\beta}_{ij} \otimes \boldsymbol{\Pi}_{\beta_{ij}}\right) \boldsymbol{\beta}_{ij} \\ &= \left(\boldsymbol{\Pi}_{\beta_{ij}} \otimes \boldsymbol{\beta}_{ij}\right) \left(\boldsymbol{\beta}_{ij} \otimes 1\right) + \left(\boldsymbol{\beta}_{ij} \otimes \boldsymbol{\Pi}_{\beta_{ij}}\right) \left(1 \otimes \boldsymbol{\beta}_{ij}\right) \\ &= \left(\boldsymbol{\Pi}_{\beta_{ij}} \boldsymbol{\beta}_{ij} \otimes \boldsymbol{\beta}_{ij}\right) + \left(\boldsymbol{\beta}_{ij} \otimes \boldsymbol{\Pi}_{\beta_{ij}} \boldsymbol{\beta}_{ij}\right) = \boldsymbol{0}. \end{aligned} \tag{5.17}$$

□

Proposition 2. *Assuming that no collisions happen i.e.* $d_{ij} \geq d_{\min} > 0, \forall i, j \in \{1, 2, ..., N\}$, *then the resulting input* $\boldsymbol{u}_i$ *is always bounded.*

Proof. w.l.o.g. considering $k = 2$, from (5.14) and (5.16):

$$\boldsymbol{u}_i^T = \sum_{j \in \mathcal{N}_i} [\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]^T \frac{1}{d_{ij}} \left(\boldsymbol{\Pi}_{\beta_{ij}} \otimes \boldsymbol{\beta}_{ij} + \boldsymbol{\beta}_{ij} \otimes \boldsymbol{\Pi}_{\beta_{ij}} \right) \tag{5.18}$$

and

$$\begin{aligned} \|\boldsymbol{u}_i\|_2 &\leq \sum_{j \in \mathcal{N}_i} \|\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]\|_2 \frac{2}{D_{\min}} \left\|\boldsymbol{\Pi}_{\beta_{ij}} \otimes \boldsymbol{\beta}_{ij}\right\|_2 \\ &\leq \frac{8}{D_{\min}} |\mathcal{N}_i| \end{aligned} \tag{5.19}$$

where we used the subadditivity property of norms, the facts that $\|\boldsymbol{v}_{4i}\|_2 \leq \|\mathbf{v}_4\|_2 = 1$, $\|\boldsymbol{X} \otimes \boldsymbol{Y}\|_2 = \|\boldsymbol{X}\|_2 \cdot \|\boldsymbol{Y}\|_2$, $\|\boldsymbol{\Pi}_\beta\|_2 = 1$ and $\|\boldsymbol{\beta}_{ij}\|_2 = 1$. □

Finally, the active sensing gradient control (5.14) requires each robot i to only know quantities which are locally available or communicated by neighboring robots, i.e.,

- $\boldsymbol{v}_{4i}$: the components of the eigenvector corresponding to the i-th robot itself, which can be estimated in a distributed way by suitably modifying the distributed power iteration method (see e.g. [Malli et al., 2021; Yang et al., 2010]) as done e.g. in [Zelazo et al., 2015];
- $\boldsymbol{v}_{4j} \forall j \in \mathcal{N}_i$: which can be communicated by neighboring robots;
- d_{ij}: for which an estimate $\hat{d}_{ij}$ can be computed from the estimated positions; $\hat{\boldsymbol{p}}_i$, locally available, and $\hat{\boldsymbol{p}}_j$ which can be communicated by the neighbors
- $\boldsymbol{\beta}_{ij}$ which is measured.

In the next section, we show how to embed the active sensing action (5.14) within a primary task of formation control.

5.4 Including formation control through CLFs

We establish a hierarchical framework that prioritizes position-based formation control over active sensing by using distributed QP-based Control Lyapunov Functions [Ames et al., 2019] (see Sect. 3.3).

The position-based formation control task considered is defined based on the error associated with each edge, denoted as $\boldsymbol{e}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i - \boldsymbol{p}_{ij}^d$. Here, $\boldsymbol{p}_{ij}^d$ represents the

desired relative position between robot i and j. We also define the desired position vector as $\boldsymbol{p}^d = \begin{bmatrix}(\boldsymbol{p}_1^d)^T & ... & (\boldsymbol{p}_N^d)^T\end{bmatrix}^T$, which relates to the relative desired position vector $\boldsymbol{p}_{\mathcal{E}}^d = \text{stack}(\boldsymbol{p}_{ij}^d)$ through $\boldsymbol{p}_{\mathcal{E}}^d = \boldsymbol{E}_3^T \boldsymbol{p}^d$. It is worth noting that $\boldsymbol{p}^d$ is defined up to a common translation. Then, the position error is $\boldsymbol{e} = \boldsymbol{p} - \boldsymbol{p}^d = \begin{bmatrix}\boldsymbol{e}_1^T & ... & \boldsymbol{e}_N^T\end{bmatrix}^T$ with dynamics $\dot{\boldsymbol{e}} = \boldsymbol{u} - \dot{\boldsymbol{p}}^d$.

We consider the following Laplacian potential function:

$$V(\boldsymbol{e}) = \boldsymbol{e}^T \boldsymbol{L}_3 \boldsymbol{e} = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2^2 . \tag{5.20}$$

We emphasize that, while this function is only positive semidefinite i.e. $V(\boldsymbol{w}) = 0$ for any vector $\boldsymbol{w} \in \text{span}(\boldsymbol{U})$ representing a common translation, it is positive definite with respect to the desired equilibrium set $\{\boldsymbol{e}_1 = \boldsymbol{e}_2 = ... = \boldsymbol{e}_N\}$. For more flexibility in the convergence requirements, one can consider as potential the following function

$$V(\boldsymbol{e}) = \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \alpha_{ij}(V_{ij}) = \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2). \tag{5.21}$$

with α_{ij} being class $\mathcal{K}$ functions.

Let us define $\boldsymbol{c}_i(\boldsymbol{\zeta}) := (L_{\boldsymbol{g}_i} L_{\boldsymbol{f}} \lambda_4)^T / \|L_{\boldsymbol{g}_i} L_{\boldsymbol{f}} \lambda_4\|_2$ if $\|L_{\boldsymbol{g}_i} L_{\boldsymbol{f}} \lambda_4\|_2 \neq 0$ and $\boldsymbol{c}_i(\boldsymbol{\zeta}) = \boldsymbol{0}$ otherwise (vector $\boldsymbol{c}_i(\boldsymbol{\zeta})$ is thus the unit-norm direction of the active sensing control (5.14)). The centralized QP, including the active sensing task, is formulated as:

$$\begin{aligned} &\min_{\boldsymbol{u}_i, i=1,...,N} \; \frac{1}{2} \sum_{i=1}^{N} \left[(\boldsymbol{c}_i(\boldsymbol{\zeta})^T (\boldsymbol{u}_i - k\boldsymbol{c}_i(\boldsymbol{\zeta})))^2 + \eta \|\boldsymbol{\Pi}_{c_i} \boldsymbol{u}_i\|_2^2 \right] \\ &\text{s.t.} \quad \sum_{i=1}^{N} \left[\left(\sum_{j\in\mathcal{N}_i} \frac{\partial \alpha_{ij}(V_{ij})}{\partial V_{ij}} \frac{(\boldsymbol{e}_i - \boldsymbol{e}_j)}{V_{ij}} \right)^T \boldsymbol{u}_i + \sum_{j\in\mathcal{N}_i} \alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2) \right] \leq 0 \end{aligned} \tag{5.22}$$

where $\boldsymbol{\Pi}_{c_i} = \boldsymbol{I}_3 - \boldsymbol{c}_i \boldsymbol{c}_i^T$ is the orthogonal projector onto the plane perpendicular to $\boldsymbol{c}_i$ and $\eta > 0$. For future convenience we also define $\bar{\boldsymbol{e}}_{ij} := \frac{\partial \alpha_{ij}(V_{ij})}{\partial V_{ij}} \frac{(\boldsymbol{e}_i - \boldsymbol{e}_j)}{V_{ij}}$.

The first term in the cost function aims at achieving the same information gain as the one obtained by $\boldsymbol{u}_i = k\boldsymbol{c}_i(\boldsymbol{\zeta})$ and, therefore, it represents the active sensing task. The second (regularization) term in the cost function is meant to address two issues, namely (i) avoiding excessive inputs in the direction orthogonal to the active sensing task due to constraint satisfaction, and (ii) obtaining a strongly convex cost function. Indeed, the Hessian of the cost function in (5.22) is given by $\boldsymbol{H}_{QP} := \text{diag}(\boldsymbol{H}_{QP,i}) = \text{diag}(\boldsymbol{c}_i \boldsymbol{c}_i^T + \eta \boldsymbol{\Pi}_{c_i}) \succ \boldsymbol{0}$. Furthermore, to avoid undesirable oscillatory behaviours close to convergence due to

active sensing, one can define the gain k to be given by the product of two terms $k = k_c k_\sigma(\sum_{j\in\mathcal{N}_i}\alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2))$: a constant term k_c and, given a sigmoid function $\sigma(\cdot)$ of the error, $k_\sigma = \sigma(\sum_{j\in\mathcal{N}_i}\alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2))$ so that the active sensing action goes to zero when the task error is approaching zero.

Our objective now is to transform this problem into one that is suitable for distributed implementation. Inspired by [Tan & Dimarogonas, 2022] (see also Sect. 3.5.2), we compute the analytical expression for the solution of the QP. First, we define $\boldsymbol{a} = \begin{bmatrix}\boldsymbol{a}_1^T & \dots & \boldsymbol{a}_N^T\end{bmatrix}^T$, with $\boldsymbol{a}_i := \sum_{j\in\mathcal{N}_i}\bar{\boldsymbol{e}}_{ij}$, and $b = \sum_{i=1}^N b_i$, with $b_i := \sum_{j\in\mathcal{N}_i}\alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2^2)$, so that the constraint can be written as $\sum_{i=1}^N(\boldsymbol{a}_i^T\boldsymbol{u}_i + b_i) \leq 0$. We point out that, in absence of input limits, the constraint is always feasible, i.e. $\boldsymbol{a} = \boldsymbol{0}$ implies $b = 0$, in which case the constraint is trivially satisfied. The analytical solution to the QP can be obtained by using the Karush–Kuhn–Tucker (KKT) optimality conditions. Let us define the Lagrangian of the problem

$$\mathcal{L} := \frac{1}{2}\sum_{i=1}^N\left[\left(\boldsymbol{c}_i(\boldsymbol{\zeta})^T(\boldsymbol{u}_i - k\boldsymbol{c}_i(\boldsymbol{\zeta})\right)^2 + \eta\,\|\boldsymbol{\Pi}_{c_i}\boldsymbol{u}_i\|_2^2\right] + \lambda_{\mathcal{L}}(\boldsymbol{a}^T\boldsymbol{u} + b), \tag{5.23}$$

with $\lambda_{\mathcal{L}}$ being the Lagrange multiplier. The resulting KKT conditions, using * to indicate the optimal solution, are:

$$\begin{cases}\frac{\partial\mathcal{L}}{\partial\boldsymbol{u}_i} = (\boldsymbol{u}_i^* - k\boldsymbol{c}_i)^T\boldsymbol{c}_i\boldsymbol{c}_i^T + \eta(\boldsymbol{u}_i^*)^T\boldsymbol{\Pi}_{c_i} + \lambda_{\mathcal{L}}^*\boldsymbol{a}_i^T = \boldsymbol{0} \quad \forall i \\ \sum_{i=1}^N(\boldsymbol{a}_i^T\boldsymbol{u}_i^* + b_i) \leq 0 \\ \lambda_{\mathcal{L}}^* \geq 0 \\ \lambda_{\mathcal{L}}^* = 0 \quad \text{if} \quad \sum_{i=1}^N(\boldsymbol{a}_i^T\boldsymbol{u}_i^* + b_i) < 0\end{cases}. \tag{5.24}$$

From the first equation one obtains:

$$\boldsymbol{u}_i^* = \boldsymbol{H}_{QP,i}^{-1}(k\boldsymbol{c}_i - \lambda_{\mathcal{L}}^*\boldsymbol{a}_i) \tag{5.25}$$

from which two cases are possible: 1) the constraint is not active at the unconstrained solution $k\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{c}_i$, hence from the last condition one has $\lambda_{\mathcal{L}}^* = 0$ and $\boldsymbol{u}_i^* = k\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{c}_i$; 2) the constraint is active, hence substituting (5.25) into the constraint equation with equality yields

$$\lambda_{\mathcal{L}}^* = \frac{\sum_{i=1}^N(\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}k\boldsymbol{c}_i + b_i)}{\sum_{i=1}^N\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{a}_i}. \tag{5.26}$$

Substituting back in (5.25), one obtains the complete solution for the two cases as:

$$\boldsymbol{u}_i = k\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{c}_i - \max\left(0, \frac{\sum_{i=1}^N(k\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{c}_i + b_i)}{\sum_{i=1}^N\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{a}_i}\right)\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{a}_i. \tag{5.27}$$

Since $\boldsymbol{H}_{QP,i} \succ 0$, a zero denominator implies $\boldsymbol{a}_i = \boldsymbol{0}\ \forall i$, that is, accomplishment of the formation control task. Hence, in this case the input can be set to zero. Otherwise, the first

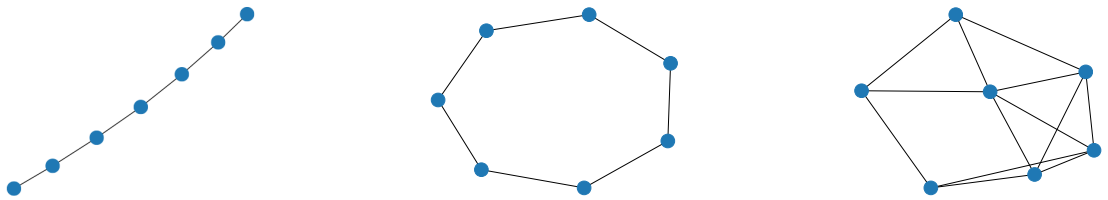
term represents the solution to the unconstrained problem. The term in the numerator $\sum_{i=1}^{N}(k\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{c}_i + b_i)$ is negative when the constraint is satisfied by the solution to the unconstrained problem. Since the term at denominator is always greater than zero, then the second term is different from zero only when the constraint is active and it acts to correct the input in order to satisfy the constraint.

The two terms $\sum_{i=1}^{N}(k\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{c}_i + b_i)$ and $\sum_{i=1}^{N}\boldsymbol{a}_i^T\boldsymbol{H}_{QP,i}^{-1}\boldsymbol{a}_i$ are not locally available but one can estimate their average through dynamic consensus, e.g. [Freeman et al., 2006]. Finally notice that the terms $\boldsymbol{a}_i$ and b_i depend on the *estimated positions* of robot i and its neighbors, which must then be estimated as accurately as possible to correctly achieve the formation task. This is achieved by the gradient-based active sensing strategy embedded in the cost function of (5.22).

5.5 Simulation results

In this section, we show the effectiveness of our approach through extensive numerical simulations. We compare the results obtained applying the proposed control law (5.27), which achieves the desired formation while performing active sensing against a control law which only implements formation control by satisfying the constraint in (5.22) while minimizing the input norm. For convenience, we refer to the two methods, respectively, as AS and CLF-only. We consider a group of $N = 7$ drones, which can sense the neighboring robots without any sensing constraints and three different graph topologies with different levels of connectivity. We performed a set of 50 simulations starting from random initial positions for each of the following graph topologies (ordered in increasing connectivity level): line graph, cycle graph and 1-redundantly bearing rigid (RBR) graph [Trinh et al., 2019] (Fig. 5.1). Notice that the first two graphs are not bearing rigid. The drones initial estimated position is drawn from a Gaussian distribution centered around the real position with standard deviation $0.8m^2$ along each axis. As desired formation, we chose to have the drones equally spaced along a circle lying on a plane parallel to the x-y plane.

The decrease of the CLF imposed in the constraint is the same for CLF-only and AS, hence, the converge speed of the formation is not of particular interest. We instead focus on the (more relevant) evaluation of the steady state formation error at convergence, and of the estimation error. In the simulations, we used the following parameters: the observer gain is $k_e = 0.1$, the forgetting factor is $\rho = 0.04$, the OG is initialized as a matrix weighted Laplacian with matrix weights $0.5\boldsymbol{I}_3$, the active sensing gain is $k = 0.2$,



(a) Line graph (b) Cycle graph (c) 1-RBR graph

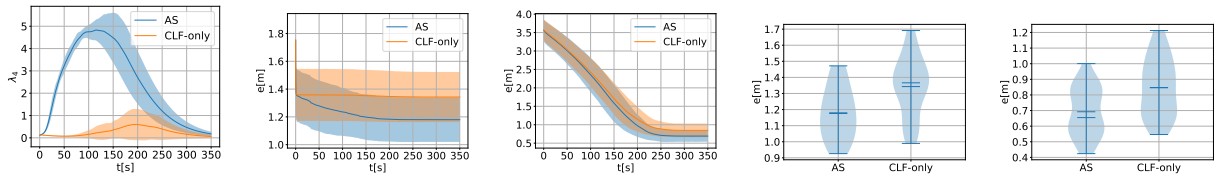
Figure 5.1: Graphs corresponding to the sets of simulations.

the regularization term is $\eta = 0.01$. We also implemented a control barrier function, which filters the inputs provided by the proposed controller to ensure collision-free trajectories [Wang et al., 2017].

We present the results for the line graph in Fig.5.2, the cycle graph in Fig.5.3, and the 1-redundant bearing rigid graph in Fig. 5.4. In each case, the figures are arranged from left to right to display: (a) the time evolution of $\lambda_4(t)$, (b) the estimation error, (c) the formation error, (d) and (e) the violin plot respectively for the estimation and formation error at the end of the trajectory. As a reminder, $\lambda_4(t)$ is the minimum non-zero eigenvalue of the OG and it is a measure of the collected information which affects the *worst case convergence* rate of the employed localization observer. The plots (a), (b) and (c) show the average trajectories along with the standard deviation of the results across simulations.

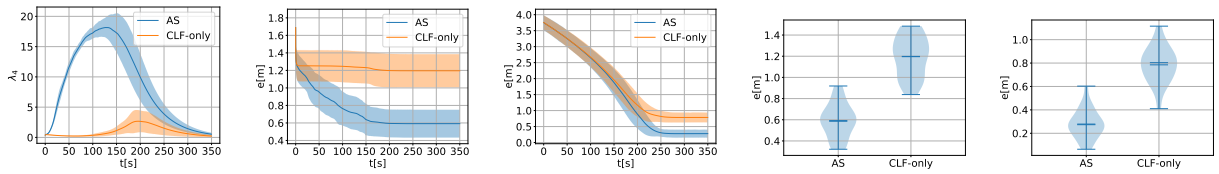
The results clearly show that the use of active sensing leads to higher acquired information and, thus, a more accurate localization and, as a consequence, a better performance for the formation control. The use of active sensing provides benefits, particularly in scenarios where the connectivity is lower, as for the line and cycle graphs. In these cases, the active sensing task reduces both the estimation and formation errors. Conversely, in highly connected and rigid graphs like the 1-redundantly rigid graph, the improvement margin is quite small. Indeed, in this case, rigidity of the graph greatly simplifies the localization task since the robot group becomes "instantaneously localizable" (up to a scale factor). This is also evident from Fig.5.4(c), where the information growth is considerably high even when active sensing is not used (CLF-only case).

An example of the trajectories followed using active sensing can be seen at the following link.



(a) Trajectory of λ_4 (b) Estimation error norm (c) Formation error norm (d) Violin plot of the estimation error norm at the final time (e) Violin plot of the formation error norm at the final time

Figure 5.2: Line graph



(a) Trajectory of λ_4 (b) Estimation error norm (c) Formation error norm (d) Violin plot of the estimation error norm at the final time (e) Violin plot of the formation error norm at the final time

Figure 5.3: Cycle graph

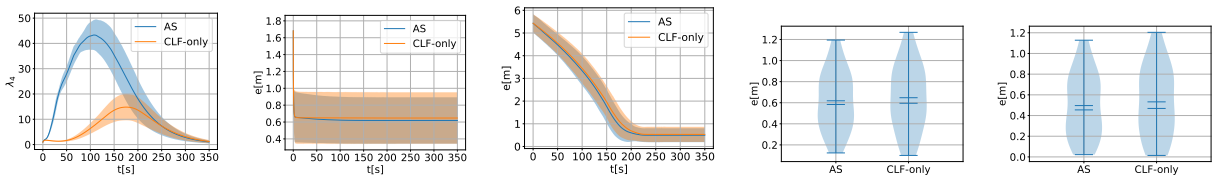
5.6 Extension to $\mathbb{R}^3 \times \mathbb{S}^1$

We now shift our focus to the scenario where the robots model is governed by (4.1) in $\mathbb{R}^3 \times \mathbb{S}^1$. In this case, each drone is able to measure the *body-frame* relative bearing with respect to its neighbors, expressed as:

$$\boldsymbol{\beta}_{ij} := \boldsymbol{R}_i^T \frac{\boldsymbol{p}_{ij}}{d_{ij}} \in \mathbb{S}^2 \tag{5.28}$$

with $\boldsymbol{R}_i = \boldsymbol{R}_z(\psi_i)$.

With the inclusion of sensing constraints in the $\mathbb{R}^3 \times \mathbb{S}^1$ representation, each robot i is assumed capable of measuring the relative bearing with respect to another robot j $\boldsymbol{\beta}_{ij}^n$ when $d_{\min} \leq d_{ij} \leq d_{\max}$, with $d_{\min}$ and $d_{\max}$ being minimum and maximum sensing range **and** the j-th robot is in the FoV of the i-th robot. This latter constraint is modeled as the condition $c_{ij} > c_{\min}$, where $c_{ij} = \boldsymbol{e}_1^T \boldsymbol{\beta}_{ij}$ is the cosine among the i-th robot x body axis (which the camera is assumed to be aligned with) and the bearing (see Fig. 4.1), and $c_{\min}$ is the minimum cosine allowed by the limited camera FoV (corresponding to the maximum angle $\alpha_{\max}$).



(a) Trajectory of λ_4

(b) Estimation error norm

(c) Formation error norm

(d) Violin plot of the estimation error norm at the final time

(e) Violin plot of the formation error norm at the final time

Figure 5.4: 1-Redundantly Rigid Graph.

Given the incorporation of sensing constraints, the induced sensing graph becomes state dependent. Moreover, since FoV sensing constraints are also accounted for, the induced sensing graph becomes directed, while the communication graph corresponds to its undirected counterpart.

5.7 Active sensing control in $\mathbb{R}^3 \times \mathbb{S}^1$

In this section, we adapt the previously formulated gradient-based active sensing control law (5.14) to accommodate body-frame bearing measurements and sensing limitations. Similar to the preceding chapter, we concentrate on the position-based partition of the OG, excluding the partitions involving yaw. Specifically, we define the following dynamics for the OG:

$$\dot{\boldsymbol{G}}_\beta(t) = -\rho \boldsymbol{G}_\beta(t) + \boldsymbol{L}_{w\beta}(t), \tag{5.29}$$

where

$$\boldsymbol{L}_{w\beta} := \boldsymbol{E}_3 \operatorname{diag}(w_{ij} \boldsymbol{R}_i \boldsymbol{\Pi}_{\beta_{ij}} \boldsymbol{R}_i^T) \boldsymbol{E}_3^T \tag{5.30}$$

and w_{ij} represents a scalar differentiable perception weight associated to each edge accounting for sensing limitations as detailed in Sect. 2.4 and it is defined as (2.38). It is worth noting that in the ideal case, $\boldsymbol{R}_i \boldsymbol{\Pi}_{\beta_{ij}} \boldsymbol{R}_i^T = \boldsymbol{\Pi}_{\bar{\beta}_{ij}} = \boldsymbol{\Pi}_{\bar{\beta}_{ji}}$. However, since the rotation matrix $\boldsymbol{R}_i$ is unknown, we substitute it with an estimate of it $\hat{\boldsymbol{R}}_i$. To simplify the notation used in the following, we will refer to $\hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ij}} = \hat{\boldsymbol{R}}_i \boldsymbol{\Pi}_{\beta_{ij}} \hat{\boldsymbol{R}}_i^T$ and $\hat{\bar{\boldsymbol{\beta}}}_{ij} = \hat{\boldsymbol{R}}_i \boldsymbol{\beta}_{ij}$.

The rest of the discussion in Sect. 5.3 remains unaltered, except for the computation of the gradients of the weights and the inclusion of the same gradient law to control the

yaw. The resulting input is given by:

$$\begin{cases} \boldsymbol{v}_i^T = \sum_{j\in\mathcal{N}_i} [\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]^T \left[\frac{w_{ij}}{d_{ij}} \left(\hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ij}} \otimes \hat{\bar{\boldsymbol{\beta}}}_{ij} + \hat{\bar{\boldsymbol{\beta}}}_{ij} \otimes \hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ij}} \right) + \mathrm{vec}(\hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ij}}) \frac{\partial w_{ij}}{\partial \boldsymbol{p}_i} \right. \\ \quad \left. + \frac{w_{ji}}{d_{ij}} \left(\hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ji}} \otimes \hat{\bar{\boldsymbol{\beta}}}_{ji} + \hat{\bar{\boldsymbol{\beta}}}_{ji} \otimes \hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ji}} \right) + \mathrm{vec}(\hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ji}}) \frac{\partial w_{ji}}{\partial \boldsymbol{p}_i} \right] \hat{\boldsymbol{R}}_i \\ w_i = \sum_{j\in\mathcal{N}_i} [\boldsymbol{v}_{4ij} \otimes \boldsymbol{v}_{4ij}]^T \left[\mathrm{vec}(\hat{\boldsymbol{\Pi}}_{\bar{\beta}_{ij}}) \frac{\partial w_{ij}}{\partial \psi_i} + \mathrm{vec}(\hat{\boldsymbol{R}}_i (\boldsymbol{S}_{e_3} \boldsymbol{\Pi}_{\beta_{ij}} + \boldsymbol{\Pi}_{\beta_{ij}} \boldsymbol{S}_{e_3}^T) \hat{\boldsymbol{R}}_i^T) \right] \end{cases} . \quad (5.31)$$

The active sensing gradient control (5.31) requires each robot i to know only quantities which are locally available or communicated by neighboring robots, i.e.,

- $\boldsymbol{v}_{4i}$: the components of the eigenvector corresponding to the i-th robot itself, which can be estimated in a distributed way by suitably modifying the distributed power iteration method;
- $\boldsymbol{v}_{4j} \forall j \in \mathcal{N}_i$: which can be communicated by neighboring robots;
- d_{ij}: for which an estimate $\hat{d}_{ij}$ can be computed from the estimated positions; $\hat{\boldsymbol{p}}_i$, locally available, and $\hat{\boldsymbol{p}}_j$ whose estimate can be communicated by the neighbors
- ψ_j: whose estimate $\hat{\psi}_j$ can be communicated by the neighboring robot.
- $\boldsymbol{\beta}_{ij}$ which is measured.
- $\boldsymbol{\beta}_{ji}$ which can be communicated.

5.8 Including an additional task through CLFs

Here, similar to Sect. 5.4, we introduce an additional task of formation control using a CLF. This time, to reduce the communication burden we impose a local CLF constraint. Specifically, each robot must satisfy: Manca La rotation $\mathrm{G}_i^{\mathrm{ND}} \left(\sum_{j\in\mathcal{N}_i} \bar{\boldsymbol{e}}_{ij} \right)^T \boldsymbol{u}_i + \sum_{j\in\mathcal{N}_i} \alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2) \leq 0$.(5.32)By summing all terms from the preceding inequality, we ensure the original inequality is met, resulting in a more conservative yet less complex solution that does not necessitate any dynamic consensus algorithm:

$$\sum_{i=1}^{N} \left(\sum_{j\in\mathcal{N}_i} \bar{\boldsymbol{e}}_{ij} \right)^T \boldsymbol{u}_i + \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2) \leq 0 \quad (5.33)$$

For convenience, let us denote $\bar{\boldsymbol{e}}_i = \sum_{j\in\mathcal{N}_i} \bar{\boldsymbol{e}}_{ij}$. The QP which needs to be solved is

$$\begin{aligned} \min_{\boldsymbol{u}_i, i=1,\dots,N} . \quad & \frac{1}{2}\sum_{i=1}^{N}\left[\left(\boldsymbol{c}_i(\boldsymbol{\zeta})^T(\boldsymbol{u}_i - k\boldsymbol{c}_i(\boldsymbol{\zeta})\right)^2 + \eta\left\|\boldsymbol{\Pi}_{c_i}\boldsymbol{u}_i\right\|_2^2\right] \\ \text{s.t.} \quad & \left(\bar{\boldsymbol{e}}_i\right)^T\boldsymbol{u}_i + \sum_{j\in\mathcal{N}_i}\alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2) \le 0 \quad \forall i \in \{1,2,...,N\} \end{aligned} \tag{5.34}$$

which can be trivially solved in a decentralized way, with each robot solving a local problem

$$\begin{aligned} \min_{\boldsymbol{u}_i} . \quad & \frac{1}{2}\left[\left(\boldsymbol{c}_i(\boldsymbol{\zeta})^T(\boldsymbol{u}_i - k\boldsymbol{c}_i(\boldsymbol{\zeta})\right)^2 + \eta\left\|\boldsymbol{\Pi}_{c_i}\boldsymbol{u}_i\right\|_2^2\right] \\ \text{s.t.} \quad & \bar{\boldsymbol{e}}_i^T\boldsymbol{u}_i + \sum_{j\in\mathcal{N}_i}\alpha_{ij}(\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2) \le 0 \end{aligned} \tag{5.35}$$

It is important to highlight that, as in the previous chapter, although the active sensing discourage edge loss, global graph connectivity can still be compromised, especially given the higher priority of the formation control task. Therefore, we include the distributed CBF presented in Chapter 3,which minimally adjusts the desired input derived from (5.35) to maintain connectivity.

The resultant input, derived from a combination of active sensing and the formation control task, prioritizes the latter. To ensure that the connectivity maintenance CBF does not interfere with the higher priority task when not necessary, in the connectivity CBF-QP, one can adjust the weight of the deviation from the desired input based on the direction of deviation. Specifically, rather than employing the conventional unweighted least squares cost:

$$J = \sum_{i=1}^{N}\left\|\boldsymbol{u}_i - \boldsymbol{u}_i^d\right\|^2 \tag{5.36}$$

one can define the following weighted least square

$$J = \sum_{i=1}^{N}(\boldsymbol{u}_i - \boldsymbol{u}_i^d)^T(\gamma_1\bar{\boldsymbol{e}}_i\bar{\boldsymbol{e}}_i^T + \gamma_2(\boldsymbol{I} - \bar{\boldsymbol{e}}_i\bar{\boldsymbol{e}}_i^T))(\boldsymbol{u}_i - \boldsymbol{u}_i^d) \tag{5.37}$$

with $\gamma_1 \gg \gamma_2$.

It is important to note that this section is still under development, as we intend to explore additional tasks beyond formation control. Ideally, we aim to consider tasks that allow for more degrees of freedom in trajectory optimization, such as controlling a leader within the group or a subset of robots, while leaving others free to optimize their trajectories.

5.9 Conclusions

In this chapter, we have presented a novel distributed control strategy for bearing formations, designed to maximize information acquisition for cooperative localization. Our active sensing strategy is based on the minimum nonzero eigenvalue of a weighted Observability Gramian with a forgetting factor as an information measure.

This active sensing strategy can be combined with other additional tasks of interest (a position-based formation control task in our case) by making use of control Lyapunov functions. This integration enables the execution of higher-level tasks while concurrently enhancing the minimum level of "localizability" for the robot group, thereby improving the overall task execution.

Numerical simulations over graphs with different levels of connectivity demonstrated the benefits of the approach, in particular for non-rigid graphs.

Additionally, we extended the algorithm to accommodate the representation of robot states in $\mathbb{R}^3 \times \mathbb{S}^1$ while addressing sensing limitations. This extension necessitated the implementation of a connectivity maintenance algorithm to safeguard against the loss of global connectivity and the incorporation of a suitable observer. Although this extension is still under development, our plans include integrating additional tasks beyond formation control and addressing challenges such as obstacles and the consequent occlusions. Moreover, we envision conducting realistic simulations and experiments. We believe that this framework holds significant potential to enhance the flexibility of robot formations navigating environments based on camera feedback.

Chapter 6

Active Sensing for Multi-Target Tracking

This chapter considers the problem of persistently monitoring a set of moving targets using a team of aerial vehicles. Each agent in the network is assumed equipped with a down-looking camera with limited range and field of view (FoV) providing relative bearing measurements and it implements an *information consensus filter* (ICF) to estimate the state of the target(s) in a distributed way. We then propose a distributed control scheme that allows maintaining a prescribed minimum PE level so as to ensure filter convergence. At the same time, the agents in the group are also allowed to perform additional tasks of interest while maintaining a collective observability of the target(s). In order to enforce the observability constraint, we leverage two main tools: (*i*) the weighted *observability gramian* with a forgetting factor as a measure of the cumulative acquired information, and (*ii*) the use of *high order control barrier functions* (HOCBF) as a mean to maintain a minimum level of observability for the targets.

Target tracking is a classical topic in the multi-robot community in which a group of (possibly mobile) sensors needs to cooperatively track the position of a moving target. Each sensor can obtain a measurement of the target and fuse it with the measurements from the other sensors to obtain a better estimate [Battistelli & Chisci, 2014; Kamal et al., 2013; Olfati-Saber, 2009]. Mobile sensors can also optimize their position/motion so as to maximize the information collected about the target state [Jacquet et al., 2022; Morbidi & Mariottini, 2012], thus improving the localization accuracy.

In this work, differently from many previous works on this subject (e.g., [Jacquet et al., 2022; Morbidi & Mariottini, 2012]), we consider a situation in which the target localization is not necessarily the only task for the robot group. This motivates to only enforce maintenance of a *minimum level* of *persistency of excitation* for localizing the moving target(s) so that the group can exploit its redundancy for realizing additional tasks of interest.

Most previous works on this subject define the active sensing task as an optimization problem aimed at *maximizing* the collected information. A notable exception is the recent paper [Coleman et al., 2024]. Similar to our approach, they employ CBFs to enforce observability, thus aiding the convergence of the state estimate to the true state while accommodating the original control objectives. Their focus lies on single-target tracking using a single robot with distance measurements to the target. They express the observability constraint through the determinant of the Observability matrix, a quantity directly influenced by inputs, necessitating the utilization of *integral* CBFs (iCBFs).

Similarly, we only require maintenance of a *minimum level of information* so as to ensure a proper convergence of the filter used for estimating the target state. However, we opt for the Observability Gramian (OG) due to its suitability for the task, providing a measure of cumulative past information. Additionally, we extend our analysis to multi-robot systems capable of tracking multiple moving targets and acquiring relative bearing measurements rather than distances. A notable advantage of leveraging the OG is its inherent structure, which facilitates distributed computations, as we will show.

Furthermore, another relevant work is [J. Li et al., 2022], which explores active target tracking using a single drone against a malicious aerial vehicle using only bearing measurements. They employed a pseudo-linear Kalman Filter similar to the one we extend to the multi-agent scenario. Both approaches assume a constant velocity target model and optimize the follower's trajectory based on the Fisher Information Matrix (FIM).

The main contribution of this chapter is the design of a distributed control for the multi-UAV system based on HOCBFs able to guarantee a minimum level of PE (which is necessary for the ICF convergence) while also allowing the execution of other tasks of interest. The rest of the chapter is structured as follows: in Section 6.1, the considered system dynamics, multi-robot interaction and sensor model are introduced. The employed ICF is presented in Section 6.2. In Section 6.3, we define the OG dynamics, then, in Section 6.4, we describe the algorithm used to achieve the persistent monitoring task. The results presented in Section 6.6 validate the approach, and conclusions are drawn in Section 6.7.

This work was published in the "*IEEE Robotics and Automation Letters*" [Balandi et al., 2023].

6.1 Modeling

We consider a group of N drones that need to localize M possibly moving target robots. It is assumed that the drones are able to localize themselves in a common frame, whereas the targets need to be localized by the drones using relative bearing measurements from onboard sensors. For example, the targets could be a set of ground robots unable to autonomously localize themselves because of limited sensing, cluttered environment, and so forth. Each drone can only communicate with its neighbors according to a fixed, undirected and connected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. The set of neighbors of the i-th robot is denoted as usual with $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. Since the communication is bidirectional $(i,j) \in \mathcal{E} \iff (j,i) \in \mathcal{E}$. The i-th drone is modeled as a single integrator with position $\boldsymbol{p}_i \in \mathbb{R}^3$ and velocity input $\boldsymbol{u}_i \in \mathbb{R}^3$ such that

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i \quad i = 1, \dots, N. \tag{6.1}$$

In the following, we will also refer to the aggregate drones positions as $\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1^T & \dots & \boldsymbol{p}_N^T \end{bmatrix}^T$ and analogously for the inputs $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1^T & \dots & \boldsymbol{u}_N^T \end{bmatrix}^T$. The r-th target position and velocity are indicated respectively as $\boldsymbol{\xi}_r$ and $\boldsymbol{\mu}_r$, and their motion model need not be known by the drones.

Each drone is assumed equipped with a down-looking onboard camera that can acquire a relative bearing measurement

$$\boldsymbol{\beta}_{ir} = \frac{\boldsymbol{p}_{ir}}{d_{ir}} \quad \in \mathbb{S}^2, \tag{6.2}$$

with respect to the target(s) in the FoV, where $\boldsymbol{p}_{ir} = \boldsymbol{\xi}_r - \boldsymbol{p}_i$ and $d_{ir} = \|\boldsymbol{\xi}_r - \boldsymbol{p}_i\|$. The bearing is obtained by projecting the noisy image plane point $\bar{\boldsymbol{p}}_{ir} = \begin{bmatrix} \bar{x}_{ir} & \bar{y}_{ir} & 1 \end{bmatrix}^T$ onto the unit sphere. A bearing measurement $\boldsymbol{\beta}_{ir}$ with respect to the r-th target is considered available if the r-th target is within a certain range w.r.t. the i-th drone, i.e. $d_{\min} < d_{ir} < d_{\max}$, and inside the FoV of the i-th drone, i.e. $-\bar{x}_M \leq \bar{x}_{ir} \leq \bar{x}_M$ and $-\bar{y}_M \leq \bar{y}_{ir} \leq \bar{y}_M$, where $\bar{x}_M$ and $\bar{y}_M$ are the FoV limits (see Fig. 6.1).

6.2 Information consensus filter

We base our estimation strategy for the target state on the ICF originally presented in [Kamal et al., 2013], for which, contrarily to the classical Kalman Consensus Filter (KCF)

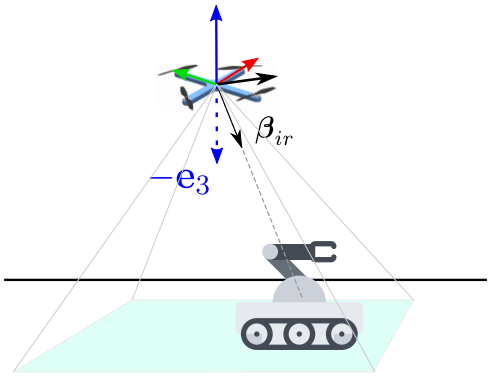


Figure 6.1: The bearing $\boldsymbol{\beta}_{ir}$ is the unit vector from the drone to the target. In the figure a camera with limited FoV is pointing in the negative z direction, represented by the unit vector $-\boldsymbol{e}_3$.

[Olfati-Saber, 2009], only collective observability of the target(s) is required [Battistelli & Chisci, 2014] (i.e., each robot is not required to individually measure the target). We will show that the a virtual measurement equation can be defined which can be linearized by output injection and that, thank to this, it is possible to obtain a uniformly globally exponentially stable observer under suitable PE conditions for the nominal system.

Since we do not assume knowledge of the target motion model, a simple constant velocity model is used for estimation purposes:

$$\boldsymbol{x}_r(k+1) = \boldsymbol{A}_d \boldsymbol{x}_r(k) + \boldsymbol{\gamma}(k) \tag{6.3}$$

where $\boldsymbol{x}_r = \begin{bmatrix} \boldsymbol{\xi}_r{}^T & \boldsymbol{\mu}_r{}^T \end{bmatrix}^T$ and $\boldsymbol{A}_d = \begin{bmatrix} \boldsymbol{I} & \Delta T \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$, with ΔT being the discretization step, while $\boldsymbol{\gamma}(k) \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ is Gaussian process noise with a positive definite covariance matrix $\boldsymbol{Q}$. The observer is not directly built on the expression (6.2) as output function but, instead, as done also in other works [Hamel & Samson, 2016], we consider an output equation linearized via output injection:

$$\boldsymbol{z}_{ir}(k) = \boldsymbol{\Pi}_{\beta_{ir}}(k) \boldsymbol{\xi}_r(k) = \boldsymbol{C}_{ir}(k) \boldsymbol{x}_r \tag{6.4}$$

where $\boldsymbol{\Pi}_{\beta_{ir}} := \boldsymbol{I}_3 - \boldsymbol{\beta}_{ir} \boldsymbol{\beta}_{ir}^T$ is the orthogonal bearing projector and $\boldsymbol{C}_{ir}(k) = \begin{bmatrix} \boldsymbol{\Pi}_{\beta_{ir}}(k) & \boldsymbol{0} \end{bmatrix}$. Also, although (6.4) is nonlinear, it only depends on a *measured* (nonlinear) function of the state (the bearing $\boldsymbol{\beta}_{ir}$).

As in [Hamel & Samson, 2016], the measurement confidence gain $\boldsymbol{R}^{-1}$ associated to each measurement is simply taken as $\boldsymbol{R}^{-1} = r^{-1} \boldsymbol{I}$, with $r^{-1} > 0$ being a tunable gain. This is possible because, from the perspective of the filter stability, this matrix is simply

Algorithm 1 - ICF at agent i relative to target r at time step k

Input: Prior state estimate $\boldsymbol{x}_{ir}^{-}(k)$, prior information matrix $\boldsymbol{W}_{ir}^{-}(k)$, observation matrix $\boldsymbol{C}_{ir}(k)$, consensus rate parameter ϵ, total consensus iterations K_c and process covariance $\boldsymbol{Q}$.

1) If $\mu_{ir}(k) = 1$, get measurement vector $\boldsymbol{z}_{ir}$ and output information matrix $\boldsymbol{R}^{-1}$

2) Compute the local information matrix $\bar{\boldsymbol{V}}_{ir}(k)$ and vector $\bar{\boldsymbol{v}}_{ir}(k)$

$$\bar{\boldsymbol{V}}_{ir}(k) \leftarrow \frac{1}{N}\boldsymbol{W}_{ir}^{-}(k) + r^{-1}\mu_{ir}(k)\begin{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{6.6}$$

$$\bar{\boldsymbol{v}}_{ir}(k) \leftarrow \frac{1}{N}\boldsymbol{W}_{ir}^{-}(k)\boldsymbol{x}_{ir}^{t-}(k) + r^{-1}\mu_{ir}(k)\begin{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k)\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{p}_i(k) \tag{6.7}$$

3) Perform K_c round of dynamic average consensus on the information matrix and vector
Let $\boldsymbol{V}_{ir}^{0}(k) = \boldsymbol{V}_{ir}^{K_c}(k-1)$ and $\boldsymbol{v}_{ir}^{0}(k) = \boldsymbol{v}_{ir}^{K_c}(k-1)$
for $\kappa = 1$ **to** K_c **do**
  a) Send $\boldsymbol{V}_{ir}^{\kappa-1}(k)$ and $\boldsymbol{v}_{ir}^{\kappa-1}(k)$ to all neighbors $j \in \mathcal{N}_i$
  b) Receive $\boldsymbol{V}_{ir}^{\kappa-1}(k)$ and $\boldsymbol{v}_{ir}^{\kappa-1}(k)$ from all neighbors $j \in \mathcal{N}_i$
  c) Update the average consensus on $\bar{\boldsymbol{V}}_{ir}(k)$ and $\bar{\boldsymbol{v}}_{ir}(k)$
end

4) Compute a posteriori state estimate and information matrix for time k

$$\boldsymbol{x}_{ir}^{+} \leftarrow (\boldsymbol{V}_{ir}^{K_c})^{\dagger}\boldsymbol{v}_{ir}^{K_c} \tag{6.8}$$

$$\boldsymbol{W}_{ir}^{+} \leftarrow N\boldsymbol{V}_{ir}^{K_c} \tag{6.9}$$

5) Predict for next time step $(k+1)$

$$\boldsymbol{W}_{ir}^{-}(k+1) \leftarrow ((\boldsymbol{A}_d^{-T}\boldsymbol{W}_{ir}^{+}(k)\boldsymbol{A}_d^{-1})^{\dagger} + \boldsymbol{Q})^{-1} \tag{6.10}$$

$$\boldsymbol{x}_{ir}^{-}(k+1) \leftarrow \boldsymbol{A}_d\boldsymbol{x}_{ir}^{+}(k) \tag{6.11}$$

Output: State estimate $\boldsymbol{x}_{ir}^{+}(k)$ and information matrix $\boldsymbol{W}_{ir}^{+}(k)$.

required to be a positive semi-definite matrix. Notice that, in this case the information associated to a single measurement is

$$\boldsymbol{C}_{ir}(k)^T\boldsymbol{R}^{-1}\boldsymbol{C}_{ir}(k) = \begin{bmatrix}r^{-1}\boldsymbol{\Pi}_{\beta_{ir}}(k) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{6.5}$$

where we used the idempotence (i.e. $\boldsymbol{\Pi}_{\beta_{ir}}(k)\boldsymbol{\Pi}_{\beta_{ir}}(k) = \boldsymbol{\Pi}_{\beta_{ir}}(k)$) and symmetry of the orthogonal projector.

The algorithm is described in Algorithm. 1. The steps of the algorithm are as follows: 1) a new measurement $\boldsymbol{z}_{ir}(k)$ is obtained; 2) in (6.6) and (6.7), the **local** information matrix $\bar{\boldsymbol{V}}_{ir}(k)$ and information vector $\bar{\boldsymbol{v}}_{ir}(k)$ are obtained from the *prior* information matrix $\boldsymbol{W}_{ir}^{-}$ and the local measurement information ; 3) K_c rounds of consensus on the

information matrix and the information vector are performed; 4) in (6.8) and (6.9), the updated state estimate $\hat{\boldsymbol{x}}_{ir}^{\tau+} = (\boldsymbol{V}_{ir}^{K_c})^{\dagger}\boldsymbol{v}_{ir}^{K_c}$, with $\dagger$ indicating the Moore-Penrose pseudo-inverse, and information matrix $\boldsymbol{W}_{ir}^{+} = N\boldsymbol{V}_{ir}^{K_c}$ are obtained and, finally; 5) in (6.11) and (6.10), the usual prediction step is performed.

The use of the linear time-varying expression (6.4) in the ICF requires some care. First, thanks to the orthogonal projector properties, one has

$$\begin{aligned} &\boldsymbol{\Pi}_{\beta_{ir}}(k)\left(\boldsymbol{\xi}_r(k) - \boldsymbol{p}_i(k)\right) = 0 \\ &\implies \boldsymbol{z}_{ir}(k) = \boldsymbol{\Pi}_{\beta_{ir}}(k)\boldsymbol{\xi}_r(k) = \boldsymbol{\Pi}_{\beta_{ir}}(k)\boldsymbol{p}_i(k). \end{aligned} \tag{6.12}$$

The update step of the original ICF [Kamal et al., 2013], by using (6.12), can then be modified for our case as

$$\begin{aligned} \boldsymbol{v}_{ir}^{0} &= \frac{1}{N}\boldsymbol{W}_{ir}^{-}(k)\hat{\boldsymbol{x}}_{ir}^{-}(k) + r^{-1}\mu_{ir}(k)\begin{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k)\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{z}_{ir}(k) \\ &= \frac{1}{N}\boldsymbol{W}_{ir}^{-}(k)\hat{\boldsymbol{x}}_{ir}^{-}(k) + r^{-1}\mu_{ir}(k)\begin{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k)\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k)\boldsymbol{p}_i(k) \\ &= \frac{1}{N}\boldsymbol{W}_{ir}^{-}(k)\hat{\boldsymbol{x}}_{ir}^{-}(k) + r^{-1}\mu_{ir}(k)\begin{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k)\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{p}_i(k) \end{aligned} \tag{6.13}$$

where $\hat{\boldsymbol{x}}_{ir}^{-}(k)$ is the prior state estimate and $\mu_{ir}(k) = 1$ if the robot i can sense the target r, otherwise $\mu_{ir}(k) = 0$. Note that the final expression in (6.13) only depends on known quantities, while $\boldsymbol{z}_{ir}(k)$ in (6.4) does not (thus showing the advantage of formulation (6.13)).

The stability of the employed filter can then be shown under the following assumptions (that will be discussed hereafter):

Assumption 2. *No collision drone-target occurs, so that the bearing measurements are always well-defined.*

Assumption 3. *The target state is collectively observable, i.e., the discrete-time Observability Gramian is full-rank*

$$\sum_{k=0}^{K}\sum_{i=1}^{N}\left(\boldsymbol{A}_d^k\right)^T\begin{bmatrix}\boldsymbol{\Pi}_{\beta_{ir}}(k) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{A}_d^k \succeq \mu_1\boldsymbol{I} \tag{6.14}$$

for some $\mu_1 > 0$, which reduces to

$$\sum_{k=0}^{K}\sum_{i=1}^{N}\boldsymbol{\Pi}_{\beta_{ir}}(k) \succeq \mu_2\boldsymbol{I} \tag{6.15}$$

for some $K, \mu_2 > 0$ (see Lemma 2.3 in [Hamel & Samson, 2016]).

Remark 7. *Assumption 3 is satisfied if there exists either a single persistently exciting direction $\boldsymbol{\beta}_{ir}$ or at least two non-collinear directions $\boldsymbol{\beta}_{ir}$ and $\boldsymbol{\beta}_{jr}$ [Hamel & Samson, 2016].*

Assumption 4. *The information matrix is initialized so that $\boldsymbol{W}_{ir}(0) \succeq \mu_3 \boldsymbol{I}$ for some $\mu_3 > 0$, $i = 1, \ldots, N$.*

Under these assumptions, since the system became linear time-varying using output-injection, the proof provided in [Battistelli & Chisci, 2014] can be used, stating that the weighted squared error vector, whose i-th component is $[\boldsymbol{L}_r(k)]_i = \boldsymbol{e}_{ir}(k)^T \boldsymbol{W}_{ir}^-(k)\boldsymbol{e}_{ir}(k)$, with $\boldsymbol{e}_{ir}(k) = \boldsymbol{x}_r(k) - \hat{\boldsymbol{x}}_{ir}^-$, converges to zero exponentially fast for the nominal (i.e. constant velocity and noise free) system.

Then since, owing to Assumptions 3 and 4, the information matrices $\boldsymbol{W}_{ir}^-$ are bounded from below and from above, i.e. $\gamma_1 \boldsymbol{I} \preceq \boldsymbol{W}_{ir}^- \preceq \gamma_2 \boldsymbol{I}$ with $\gamma_1, \gamma_2 \geq 0$, it follows that, also the estimation error converges to zero exponentially fast, and thus the observer is uniformly globally exponentially stable. We note that, in general, the real velocity of the target will not be constant but, since the unknown acceleration enters the error dynamics linearly and the observer is uniformly globally exponentially stable, the observer is *input-to-state stable* with respect to perturbations in the velocity dynamics, hence bounded accelerations will only cause bounded estimation errors (as expected and desired).

Remark 8. *The state estimation process employs a second-order model to incorporate target velocity estimation, a crucial aspect for subsequent control tasks, as elaborated upon in the following section. Possible improvements in terms of estimation performance could be obtained using a higher order model, e.g. constant acceleration. We selected the second-order model due to its broad applicability and prevalence in target tracking scenarios, particularly when detailed information about target motion is lacking. Notably, our approach does not presuppose knowledge of the exact motion model of the target. Should knowledge of the target model become available, leveraging it directly could further enhance estimation performance.*

Remark 9. *We point out that, for a generic double integrator system with unknown acceleration, the position and velocity are unobservable from a single bearing measurement. Hence, for the filter to have acceptable performance, one of the two following assumptions needs to be verified: 1) the target has a small acceleration so that the ultimate bound of the error system is small, or 2) at least two drones are measuring non collinear bearings*

w.r.t. the target, so that the position of the target is effectively measured and the target state is observable.

We finally comment about the three Assumptions 2–4: Assumption 2 can be trivially met by adding a constraint on the UAV/target minimum distance (as done in the case study of this work); Assumption 3 is a Persistency of Excitation (PE) condition that is enforced at runtime by the algorithm proposed in the following sections; Assumption 4 is only an initialization condition.

6.3 Information measures

As previously mentioned, we aim at enforcing the satisfaction of the PE condition in Assumption 3 necessary for the filter to converge in presence of sensing limitations. For this purpose, in this section, we design the dynamics of the corresponding continuous-time OG, keeping into account the sensing limitations by weighting the acquired information through suitable perception weights which decrease approaching the sensing limits, as done also in the previous chapters. Furthermore, to maintain the OG bounded and to keep into account model uncertainties, we introduce a forgetting factor.

The OG representing the information acquired about a target position until time t, indicated as $\boldsymbol{G}_{or}(t) \in \mathbb{R}^{3\times3}$, can be expressed as:

$$\boldsymbol{G}_{or}(t) = \boldsymbol{G}_{or}(t_0) + \int_{t_0}^{t} \sum_{i=1}^{N} \boldsymbol{\Pi}_{\beta_{ir}}^{T} \boldsymbol{\Pi}_{\beta_{ir}} \, d\tau = \boldsymbol{G}_{or}(t_0) + \int_{t_0}^{t} \sum_{i=1}^{N} \boldsymbol{\Pi}_{\beta_{ir}} \, d\tau \tag{6.16}$$

where we used the orthogonal projector properties. Notice that this matrix is the continuous-time analogous of the matrix appearing in (6.15). The OG is a positive semi-definite matrix and it is invertible if and only if the position of the target is observable along the trajectory. The minimum eigenvalue λ_{1r} of the OG can then be taken as an observablity metric: it quantifies how far is the target position from being unobservable [Krener & Ide, 2009].

The integrand of the OG is a positive semi-definite matrix and the OG is monotonically increasing in time. Since, in this work, we are only concerned about maintaining λ_{1r} above a minimum threshold, we introduce a forgetting factor in the OG dynamics that makes the information exponentially decaying in absence of new measurements. This also allows to take into account the uncertainty on the model. To take into account the sensing limits of the drones, we introduce weights on the information acquired at time t. The dynamics

of the weighted OG with forgetting factor can then be written as:

$$\dot{\boldsymbol{G}}_{or} = -\rho \boldsymbol{G}_{or} + r^{-1} \sum_{i=1}^{N} w_{ir} \boldsymbol{\Pi}_{\beta_{ir}} \tag{6.17}$$

where $\rho > 0$ is the forgetting factor, $w_{ir} \in [0, 1]$ is a differentiable weight function defined in (2.38), used to encode sensing constraints in the information dynamics, and, as before, r^{-1} is the information gain associated to the measurement. Note that, the OG first derivative only depends on the drone positions and not on their velocities. Hence, any function of the entries of the OG would need to be derived twice for the velocity of the drones (i.e. the input of the system) to appear, i.e. it is a function of relative degree 2.

6.3.1 Perception Awareness

As mentioned in the previous section, the drone sensing limitations are taken into account by weighting the OG. In (6.17), w_{ir} is a scalar differentiable quantity used by the i-th drone to weight the information acquired about the r-th target. The weight w_{ir} smoothly varies from 1, inside the sensing limit region, to 0, outside the sensing limit region. The information artificially decreases in case the target approaches the maximum sensing range or angle of the FoV. The weight $w_{ir} = w_{Dir} w_{\beta ir}$ are designed to be non vanishing at the sensing limits as defined in (2.38).

The weights w_{Dir} and $w_{\beta ir}$ do not (purposely) vanish to zero when approaching the sensing limits. The idea being to allow the drones temporarily losing a measurement when enough information is available. The non-zero gradient of the weights can then be exploited by the drones to possibly move back towards the targets for reacquiring information. This choice also allows drones which are not currently acquiring measurements to use the group-level knowledge about the target position, and their own weight gradient, to obtain a measurement in the future.

6.4 Persistent target monitoring

In this section, we show how to enforce the maintenance of the unobservability index over a certain threshold by using *high order control barrier functions* (HOCBF), see Sect. 3.4, in order to achieve the persistent target monitoring task. We start by formulating the general problem in a centralized form and we then discuss how it can be solved in a distributed way.

6.4.1 Centralized formulation

As previously mentioned, the goal is to enforce Assumption 3. This can be obtained by ensuring that the minimum eigenvalue λ_{1r} of the OG remains over a prescribed threshold. For this purpose, we define the safe set (see Sect. 3.4) as

$$\mathcal{C}_0^r = \{\boldsymbol{\zeta} \in \mathbb{R}^{3N+(3+9)} : h_r(\boldsymbol{\zeta}) = \lambda_{1r}(\boldsymbol{\zeta}) - \epsilon_\psi \geq 0\} \tag{6.18}$$

with $\epsilon_\psi > 0$. The state $\boldsymbol{\zeta}$ consists of the stack of the drone positions, of the target position and of the vectorized OG, i.e.,

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{p}^T & \boldsymbol{\xi}_r^T & \text{vec}\,(\boldsymbol{G}_{or})^T \end{bmatrix}^T \tag{6.19}$$

where $\text{vec}(\cdot)$ is the vectorization operator and $\boldsymbol{p}$ aggregates all the drones positions. The corresponding system dynamics are

$$\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{f}(\boldsymbol{\zeta}, t) + \boldsymbol{G}(\boldsymbol{\zeta})\boldsymbol{u} \tag{6.20}$$

with

$$\boldsymbol{f}(\boldsymbol{\zeta}, t) = \begin{bmatrix} \mathbf{0}_{3N} \\ \boldsymbol{\mu}(t) \\ \text{vec}\left(-\rho \boldsymbol{G}_{or} + b \sum_{i=1}^N w_{ir} \boldsymbol{\Pi}_{\beta_{ir}}\right) \end{bmatrix} \tag{6.21}$$

where, with a slight abuse of notation, we only indicated the direct time dependency, and

$$\boldsymbol{G}(\boldsymbol{\zeta}) = \begin{bmatrix} (\mathbf{1}_N \otimes \boldsymbol{I}_3)^T & \mathbf{0}_{3\times 3N}^T & \mathbf{0}_{9\times 3N}^T \end{bmatrix}^T. \tag{6.22}$$

As previously mentioned, a function of the entries of the OG (such as $h_r(\boldsymbol{\zeta})$ in our case) has relative degree $\varrho = 2$ w.r.t. the system input $\boldsymbol{u}$, hence the need to resort to HOCBFs. In our case,

$$\begin{aligned} \psi_{0r}(\boldsymbol{\zeta}) &= \lambda_{1r}(\boldsymbol{\zeta}) - \epsilon_\psi \\ \psi_{1r}(\boldsymbol{\zeta}) &= \frac{\partial \lambda_{1r}(\boldsymbol{\zeta})}{\partial(\text{vec}(\boldsymbol{G}_{or}))} \text{vec}(\dot{\boldsymbol{G}}_{or}) + \alpha_1^\psi(\lambda_{1r}(\boldsymbol{\zeta}) - \epsilon_\psi) \end{aligned} \tag{6.23}$$

where we used a linear extended class $\mathcal{K}$ function with $\alpha_1^\psi > 0$ and $\frac{\partial \lambda_{1r}}{\partial(\text{vec}(\boldsymbol{G}_{or}))} = \boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T$, with $\boldsymbol{v}_{1r}$ being the eigenvector of the OG associated to λ_{1r}.

The centralized QP that needs to be solved is

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathcal{U}} \quad & \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{u}^d\right\|_2^2 \\ \text{s.t.} \quad & \sum_{i=1}^N L_{Gi}\psi_{1r}(\boldsymbol{\zeta})\boldsymbol{u}_i + L_f\psi_{1r}(\boldsymbol{\zeta}) + \alpha_2^\psi \psi_{1r}(\boldsymbol{\zeta}) \geq 0 \end{aligned} \tag{6.24}$$

where, again, we chose a linear extended class $\mathcal{K}$ function with $\alpha_2^{\psi} > 0$. Also,

$$L_{Gi}\psi_{1r}(\boldsymbol{\zeta}) = r^{-1}\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T \left(\operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}}) \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} + w_{ir} \frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{p}_i} \right) \tag{6.25}$$

and

$$\begin{aligned} L_f\psi_{1r}(\boldsymbol{\zeta}) &= \left(\operatorname{vec}(\dot{\boldsymbol{G}}_{or})^T \boldsymbol{H}_{\lambda r} + (\alpha_1^{\psi} - \rho)\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T \right) \operatorname{vec}(\dot{\boldsymbol{G}}_{or}) \\ &\quad + r^{-1}\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T \sum_{i=1}^{N} \left(\operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}}) \frac{\partial w_{ir}}{\partial \boldsymbol{\xi}_r} + w_{ir} \frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{\xi}_r} \right) \boldsymbol{\mu}_r \\ &= c_r(\boldsymbol{\zeta}) + \sum_{i=1}^{N} d_{ir}(\boldsymbol{\zeta}) \end{aligned} \tag{6.26}$$

where

$$\boldsymbol{H}_{\lambda r} := \frac{\partial^2 \lambda_{1r}}{\partial \operatorname{vec}(\boldsymbol{G}_{or}) \operatorname{vec}(\boldsymbol{G}_{or})^T} = \boldsymbol{K}_3 \left(\boldsymbol{Y}_{1r}^{\dagger} \otimes \boldsymbol{v}_{1r}\boldsymbol{v}_{1r}^T + \boldsymbol{v}_{1r}\boldsymbol{v}_{1r}^T \otimes \boldsymbol{Y}_{1r}^{\dagger} \right) \tag{6.27}$$

with $\boldsymbol{Y}_{1r} := \lambda_{1r}\boldsymbol{I} - \boldsymbol{G}_r$, $\boldsymbol{K}_3$ being the commutation matrix [Magnus, 1985] and "†" being the Moore-Penrose pseudoinverse. Also, we split $L_f\psi_{1r}(\boldsymbol{\zeta})$ in the separable part $\sum_{i=1}^{N} d_{ir}(\boldsymbol{\zeta})$ and the non-separable one $c_r(\boldsymbol{\zeta})$ (the reason will be clearer later), defined as follows:

$$c_r(\boldsymbol{\zeta}) := \left(\operatorname{vec}(\dot{\boldsymbol{G}}_{or})^T \boldsymbol{H}_{\lambda r} + (\alpha_1^{\psi} - \rho)\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T \right) \operatorname{vec}(\dot{\boldsymbol{G}}_{or}) \tag{6.28}$$

and

$$d_{ir}(\boldsymbol{\zeta}) := r^{-1}\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T \left(\operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}}) \frac{\partial w_{ir}}{\partial \boldsymbol{\xi}_r} + w_{ir} \frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{\xi}_r} \right) \boldsymbol{\mu}_r \tag{6.29}$$

The QP problem in (6.24) is centralized. Our goal is to have each drone solving a local QP using only local quantities, such that the collective solution of the local QPs results in the satisfaction of the centralized constraint in (6.24). In the next subsection, we show how to solve this problem in a distributed way.

We also point out that, as in Chapter 4, we use a smooth approximation of the minimum function (see Remark 4).

6.4.2 Distributed persistent target monitoring

To guarantee satisfaction of the constraint in (6.24) first note that, from (6.26), $L_f\psi_{1r}(\boldsymbol{\zeta})$ can be split in a part local to each robot, $d_{ir}(\boldsymbol{\zeta})$, and a part that is not already separated $c_r(\boldsymbol{\zeta})$. A possible strategy to satisfy the previous constraint is, for each drone, to consider the following constraint in its local QP:

$$L_{Gi}\psi_{1r}(\boldsymbol{\zeta})\boldsymbol{u}_i + d_{ir}(\boldsymbol{\zeta}) \geq -k_i(\boldsymbol{\zeta}) \left(c_r(\boldsymbol{\zeta}) + \alpha_2^{\psi}\psi_{1r}(\boldsymbol{\zeta}) \right) \tag{6.30}$$

where the weights $k_i(\boldsymbol{\zeta})$ need to sum up to 1, i.e. $\sum k_i = 1$ (see Sect. 3.5.1). Then, taking the sum for each robot of the left hand side of the inequality in (6.30) yields

$$\begin{aligned} &\sum_{i=1}^{N} L_{Gi}\psi_{1r}(\boldsymbol{\zeta})\boldsymbol{u}_i + \sum_{i=1}^{N} d_{ir}(\boldsymbol{\zeta}) \\ &\geq -\left(\sum_{i=1}^{N} k_i(\boldsymbol{\zeta})\right)\left(c_r(\boldsymbol{\zeta}) + \alpha_2^{\psi}\psi_{1r}(\boldsymbol{\zeta})\right) = -\left(c_r(\boldsymbol{\zeta}) + \alpha_2^{\psi}\psi_{1r}(\boldsymbol{\zeta})\right) \end{aligned} \tag{6.31}$$

which satisfies the centralized constraint in (6.24). The most trivial choice for the weights is $k_i(\boldsymbol{\zeta}) = \frac{1}{N}$, which divides equally the constraint among the robots, but other choices are also possible [Lindemann & Dimarogonas, 2020]. In the local constraint in (6.30), some of the variables are not directly locally available but they can be estimated in a decentralized way:

- $\sum_{i=1}^{N} w_{ir}\boldsymbol{\Pi}_{\beta_{ir}}$: this quantity appears in $\mathrm{vec}(\dot{\boldsymbol{G}}_{or})$ (see (6.17)) and can be computed in a distributed way by dynamic average consensus [Kia et al., 2019] and multiplying the average by the number of drones;

- $\boldsymbol{G}_{or}$: Each drone has its own copy $\boldsymbol{G}_{oir}$ of the information collected about the target $\boldsymbol{G}_{or}$, which is obtained by integrating (6.17). The matrices $\boldsymbol{G}_{oir}$ starts from the same initial conditions and have the same dynamics up to the consensus error on $\sum_{i=1}^{N} w_{ir}\boldsymbol{\Pi}_{\beta_{ir}}$. In order to have consistency across the network we add a consensus term to the OG dynamics:

$$\dot{\boldsymbol{G}}_{oir} = -\rho\boldsymbol{G}_{oir} + \sum_{i=1}^{N} b w_{ir}\boldsymbol{\Pi}_{\beta_{ir}} + \sum_{j\in\mathcal{N}_i}\left(\boldsymbol{G}_{ojr} - \boldsymbol{G}_{oir}\right). \tag{6.32}$$

  Then, also the quantities λ_{1r} and $\boldsymbol{v}_{1r}$ are available to all the drones;

- $\boldsymbol{\xi}_r^+$, $\boldsymbol{\mu}_r^+$: every drone has its own posterior estimate $\hat{\boldsymbol{\xi}}_{ir}^+$ and $\hat{\boldsymbol{\mu}}_{ir}^+$ provided by the ICF.

A direct implementation of (6.30) would imply that, when $L_{Gi}\psi_{1r}(\boldsymbol{\zeta})$ approaches zero, the local QP may result infeasible. In order to solve this issue, considering the fact that the constraint is not safety-critical, we add a slack variable to soften the constraint

$$\begin{aligned} \min_{\boldsymbol{u}_i\in\mathcal{U}_i,\delta_{ir}} \quad & \frac{1}{2}\left\|\boldsymbol{u}_i - \boldsymbol{u}_i^d\right\|_2^2 + \frac{1}{2}K_\delta w_{ir}\delta_{\psi_{ir}}^2 \\ \text{s.t.} \quad & L_{Gi}\psi_{1r}(\boldsymbol{\zeta})\boldsymbol{u}_i + d_{ir}(\boldsymbol{\zeta}) + \delta_{\psi_{ir}} \geq -\frac{1}{N}\left(c_r(\boldsymbol{\zeta}) + \alpha_2^{\psi}\psi_{1r}(\boldsymbol{\zeta})\right) \\ & r = 1,\ldots,M. \end{aligned} \tag{6.33}$$

Notice that the slack variable in the cost function is weighted by the product of a high gain K_δ and the weight w_{ir}. The reason of this is two-fold: 1) to avoid numerical problems when $L_{Gi}\psi_{1r}(\boldsymbol{\zeta})$ is very small (because of a very small weight w_{ir}); 2) to relax the constraint for drones which are not very close to the target. This in practice means that if some drones are already observing the target, the drones with very small weight will ignore the constraint and only track the desired input (representative of any additional task).

For the formation to track multiple targets, we simply take the intersection of the safe sets corresponding to each target, i.e. we add a linear inequality constraint for each target and extend the state $\boldsymbol{\zeta}$ with the other targets state. Notice that, with the proposed formulation, if the collected information becomes high enough, since the constraint is satisfied at the current time, the drones could stop following a target r and just track the desired input. In fact, as in previous chapters, active sensing discourages loosing connections but it does not prevent from loosing connectivity. In this situation, the weight of each robot w.r.t. target r could become very small and it may happen that no drone would then be able to reach the target again. In order to avoid this issue, we add another CBF for ensuring that $h_{wr} := \sum_{i=1}^{N} w_{ir} - \epsilon_w \geq 0$ for $r = 1, ..., M$ and $\epsilon_w > 0$. The additional CBF constraint can be added to the local QP as in the previous case:

$$
\begin{aligned}
\min_{\substack{\boldsymbol{u}_i \in \mathcal{U}_i, \\ \delta_{\psi_{ir}}, \delta_{w_{ir}}}} \quad & \frac{1}{2}\left\|\boldsymbol{u}_i - \boldsymbol{u}_i^d\right\|_2^2 + \frac{K_\delta}{2}\sum_{r=1}^{M} w_{ir}\delta_{\psi_{ir}}^2 + \frac{K_\delta}{2}\sum_{r=1}^{M} w_{ir}\delta_{w_{ir}}^2 \\
\text{s.t.} \quad & L_{Gi}\psi_{1r}(\boldsymbol{\zeta})\boldsymbol{u}_i + d_{ir}(\boldsymbol{\zeta}) + \delta_{\psi_{ir}} \geq -\frac{1}{N}\left(c_r(\boldsymbol{\zeta}) + \alpha_2^{\psi}\psi_{1r}(\boldsymbol{\zeta})\right) \\
& \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}\boldsymbol{u}_i + \frac{\partial w_{ir}}{\partial \boldsymbol{\xi}_r}\boldsymbol{\mu}_r + \delta_{w_{ir}} \geq -\frac{\alpha_1^w}{N}\left(\sum_{i=1}^{N} w_{ir} - \epsilon_w\right) \\
& r = 1, \ldots, M
\end{aligned}
\tag{6.34}
$$

where $\frac{1}{N}\sum_{i=1}^{N} w_{ir}$ is obtained through average consensus.

Remark 10. *Depending on the design of the weights, this constraint does not necessarily imply that one of the drones is forced to continuously observe the target. Rather, it ensures that the drones remain close enough to the target so that they can exploit the gradient information in the weights for approaching and measuring again the target whenever necessary.*

6.5 Analysis of the proposed control law

In this section, we provide some analytical results regarding the proposed algorithm. To simplify the analysis, we do not consider input saturations. First, we discuss the effect of introducing the slack variables and we establish that the slack variable associated with a certain constraint can attain a value of precisely zero if and only if the optimal solution, excluding that particular constraint, remains feasible for the constraint itself. This is a classical result in optimization theory, stemming from the smoothness of the cost function when incorporating the slack variable in the cost function through its squared norm [Kerrigan & Maciejowski, 2000].

We define the constraints imposed by the CBF as follows: the weight constraint (second constraint in (6.34)) $g^w_{ir}(\boldsymbol{\zeta}, \boldsymbol{u_i}) \leq 0$ and the information constraint (first constraint in (6.34)) $g^\psi_{ir}(\boldsymbol{\zeta}, \boldsymbol{u_i}) \leq 0$:

$$g^w_{ir}(\boldsymbol{\zeta}, \boldsymbol{u_i}) := -\frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}(\boldsymbol{u}_i - \boldsymbol{\mu}_r) - \delta_{w_{ir}} - \frac{\alpha^w_1}{N}(\sum_{i=1}^{N} w_{ir} - \epsilon_w) \tag{6.35}$$

$$\begin{aligned} g^\psi_{ir}(\boldsymbol{\zeta}, \boldsymbol{u_i}) &:= -L_{gi}\psi_{1r}(\boldsymbol{\zeta})\boldsymbol{u}_i - \delta_{\psi_{ir}} - d_{ir}(\boldsymbol{\zeta}) - \frac{1}{N}(c_r(\boldsymbol{\zeta}) + \alpha^\psi_2 \psi_{1r}(\boldsymbol{\zeta})) = \\ &= -b(\boldsymbol{v}^T_{1r} \otimes \boldsymbol{v}^T_{1r}) \left(\operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}}) \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} + w_{ir} \frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{p}_i} \right) (\boldsymbol{u}_i - \boldsymbol{\mu}_r) - \delta_{\psi_{ir}} - \frac{1}{N} \operatorname{vec}(\dot{\boldsymbol{G}}_r)^T \boldsymbol{H}_{\lambda_{1r}} \operatorname{vec}(\dot{\boldsymbol{G}}_r) - \\ &- \frac{1}{N}(\alpha^\psi_1 \alpha^\psi_2 - \rho)(\boldsymbol{v}^T_{1r} \otimes \boldsymbol{v}^T_{1r}) \operatorname{vec}(\dot{\boldsymbol{G}}_r) - \frac{\alpha^\psi_1 \alpha^\psi_2}{N}(\lambda_{1r} - \epsilon_\psi) \end{aligned} \tag{6.36}$$

Hereafter, we use $\delta_{w/\psi_{ir}}$ to indicate that the equation must hold for both the slack variables of the weight and the information constraints.

We define the Lagrangian associated to the QP as:

$$\mathcal{L} = \frac{1}{2} \left\| \boldsymbol{u}_i - \boldsymbol{u}^d_i \right\|^2 + \sum_{r=1}^{M} \left[\frac{K_\delta}{2} w_{ir} \left(\delta^2_{w_{ir}} + \delta^2_{\psi_{ir}} \right) + \lambda^w_{ir} g^w_{ir} + \lambda^\psi_{ir} g^\psi_{ir} \right] \tag{6.37}$$

where λ^w_{ir} and λ^ψ_{ir} are Lagrange multipliers and, analogously to the slack variables, we may use $\lambda^{w/\psi}_{ir}$. The Karush–Kuhn–Tucker (KKT) optimality conditions associated to the

QP are the following:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_i}(\boldsymbol{u}_i^*, \delta_{w/\psi_{ir}}^*, \lambda_{ir}^{w/\psi *}) &= 0 \\
\frac{\partial \mathcal{L}}{\partial \delta_{w/\psi_{ir}}}(\boldsymbol{u}_i^*, \delta_{w/\psi_{ir}}^*, \lambda_{ir}^{w/\psi *}) &= 0 \\
g_{ir}^w(\boldsymbol{u}_i^*, \delta_{w_{ir}}^*) &\leq 0 \\
g_{ir}^\psi(\boldsymbol{u}_i^*, \delta_{\psi_{ir}}^*) &\leq 0 \\
\lambda_{ir}^{w*} = 0 \quad \text{if} \quad g_{ir}^w(\boldsymbol{u}_i^*, \delta_{w_{ir}}^*) &< 0 \\
\lambda_{ir}^{\psi*} = 0 \quad \text{if} \quad g_{ir}^\psi(\boldsymbol{u}_i^*, \delta_{\psi_{ir}}^*) &< 0 \\
\forall r \in \{1, .., M\}&
\end{aligned} \tag{6.38}$$

where we use the '$*$' symbol to indicate the value of the variable at the optimal solution of the QP.

From the first condition, and using:

$$\begin{aligned}
\frac{\partial g_{ir}^w}{\partial \boldsymbol{u}_i} &= -\frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} \\
\frac{\partial g_{ir}^\psi}{\partial \boldsymbol{u}_i} &= -b(\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T) \left(\operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}}) \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} + w_{ir} \frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{p}_i} \right)
\end{aligned} \tag{6.39}$$

we have:

$$\begin{aligned}
\boldsymbol{u}_i^* - \boldsymbol{u}_i^d &= \sum_{r=1}^{N} \left(\lambda_i^{w*} \frac{\partial g_{ir}^w}{\partial \boldsymbol{u}_i} + \lambda_i^{\psi*} \frac{\partial g_{ir}^\psi}{\partial \boldsymbol{u}_i} \right)^T = \\
&= \sum_{r=1}^{M} \left[\left(\lambda_{ir}^{w*} + \lambda_{ir}^{\psi*} b(\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T) \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}}) \right) \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} + \lambda_{ir}^{\psi*} b(\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T) w_{ir} \frac{\partial \operatorname{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{p}_i} \right]^T
\end{aligned} \tag{6.40}$$

From the second KKT condition, we have:

$$\lambda_{ir}^{w/\psi *} = K_\delta w_{ir} \delta_{w/\psi_{ir}}^*. \tag{6.41}$$

Since the chosen weights are described by Gaussian functions, they only vanish asymptotically. To simplify the analysis, let us assume that the weights vanish precisely outside the sensing range. Considering one of the constraints associated with a particular slack variable $\delta_{w/\psi_{ir}}$, from (6.41), the slack variable can only be zero if $\lambda_{ir}^{w/\psi *} = 0$. Thus, substituting $\lambda_{ir}^{w/\psi *} = 0$ into (6.40), the slack variables can all be proven zero only if the desired

input $\boldsymbol{u}^d$ satisfies all constraints. For instance, if $\lambda_{ir}^w > 0$, implying $w_{ir}^w > 0$ and $\delta_{w_{ir}} > 0$, and assuming there is only one target, then the slack variable $\delta_{\psi_{ir}}^* = 0$ holds true if and only if

$$\boldsymbol{u}_i^* = \boldsymbol{u}_i^d + \lambda_{ir}^{w*} \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}^T = \boldsymbol{u}_i^d + K_\delta w_{ir} \delta_{w_{ir}}^* \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}^T \tag{6.42}$$

satisfies the inequality $g_{ir}^\psi \leq 0$.

In summary, unless the desired input satisfies all constraints, the constraints will inevitably be slightly violated by a quantity that can be made arbitrarily small depending on the value K_δ [Kerrigan & Maciejowski, 2000]. If ensuring the slack variable being proven zero is crucial, one may consider incorporating the slack variable into the cost function in terms of the 1-norm $\|\cdot\|_1$. In practical terms, this implies adding the slack variable to the cost function as a linear function associated with a constraint on the slack variable being greater than zero. Below is an illustrative example:

$$\begin{aligned} \min_{\boldsymbol{u}_i \in \mathcal{U}_i, \delta_{ir}} \quad & \frac{1}{2} \left\| \boldsymbol{u}_i - \boldsymbol{u}_i^d \right\|_2^2 + K_\delta w_{ir} \delta_{ir} \\ \text{s.t.} \quad & L_{gi} \psi_{1r}(\boldsymbol{x}) \boldsymbol{u}_i + d_i(\boldsymbol{x}) + \delta_{ir} \geq -\frac{1}{N} \left(c(\boldsymbol{x}) + \psi_{1r}(\boldsymbol{x}) \right) \\ & \delta_{ir} \geq 0. \end{aligned} \tag{6.43}$$

In this scenario, if the constraint is feasible (and $w_{ir} \neq 0$), then the slack variables will provably be zero for an appropriate choice of the value of K_δ [Kerrigan & Maciejowski, 2000]. As a downside, we point out that, in this case the control law may not be Lipschitz continuous anymore.

We also point out that deadlocks (i.e. an equilibrium of system dynamics that causes robots to come to a standstill before reaching their goals) may occur. We consider separately different cases of interest in which there are two targets. The following analysis is far from complete, since there exist cases which are too complex either to analyze in a rigorous way or to provide interpretable conclusions.

CASE 1: The two weight constraints are active, i.e., $g_{ir}^w = 0$, while the information constraints g_{ir}^ψ are inactive.

In this case, $\lambda_{ir}^{w*} > 0$, which implies $w_{ir} > 0$ and $\delta_{w_{ir}}^* > 0$, for $r = 1, 2$. If $w_{i1} = 1$ and $0 < w_{i2} < 1$, then $\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i} = 0$, which from (6.40) and substituting $\boldsymbol{u}_i^* = \boldsymbol{0}$, a deadlock is verified if:

$$\lambda_{i2}^{w*} \frac{\partial w_{i2}}{\partial \boldsymbol{p}_i} + \boldsymbol{u}_i^d = K_\delta w_{i2} \delta_{i2}^{w*} \frac{\partial w_{i2}}{\partial \boldsymbol{p}_i} + \boldsymbol{u}_i^d = 0. \tag{6.44}$$

Here, if for example $\boldsymbol{u}_i^d = 0$, and recalling that $0 < w_{i2} < 1$, which implies $\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i} \neq \mathbf{0}$ and $\delta_{i2}^{w*} > 0$, then the deadlock cannot occur. Practically speaking, this means that the input will be used to satisfy the constraint. For $\boldsymbol{u}_i^d \neq 0$, this cannot be verified if a constraint on the desired input limits is considered and the value of K_δ is big enough. If both $0 < w_{ir} < 1$ for $r = 1, 2$, then from (6.40):

$$\boldsymbol{u}_i^* = \sum_{r=1}^{2} \lambda_{ir}^{w*} \frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}^T + \boldsymbol{u}_i^d. \tag{6.45}$$

For a deadlock we have $\boldsymbol{u}_i^* = \mathbf{0}$, then using (6.41):

$$K_\delta w_{i1} \delta_{i1}^{w*} \frac{\partial w_{i1}}{\partial \boldsymbol{p}_i}^T + \boldsymbol{u}_i^d = -K_\delta w_{i2} \delta_{i2}^{w*} \frac{\partial w_{i2}}{\partial \boldsymbol{p}_i}^T. \tag{6.46}$$

From the fact that the constraint g_{ir}^w are active, it follows:

$$\delta_{w_{ir}}^* = -\frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}(\boldsymbol{u}_i^* - \boldsymbol{\mu}_r) - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{ir} - \epsilon_w\right). \tag{6.47}$$

Plugging back in (6.46):

$$\begin{aligned} &K_\delta w_{i1}\left[\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i}(\boldsymbol{\mu}_1 - \boldsymbol{u}_i^*) - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{i1} - \epsilon_w\right)\right]\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i} + \boldsymbol{u}_i^d \\ &= -K_\delta w_{i2}\left[\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i}(\boldsymbol{\mu}_2 - \boldsymbol{u}_i) - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{i2} - \epsilon_w\right)\right]\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i} \end{aligned} \tag{6.48}$$

For example for $\boldsymbol{u}_i^d = \mathbf{0}$, from (6.46), this means:

$$w_{i1}\left[\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i}\boldsymbol{\mu}_1 - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{i1} - \epsilon_w\right)\right]\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i} = -w_{i2}\left[\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i}\boldsymbol{\mu}_2 - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{i2} - \epsilon_w\right)\right]\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i} \tag{6.49}$$

Hence, to have a deadlock one needs $\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i}$ and $\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i}$ to be colinear (as one would expect) and

$$w_{i1}\left\|\left[\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i}\boldsymbol{\mu}_1 - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{i1} - \epsilon_w\right)\right]\right\| = w_{i2}\left\|\left[\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i}\boldsymbol{\mu}_2 - \frac{\alpha_1}{N}\left(\sum_{i=1}^{N} w_{i2} - \epsilon_w\right)\right]\right\| \tag{6.50}$$

with proper sign, depending on $\frac{\partial w_{ir}}{\partial \boldsymbol{p}_i}$.

CASE 2: The two information constraints are active, while the weights constraints g_{ir}^w are inactive.

In this case, $\lambda_{ir}^{\psi*} > 0$, which implies $w_{ir} > 0$ and $\delta_{\psi_{ir}}^{*} > 0$, for $r = 1, 2$. If $w_{i1} = 1$ and $0 < w_{i2} < 1$, then $\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i} = 0$, which from (6.40) and substituting $\boldsymbol{u}_i^* = \mathbf{0}$, a deadlock is verified if:

$$\begin{aligned} &\boldsymbol{u}_i^d + b\sum_{r=1}^{2} \lambda_{ir}^{\psi*}(\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T)\left[\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{ir}})\frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} + w_{ir}\frac{\partial\,\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{p}_i}\right]^T = \\ &= \boldsymbol{u}_i^d + K_\delta b\sum_{r=1}^{2} w_{ir}\delta_{\psi_{ir}}^{*}(\boldsymbol{v}_{1r}^T \otimes \boldsymbol{v}_{1r}^T)\left[\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{ir}})\frac{\partial w_{ir}}{\partial \boldsymbol{p}_i} + w_{ir}\frac{\partial\,\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{ir}})}{\partial \boldsymbol{p}_i}\right]^T = 0 \end{aligned} \tag{6.51}$$

One may also use the fact that the constraint is active to remove the slack variable from the expression, but the expression becomes very complex. If one considers the case $\boldsymbol{u}_i^d = \mathbf{0}$, then for a deadlock to be possible it is required the existence of a α such that:

$$\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{i1}})\frac{\partial w_{i1}}{\partial \boldsymbol{p}_i} + w_{i1}\frac{\partial\,\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{i1}})}{\partial \boldsymbol{p}_i} = \alpha\left[\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{i2}})\frac{\partial w_{i2}}{\partial \boldsymbol{p}_i} + w_{i2}\frac{\partial\,\mathrm{vec}(\boldsymbol{\Pi}_{\beta_{i2}})}{\partial \boldsymbol{p}_i}\right]. \tag{6.52}$$

Analogously, one may consider the case of mixed information-weight active constraints.

6.6 Simulation results

In this section we validate the proposed approach via a series of simulations. We consider N drones localized in a common frame (e.g., using GPS or by running a distributed localization algorithm such as [Luft et al., 2018]). They have to estimate the positions of M ground robots, which are unable to localize themselves, because of their limited sensing capabilities and/or moving in a GPS-denied environment. In this case, the drones can act as a mobile localization system for the ground robots and they can exploit the group redundancy for optimizing their motion and achieving any other task of interest (e.g. formation control). The drones communicate with each other according to a fixed, undirected and connected communication graph, as explained in Section 6.1. The simulations are implemented in Python and the solver used for the QP is OSQP [Stellato et al., 2020]. We show three representative scenarios: 1) a single target moving with non-constant velocity, 2) the same target motion but with the drones having a secondary task of formation control, 3) a multi-target scenario, with targets moving at non-constant velocity. For each case, we perform 20 simulations, with the initial drone positions sampled from a uniform distribution near the origin. Knowledge of the fact that the targets are moving on the xy plane is never used by the algorithm (which indeed would also work for a generic 3D

motion). For the problem to be feasible, the maximum velocity of the targets is smaller than the maximum velocity of the drones. Each drone runs an ICF for each target and solves the QP (6.34). For each case, we report the plots of average results for the CBFs ψ_{0r}, ψ_{1r}, h_{wr} and of the norm of the position estimation error $\|\boldsymbol{e}_p(t)\|_2$ across all drones and all simulations. We also plot the minimum and maximum values at each time instant showing a shadowed area between the two. Note that if the CBFs remain non-negative then the corresponding constraints are satisfied. A red dashed line puts in evidence the minimum threshold at zero. We check that at $t = 0$ each target is visible by at least 2 drones. The drones run a collision avoidance algorithm using CBFs as in [Wang et al., 2017], but as the topology of the communication graph is fixed, it is not implemented in a distributed way. The minimum inter-agent distance is set to 1.0 m, and the velocities of the drones satisfy $\|\boldsymbol{u}_i\| \in [-3, 3]$ m/s.
Other parameters common to all the simulations are the following: forgetting factor $\rho = 0.7$, threshold $\epsilon_\psi = 0.1$, threshold $\epsilon_w = 1$, slack variable weight $K_\delta = 10^5$, number of consensus iterations per step $K_c = 1$, information gain $r^{-1} = 2$. For all targets, we initialize the estimated position $\hat{\boldsymbol{\xi}}_{ir}^-$ randomly and the velocity $\hat{\boldsymbol{\mu}}_{ir}^-$ to zero, $\boldsymbol{W}_{ir}^-$ to $\boldsymbol{I}_3$ and $\boldsymbol{G}_r$ to $\epsilon_\psi \boldsymbol{I}_3$. The system covariance $\boldsymbol{Q}$ is $\text{diag}(0.01 \cdot \mathbf{1}_3, \mathbf{1}_3)$. The measurements acquired by the drones are affected by Gaussian noise acting on the image plane with zero mean and covariance $\boldsymbol{R} = 5 \cdot 10^{-5} \boldsymbol{I}_2$. The reader can find attached to the paper a video with representative simulations.

6.6.1 Case 1: Single target, no additional tasks

In this case, we consider $N = 6$ and $M = 1$. The drones do not have an additional task besides the target estimation one (namely $\boldsymbol{u}_i^d = \mathbf{0}$ $\forall i \in \{1, \dots, 6\}$). The trajectory of the target, starting from the origin, is the eight-shape (Figs. 6.2 and 6.3e) defined by $\boldsymbol{\xi}_1(t) = [A \sin(\omega t), A \sin(\omega t) \cos(\omega t), 0]^T$, with $\omega = 0.12$ rad/s, $A = 10$ m. Notice that the velocity is far from being constant: hence, two non-colinear bearings are necessary as per Remarks 7 and 9. In each of the 20 realizations, the drones initial positions are generated by uniformly sampling a box of sizes $8 \times 8 \times 1$ m centered at $[0, 0, 2]^T$ m.

The results are reported in Fig. 6.3. Figures 6.3a, 6.3b and 6.3c depict the mean CBFs. In these plots, three peaks are clearly visible: the first is due to initial conditions, since the target starts at the origin where it can be sensed by the majority of the drones with high weights w_{i1} (see Fig.6.3c, which shows the sum of the weights with offset given by ϵ_w), and hence the information rapidly grows. The second and third peaks refer, respectively,

to the target going back from the origin after half a period, where it is visible again by the drones which did not follow it, and the target completing a period of its motion and passing again through the origin. In between the peaks, the constraints are active and few drones follow the target. The actual number of moving drones depends also on ϵ_w, which can be tuned depending on the application. For example, in this case it can be used to impose that at least two drones follow the target by choosing $\epsilon_w > 1$. The trajectories of drones and target at the final instant of one of the 20 simulations are depicted in Fig. 6.3e. As one can see, the CBFs have non-negative values, although some slight violations may happen because of estimation noise and the use of slack variables. Fig. 6.3d depicts the average of the norm of the position error with maximum and minimum values for each time instant. In correspondence of the peaks in the CBFs plots, one can observe a slight decrease of the error, because the target is being observed by more drones.

6.6.2 Case 2: single target, formation control task

In this case, we consider the same scenario as before, but adding as secondary task a bearing-only formation control [Zhao & Zelazo, 2015], which does not constrain scale and barycenter of the formation and which is used to provide the desired inputs $\boldsymbol{u}_i^d$. The results are reported in Fig. 6.4, while Fig. 6.2 depicts the trajectories of a simulation of case 2 at the final instant. From Figs. 6.4a, 6.4b and 6.4c, we can draw the same considerations as in the previous case, with the CBFs having in general higher values. This is because the drones observing the target tend to steer the formation closer to the target. This is confirmed also by Fig. 6.4d, where we can see that the estimation error remains lower than the previous case. When the target moves away from the origin, 2 or 3 drones follow it and the rest of the group implements the input given by the formation control task. In Fig. 6.4e, we also show the average of the norm of the bearing errors related to the formation task, with maximum and minimum values for each time t. The bearing error initially rapidly decreases, and it then remains limited, although not zero on average. The reason being that the formation control provides a *desired* input which is then filtered by the CBFs. This is also due to the suboptimality of the distributed CBFs here implemented w.r.t. the centralized QP.

6.6.3 Case 3: multi-target, no additional tasks

In this case, we consider $N = 10$ and $M = 3$ with $\boldsymbol{u}_i^d = \mathbf{0}\ \forall i \in \{1, \dots, 10\}$. The

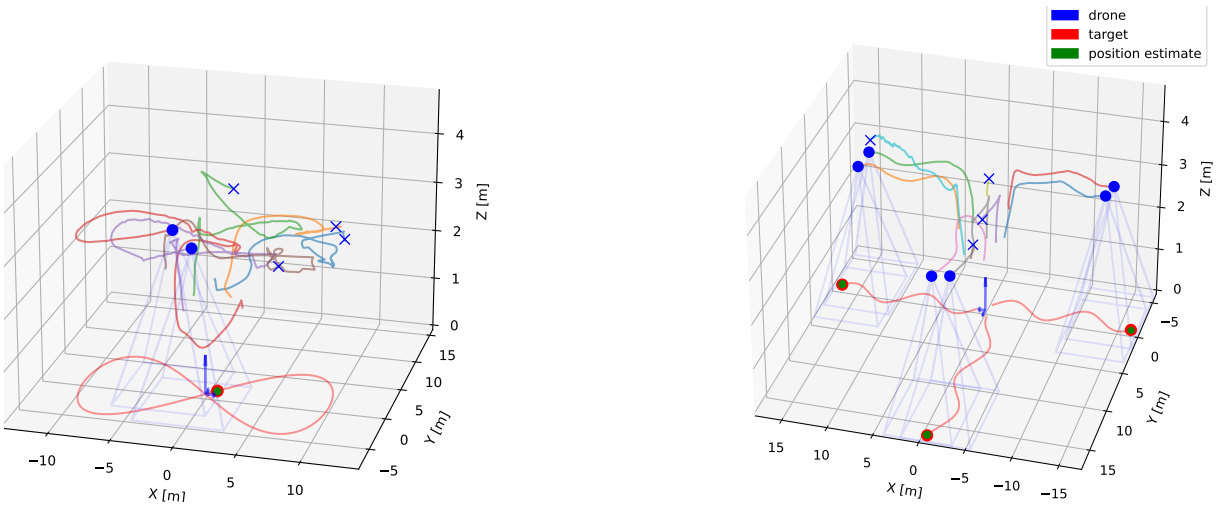
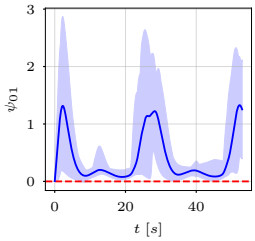


Figure 6.2: Trajectories at the final instant of simulation examples of case 2 (left) and case 3 (right). The blue 'X' represent drones which are not sensing any target, the blue thin lines represent the FoVs.

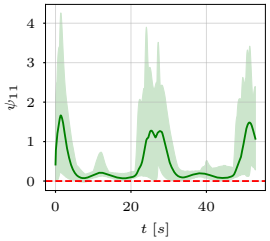
target initial positions are $\boldsymbol{\xi}_1(0) = [1, 0, 0]^T m$, $\boldsymbol{\xi}_2(0) = [-0.5, 0.866, 0]^T m$ and $\boldsymbol{\xi}_3(0) = [-0.5, -0.866, 0]^T m$, and they move with sinusoidal motion in the $+x$, $+y$, and $-x$ directions respectively (see Fig.6.2). Hence, again, they do not satisfy the constant velocity condition. The results are reported in Fig. 6.5, where in each subfigure the top plot refers to target 1, the middle plot to target 2 and the bottom plot to target 3. Initially, the targets are seen by the majority of the drones, hence the minimum eigenvalue of the OG rapidly increases and the weights related to the targets are high, as it can be seen from Figures 6.5a, 6.5b and 6.5c. After these initial peaks, the CBFs decrease approaching zero without crossing it (slight exceptions are possible again due to estimation noise and the use of slack variables). This phase corresponds to the situation in which the drones have to "split" and move from their original positions to follow different targets. Also, in this case the minimum number of drones required to estimate the state of a target respecting the constraints is 2. The final time of a simulation is depicted in Fig. 6.2.

6.6.4 A problematic case

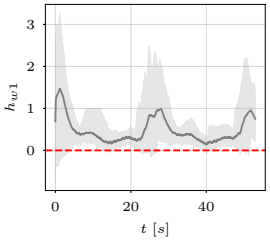
A limitation of the presented approach verifies when the drones need to split to track different targets and the drones are centered w.r.t. the targets at the moment in which both the constraints become active. Some undesirable oscillations may occur in the drone motion, as the drones struggle to determine which target to pursue. This issue arises



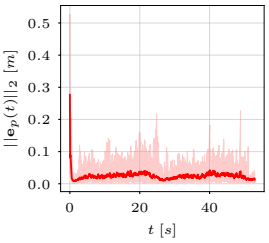
(a) ψ_{01}



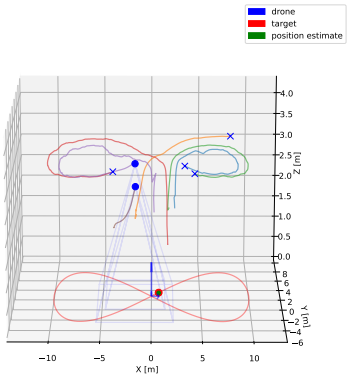
(b) ψ_{11}



(c) h_{w1}



(d) Norm of position errors



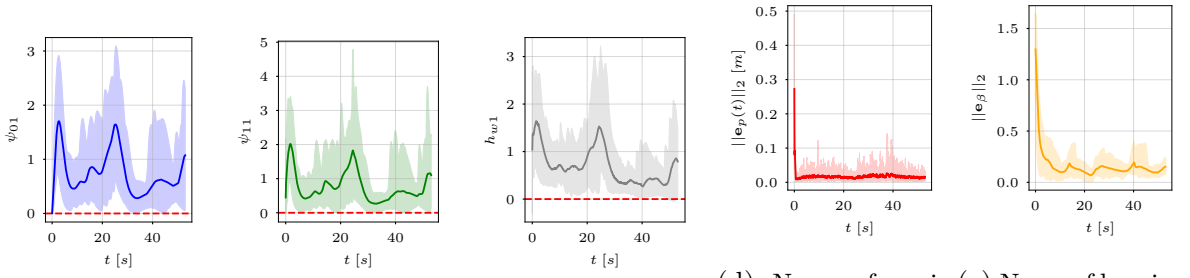
(e) 3D trajectories

Figure 6.3: Case 1: mean, maximum and minimum values for each t.

because of the lack of centralized optimality of the employed distributed CBFs and could be addressed by adopting the distributed CBF approach presented in Chapter 7, which is able to track the centralized optimal solution.

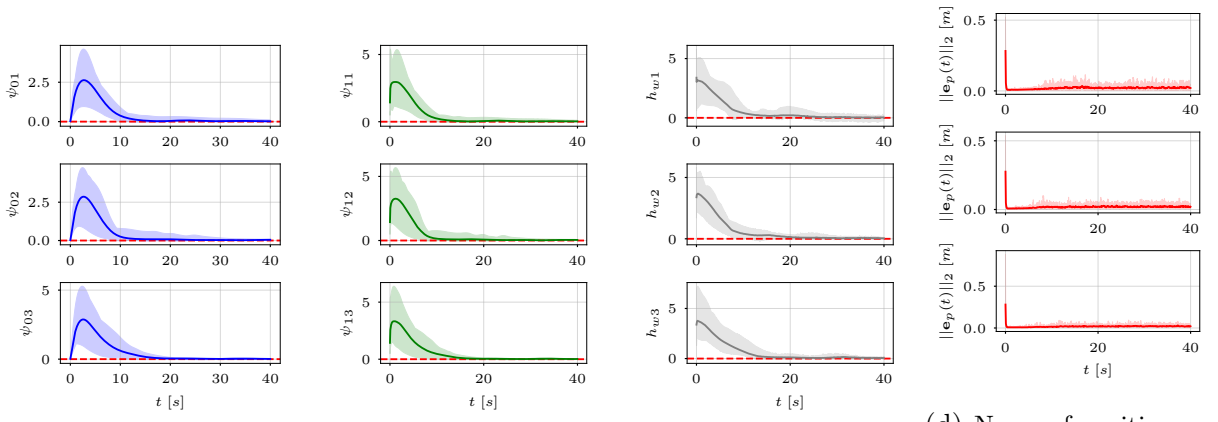
To demonstrate this fact, we simulate 4 drones and 2 targets. All the drones at the start of the simulation are in between the two targets. The parameters used in this simulation are $D_m = 0.5\ m$, $D_m^{th} = 1\ m$, $D_M = 3\ m$, $D_M^{th} = 2.5\ m$, $\sigma_{D_M}^2 = 0.5m^2$, $\sigma_{D_m}^2 = 0.1m^2$, $\alpha_M = 50$ deg, $\alpha_M^{th} = 40$ deg, $\sigma_\beta^2 = 0.01\ rad^2$, while the other parameters are the same of the first simulation, except for $\epsilon_\psi = 0.08$, $\epsilon_w = 1.1$. We do not considered measurement noise in this case to obtain smoother plots and highlight better the main results.

In Fig.6.6, we show the results obtained using the proposed **distributed controller**, while in Fig.6.7 we report the results obtained by using a **centralized controller** which solves the centralized QP. The initial conditions are the same in the two cases. In particular, in Fig.6.6a, the resulting 3D trajectory is shown, where it is visible that the targets



(a) ψ_{01} (b) ψ_{11} (c) h_{w1} (d) Norm of position errors (e) Norm of bearing errors

Figure 6.4: Case 2: mean, maximum and minimum values for each t.



(a) ψ_{01}, ψ_{02}, ψ_{03} (b) ψ_{11}, ψ_{12}, ψ_{13} (c) h_{w1}, h_{w2}, h_{w3} (d) Norm of position errors

Figure 6.5: Case 3: mean, maximum and minimum values for each t.

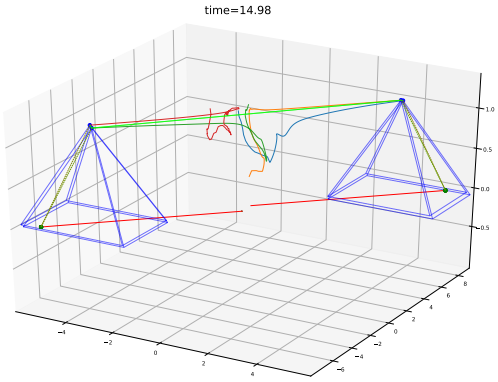
move in opposite direction, while the drones start in the middle among the two. By the plot of the weights (Fig.6.6b and Fig.6.6c), the targets are visible by all the drones with relatively high weights. As shown in the plot of the input of drone 1 and 3 (Fig.6.6f), the input becomes very discontinuous around $t = 3s$, when the constraints become active. The impact of this situation on the inputs of the drones also highly depends on the speed of the targets.

In the results obtained by using the centralized controller, starting from the same initial conditions, we have that, again, around $t = 3s$, the constraints become active and there is a peak in the inputs, but with limited oscillations. This can be explained by the fact that the drones, in the centralized case, are aware that one of the two constraints can also be satisfied by the other drone inputs. This simulation highlights a limitation of the presented approach due to the subotimality of the employed distributed CBFs. This fact, along with the distributed implementation of connectivity maintenance algorithms (Chapter 7), pushed us to explore distributed CBFs formulations converging to centralized optimality (Chapter 7).

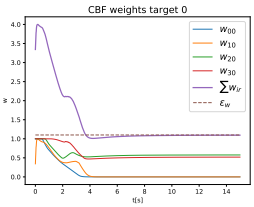
6.6.5 Discussion

From the reported results, it emerged that the multi-target tracking cases still present some limitations. Specifically, when the drones need to split to track different targets and the drones are centered w.r.t. the targets at the moment in which the constraints become active, some undesirable oscillations may occur in the drone motion. This issue arises because each drone lacks knowledge of the inputs applied by the other drones: indeed, a *centralized* implementation of the proposed strategy does not encounter the same problem. Better performance, both in terms of constraint fulfillment and optimality of secondary tasks, could be achieved by leveraging the solution for distributed CBFs described in Chapter 7, which allows to track the centralized optimality and by combining the multiple CBF constraints using a smooth minimum function as done in [Fernandez-Ayala et al., 2023].

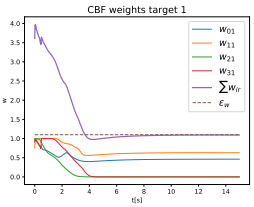
Another limitation of the approach is the influence of estimation errors on the fulfillment of the constraints. While some robustness of the control law to noise in the model is provided by CBFs [Xu et al., 2015], uncertainties in the estimation of the current estimated position could negatively impact constraint satisfaction. It would be interesting to address this issue by adding robustifying terms in the CBF framework, considering the current estimation uncertainty represented by the dispersion matrix $(\boldsymbol{W}_{ir}^{+})^{-1}$.



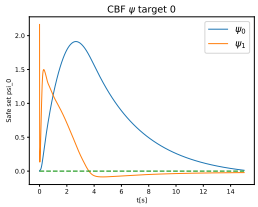
(a) Trajectories followed during the simulation with distributed controller.



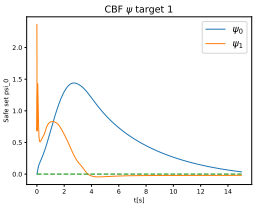
(b) CBF on sum of weights target 0



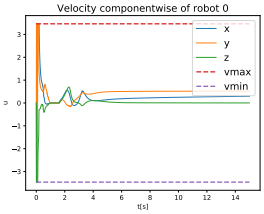
(c) CBF on sum of weights target 1



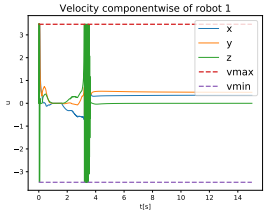
(d) CBFs on information ψ_{00} and ψ_{01}.



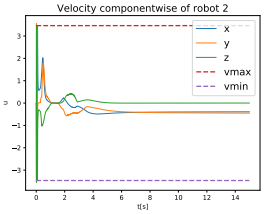
(e) CBFs on information ψ_{00} and ψ_{01}.



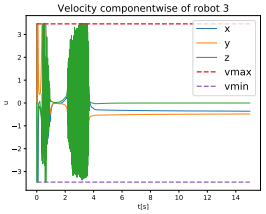
(f) Input drone 0



(g) Input drone 1



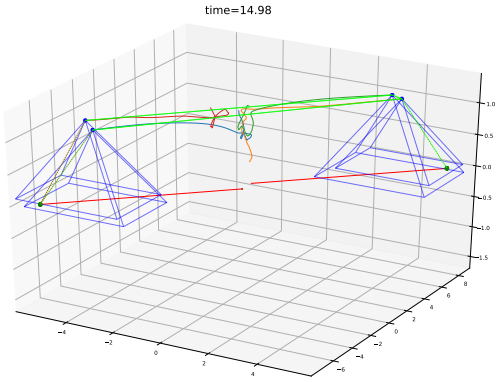
(h) Input drone 2



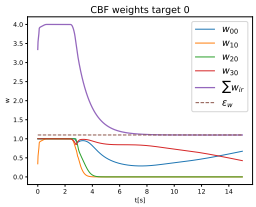
(i) Input drone 3

Figure 6.6: Simulation example of a problematic case with the distributed controller.

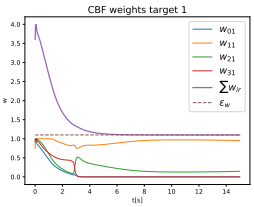
Finally, recall that the drones have no information in advance about the real motion of the targets. For this reason, the feasibility of a scenario cannot be stated a priori, since it could happen a *worst-case scenario*, where all targets have diverging motions, as well as a *best-case scenario*, in which all the targets move near to each other in the same direction and they can be sensed by the minimum number of drones.



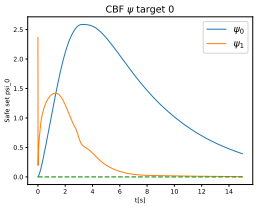
(a) Trajectories followed during the simulation with centralized controller.



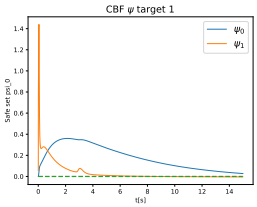
(b) CBF on sum of weights target 0



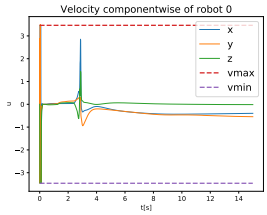
(c) CBF on sum of weights target 1



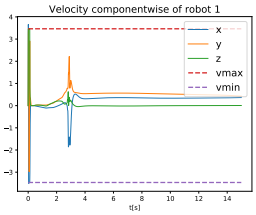
(d) CBFs on information ψ_{00} and ψ_{01}.



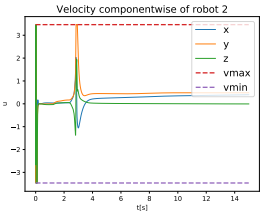
(e) CBFs on information ψ_{00} and ψ_{01}.



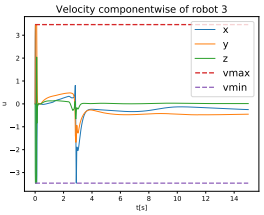
(f) Input drone 0



(g) Input drone 1



(h) Input drone 2



(i) Input drone 3

Figure 6.7: Simulation example of a problematic case with the centralized controller.

6.7 Conclusions

In this chapter, we proposed a distributed persistent monitoring scheme for estimating the state of one or multiple moving target(s) from bearing measurements by employing an *information consensus filter*. The filter is uniformly globally exponentially stable if a Persistency of Excitation condition is met. The main contribution of this work is to guarantee that such a PE condition is met also in the presence of sensing constraints while potentially achieving other tasks of interest. This is achieved by relying on two main tools, namely the distributed High Order Control Barrier Functions, used to enforce

the invariance of a set, and the weighted Observability Gramian with forgetting factor, which is used to quantify the persistency of excitation. The approach has been validated via numerical simulations. In addition to the discussion provided in Sect. 6.6.5, in the future we plan to consider more complex secondary tasks, as well as a time-varying graph topology with a connectivity maintenance constraint.

Chapter 7

Distributed CBFs with Application to Connectivity Maintenance

In this chapter, we present a framework for the distributed implementation of QP-based controllers, building upon and rectifying a significant limitation in a previously presented approach [Tan & Dimarogonas, 2021]. The framework is primarily designed for the distributed implementation of Control Barrier Functions (CBFs). By improving over some limitations in the current state-of-the-art, we allow their application to global connectivity maintenance in the presence of communication and sensing constraints.

Communication and sensing interactions within robot teams are commonly modeled using graph theory, and the connectivity of the interaction graph is vital for information flow and for the convergence of cooperative algorithms, such as for localization. However, real-world scenarios often involve limitations in communication and relative sensing due to factors like maximum range, limited Field of View (FoV), and others. Consequently, motion strategies unaware of these limitations can lead to graph disconnections and mission failure. This challenge has prompted extensive research on connectivity maintenance to ensure mission fulfillment despite sensing and communication constraints [M. Bernard et al., 2023; Capelli & Sabattini, 2020; Robuffo Giordano et al., 2013; Sabattini et al., 2013; Trakas et al., 2022; Yang et al., 2010].

Connectivity maintenance strategies can be categorized as either *local* [Dimarogonas & Johansson, 2008; Ji & Egerstedt, 2007; Restrepo et al., 2021] and *global* [M. Bernard et al., 2023; Capelli & Sabattini, 2020; Robuffo Giordano et al., 2013; Sabattini et al., 2013; Trakas et al., 2022]. Local approaches focus on maintaining all local connections but can severely restrict robot mobility. In contrast, global approaches leverage algebraic graph theory concepts, such as the connectivity (or Fiedler) eigenvalue, allowing robots to create and lose edges dynamically while preserving global graph connectivity, offering greater mobility at the expense of increased complexity.

Preserving connectivity in the sensing graph can be even more challenging due to

stricter sensing constraints, sometimes resulting in a directed sensing graph. Therefore, there is a demand for connectivity maintenance algorithms that minimally impact the execution of higher-level missions. While most connectivity maintenance works rely on gradient descent of potential functions [M. Bernard et al., 2023; Robuffo Giordano et al., 2013; Sabattini et al., 2013; Trakas et al., 2022], recent works [Capelli & Sabattini, 2020; Ong et al., 2023] have employed (centralized) Control Barrier Functions (CBFs) to introduce minimal modifications to nominal controllers, ensuring both constraint satisfaction and performance optimization [Ames et al., 2019]. Compared to methods based on potential functions, CBF-based approaches offer greater freedom of movement while respecting constraints, which is a critical advantage in multi-robot coordination.

This work has been published in [De Carli, Salaris, & Robuffo Giordano, 2024].

7.1 Formation model

We consider a group of N quadrotor UAVs equipped with onboard Inertial Measurement Units (IMUs), calibrated cameras and relative distance sensors, which are able to exchange data over a radio communication channel. As in some of the previous chapters, we consider a simplified kinematic model in $\mathbb{R}^3 \times \mathbb{S}^1$ for the i-th quadrotor with body-frame velocities and yaw rate commands $\boldsymbol{u}_i := \begin{bmatrix} \boldsymbol{v}_i^T & \omega_i \end{bmatrix}^T$:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ \omega_i \end{bmatrix} = \boldsymbol{T}(\psi_i)\boldsymbol{u}_i \tag{7.1}$$

We denote the state of the i-th robot as $\boldsymbol{q}_i = \begin{bmatrix} \boldsymbol{p}_i^T & \psi_i \end{bmatrix}^T$. For convenience, we also denote the stack of the state of each robot as $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_1^T & \dots & \boldsymbol{q}_N^T \end{bmatrix}^T \in (\mathbb{R}^3 \times \mathbb{S}^1)^N$ and analogously for the input $\boldsymbol{u}$. Then, the full dynamics can be written as

$$\dot{\boldsymbol{q}} = \bar{\boldsymbol{T}}(\boldsymbol{\psi})\boldsymbol{u}, \tag{7.2}$$

with $\bar{\boldsymbol{T}}(\boldsymbol{\psi}) := \mathrm{diag}(\boldsymbol{T}(\psi_i))$ being a block-diagonal matrix.

The robots are assumed to be equipped with a camera-like sensor with limited range and FoV and a sensor providing relative distance measurements. In particular, when robot j is visible by robot i, we assume that robot i can obtain a measurement of the

relative position of robot j in its own body-frame, i.e.:

$$ {}^{i}\boldsymbol{p}_{ij} = \boldsymbol{R}_i^T(\boldsymbol{p}_j - \boldsymbol{p}_i) \tag{7.3} $$

From such measurements, assuming that the sensing graph remains weakly connected, the robots are able to estimate their relative orientation, see e.g. Chapter 8 and [M. Bernard et al., 2023]. This motivates the necessity for connectivity maintenance when sensing constraints are present. We will discuss further about the relative yaw estimation from body-frame position measurements in the next chapter. Because of field of view limitations, the induced sensing graph is directed, while we assume bidirectional communication among the robots, inducing an undirected communication graph.

7.2 Distributed control barrier functions

In this section, we describe our contribution concerning distributed CBFs, which extends the approach presented in [Tan & Dimarogonas, 2021]. In particular, we are able to lift the restrictive assumption concerning the local Lie derivative of the control barrier function not vanishing [Tan & Dimarogonas, 2021] (see also Sect. 3.5.3). In the following, we use the global connectivity maintenance task to show the importance of lifting such assumption.

First, we give again a brief introduction to the problem formulation. For more details refer to Sect. 3.5.3. Then, we describe the proposed algorithm. The first part of the algorithm is the same as described in [Tan & Dimarogonas, 2021] and summarized in Sect. 3.5.3. We will point out from which point on the two algorithms will differ.

7.2.1 Problem statement

Consider a system in control-affine form:

$$ \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\xi} \tag{7.4} $$

with state $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_1^T & \dots & \boldsymbol{x}_N^T\end{bmatrix}^T \in \mathbb{R}^n$ and input $\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{\xi}_1^T & \dots & \boldsymbol{\xi}_N^T\end{bmatrix}^T \in \mathbb{R}^m$, and dynamics in the form of $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{f}_1(\boldsymbol{x}_1)^T & \dots & \boldsymbol{f}_N(\boldsymbol{x}_N)^T\end{bmatrix}^T$ and $\boldsymbol{g}(\boldsymbol{x}) := \mathrm{diag}(\boldsymbol{g}_i(\boldsymbol{x}_i))$ block-diagonal. Note that (7.2) belongs to this class of systems.

We consider a safe set $\mathcal{C}$ which is described by the zero superlevel set of a differentiable

function h which we want to render forward invariant and attractive. For example, in the connectivity maintenance case one wants to maintain $\lambda_2 \geq \epsilon$ with $\epsilon > 0$ and λ_2 being the connectivity eigenvalue (see Sect. 2.2), hence, one can define $h(\boldsymbol{x}) := \lambda_2(\boldsymbol{x}) - \epsilon$.

The class of constraints that we consider is the same as those considered in [Tan & Dimarogonas, 2021], that is, functions $h(\boldsymbol{x})$ such that the CBF parameters of (3.11) are locally obtainable, meaning that (3.11) can be expressed as:

$$\sum_{i=1}^{N} \boldsymbol{a}_i^T(\boldsymbol{x}_i^{\mathcal{N}_i})\boldsymbol{\xi}_i + \sum_{i=1}^{N} b_i(\boldsymbol{x}_i^{\mathcal{N}_i}) \leq 0 \tag{7.5}$$

where $\boldsymbol{x}_i^{\mathcal{N}_i} := \begin{bmatrix} \boldsymbol{x}_i^T & \{\boldsymbol{x}_j^T\}_{j\in\mathcal{N}_i} \end{bmatrix}^T$ is the stack of the state of robot i itself and the one of its neighbors. Notice that, $\boldsymbol{a}_i^T = -L_{gi}h$ while, b_i can vary based on the type of constraint considered and design choice, with the general expression being $b_i = -\kappa_i(\boldsymbol{x})(\alpha(h(\boldsymbol{x})) + L_{f_i}h(\boldsymbol{x}))$. For connectivity maintenance of robots modeled as (7.1), one could consider $b_i = -\kappa_i(\boldsymbol{x})\alpha(h(\boldsymbol{x}))$, with $\kappa_i(\boldsymbol{x})$ being any partition of the unity, i.e. $\sum_{i=1}^{N}\kappa_i(\boldsymbol{x}) = 1$, for example $\kappa_i = \frac{1}{N}$. Also, as shown in, e.g., [Yang et al., 2010], by estimating in a distributed way the Fiedler eigenpair, the gradients for global connectivity maintenance can be computed locally, as required by (7.5).

7.2.2 QP formulation

We assume that, a nominal controller provides a Lipschitz continuous desired input $\boldsymbol{\xi}^d$ which does not need to satisfy the condition in (3.11). Then, the following Quadratic Program (QP) is formulated in order to modify in a minimally invasive way the nominal controller, so that condition (3.11) is always satisfied

$$\begin{aligned} \min_{\boldsymbol{\xi}\in\mathbb{R}^m} \cdot \quad & \frac{1}{2}\sum_{i=1}^{N}\left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \sum_{i=1}^{N}\boldsymbol{a}_i^T\boldsymbol{\xi}_i + \sum_{i=1}^{N} b_i \leq 0, \end{aligned} \tag{7.6}$$

where we omitted the state dependency of $\boldsymbol{a}_i$ and b_i for the sake of readability. However, we remind that these terms change over time as the state evolves along the system trajectories. We also point out that, here, the problem is formulated as a least square problem minimizing the control input for simplicity, but the extension to a more generic quadratic cost function (still preserving the decoupling between the robot inputs) is straightforward.

Defining $\bar{\boldsymbol{a}} := \begin{bmatrix} \boldsymbol{a}_1^T & \dots & \boldsymbol{a}_N^T \end{bmatrix}^T$ and $\bar{b} := \sum_{i=1}^{N} b_i$, we make the following assumption.

Assumption 5. *The QP* (7.6) *is feasible, i.e.* $\bar{b} \leq 0$ *whenever* $\bar{\boldsymbol{a}} = 0$.

The QP problem in (7.6) is centralized. Our goal is to obtain a distributed algorithm which asymptotically converges to the time-varying centralized optimal solution of the QP while always enforcing the safety constraint. To accomplish this task, in [Tan & Dimarogonas, 2021], the following QP was considered

$$\begin{aligned} \min_{(\boldsymbol{\xi},\boldsymbol{y})\in\mathbb{R}^{m+N}} \cdot \quad & \frac{1}{2}\sum_{i=1}^{N}\left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{\xi}_i + \sum_{j\in\mathcal{N}_i}(y_i - y_j) + b_i \leq 0, \quad \forall i \in \mathcal{V} \end{aligned} \tag{7.7}$$

where $\boldsymbol{y} = \begin{bmatrix} y_1, \dots y_N \end{bmatrix}^T \in \mathbb{R}^N$ is an auxiliary variable, with the element y_i associated to robot i. The equivalence among (7.7) and (7.6) was shown in [Tan & Dimarogonas, 2021] and we discussed it in detail in Sect. 3.5.3. In particular, this implies that each solution $(\boldsymbol{\xi}', \boldsymbol{y}')$ satisfying the constraint in (7.7), also satisfies the constraint in the original QP (7.6) and viceversa. The constraints in the previous QP can be equivalently written in matrix form as:

$$\bar{\boldsymbol{A}}\boldsymbol{\xi} + \boldsymbol{L}\boldsymbol{y} + \boldsymbol{b} \leq 0 \tag{7.8}$$

with $\bar{\boldsymbol{A}} = \text{diag}(\boldsymbol{a}_i^T)$, $\boldsymbol{L}$ is the unweighted Laplacian matrix of the time-varying undirected graph and $\boldsymbol{b} = \begin{bmatrix} b_1 & \dots & b_N \end{bmatrix}^T$. One may, then, formulate the following QP

$$\begin{aligned} \min_{(\boldsymbol{\xi}_i,\boldsymbol{y})\in\mathbb{R}^{m_i+N}} \cdot \quad & \frac{1}{2}\left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{\xi}_i + \sum_{j\in\mathcal{N}_i}(y_i - y_j) + b_i \leq 0 \end{aligned}. \tag{7.9}$$

The challenge here is that consistency of $\boldsymbol{y}$ needs to be preserved across the robots. To solve this issue, in [Tan & Dimarogonas, 2021] the Karush-Kuhn-Tucker (KKT) optimality conditions of the problem (7.7) were studied and a distributed adaptive law for updating $\boldsymbol{y}$ was proposed so that $\boldsymbol{y}$ converges to the optimal $\boldsymbol{y}^*$. Hence, each robot solves (7.9) but with only $\boldsymbol{\xi}_i$ as decision variables, while $\boldsymbol{y}$ is fixed in the QP. Let $\boldsymbol{l}_i$ represent the i-th row of the Laplacian $\boldsymbol{L}$, the optimal solution $\boldsymbol{y}^*$ from the KKT conditions needs to satisfy the following condition $\forall i \in \mathcal{V}$:

$$\begin{cases} \boldsymbol{a}_i^T\boldsymbol{\xi}_i^d + \boldsymbol{l}_i^T\boldsymbol{y}^* + b_i \leq 0, & \text{if} \quad \bar{\boldsymbol{a}}^T\boldsymbol{\xi}^d + \bar{b} \leq 0 \\ \boldsymbol{a}_i^T\boldsymbol{\xi}_i^d + \boldsymbol{l}_i^T\boldsymbol{y}^* + b_i = k\boldsymbol{a}_i^T\boldsymbol{a}_i, & \text{if} \quad \bar{\boldsymbol{a}}^T\boldsymbol{\xi}^d + \bar{b} > 0 \end{cases} \tag{7.10}$$

with

$$k = (\bar{\boldsymbol{a}}^T \boldsymbol{\xi}^d + \bar{b}) / \left\| \bar{\boldsymbol{a}} \right\|_2^2 . \tag{7.11}$$

Since $k \leq 0$ when $\bar{\boldsymbol{a}}^T \boldsymbol{\xi}^d + \bar{b} \leq 0$, a sufficient condition for $\boldsymbol{y}^*$ is simply to satisfy (7.10) as

$$\boldsymbol{a}_i^T \boldsymbol{\xi}_i^d + \boldsymbol{l}_i^T \boldsymbol{y}^* + b_i = k \boldsymbol{a}_i^T \boldsymbol{a}_i \quad \forall i \in \mathcal{V}, \tag{7.12}$$

which can be rewritten in matrix form as

$$\boldsymbol{L}\boldsymbol{y}^* = k \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_N^T \boldsymbol{a}_N \end{bmatrix} - \bar{\boldsymbol{A}} \boldsymbol{\xi}^d - \boldsymbol{b}. \tag{7.13}$$

At this stage we can highlight the difference between what has been proposed in [Tan & Dimarogonas, 2021] and our contribution. In [Tan & Dimarogonas, 2021], local variables are defined as

$$k_i = \frac{1}{\boldsymbol{a}_i^T \boldsymbol{a}_i} (\boldsymbol{a}_i^T \boldsymbol{\xi}_i^d + \boldsymbol{l}_i^T \boldsymbol{y}^* + b_i) \tag{7.14}$$

and it was shown that if $k_i = k_j$, $\forall (i,j) \in \mathcal{E}$, then condition (7.10) is satisfied. Hence it was proposed to update the variables $\boldsymbol{y}$ according to the following finite time modified consensus:

$$\dot{\boldsymbol{y}} = -k_0 \text{sign}(\boldsymbol{L}\boldsymbol{k}) \tag{7.15}$$

with $\boldsymbol{k} = \begin{bmatrix} k_1 & \dots & k_N \end{bmatrix}^T$. A crucial point is that, due to (7.14), this approach requires $\|\boldsymbol{a}_i(t)\| > 0 \ \forall t, \ \forall i$ which, as also acknowledged in [Tan & Dimarogonas, 2021], can be quite restrictive. Indeed, as it will be discussed in the next section, this assumption is often not verified also for the standard problem of connectivity maintenance.

To cope with this issue, we propose the following solution. Each robot can estimate k in (7.11) with two dynamic average consensus estimators, one for computing the numerator average $n_{\text{avg}} := 1/N \sum_{i=1}^N ({\boldsymbol{a}_i}^T \boldsymbol{\xi}_i^d + b_i)$ and one for computing the denominator average $d_{\text{avg}} := 1/N \sum_{i=1}^N \left\| \boldsymbol{a}_i \right\|^2$, from which each robot can obtain its estimate of k as $k_i = n_{\text{avg}}/d_{\text{avg}}$. Then, the only unknown left in (7.13) is $\boldsymbol{y}^*$. While solving a linear system in a distributed manner is generally challenging, leveraging the sparsity of the Laplacian matrix and its positive semidefinite nature enables a distributed solution for finding $\boldsymbol{y}^*$ in (7.13) through the simulation of the following dynamical system:

$$\dot{\boldsymbol{y}}(t) = -k_y \left(\boldsymbol{L}\boldsymbol{y}(t) - \boldsymbol{r}(t) \right) - k_{\text{avg}} \bar{y}(t) \boldsymbol{1}_N \tag{7.16}$$

with

$$\boldsymbol{r} := \text{diag}(k_i) \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_N^T \boldsymbol{a}_N \end{bmatrix} - \bar{\boldsymbol{A}}\boldsymbol{\xi}^d - \boldsymbol{b} \tag{7.17}$$

and $\bar{y} = \frac{1}{N}\mathbf{1}_N^T \boldsymbol{y}$ is the mean of $\boldsymbol{y}$, which can be estimated with a dynamic average consensus estimator. This system corresponds to the continuous-time version of the Richardson iteration [Bertsekas & Tsitsiklis, 2015], which is used to solve linear systems in parallel computing schemes where the matrix to be inverted is positive definite (in this case $\boldsymbol{L}+\mathbf{1}_N\mathbf{1}_N^T$). An alternative, which potentially may lead to faster convergence is represented by the Jacobi over-relaxation method [Bertsekas & Tsitsiklis, 2015]. We point out that (7.16) is fully distributed, in fact, each robot implements:

$$\dot{y}_i = -k_y \left(\sum_{j \in \mathcal{N}_i(t)} (y_i - y_j) - k_i \boldsymbol{a}_i^T \boldsymbol{a}_i + \boldsymbol{a}_i^T \boldsymbol{\xi}_i^d + b_i \right) - k_{\text{avg}} \bar{y}. \tag{7.18}$$

Notice that, the only equilibrium of this system is given by the minimum-norm solution to (7.13). We also point out that, the average of $\boldsymbol{y}$ is not relevant as it has no effect on the QP constraint since $\boldsymbol{L}\bar{y}\mathbf{1}_N = \mathbf{0}$. We add the second term in (7.16) with the purpose of guaranteeing boundedness of $\bar{y}$. We also point out that $\boldsymbol{L}\mathbf{1}_N \subseteq \mathbf{1}_N$ for all t, meaning that the consensus subspace is infinitesimally $\boldsymbol{L}$-invariant [Bullo, 2023; De Pasquale et al., 2023].

Let $\boldsymbol{y}_\perp := \boldsymbol{y} - \bar{y}\mathbf{1}_N$ and analogously $\boldsymbol{r}_\perp$, and consider the error $\boldsymbol{e}_\perp := \boldsymbol{L}\boldsymbol{y}_\perp - \boldsymbol{r}_\perp$, then:

$$\begin{aligned} \dot{\boldsymbol{e}}_\perp(t) &= \boldsymbol{L}\left(-k_y(\boldsymbol{L}(t)\boldsymbol{y}_\perp(t) - \boldsymbol{r}_\perp(t))\right) - \dot{\boldsymbol{r}}_\perp(t) \\ &= -k_y \boldsymbol{L}\boldsymbol{e}_\perp - \dot{\boldsymbol{r}}_\perp(t) \\ \dot{y}_{\text{avg}}(t) &= -k_{\text{avg}}\bar{y}(t) + k_y \bar{r}(t) \end{aligned} \quad . \tag{7.19}$$

where in the first equation the average dynamics of $\dot{\boldsymbol{y}}_\perp$ are removed as they are in the null space of $\boldsymbol{L}$. Then, the error $\boldsymbol{e}_\perp$ is practically exponentially stable with rate $c := k_y \min_{\tau\in[t_0,t]} \lambda_2(t)$ and $\boldsymbol{y}_\perp$ converges to a ball of size $\frac{1}{c}\sup_{\tau\in[t_0,t]} ||\dot{\boldsymbol{r}}_\perp||$ centered in $\boldsymbol{y}_\perp^* = \boldsymbol{L}^\dagger \boldsymbol{r}$, with $\dagger$ indicating the Moore-Penrose pseudoinverse, which gives the minimum-norm solution to equation (7.13). The origin of the average $\bar{y}$ is practically exponentially stable as well and, in particular, what is of interest in this case is the fact that it remains bounded. Also, notice that, with the proposed solution, it is *not required* to have $\|\boldsymbol{a}_i\| > 0 \ \forall i$. We point out that, we did not consider the dynamics of the dynamic average consensus used to estimate $\bar{y}(t)$ in the previous analysis. In practice, since it is not required for $\boldsymbol{y}$ to be

zero-mean, it simply needs to be bounded, then k_{avg} can be chosen to be much smaller than k_y so that the effects of the disagreement in computing $\bar{y}$ are negligible.

Remark 11. *This approach can be extended to the case of multiple CBF constraints using soft minimum functions as shown in [Fernandez-Ayala et al., 2023].*

Interestingly, solving the QP (7.9) without auxiliary variables is equivalent to pre-allocation schemes such as in [Balandi et al., 2023; Lindemann & Dimarogonas, 2020; Wang et al., 2017], which maintains safety but leads to suboptimal solutions. The role of the auxiliary variables is, in fact, to optimally allocate the constraint among the robots.

We also point out that, for simplicity of exposition, we considered a cost function of the type $\sum_{1=1}^{N} \left\| \boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d \right\|_2^2$. The extension to a more generic quadratic cost function $\sum_{i=1}^{N} \frac{1}{2} \boldsymbol{\xi}_i^T \boldsymbol{H}_i \boldsymbol{\xi} + \boldsymbol{w}_i^T \boldsymbol{\xi}_i$ with $\boldsymbol{H}_i \succ 0$ is straightforward, see [De Carli et al., 2023] for the expression of k in this case.

While, in [Tan & Dimarogonas, 2021], convergence in finite time to the time-varying optimal solution was achieved, in the proposed solution, we only achieve practical exponential stability of the optimal solution. But, as already mentioned, our method removes any limitation concerning $\|\boldsymbol{a}_i\|$.

7.3 Connectivity maintenance

As we stated in Sect. 2.2, a well-known result in graph theory is that an undirected graph is connected if and only if $\lambda_2 > 0$. When the connectivity of the graph depends on sensing and communication constraints, e.g. limited range and field of view, a classical approach to ensure connectivity is summarized below by considering several previous works on this subject [M. Bernard et al., 2023; Capelli & Sabattini, 2020; Ong et al., 2023; Robuffo Giordano et al., 2013; Sabattini et al., 2013; Trakas et al., 2022]:

- design a weighted adjacency matrix $\boldsymbol{A}_w = [a_{ij}]$, such that the weight a_{ij} smoothly goes to zero when the edge between robot i and robot j approaches disconnection (because of the sensing/communication constraints);
- estimate in a distributed way the Fiedler eigenpair, i.e. $(\lambda_2, \boldsymbol{v}_2)$ of the corresponding weighted Laplacian matrix $\boldsymbol{L}_w$ [Malli et al., 2021; Yang et al., 2010];
- design a control strategy, e.g. based on potential functions [M. Bernard et al., 2023; Robuffo Giordano et al., 2013; Sabattini et al., 2013; Trakas et al., 2022] or based

on CBFs [Capelli & Sabattini, 2020; Ong et al., 2023], which ensures that $\lambda_2(t) \geq \epsilon$ with $\epsilon > 0$ $\forall t$.

A fundamental assumption is that, at the initial time t_0, the undirected graph $\mathcal{G}_u$ is connected, i.e., $\lambda_2(t_0) > 0$. We point out that an approach based on CBFs, as in the present work, does not require $\lambda_2(t_0) \geq \epsilon$, in fact, the safe set can be shown to be asymptotically stable under the effect of the CBF [Ames et al., 2019; Capelli & Sabattini, 2020]. In this work, we consider distance constraints, expressed as $d_{\min} \leq d_{ij} \leq d_{\max}$, where $d_{ij} = \|\boldsymbol{p}_j - \boldsymbol{p}_i\|$ and field of view constraints $c_{ij} = \boldsymbol{\beta}_{ij}^T \boldsymbol{e}_1 \geq c_{\min}$ where $\boldsymbol{\beta}_{ij} = {}^i\boldsymbol{p}_{ij}/d_{ij}$ is the relative body-frame bearing and $\boldsymbol{e}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. We consider symmetric weights, i.e. $a_{ij} = a_{ji}$, given by $a_{ij} = \bar{a}_{ij} + \bar{a}_{ji}$, with $\bar{a}_{ij} = w_{d_{ij}} w_{b_{ij}}$, where $w_{d_{ij}}$ (resp. $w_{b_{ij}}$) is a continuously differentiable weight which smoothly vary from 1 to 0 as the distance (resp. FoV) limit of the sensor is being approached. The weights used are the ones in (2.39).

Remark 12. *While the field of view constraint results in a directed sensing graph, the bidirectional communication allows to define $a_{ij} = \bar{a}_{ij} + \bar{a}_{ji}$. This approach makes possible to perform connectivity maintenance on a undirected graph, ensuring weak connectivity of the sensing graph.*

Then, the gradient of the Fiedler eigenvalue with respect to the robots position can be expressed as follows [Yang et al., 2010]:

$$\frac{\partial \lambda_2}{\partial \boldsymbol{q}_i} = \sum_{j \in \mathcal{N}_i} (v_{2j} - v_{2i})^2 \frac{\partial a_{ij}}{\partial \boldsymbol{q}_i} \tag{7.20}$$

where v_{2i} is the i-th component of the Fiedler eigenvector $\boldsymbol{v}_2$ and v_{2i} as well as λ_2 are computed in a distributed way using the algorithm presented in [Malli et al., 2021] and a discrete-time PI average consensus [Kia et al., 2019]. Then in order to maintain $\lambda_2 \geq \epsilon$, each robot solves the following QP:

$$\begin{aligned} \min_{\boldsymbol{u}_i \in \mathbb{R}^4} \cdot \quad & \frac{1}{2} \left\| \boldsymbol{u}_i - \boldsymbol{u}_i^d \right\|_2^2 \\ \text{s.t.} \quad & - \sum_{j \in \mathcal{N}_i} \left((v_{2j} - v_{2i})^2 \frac{\partial a_{ij}}{\partial \boldsymbol{q}_i} \right) \boldsymbol{T}(\psi_i) \boldsymbol{u}_i - \frac{1}{N} \alpha_0 (\lambda_2 - \epsilon)^3 \\ & + \sum_{j \in \mathcal{N}_i} (y_i - y_j) \leq 0, \end{aligned} \tag{7.21}$$

where we chose to use the extended class $\mathcal{K}$ function $\alpha(x) = \alpha_0 x^3$ with $\alpha_0 > 0$ and $\boldsymbol{y}$ is computed according to (7.16). Notice that, as mentioned in Sec. 7.2, it can happen that

$\frac{\partial \lambda_2}{\partial \boldsymbol{q}_i} = \mathbf{0}$ but $\frac{\partial \lambda_2}{\partial \boldsymbol{q}} \neq \mathbf{0}$, for example when all the weights of the i-th robot are already at their maximum or when $v_{2i} = v_{2j}\ \forall j \in \mathcal{N}_i$. In this case, the global QP is still feasible and the i-th robot can simply set $\boldsymbol{u}_i = \boldsymbol{u}_i^d$. Here, the assumption $\boldsymbol{a}_i \neq 0$, made in [Tan & Dimarogonas, 2021], as in many other applications, is not verified, hence, motivating the alternative approach that we propose.

The algorithm requires the following quantities to be communicated:

- 7 scalar variables for the Power Iteration method [Malli et al., 2021] and related consensus
- 4 scalar variables to estimate k_i using two PI consensus
- 1 scalar variables to estimate $\boldsymbol{y}$ in (7.16) and its average.

Also, either N (which is used in (7.21)) is fixed and known by all robots, or it may be estimated using consensus. As previously mentioned, other possibilities can be used for $\kappa_i(x)$, other than $\kappa_i = 1/N$, which may be computed using a consensus e.g. see [Lindemann & Dimarogonas, 2020].

7.4 Simulation results

This section presents the results of a simulation of $T = 80s$ with $N = 6$ quadrotors, starting from a connected random configuration. The nominal input to the quadrotors is provided by the active sensing controller proposed in [De Carli et al., 2023], which generates trajectories that are exciting for the robot localization but unaware of the sensing constraints. We point out that these trajectories are particularly challenging in our context because the active sensing [De Carli et al., 2023] forces the relative positions among the robots to change very rapidly and, as a consequence, the Laplacian entries and eigenvalues, making it harder to estimate the Fiedler eigenpair and track the optimal solution. The resulting input of each agent is filtered by the connectivity maintenance CBF by solving (7.21) and updating y_i according to (7.18). A lower-level safety layer implements collision avoidance using another CBF. The connectivity maintenance runs @25Hz, with the corresponding consensus running @150Hz, while the estimation of the Fiedler eigenpair runs @250Hz and the corresponding consensus run @750Hz. A quadrotor can sense its neighbors if they lie within the distance range $[1, 4.5]m$, with a maximum FoV angle set at 60 degrees. The minimum threshold for the Fiedler eigenvalue λ_2 is set

to $\epsilon = 0.2$ and $\alpha_0 = 100$. The norm of the desired input velocity is saturated at $1m/s$ and the yaw-rate at $1rad/s$. The parameters for the power iteration proposed in [Malli et al., 2021] are set to $\beta = 100$ and $\gamma = 40$, while the consensus used is a discrete-time PI from (31) in [Kia et al., 2019] with gains $k_p = 0.05$, $k_I = 0.008$ and $\rho = 0.9$. In (7.18), we use $k_y = 50$ and it runs @250Hz. The robots start to move after $1s$ for allowing the power iteration to reach a satisfactory convergence before starting to move.

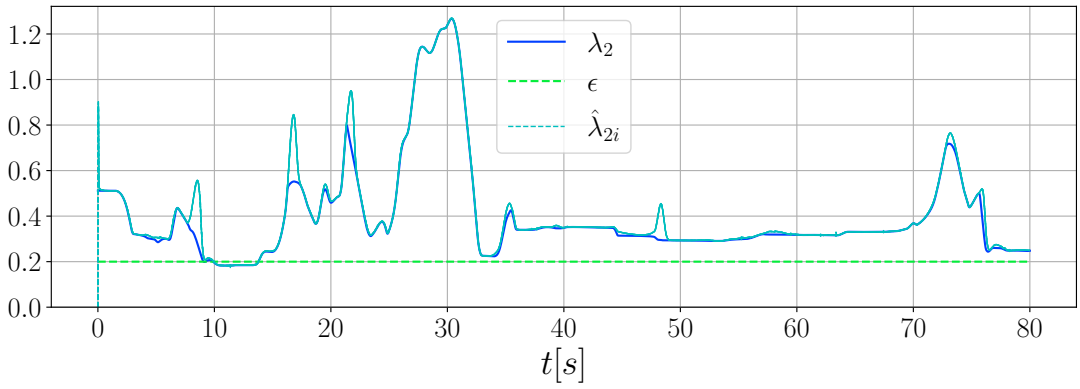


Figure 7.1: The plot represents λ_2, the lower bound ϵ and the estimates of the Fiedler eigenvalue $\hat{\lambda}_{2i}$ for each robot (in dashed lines). Notice that for most of the time the estimates are overlapping the real λ_2.

In Fig. 7.1, we report the Fiedler eigenvalue λ_2 along with, the lower bound ϵ and the local estimations for each robot (in dashed lines). We point out that, when λ_3 (and possibly λ_4) get close to λ_2, the estimation becomes much less reliable (as well-known) with the estimate $\hat{\mathbf{v}}_2$ approaching the hyper-plane spanned by $\mathbf{v}_2$-$\mathbf{v}_3$ (and possibly $\mathbf{v}_4$).

We verify the tracking of the optimal $\boldsymbol{y}^*$ in Fig. 7.2 by plotting $\boldsymbol{y}_\perp$ and the solution $\boldsymbol{L}^\dagger \boldsymbol{r}$ to $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{r}$ as ground truth.

In order to verify the convergence of the distributed solution to the centralized one, we run in parallel the distributed CBFs (7.21) and the corresponding centralized one (7.6), whose inputs are not applied and only used as a comparison. The comparative analysis of the resulting inputs is presented in Fig. 7.3, where it can be noticed the optimal solution is tracked very closely for most of the time.

7.5 Conclusions

We have presented a framework for a class of distributed QP-based controllers that typically arise when employing Control Barrier Functions, and we have addressed a significant

limitation of the recent literature in this field. Each agent solves a local QP and locally adapts an auxiliary variable. The solutions of the local QPs asymptotically converge to a neighborhood of the centralized optimal solution. This improvement allows us to apply distributed CBFs to the problem of global connectivity maintenance for quadrotor formations, accounting for distance and field of view constraints. The effectiveness of the approach is shown by comparing the obtained inputs with the ones provided by the corresponding centralized CBF. In future, we plan to perform experiments using the proposed framework, possibly performing a comparison with the classical approach based on potential functions.

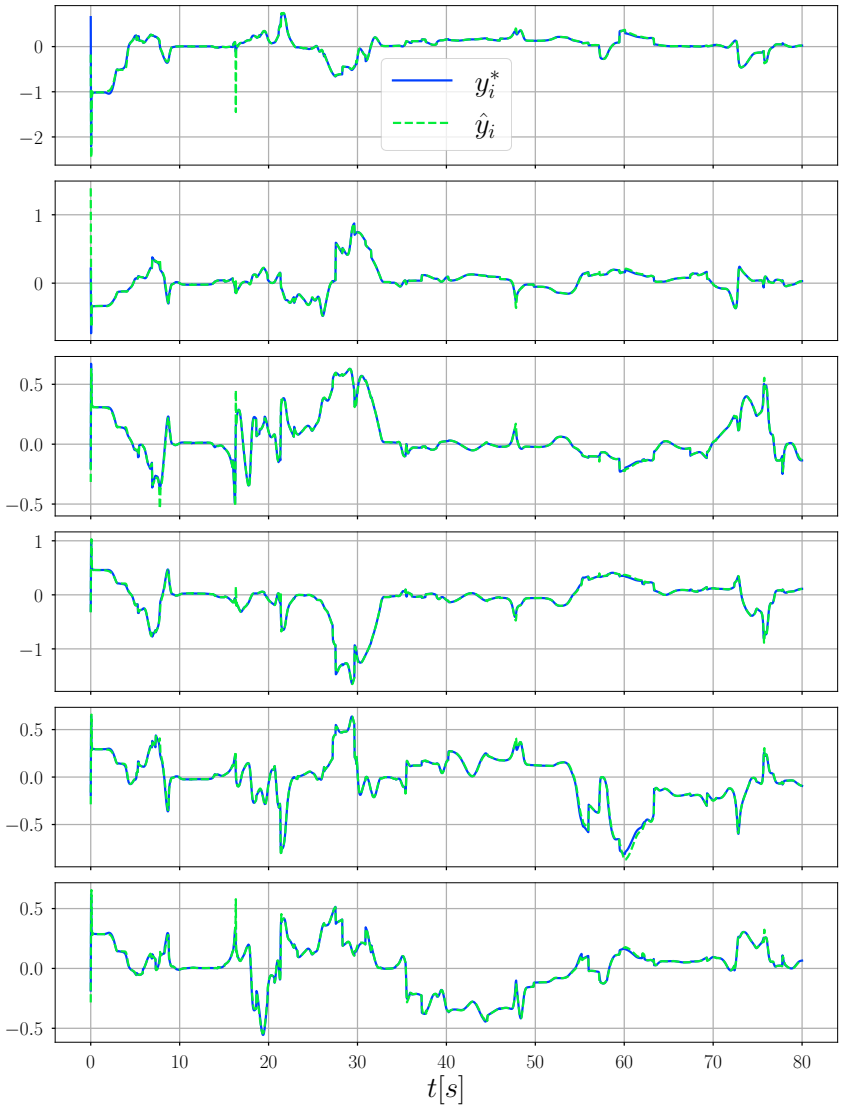


Figure 7.2: The plot shows, for each robot, the estimate $\hat{y}_i$ given by (7.18) (in dashed green) tracking the optimal y_i^* (in solid blue).

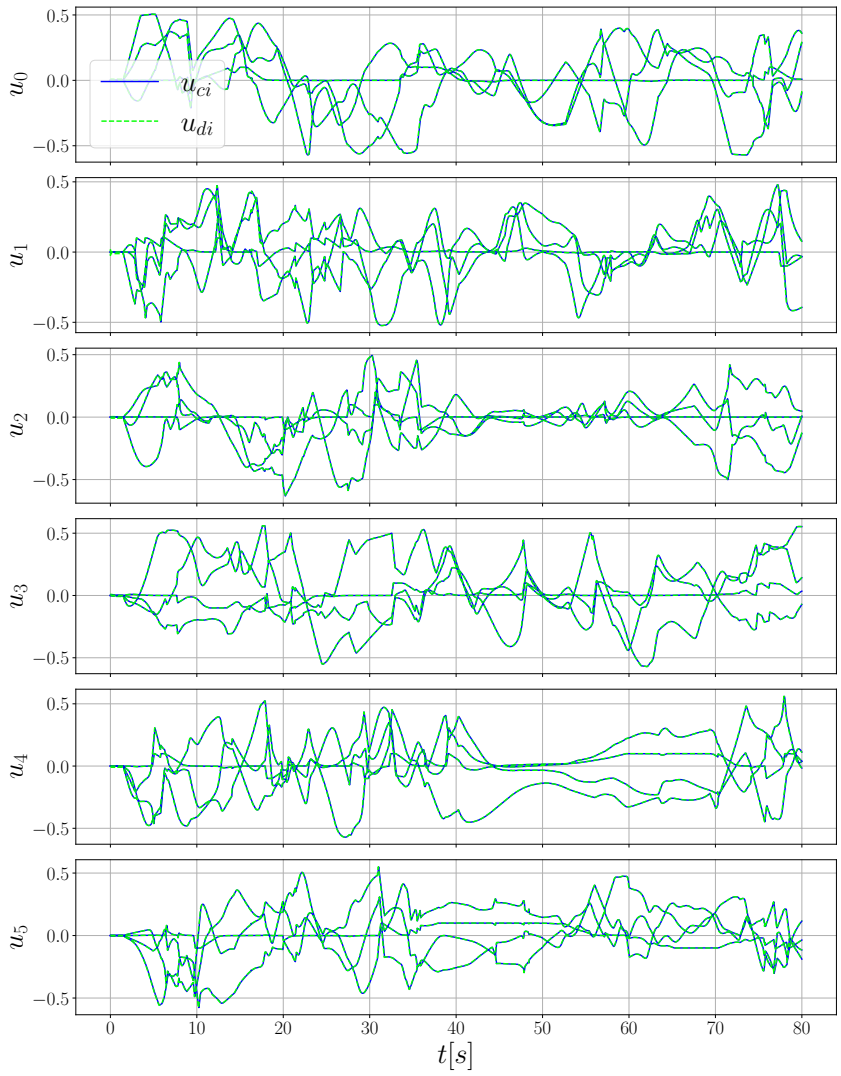


Figure 7.3: Comparison among the inputs obtained solving the QP in a distributed way (7.21) (in dashed green) and in a centralized way (**??**) (in solid blue). Each plot contains each input components for a certain drone. One can see that after a short initial transient, the trajectories are almost superimposed for most of the time.

Part IV

Distributed Observer Design

Chapter 8

Cooperative Localization from Directed Body-Frame Position Measurements

Cooperative control challenges often demand substantial coordination among robots, mandating a shared comprehension of specific physical quantities. This often entails establishing a consensus on a *shared frame* to facilitate the exchange of information initially captured in *local frames*.

Absence of shared orientation necessitates either global orientation estimation [Boumal et al., 2013; B.-H. Lee & Ahn, 2016a, 2016b; Zhao & Zelazo, 2015] or coordinate frame alignment [Montijano, Zhou, et al., 2014; Oh & Ahn, 2013]. Previous studies primarily address orientation estimation when relative orientation measurements are available [Boumal et al., 2013; B.-H. Lee & Ahn, 2016a, 2016b; Van Tran & Ahn, 2020; Zhao & Zelazo, 2015]. Fewer works, however, tackle scenarios where solely body-frame relative position or bearing measurements are accessible [Leonardos et al., 2019; X. Li et al., 2019; Schiano et al., 2016; Van Tran et al., 2018]. Typically, agents' orientation can be estimated uniquely up to a coordinated rotation. Estimating this unknown coordinated rotation becomes feasible if at least one agent in the network measures its orientation relative to the world frame.

In prior works [B.-H. Lee & Ahn, 2016a, 2016b; Van Tran & Ahn, 2020], algorithms are proposed to estimate the global orientation of individual agents from relative orientation measurements, demonstrating almost global convergence. These algorithms rely on defining linear dynamics depending on a matrix which is shown to be *similar* to a Laplacian matrix. The authors consider both the case in which the orientation belongs to the unit sphere $\mathbb{S}^1$ [B.-H. Lee & Ahn, 2016b] and the case in which the orientation belongs to $SO(3)$ [B.-H. Lee & Ahn, 2016a; Van Tran & Ahn, 2020]. Addressing noise-corrupted relative orientation measurements, [Boumal et al., 2013] introduces a locally

optimal maximum likelihood estimator.

In another approach by [Schiano et al., 2016], an observer estimates the position and yaw orientation of a group of robots in a shared frame using body-frame bearing measurements. This method utilizes gradient descent for bearing rigid formations, exhibiting local convergence. Meanwhile, works such as [X. Li et al., 2019; Van Tran et al., 2018], present algorithms for estimating orientation in a shared frame from body-frame bearing measurements, relying on an *undirected sensing graph* to recover relative orientations among robots from reciprocal measurements.

In this chapter, we propose an observer to estimate in a common frame the position and yaw orientation of a group of robots from *body-frame* relative position measurements. The state of the robots is represented by their position and yaw orientation ($\mathbb{R}^3 \times \mathbb{S}^1$). The graph representing the sensing interaction among the robots is directed and it is only required to be weakly connected in addition to satisfy certain *persistency of excitation* conditions.

Our scheme comprises three distinct components:

1. An adaptive observer estimating relative yaw corresponding to persistently exciting edges derived from body-frame relative position measurements.

2. Modification of the observer introduced in [B.-H. Lee & Ahn, 2016b] to estimate yaw in a shared frame from the estimated relative yaws. Notably, our adaptation accounts for varying orientation and imperfect measurements, unlike [B.-H. Lee & Ahn, 2016b].

3. An observer estimating positions in a shared frame, utilizing estimated yaw orientations and body-frame relative position measurements.

We point out that considering only planar orientation is justified by the fact that typically, the attitude of UAVs can be retrieved from Inertial Measurement Units (IMUs) measurements using well-established observers such as [Mahony et al., 2008], while the same is often not true for the yaw orientation, particularly indoors.

The proposed scheme assumes an undirected communication graph but, unlike previous works [X. Li et al., 2019; Van Tran et al., 2018], it does not require the assumption of an undirected sensing graph. This relaxed assumption is crucial for practical robotic applications, considering constraints posed by sensors with limited field of view (FoV). Instead, we introduce a *persistency of excitation* (PE) condition, requiring the robots to

have non-zero velocity on the x-y plane, which is a mild assumption in robotic applications.

This work has been published in [De Carli, Restrepo, et al., 2024].

8.1 System model

As in previous chapters, we consider a group of N UAVs modeled as simple first order kinematic system with *known body-frame* velocities and yaw rate commands $\boldsymbol{u}_i := \begin{bmatrix}\boldsymbol{v}_i^\top & \omega_i\end{bmatrix}^\top \in \mathbb{R}^4$:

$$\begin{bmatrix}\dot{\boldsymbol{p}}_i \\ \dot{\psi}_i\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}_i & \boldsymbol{0} \\ \boldsymbol{0} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_i \\ \omega_i\end{bmatrix} \tag{8.1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^3$ is the robot position, $\psi_i \in (-\pi, \pi]$ is the yaw angle and $\mathbf{R}_i := \mathbf{R}_z(\psi_i) \in SO(3)$ is the associated rotation matrix around the z-axis, with $SO(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times3} | \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}_3, \det(\boldsymbol{R}) = 1\}$. Additionally, we assume $\boldsymbol{v}_i \in \Upsilon_i \subset \mathbb{R}^3$ (and $\omega_i \in \mathcal{W}_i \subset \mathbb{R}$), where Υ_i (and $\mathcal{W}_i$) is a compact set, and both $\boldsymbol{v}_i$ and ω_i are continuously differentiable.

Each robot is assumed to be capable of exchanging data over a communication channel and to have a camera sensor and a sensor providing relative distance measurements (e.g., an RGB-D camera). These sensors combined provide the relative position of robot j in the i-th robot body-frame:

$${}^i\boldsymbol{p}_{ij} := \boldsymbol{R}_i^\top (\boldsymbol{p}_j - \boldsymbol{p}_i) \tag{8.2}$$

We model the interactions among the robots using a *directed sensing* graph $\mathcal{G}_s := (\mathcal{V}, \mathcal{E}_s)$ and an *undirected communication* graph $\mathcal{G}_c := (\mathcal{V}, \mathcal{E}_c)$, where $\mathcal{V} = \{0, 1, ..., N-1\}$ is the vertex set and $\mathcal{E}_* \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. A directed sensing edge $e_{sk} = (i, j) \in \mathcal{E}_s$ implies that robot i can sense robot j, the opposite is not necessarily true. This assumption is motivated by the fact that employing a camera sensor naturally induces a directed sensing graph due to FoV limitations. We consider the communication graph to correspond to the undirected counterpart of the sensing graph, hence, if robot i can measure robot j then the two robots can communicate in a bidirectional way and there exist both the edges $e_{ck} = (i, j) \in \mathcal{E}_c$ and $e_{ck} = (j, i) \in \mathcal{E}_c$.

8.2 Observer design

For robots to be able to cooperate, it is of paramount importance to be localized one relative to the other. Furthermore, it is also highly desirable to share a common frame in

which to express shared physical quantities to facilitate the exchange of locally available information. In this section, we present a *cascaded observer* designed to estimate the position and orientation of robots within a common frame.

The proposed observer is composed by the following three systems in cascade: (i) an adaptive observer which is used to estimate the relative yaw orientation $\psi_{ij} := \psi_j - \psi_i$, among each couple of neighboring robots from the body-frame relative position measurements, (ii) an observer which uses the estimated relative yaws to estimate the yaw ψ_i of each robot in a common frame and (iii) an observer which uses the estimated yaw measurements in a common frame and the body-frame position measurements to estimate the position $\boldsymbol{p}_i$ of each robot in a common frame. The stability of each observer and of their interconnection is studied.

8.2.1 Relative state observer

In this section, we formulate the adaptive observer used to estimate the relative yaw orientation among each pair of neighboring robots. First of all, we consider the relative state among two robots given by $({}^i\boldsymbol{p}_{ij}, \psi_{ij})$ and we write down the relative state dynamics:

$$\begin{aligned} {}^i\dot{\boldsymbol{p}}_{ij} &= -\boldsymbol{v}_i - \boldsymbol{S}_{e_3}\omega_i {}^i\boldsymbol{p}_{ij} + {}^i\boldsymbol{R}_j\boldsymbol{v}_j \\ \dot{\psi}_{ij} &= \omega_{ij} \end{aligned} \tag{8.3}$$

where $\boldsymbol{S}_{e_3} := [\boldsymbol{e}_3]_\times$[1], ${}^i\boldsymbol{R}_j := \boldsymbol{R}_i^\top \boldsymbol{R}_j = \boldsymbol{R}_z(\psi_{ij})$ and $\omega_{ij} := \omega_j - \omega_i$.

We point out that ${}^i\boldsymbol{p}_{ij}$ is directly measured, $\boldsymbol{v}_j$ and ω_j can be communicated, the only unknown is ψ_{ij}, which affects the dynamics of ${}^i\boldsymbol{p}_{ij}$. The idea is then to reformulate the dynamics of ${}^i\boldsymbol{p}_{ij}$ for the relative orientation to appear in a more convenient way by changing parameterization so that we can formulate an adaptive observer. In particular, we define $:= \begin{bmatrix}\cos(\psi_{ij}) & \sin(\psi_{ij})\end{bmatrix}^\top \in \mathbb{S}^1$, which is a representation of the relative orientation among the two robots using sphere coordinates, with $\mathbb{S}^d := \{\boldsymbol{x} \in \mathbb{R}^{d+1} | \, \|\boldsymbol{x}\| = 1\}$.

We can now rewrite the relative position dynamics as:

$${}^i\dot{\boldsymbol{p}}_{ij} = -\boldsymbol{v}_i - \boldsymbol{S}_{e_3}\omega_i {}^i\boldsymbol{p}_{ij} + v_{jz}\boldsymbol{e}_3 + \bar{\boldsymbol{V}}_j \tag{8.4}$$

with

$$\bar{\boldsymbol{V}}_j(t) := \begin{bmatrix} v_{jx}(t) & -v_{jy}(t) \\ v_{jy}(t) & v_{jx}(t) \\ 0 & 0 \end{bmatrix} \tag{8.5}$$

1. $[\cdot]_\times$ is defined such that $[\boldsymbol{x}]_\times \boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$ is the vector cross product.

and v_{jx}, v_{jy} and v_{jz} representing respectively the x, y and z components of $\boldsymbol{v}_j$. It is worth noting that, as $\bar{\boldsymbol{V}}_j$ is a linear transformation of $\boldsymbol{v}_j$, and given the compactness of Υ_j and the continuous differentiability of $\boldsymbol{v}_j$, $\bar{\boldsymbol{V}}_j$ is uniformly bounded and globally Lipschitz over Υ_j. Consequently, both $\left\|\bar{\boldsymbol{V}}_j(t)\right\|$ and $\left\|\dot{\bar{\boldsymbol{V}}}_j(t)\right\|$ are uniformly bounded. Contrary to the basic adaptive setup, the unknown is not constant, but its dynamics are given by

$$\dot{\boldsymbol{z}}_{\psi_{ij}} = \begin{bmatrix} -\omega_{ij}\sin(\omega_{ij}t) \\ \omega_{ij}\cos(\omega_{ij}t) \end{bmatrix} = \boldsymbol{\Omega}_{ij} \tag{8.6}$$

with skew-symmetric $\boldsymbol{\Omega}_{ij} := \begin{bmatrix} 0 & -\omega_{ij} \\ \omega_{ij} & 0 \end{bmatrix}$.

Then, we formulate an adaptive observer as follows:

$$\begin{aligned} {}^{i}\dot{\hat{\boldsymbol{p}}}_{ij} &= -\boldsymbol{v}_i - \boldsymbol{S}_{e_3}\omega_i {}^{i}\boldsymbol{p}_{ij} + v_{jz}\boldsymbol{e}_3 + \bar{\boldsymbol{V}}_j - k_{pe}\,{}^{i}\boldsymbol{e}_{p_{ij}} \\ \dot{\hat{\boldsymbol{z}}}_{\psi_{ij}} &= \boldsymbol{\Omega}_{ij} - k_\psi \bar{\boldsymbol{V}}_j^{\top i}\boldsymbol{e}_{p_{ij}} \end{aligned} \tag{8.7}$$

with ${}^{i}\boldsymbol{e}_{p_{ij}} = {}^{i}\hat{\boldsymbol{p}}_{ij} - {}^{i}\boldsymbol{p}_{ij}$ and $k_{pe}, k_\psi > 0$.

Moreover, if both the directed edges (i,j) and (j,i) exist, the update of the relative yaw can benefit from both edges. One can consider that $\boldsymbol{z}_{\psi_{ji}} = \boldsymbol{Z}$, with $\boldsymbol{Z} := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. Then, the update for the bidirectional edges can be written as:

$$\begin{aligned} {}^{i}\dot{\hat{\boldsymbol{p}}}_{ij} &= -\boldsymbol{v}_i - \boldsymbol{S}_{e_3}\omega_i {}^{i}\boldsymbol{p}_{ij} + v_{jz}\boldsymbol{e}_3 + \bar{\boldsymbol{V}}_j - k_{pe}\,{}^{i}\boldsymbol{e}_{p_{ij}} \\ {}^{j}\dot{\hat{\boldsymbol{p}}}_{ji} &= -\boldsymbol{v}_j - \boldsymbol{S}_{e_3}\omega_j {}^{j}\boldsymbol{p}_{ji} + v_{iz}\boldsymbol{e}_3 + \bar{\boldsymbol{V}}_i\boldsymbol{Z} - k_{pe}\,{}^{j}\boldsymbol{e}_{p_{ji}} \\ \dot{\hat{\boldsymbol{z}}}_{\psi_{ij}} &= \boldsymbol{\Omega}_{ij} - k_\psi\left(\bar{\boldsymbol{V}}_j^{\top i}\boldsymbol{e}_{p_{ij}} + \boldsymbol{Z}\bar{\boldsymbol{V}}_i^{\top j}\boldsymbol{e}_{p_{ji}}\right) \end{aligned} \tag{8.8}$$

Remark 13. *We point out that, from two reciprocal measurements, if $\boldsymbol{p}_j - \boldsymbol{p}_j$ is not aligned with the z-axis, it is also possible to algebraically compute the relative yaw among the two robots. However, since, in practice, measurements are affected by noise, we avoid the algebraic computation.*

In the following stability analysis, we consider the case of a single directed edge, but the proof for the bidirectional edge follows the same lines.

We define the yaw error $\boldsymbol{e}_{\psi_{ij}} := -$, where $\boldsymbol{e}_{\psi_{ij}}$ is the chordal distance on $\mathbb{S}^1$. Then, the error dynamics are described by the following linear time-varying system:

$$\begin{bmatrix} {}^{i}\dot{\boldsymbol{e}}_{p_{ij}} \\ \dot{\boldsymbol{e}}_{\psi_{ij}} \end{bmatrix} = \begin{bmatrix} -k_{pe}\boldsymbol{I}_3 & \bar{\boldsymbol{V}}_j \\ -k_\psi \bar{\boldsymbol{V}}_j^{\top} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} {}^{i}\boldsymbol{e}_{p_{ij}} \\ \boldsymbol{e}_{\psi_{ij}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Omega}_{ij}\boldsymbol{e}_{\psi_{ij}} \end{bmatrix} \tag{8.9}$$

Theorem 4. *Since* $\left\|\bar{\boldsymbol{V}}_j(t)\right\|$ *and* $\left\|\dot{\bar{\boldsymbol{V}}}_j(t)\right\|$ *are uniformly bounded and assuming that, there exists* $T, \mu > 0$*, such that* $\bar{\boldsymbol{V}}_j(t)$*, as given by* (8.5)*, satisfies the persistency of excitation condition*

$$\int_t^{t+T} \bar{\boldsymbol{V}}_j^\top(\tau)\bar{\boldsymbol{V}}_j(\tau)\, d\tau \succeq \mu \boldsymbol{I}_2 \quad \forall t \geq 0, \tag{8.10}$$

then, $(\boldsymbol{e}_{p_{ij}}, \boldsymbol{e}_{\psi_{ij}}) = (\boldsymbol{0}, \boldsymbol{0})$ *is a globally exponentially stable equilibrium point of* (8.9)*.*

Proof. The proof uses conventional adaptive control reasoning [Besançon, 2000; Marino, 1995]. We outline the main steps demonstrating that the dynamics of do not impact the convergence proof. Consider the quadratic Lyapunov function $V({}^i\boldsymbol{e}_{p_{ij}}, \boldsymbol{e}_{\psi_{ij}}) := \frac{1}{2}{}^i\boldsymbol{e}_{p_{ij}}^\top {}^i\boldsymbol{e}_{p_{ij}} + \frac{1}{2k_\psi}\boldsymbol{e}_{\psi_{ij}}^\top \boldsymbol{e}_{\psi_{ij}}$. Its derivative is given by

$$\begin{aligned}\dot{V} &= {}^i\boldsymbol{e}_{p_{ij}}^\top \bar{\boldsymbol{V}}_j \boldsymbol{e}_{\psi_{ij}} - k_{pe}\left\|{}^i\boldsymbol{e}_{p_{ij}}\right\|^2 - \boldsymbol{e}_{\psi_{ij}}^\top \bar{\boldsymbol{V}}_j^\top {}^i\boldsymbol{e}_{p_{ij}} \\ &+ \frac{1}{k_\psi}\boldsymbol{e}_{\psi_{ij}}^\top \boldsymbol{\Omega}_{ij}\boldsymbol{e}_{\psi_{ij}} = -k_{pe}\left\|{}^i\boldsymbol{e}_{p_{ij}}\right\|^2 \leq 0\end{aligned} \tag{8.11}$$

where we used the fact that $\boldsymbol{\Omega}_{ij}$ is skew-symmetric. The negative semi-definiteness of $\dot{V}$, implies uniform stability of the origin. Leveraging Barbalat's lemma demonstrates convergence to zero of $\boldsymbol{e}_{p_{ij}}$. The PE condition (8.10) ensures convergence to zero of the error $\boldsymbol{e}_{\psi_{ij}}$. Global exponential stability follows from the linear time-varying nature of the system (see the Persistency of Excitation lemma in [Marino, 1995]). □

Remark 14. *Condition* (8.10) *is satisfied if the velocity on the xy-plane is non-zero for a time interval of non-zero measure over each period* T*. In fact,*

$$\bar{\boldsymbol{V}}_j(t)^\top \bar{\boldsymbol{V}}_j(t) = \begin{bmatrix} v_{jx}^2(t) + v_{jy}^2(t) & 0 \\ 0 & v_{jx}^2(t) + v_{jy}^2(t) \end{bmatrix}, \tag{8.12}$$

which is clearly positive definite if the velocity on the xy-plane is non-zero.

Remark 15. *We point out that, given the update law* (8.8) *for , in presence of PE, the error* $\boldsymbol{e}_{\psi_{ij}}$ *will asymptotically go to zero, but the norm of will not necessarily be* 1 *at each instant, meaning that it will not represent a valid orientation. One can, anyway, extract the closest angle* ψ_{ij} *representing the orientation after renormalizing .*

8.2.2 Common Frame Yaw Observer

In the preceding section, we introduced an observer able to estimate the relative orientation among neighboring robots. While knowledge of their relative orientation allows for

coordination between two robots, it often falls short, necessitating awareness of position and orientation within a shared frame. Among its advantages, for instance, is the ability to adapt to a time-varying graph; upon the creation of a new edge, two robots can initialize their relative orientation in an informed way.

To estimate the yaw orientations in a common frame, we leverage the algorithm proposed in [B.-H. Lee & Ahn, 2016b], designed for estimating the yaw within a shared frame based on relative yaw measurements. In our context, we substitute the relative yaw measurements with estimated relative yaw orientations obtained using (8.8) and, moreover, we consider time-varying orientations.

In the design of this observer, we represent the orientation of the i-th robot as a *unit* complex number $z_i := e^{\iota\psi_i} \in \mathbb{C}$, where $\iota = \sqrt{-1}$, and, similarly, the relative orientation between two robots as $z_{ij} := e^{\iota\psi_{ij}} = \begin{bmatrix} 1 & \iota \end{bmatrix} \boldsymbol{z}_{\psi_{ij}} \in \mathbb{C}$. While equivalent to the representation used in the previous section, we adopt this representation for analytical simplicity of the proof and coherence with [B.-H. Lee & Ahn, 2016b].

The objective of the observer is articulated as follows: for a common complex value $\alpha \in \mathbb{C}$ and robot orientations $z_i(t)$, devise an estimation law such that $\angle\hat{z}_i \to \angle z_i + \angle\alpha$ $\forall i$ as $t \to \infty$, i.e. the estimated robot orientations should converge to the real ones up to a common rotation.

The proposed estimation law takes the form:

$$\dot{\hat{z}}_i = \iota\omega_i\hat{z}_i + k_z \sum_{j\in\mathcal{N}_i} (\bar{z}_{ij}\hat{z}_j - \hat{z}_i) \tag{8.13}$$

where $k_z > 0$ and $\bar{z}_{ij}$ represents the complex conjugate of z_{ij}. It differs with respect to [B.-H. Lee & Ahn, 2016b] for the fact that the orientation of the robots is time-varying. Defining $\hat{\boldsymbol{z}} := \begin{bmatrix} \hat{z}_1, ..., \hat{z}_N \end{bmatrix}^\top$ and $\boldsymbol{\omega} := \begin{bmatrix} \omega_1, ..., \omega_N \end{bmatrix}^\top$, the observer dynamics can be written in matrix form as:

$$\dot{\hat{\boldsymbol{z}}} = \iota\,\mathrm{diag}(\boldsymbol{\omega})\hat{\boldsymbol{z}} - k_z\boldsymbol{L}_c\hat{\boldsymbol{z}} = (\iota\,\mathrm{diag}(\boldsymbol{\omega}) - k_z\boldsymbol{L}_c)\hat{\boldsymbol{z}} \tag{8.14}$$

where

$$(\boldsymbol{L}_c)_{ij} := \begin{cases} |\mathcal{N}_i| & \text{if} \quad i = j \\ -\bar{z}_{ij} & \text{if} \quad j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{8.15}$$

and $|\mathcal{N}_i|$ is the cardinality of the neighbors set of the i-th robot.

Now, we present the following Proposition, introduced in [B.-H. Lee & Ahn, 2016b] and, to better understand the following developments, we provide a sketch of the proof.

Proposition 3 ([B.-H. Lee & Ahn, 2016b])**.** *Zero is a simple eigenvalue of* $\boldsymbol{L}_c(t)$ *with a corresponding eigenvector* $\boldsymbol{z}(t)$ *if and only if the associated digraph has a spanning tree. Moreover, every eigenvalue, except for the zero eigenvalue, has strictly negative real part.*

Define the matrix $\boldsymbol{D}_z := \text{diag}(z_1, ..., z_N) \in \mathbb{C}^{N\times N}$ [2]. Notice that, since z_i is a unit complex number, then $|z_i| = 1$. Moreover, $\boldsymbol{D}_z$ is a nonsingular matrix and its inverse is $\boldsymbol{D}_z^{-1} = \boldsymbol{D}_z^H$, where we indicated with H the conjugate transpose. As $\boldsymbol{D}_z$ is a nonsingular matrix, we define the similarity transformation $\boldsymbol{L} = \boldsymbol{D}_z^H \boldsymbol{L}_c \boldsymbol{D}_z$. Since $\boldsymbol{D}_z$ is a diagonal matrix, the *off-diagonal* entries of $\boldsymbol{L}$ are:

$$\begin{aligned}(\boldsymbol{L})_{ij} &= \left(\boldsymbol{D}_z^H\right)_{ii} (\boldsymbol{L}_c)_{ij} (\boldsymbol{D}_z)_{jj} = -e^{-\iota\psi_i} e^{-\iota\psi_{ij}} e^{\iota\psi_j} \\ &= -e^{\iota(-\psi_i - \psi_{ij} + \psi_j)} = -e^{\iota 0} = -1\end{aligned} . \tag{8.16}$$

In contrast, the diagonal entries are invariant under this transformation (i.e. $e^{-\iota\psi_i}|\mathcal{N}_i|e^{\iota\psi_i} = |\mathcal{N}_i|$). This demonstrates that, $\boldsymbol{L}$ is the classical unweighted Laplacian matrix, which, as well-known [Bullo, 2020; Mesbahi & Egerstedt, 2010], is positive semi-definite and if the graph has a spanning tree, then 0 is a simple eigenvalue corresponding to the right eigenvector $\boldsymbol{1}_N = \left[1, ..., 1\right]^\top \in \mathbb{R}^N$. Since $\boldsymbol{L}_c$ and $\boldsymbol{L}$ are similar matrices, then zero is a simple eigenvalue of $\boldsymbol{L}_c$ as well. Furthermore, since $\boldsymbol{0} = \boldsymbol{L}\boldsymbol{1}_N = \boldsymbol{D}_z^H \boldsymbol{L}_c \boldsymbol{D}_z \boldsymbol{1}_N = \boldsymbol{D}_z^H \boldsymbol{L}_c \boldsymbol{z}$, it follows that, $\boldsymbol{z}$ is the right eigenvector associated to the zero eigenvalue.

Remark 16. *From* (8.8)*, each robot gets an estimate of the relative orientation with respect to both the in- and out- neighboring robots in the directed graph* $\mathcal{G}_s$*. Hence, the Laplacian* $\boldsymbol{L}$ *corresponds to the undirected graph* $\mathcal{G}_c$ *and the condition for zero to be a simple eigenvalue of* $\boldsymbol{L}_c$ *reduces to the graph* $\mathcal{G}_c$ *being connected.*

The following theorem extends the one from [B.-H. Lee & Ahn, 2016b] to the case in which the yawrate of the robots may be different from zero.

Theorem 5. *The observer* (8.14) *globally exponentially converges to* $\hat{\boldsymbol{z}}(t) = (\boldsymbol{z}^H(0)\hat{\boldsymbol{z}}(0)/N)\boldsymbol{z}(t)$*, if and only if the graph is connected.*

Proof. Consider the change of coordinates $\hat{\boldsymbol{z}} = \boldsymbol{D}_z \boldsymbol{y}$. Notice that, $y_i = \bar{z}_i \hat{z}_i$ corresponds to the rotation error on $\mathbb{S}^1$, where $y_i = 1$ represents the identity rotation $z_i = \hat{z}_i$. Unlike [B.-H. Lee & Ahn, 2016b], we consider non-zero yaw rates for the robots, resulting in (i)

2. diag$(\cdot)$ represents a diagonal matrix of the input arguments and $(\cdot)$ represents a block diagonal matrix of the input arguments

an additional term in the dynamics (8.14) and (*ii*) a time-varying change-of-coordinates matrix. Consequently:

$$\begin{aligned}\dot{\hat{\boldsymbol{z}}} &= \boldsymbol{D}_z\dot{\boldsymbol{y}} + \dot{\boldsymbol{D}}_z\boldsymbol{y} = \boldsymbol{D}_z\dot{\boldsymbol{y}} + \iota\,\mathrm{diag}(\boldsymbol{\omega})\,\mathrm{diag}(\boldsymbol{z})\boldsymbol{y} \\ &= \boldsymbol{D}_z\dot{\boldsymbol{y}} + \iota\,\mathrm{diag}(\boldsymbol{\omega})\boldsymbol{D}_z\boldsymbol{y}\end{aligned}. \tag{8.17}$$

From (8.17) and (8.14)

$$\begin{aligned}\dot{\boldsymbol{y}} &= -k_z\boldsymbol{D}_z^H\boldsymbol{L}_c\boldsymbol{D}_z\boldsymbol{y} + \iota\boldsymbol{D}_z^H\,\mathrm{diag}(\boldsymbol{\omega})\boldsymbol{D}_z\boldsymbol{y} - \iota\boldsymbol{D}_z^H\,\mathrm{diag}(\boldsymbol{\omega})\boldsymbol{D}_z\boldsymbol{y} \\ &= -k_z\boldsymbol{L}\boldsymbol{y}\end{aligned} \tag{8.18}$$

for which the consensus subspace $\{y_1 = y_2 = ... = y_N\}$ is a globally exponentially stable equilibrium set [Moreau, 2004]. In particular, since $\boldsymbol{L}$ is a symmetric Laplacian, $\boldsymbol{y}(t) \to (\mathbf{1}_N^T\boldsymbol{y}(0)/N)\mathbf{1}_N$, which corresponds to the same rotation error for each angle $\hat{z}_i$, i.e. $\mathbf{1}_N^T\boldsymbol{y}(0)/N = \alpha$. By applying again the change of coordinates:

$$\begin{aligned}&\hat{\boldsymbol{z}}(t) \to \boldsymbol{D}_z(t)\left(\frac{1}{N}\mathbf{1}_N^T\boldsymbol{y}(0)\right)\mathbf{1}_N \\ &= \left(\frac{1}{N}\mathbf{1}_N^T\boldsymbol{D}_z(0)^H\hat{\boldsymbol{z}}(0)\right)\boldsymbol{D}_z(t)\mathbf{1}_N = \left(\frac{1}{N}\boldsymbol{z}^H(0)\hat{\boldsymbol{z}}(0)\right)\boldsymbol{z}(t)\end{aligned} \tag{8.19}$$

We also note that, $\hat{\boldsymbol{z}}(t) = \mathbf{0}$ is an undesired equilibrium of the system, as it does not represent a valid orientation. But $\hat{\boldsymbol{z}}(t) \to \mathbf{0}$ if and only if $\boldsymbol{z}^H(0)\hat{\boldsymbol{z}}(0) = 0$. □

The stability proof in Theorem 5 relies on Proposition 3, assuming perfect measurements z_{ij}. However, using $\hat{z}_{ij}$ from (8.8) in place of z_{ij} may render the estimated relative orientations unrealizable. Consequently, the existence of a $\boldsymbol{z}^* := \begin{bmatrix} z_1^* & \dots & z_N^* \end{bmatrix}^\top \in N$ compatible with all estimated $\hat{z}_{ij}$ $\forall(i,j) \in \mathcal{E}$ is not guaranteed. Specifically, within a graph containing cycles, while (8.16) may hold for edges within a spanning tree, this might not hold for the remaining edges outside the spanning tree.

Let us denote $\hat{\boldsymbol{L}}_c$ as the matrix derived similarly to (8.15) but using the estimated $\hat{z}_{ij}$ and $\hat{\boldsymbol{L}} := \boldsymbol{D}_{z^*}^H\hat{\boldsymbol{L}}_c\boldsymbol{D}_{z^*}$, in which $\boldsymbol{D}_{z^*} := \mathrm{diag}(z_1^*, ..., z_N^*)$. For edges within cycles, instead of (8.16), we might encounter $\left(\boldsymbol{D}_{z^*}^H\hat{\boldsymbol{L}}_c\boldsymbol{D}_{z^*}\right)_{ij} = -e^{\iota\tilde{\psi}_{ij}}$ where $\tilde{\psi}_{ij} = \psi_j^* - \psi_i^* - \hat{\psi}_{ij}$, with $\psi_i^* = \angle z_i^*$ and $\psi_j^* = \angle z_j^*$. Consequently,

$$\left(\hat{\boldsymbol{L}}\right)_{ij} := \begin{cases} |\mathcal{N}_i| & \text{if} \quad i = j \\ -1 & \text{if} \quad (i,j) \in \mathcal{G}_\tau \\ -e^{\iota\tilde{\psi}_{ij}} & \text{if} \quad (i,j) \in \mathcal{G}_c \\ 0 & \text{otherwise} \end{cases} \tag{8.20}$$

where we used $\mathcal{G}_\tau$ and $\mathcal{G}_c$ to denote respectively a subgraph of $\mathcal{G}_c$ representing the considered spanning tree and a subgraph representing the remaining cycle edges. As $e^{\iota\tilde{\psi}_{ij}} = e^{-\iota\tilde{\psi}_{ji}}$ and $|e^{\iota\tilde{\psi}_{ij}}| = 1$, $\hat{\boldsymbol{L}}$ emerges as a Hermitian weakly diagonally dominant matrix with positive elements on the diagonal, establishing its positive semidefiniteness. Consequently, (8.18), and hence (8.14), remain stable also using imperfect measurements $\hat{z}_{ij}$.

8.2.3 Position Observer

In this section, we design an observer for the position of each robot within a common frame. This observer is added in cascade with respect to (8.14). The primary objective of this observer can be defined as follows: for a common translation $\bar{\boldsymbol{p}} \in \mathbb{R}^3$, a common rotation $\boldsymbol{R}_\alpha$ (where $\alpha \in \mathbb{C}$ remains consistent with the preceding section), alongside the real position $\boldsymbol{p}_i$ and orientation $\angle z_i$, the aim is to devise an estimation law ensuring that $\hat{\boldsymbol{p}}_i \to \bar{\boldsymbol{p}} + \boldsymbol{R}_\alpha^\top \boldsymbol{p}_i \ \forall i$ as $t \to \infty$

Each robot implements the following update law:

$$\begin{aligned}\dot{\hat{\boldsymbol{p}}}_i = \hat{\boldsymbol{R}}_i \boldsymbol{v}_i - k_p \Big[&\sum_{j \in \mathcal{N}_i^{\text{out}}} (\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_j + \hat{\boldsymbol{R}}_i{}^i \boldsymbol{p}_{ij}) \\ &+ \sum_{j \in \mathcal{N}_i^{\text{in}}} (\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_j - \hat{\boldsymbol{R}}_j{}^j \boldsymbol{p}_{ji}) \Big]\end{aligned} \tag{8.21}$$

where $k_p > 0$, $\mathcal{N}_i^{\text{in}}$ and $\mathcal{N}_i^{\text{out}}$ represent the *in-* and *out-neighbors* of the i-th robot, while $\hat{\boldsymbol{R}}_i$ and $\hat{\boldsymbol{R}}_j$ are obtained from observer (8.14) and the corresponding angle communicated among neighboring robots.

Let us denote $\bar{\boldsymbol{R}} := (\{\boldsymbol{R}_i\}_{i=1}^N) \in \mathbb{R}^{3N \times 3N}$ and $\bar{\boldsymbol{R}}_\mathcal{E} := (\{\boldsymbol{R}_k\}_{k=1}^{|\mathcal{E}|}) \in \mathbb{R}^{3|\mathcal{E}_s| \times 3|\mathcal{E}_s|}$, where $\boldsymbol{R}_k$ is the rotation matrix corresponding to the *initial node* of the edge $e_k := (i, j)$ and we use the 'hat' to indicate the estimated counterpart. The dynamics of the observer can be expressed in matrix form as follows:

$$\dot{\hat{\boldsymbol{p}}} = \hat{\bar{\boldsymbol{R}}} \boldsymbol{v} - k_p \boldsymbol{E}_3 \left(\boldsymbol{E}_3^\top \hat{\boldsymbol{p}} - \hat{\bar{\boldsymbol{R}}}_\mathcal{E}{}^b \boldsymbol{p}_\mathcal{E} \right) \tag{8.22}$$

where $\boldsymbol{E}_3 := \boldsymbol{E} \otimes \boldsymbol{I}_3$ is the incidence matrix of the directed sensing graph $\mathcal{G}_s$, $\otimes$ is the Kronecker product and ${}^b\boldsymbol{p}_\mathcal{E} := \bar{\boldsymbol{R}}_\mathcal{E}^\top \boldsymbol{E}_3^\top \boldsymbol{p}$. Similarly, we define for later use $\boldsymbol{p}_\mathcal{E} := \boldsymbol{E}_3^\top \boldsymbol{p}$.

Notice that, after convergence of the yaw observer, it holds $\hat{\boldsymbol{R}}_i = \boldsymbol{R}_\alpha^\top \boldsymbol{R}_i$ and, by denoting $\bar{\boldsymbol{R}}_\alpha := \boldsymbol{I}_N \otimes \boldsymbol{R}_\alpha$, it follows:

$$\hat{\bar{\boldsymbol{R}}}_\mathcal{E}{}^b \boldsymbol{p}_\mathcal{E} = \bar{\boldsymbol{R}}_\alpha^\top \bar{\boldsymbol{R}}_\mathcal{E} \bar{\boldsymbol{R}}_\mathcal{E}^\top \boldsymbol{p}_\mathcal{E} = \bar{\boldsymbol{R}}_\alpha^\top \boldsymbol{E}_3^\top \boldsymbol{p}. \tag{8.23}$$

Before proceiding with the main theorem, we define some quantities which will be used later. Let us denote $\bar{\boldsymbol{L}}_3 := \bar{\boldsymbol{L}} \otimes \boldsymbol{I}_3 = \boldsymbol{E}_3 \boldsymbol{E}_3^\top$, where the Laplacian matrix $\bar{\boldsymbol{L}}$ has components:

$$\left(\bar{\boldsymbol{L}}\right)_{ij} := \begin{cases} |\mathcal{N}_i^{\text{in}}| + |\mathcal{N}_i^{\text{out}}| & \text{if} \quad i = j \\ 2 & \text{if} \quad j \in \mathcal{N}_i^{\text{in}} \wedge j \in \mathcal{N}_i^{\text{out}} \\ 1 & \text{if} \quad j \in \mathcal{N}_i^{\text{in}} \oplus j \in \mathcal{N}_i^{\text{out}} \\ 0 & \text{otherwise} \end{cases} \tag{8.24}$$

with $\oplus$ denoting the exclusive disjunction. Also, define the yaw estimation error $\boldsymbol{e}_\psi := \hat{\boldsymbol{z}}_\psi - \boldsymbol{z}_{\alpha\psi}$, with $\hat{\boldsymbol{z}}_\psi := \left[\cos(\hat{\psi}_1), \sin(\hat{\psi}_1), ..., \cos(\hat{\psi}_N), \sin(\hat{\psi}_N)\right]^\top$ and $\boldsymbol{z}_{\alpha\psi} := \left[\cos(\psi_1 - \alpha), \sin(\psi_1 - \alpha), ..., \cos(\psi_N - \alpha), \sin(\psi_N - \alpha)\right]^\top$, as well as $\bar{\boldsymbol{R}}_{\alpha\mathcal{E}} := \boldsymbol{I}_{|\mathcal{E}|} \otimes \boldsymbol{R}_\alpha$.

Theorem 6. *Consider the estimation error up to a common rotation $\boldsymbol{e}_p := \hat{\boldsymbol{p}} - \bar{\boldsymbol{R}}_\alpha^\top \boldsymbol{p}$ and the observer dynamics* (8.22)*. Under the assumptions that $\mathcal{G}_c$ is connected and the inputs, as well as the measurements* (8.2)*, are uniformly bounded and continuously differentiable, then, the set $\mathcal{S} := \{\boldsymbol{e}_p | \boldsymbol{e}_{p1} = \boldsymbol{e}_{p2} = ... = \boldsymbol{e}_{pN}\}$ is input-to-state stable (ISS) with respect to the yaw error $\boldsymbol{e}_\psi$.*

Proof. The error dynamics can be written as:

$$\begin{aligned} \dot{\boldsymbol{e}}_p &= (\hat{\bar{\boldsymbol{R}}} - \bar{\boldsymbol{R}}_\alpha^\top \bar{\boldsymbol{R}})\boldsymbol{v} - k_p \boldsymbol{E}_3 \left(\boldsymbol{E}_3^\top \hat{\boldsymbol{p}} - \hat{\bar{\boldsymbol{R}}}_{\mathcal{E}}{}^b \boldsymbol{p}_{\mathcal{E}}\right) \\ &\quad + k_p \boldsymbol{E}_3 \bar{\boldsymbol{R}}_{\alpha\mathcal{E}}^\top \bar{\boldsymbol{R}}_{\mathcal{E}}{}^b \boldsymbol{p}_{\mathcal{E}} - k_p \boldsymbol{E}_3 \bar{\boldsymbol{R}}_{\alpha\mathcal{E}}^\top \bar{\boldsymbol{R}}_{\mathcal{E}}{}^b \boldsymbol{p}_{\mathcal{E}} \\ &= -k_p \bar{\boldsymbol{L}}_3 \boldsymbol{e}_p + (\hat{\bar{\boldsymbol{R}}} - \bar{\boldsymbol{R}}_\alpha^\top \bar{\boldsymbol{R}})\boldsymbol{v} + k_p \boldsymbol{E}_3 \left(\hat{\bar{\boldsymbol{R}}}_{\mathcal{E}} - \bar{\boldsymbol{R}}_{\alpha\mathcal{E}}^\top \bar{\boldsymbol{R}}_{\mathcal{E}}\right)^b \boldsymbol{p}_{\mathcal{E}} \\ &= -k_p \bar{\boldsymbol{L}}_3 \boldsymbol{e}_p + \left(\{\bar{\boldsymbol{V}}_i\} + k_p \boldsymbol{E}_3 \{{}^i\bar{\boldsymbol{P}}_{ij}\} \boldsymbol{E}_{2\otimes}^\top\right) \boldsymbol{e}_\psi \\ &= -k_p \bar{\boldsymbol{L}}_3 \boldsymbol{e}_p + \boldsymbol{B} \boldsymbol{e}_\psi \end{aligned} \tag{8.25}$$

where, in the second line, we added and subtracted $k_p \boldsymbol{E}_3 \bar{\boldsymbol{R}}_{\alpha\mathcal{E}}^\top \bar{\boldsymbol{R}}_{\mathcal{E}}{}^b \boldsymbol{p}_{\mathcal{E}}$, in the third equality, we used an analogous transformations to (8.5), i.e.

$${}^i\bar{\boldsymbol{P}}_{ij}(t) := \begin{bmatrix} {}^i p_{ijx}(t) & -{}^i p_{ijy}(t) \\ {}^i p_{ijy}(t) & {}^i p_{ijx}(t) \\ 0 & 0 \end{bmatrix}, \tag{8.26}$$

to obtain linearly the orientation error $\boldsymbol{e}_\psi$. Also, $\boldsymbol{E}_{2\otimes} := \boldsymbol{E}_\otimes \otimes \mathbf{I}_2$, where

$$(\boldsymbol{E}_\otimes)_{ik} := \begin{cases} 1 & \text{if node } i \text{ is the head of edge } e_k \\ 0 & \text{otherwise} \end{cases} \tag{8.27}$$

is the *out-incidence matrix* [Bullo, 2020]. In the last equality, we defined $\boldsymbol{B}(t) := \{\bar{\boldsymbol{V}}_i\}(t) + k_p \boldsymbol{E}_3 \{{}^i\bar{\boldsymbol{P}}_{ij}\}(t) \boldsymbol{E}_{2\otimes}^\top$. Let us denote $\boldsymbol{U} := \boldsymbol{1}_N \otimes \boldsymbol{I}_3$ and let $\boldsymbol{e}_p^\perp := \boldsymbol{e}_p - \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{e}_p$ be the error orthogonal to the consensus subspace $\mathcal{S}$ and $\bar{\boldsymbol{e}}_p := \boldsymbol{U}^\top \boldsymbol{e}_p$ be the average error (i.e. a common translation). Then, the system dynamics can be written as:

$$\dot{\boldsymbol{e}}_p^\perp(t) = -k_p \bar{\boldsymbol{L}}_3 \boldsymbol{e}_p^\perp(t) + (\boldsymbol{I}_{3N} - \boldsymbol{U}\boldsymbol{U}^\top)\boldsymbol{B}(t)\boldsymbol{e}_\psi(t) \tag{8.28}$$

$$\dot{\bar{\boldsymbol{e}}}_p(t) = \boldsymbol{U}^\top \boldsymbol{B}(t)\boldsymbol{e}_\psi(t) \tag{8.29}$$

where we used the fact that $\boldsymbol{U}^\top \bar{\boldsymbol{L}}_3 = \bar{\boldsymbol{L}}_3 \boldsymbol{U} = \boldsymbol{0}$ for undirected graphs [Bullo, 2020]. The unforced system (8.28) (i.e. for $\boldsymbol{e}_\psi(t) = \boldsymbol{0}$) is globally exponentially stable and since $\boldsymbol{B}(t)$ is uniformly bounded and continuously differentiable, then (8.28) is ISS with respect to $\boldsymbol{e}_\psi(t)$ (see [**bullo2022contraction**]). Furthermore, as shown in the previous section, $\boldsymbol{e}_\psi(t)$ is bounded $\forall t \geq 0$ and $\boldsymbol{e}_\psi(t) \to 0$ as $t \to \infty$, implying that $\boldsymbol{e}_\psi(t)$ is integrable, and, as a consequence, $\bar{\boldsymbol{e}}_p(t)$ remains bounded. □

8.3 Simulation Results

This section presents simulation results validating our theoretical findings using a group of 8 quadrotors, whose sensing graph is depicted in Fig. 8.1. The quadrotors execute a

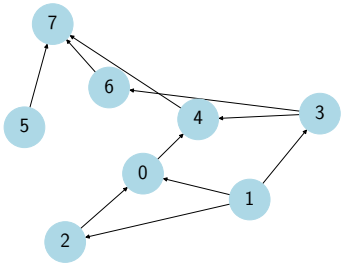


Figure 8.1: Sensing graph

formation control task with a modified version of the control law from [B.-H. Lee & Ahn, 2016b] to suit a directed sensing graph,

$$\boldsymbol{v}_i = k_c k_i \boldsymbol{\mu}_i \tag{8.30}$$

with

$$\boldsymbol{\mu}_i := \sum_{j \in \mathcal{N}_i^{\text{in}}} ({}^i\boldsymbol{p}_{ij} - \hat{\boldsymbol{R}}_i^\top (\boldsymbol{p}_j^* - \boldsymbol{p}_i^*)) + \sum_{j \in \mathcal{N}_i^{\text{out}}} (\hat{\boldsymbol{R}}_j^\top (\boldsymbol{p}_i^* - \boldsymbol{p}_j^*) - {}^j\boldsymbol{p}_{ji}), \tag{8.31}$$

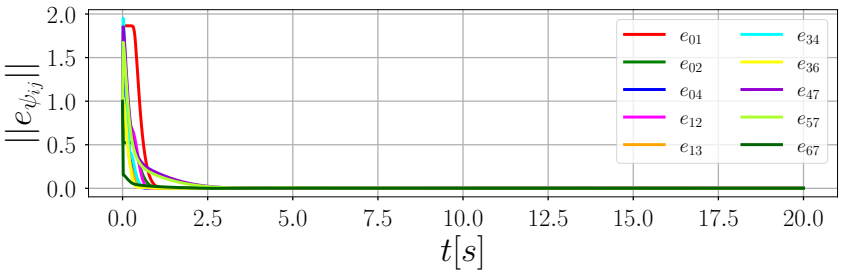
$k_c > 0$ and $k_i = 1$ if $\|\boldsymbol{\mu}_i\| \leq \nu_{\max}$, $k_i = \nu_{\max} / \|\boldsymbol{\mu}_i\|$ otherwise, where $\nu_{\max}$ is the maximum velocity norm.

The yaw is controlled independently using a spline with random coefficients. The initial estimated positions of the drones are drawn from a gaussian distribution centered around the real position of the drones and with standard deviation of $1m$, instead the initial common frame yaw estimate $\hat{\boldsymbol{z}}$ is drawn from a uniform distribution in $(-\pi, \pi]$. The relative yaw estimates $\hat{\boldsymbol{z}}_{\psi_{ij}}$ are initialized based on the initial common frame yaw estimate, ensuring consistency with the graph cycles. Consequently, $\boldsymbol{L}_c$ will be a similar matrix to a Laplacian, with an eigenvalue exactly at 0.

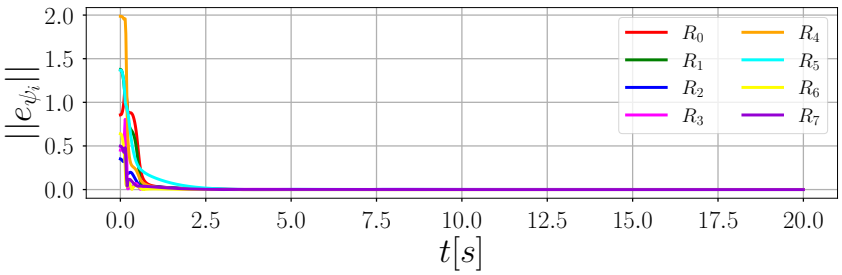
The relative state observer (8.8) is run @200Hz with gains $k_{pe} = 20$ and $k_\psi = 600$, the common frame yaw observer (8.15) @100Hz with gain $k_z = 40$ and the common frame position observer (8.21) @50Hz with gain $k_p = 1$. Results for the estimation of relative yaw, yaw in a common frame, and position in a common frame, as well as the formation error norm are depicted in Fig. 8.2.

8.4 Conclusions

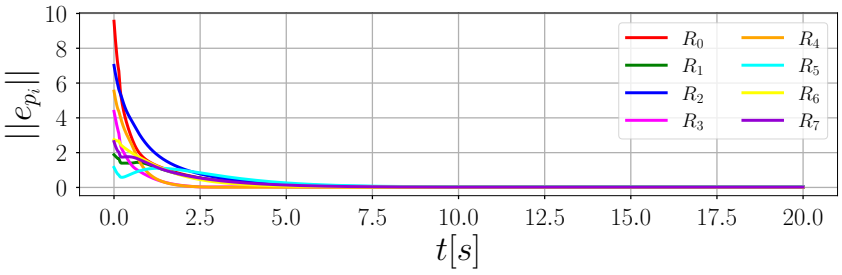
This work presents a novel observer for robots to estimate their position and yaw in a common frame from relative position measurements. We achieve this through a three-part observer that leverages persistency of excitation, eliminating the need for restrictive assumptions about the sensing graph. Future work will focus on extending this approach to bearing measurements and exploring a single-observer implementation.



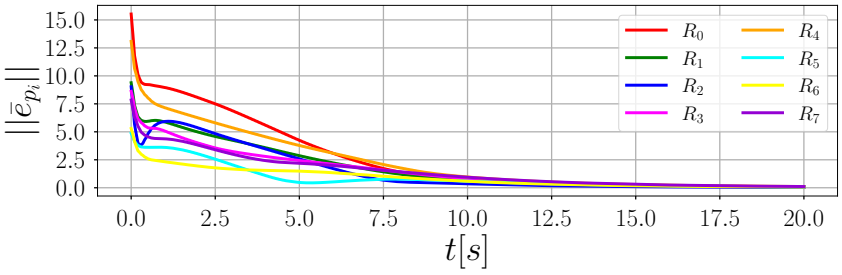
(a)



(b)



(c)



(d)

Figure 8.2: The figures respectively illustrate the following: (a) the norm of the estimation error on the relative yaw for each edge, (b) the norm of the estimation error on the yaw in a common frame for each robot, (c) the norm of the estimation error on the position in a common frame for each robot, and (d) the norm of the formation error for each robot.

Part V

Conclusions and Future Works

CONCLUSIONS AND FUTURE WORKS

In this last chapter, we wish to summarize the main contributions of this thesis highlighting some of the issues that still remain unsolved and suggesting, whenever possible, some directions for improvement and further investigation.

8.5 Summary and contributions

The initial objective of this thesis was to investigate the problem of distributed active sensing for the cooperative localization of quadrotor formations. We set out to address the challenges posed by incomplete relative measurements and the presence of sensing constraints, all while avoiding the strict requirement of infinitesimal rigidity in the formations, thereby granting the robots greater freedom of movement (Chapter 4).

While localization is a necessary condition to achieve a certain task, it is seldom the sole mission objective. Coupling active sensing with the execution of a task of interest is not trivial. Throughout this thesis, the concepts of Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs) emerged as effective tools for integrating active sensing with other tasks. We explored the possibility of encoding the task to be accomplished, such as formation control, using a CLF and imposing the decrease of the CLF as a constraint in a quadratic program. Simultaneously, the framework aimed to enhance as much as possible the system's localization performance by moving the robots along informative trajectories (Chapter 5).

The exploration of such kind of framework, integrating active sensing with other tasks using CLF and CBF, prompted us to investigate a somehow inverse problem. This scenario involves persistently localizing multiple moving targets using a group of flying robots equipped with down-looking cameras. The challenge is to ensure a minimum level of information about the targets while optimizing the accomplishment of another task, exemplified here by a formation control task (Chapter 6).

In the course of implementing persistent target monitoring and connectivity maintenance using CBFs, we identified as a critical issue in certain scenarios the loss of optimality compared to centralized solutions when implementing low-complexity distributed CBF-

based controllers, such as those presented in [Lindemann & Dimarogonas, 2020; Wang et al., 2017]. For example, a pair of flying robots observing two targets going in opposite direction would not immediately take the most reasonable decision of splitting into two groups to follow the two targets due to a sub-optimal approach in distributing the constraint. Two alternatives were proposed in [Tan & Dimarogonas, 2021, 2022], which allow to converge to the optimal centralized solution, but they have limitations, either failing to guarantee constraint satisfaction at all times or imposing overly restrictive constraints on the local lie derivative L_{G_i} of the constraint. In this thesis, we introduce an approach building upon [Tan & Dimarogonas, 2021] that overcomes the limitations related to L_{G_i}, providing a solution that ensures constraint satisfaction at all times and asymptotically converges to the optimal centralized solution (Chapter 7).

For scenarios involving simple pairwise constraints, like collision avoidance, solutions akin to [Lindemann & Dimarogonas, 2020; Wang et al., 2017] remain preferable due to their simplicity and low communication requirements. However, when dealing with constraints involving multiple robots, such as connectivity maintenance and target monitoring, it becomes paramount to minimize conservativeness and achieve a solution that closely tracks the centralized optimal solution.

More than active sensing, the foundational element of cooperative localization lies in the localization algorithm. Few works have explored the estimation in a common reference frame of both position and orientation of a group of robots from body-frame measurements with provable stability. This motivated our investigation into the design of such observers, starting with an observer estimating the yaw orientation and position of a formation in a common frame from body-frame position measurements (Chapter 8).

To conclude, the novel contributions of this thesis include:

- The design of a predictive active sensing controller for optimal cooperative localization based on distributed optimization and on a suitable cost function, which is distributed and scalable with respect to the number of robots and it optimizes the whole trajectory rather then only the geometry (Chapter 4).

- The design of a distributed reactive active sensing controller for optimal cooperative localization which allows to integrate the active sensing action while ensuring the decrease of a CLF encoding the accomplishment of the desired task (Chapter 5).

- The design of a controller which aims to ensure a minimum prescribed level of information collected by a group of flying robots about the state of one or multiple

moving targets while possibly achieving other tasks through the use of HOCBFs (Chapter 6).

- The design of a distributed CBF approach which asymptotically converges to the centralized optimal solution while always ensuring satisfaction of the constraints overcoming a limitation which was present in [Tan & Dimarogonas, 2021] (Chapter 7).

- The design of an observer to estimate in a common frame the position and yaw of a group of robots from body-frame relative position measurements, whose stability is rigorously established (Chapter 8).

8.6 Open issues and future perspectives

While the results presented in this thesis regarding active cooperative localization are promising, there exist some limitations that impact the proposed approaches. Notably, our evaluation was performed by simulations, and an experimental validation is still missing.

In the immediate future, we plan to conduct an experimental validation of the algorithm introduced in Chapter 5 for camera-based cooperative localization of quadrotors in the Fly-Crane system (Fig. 8.3) [Sanalitro et al., 2022], which I have actively contributed to setting up in our team. In fact, we believe that the controller presented in Chapter 5

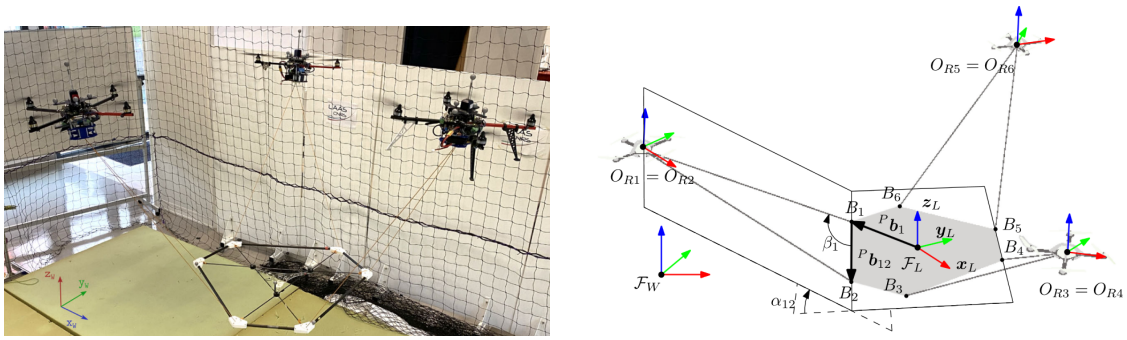


Figure 8.3: [Sanalitro et al., 2022] The Fly-Crane system is a robotic system composed by three aerial vehicles attached to a rigid platform, each one by two cables. This system allows to precisely control both the position and orientation of the platform for aerial manipulation/assembly tasks.

is more suitable than the one presented in Chapter 4 for real-world applications since

it better allows combining active sensing with another higher priority task, with lower requirements in terms of communication.

We point out that, to really take advantage of the approach proposed in Chapter 5, some redundancy with respect to the task is necessary; otherwise, the considered task may not leave much space for optimization as the trajectory can only be optimized as long as the task error (encoded by the Lyapunov function) decreases. Moreover, an interesting idea for future works could be to use *prescribed performance control* (PPC) [Bechlioulis & Rovithakis, 2008] instead of CLFs to enforce task satisfaction. In fact, prescribed performance control upper bounds the evolution of the task error to lie below a certain time-dependent but state-independent function. This would ensure that the task is accomplished in a certain prescribed amount of time while allowing the task error to locally increase, allowing to acquire more information. Furthermore, the work in Chapter 5 still needs to be finalized for the case with sensing constraints and we plan to test it considering more diverse tasks than the sole formation control. Also, more importantly, a provably stable observer to estimate the state of the robots in $\mathbb{R}^3 \times \mathbb{S}^1$ from *body-frame* relative bearing measurements is still missing. We are already working on it, combining ideas from [Z. Tang et al., 2022] and from Chapter 8.

Regarding the work on persistent monitoring of moving targets (Chapter 6), potential enhancements to the algorithm include utilizing a smooth minimum function approximation to combine multiple Control Barrier Functions (CBFs) [Fernandez-Ayala et al., 2023; Molnar & Ames, 2023] and employing an optimal approach to distribute the CBFs, as presented in Chapter 7, to allow smoother and more effective control of the system when multiple targets are splitting. Additionally, the algorithm could benefit from considering estimation uncertainty using robust CBFs [Nguyen & Sreenath, 2021; Singletary et al., 2022] for example considering some estimate of the maximum error based on the Kalman Filter covariance.

In the context of connectivity maintenance (Chapter 7) and information gradient estimation in Chapter 5, the distributed power iteration algorithm is employed to estimate the minimum nonzero eigenvalue and its associated eigenvector. This algorithm requires high frequency communication for the employed dynamic average consensus to track the associated signals. While it has seen extensive use in simulation and experiments without a real distributed setup [Malli et al., 2021; Robuffo Giordano et al., 2013; Sabattini et al., 2013; Yang et al., 2010], up to the author's knowledge, it has rarely (or not at all) been applied in a real distributed setup. Hence, experimental studies are necessary to

determine the limits of convergence when the robots do not move slowly.

The observer presented in Chapter 8 is split into three sub-components in cascade, potentially impacting performance in two ways: (*i*) the adaptive observer providing estimated relative yaw orientations, used as if they were measurements, is expected to be less robust than directly updating the yaw estimate in a common frame. Unfortunately, also due to the nonlinear nature of the $\mathbb{S}^1$, it is difficult to use the same idea of an adaptive observer, considering directly an update the of the yaw in a common frame from the local body frame position estimate associated to each edge; and (*ii*) considering the yaw observer and the position observer as two systems in feedback, using the current estimated positions in a common frame to improve the estimate of the yaw in the same frame, could potentially enhance performance but necessitates further studies on the stability of the feedback interconnection. It would be still more interesting to formulate a single observer, integrating an adaptive term, able to estimate position and yaw orientation in a common frame. Furthermore, as already mentioned, we plan to extend the ideas used for this observer to the case of body-frame relative bearing measurements, where more complexities arise to the necessity of *persistency of excitation* conditions associated also to the scale retrieval.

BIBLIOGRAPHY

Aggravi, M., Sirignano, G., Giordano, P. R., & Pacchierotti, C., (2021), Decentralized control of a heterogeneous human–robot team for exploration and patrolling, *IEEE Transactions on Automation Science and Engineering*, *19*(4), 3109–3125.

Amari, S.-I., (1998), Natural gradient works efficiently in learning, *Neural computation*, *10*(2), 251–276.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P., (2019), Control barrier functions: theory and applications, *2019 18th European control conference (ECC)*, 3420–3431.

Ames, A. D., Grizzle, J. W., & Tabuada, P., (2014), Control barrier function based quadratic programs with application to adaptive cruise control, *53rd IEEE Conference on Decision and Control*, 6271–6278.

Ames, A. D., & Powell, M., (2013), Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs, *Control of Cyber-Physical Systems: Workshop held at Johns Hopkins University, March 2013*, 219–240.

Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P., (2016), Control barrier function based quadratic programs for safety critical systems, *IEEE Transactions on Automatic Control*, *62*(8), 3861–3876.

Anderson, B., & Moore, J., (1969), New results in linear system stability, *SIAM Journal on Control*, *7*(3), 398–414.

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

Antsaklis, P. J., & Michel, A. N., (1997), *Linear systems* (Vol. 8), Springer.

Babar, M. Z., & Baglietto, M., (2021), Optimal feedback input design for dynamic nonlinear systems, *International Journal of Control*, *94*(8), 2264–2281.

Bajcsy, R., (1988), Active perception, *Proceedings of the IEEE*, *76*(8), 966–1005.

Balandi, L., De Carli, N., & Robuffo Giordano, P., (2023), Persistent monitoring of multiple moving targets using high order control barrier functions, *IEEE Robotics and Automation Letters*.

Barfoot, T. D., (2017), *State estimation for robotics*, Cambridge University Press.

Battistelli, G., & Chisci, L., (2014), Kullback–leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability, *Automatica*, *50*(3), 707–718.

Bechlioulis, C. P., & Rovithakis, G. A., (2008), Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance, *IEEE Transactions on Automatic Control*, *53*(9), 2090–2099.

Bernard, M., Pacchierotti, C., & Giordano, P. R., (2023), Decentralized connectivity maintenance for quadrotor uavs with field of view constraints, *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11111–11118.

Bernard, P., (2019), *Observer design for nonlinear systems* (Vol. 479), Springer.

Bernard, P., Andrieu, V., & Astolfi, D., (2022), Observer design for continuous-time dynamical systems, *Annual Reviews in Control*, *53*, 224–248.

Bernard, P., Mimmo, N., & Marconi, L., (2020), On the semi-global stability of an ek-like filter, *IEEE Control Systems Letters*, *5*(5), 1771–1776.

Bertsekas, D., & Tsitsiklis, J., (2015), *Parallel and distributed computation: numerical methods*, Athena Scientific.

Besançon, G., (2000), Remarks on nonlinear adaptive observer design, *Systems & control letters*, *41*(4), 271–280.

Besançon, G., (2007), *Nonlinear observers and applications* (Vol. 363), Springer.

Besançon, G., Scola, I. R., & Georges, D., (2013), Input selection in observer design for non-uniformly observable systems, *IFAC Proceedings Volumes*, *46*(23), 664–669.

Biagiotti, L., & Melchiorri, C., (2008), *Trajectory planning for automatic machines and robots*, Springer Science & Business Media.

Boffi, N. M., & Slotine, J.-J. E., (2021), Implicit regularization and momentum algorithms in nonlinearly parameterized adaptive control and prediction, *Neural Computation*, *33*(3), 590–673.

Böhm, C., Brault, P., Delamare, Q., Robuffo Giordano, P., & Weiss, S., (2022), Cop: control & observability-aware planning, *arXiv preprint arXiv:2203.06982*.

Böhm, C., Li, G., Loianno, G., & Weiss, S., (2020), Observability-aware trajectories for geometric and inertial self-calibration, *Power On and Go Robots*.

Boumal, N., Singer, A., & Absil, P.-A., (2013), Robust estimation of rotations from relative measurements by maximum likelihood, *52nd IEEE Conference on Decision and Control*, 1156–1161.

Boyd, S. P., & Vandenberghe, L., (2004), *Convex optimization*, Cambridge university press.

Briot, S., & Robuffo Giordano, P., (2019), Physical interpretation of rigidity for bearing formations: application to mobility and singularity analyses, *Journal of Mechanisms and Robotics*, *11*(3), 031006.

Bullo, F., (2023), *Contraction theory for dynamical systems* (1.1), Kindle Direct Publishing, https://fbullo.github.io/ctds

Bullo, F., (2020), *Lectures on network systems* (Vol. 1), Kindle Direct Publishing Seattle, DC, USA.

Calkins, L., Khodayi-mehr, R., Aquino, W., & Zavlanos, M. M., (2020), Sensor planning for model-based acoustic source identification, *2020 American Control Conference (ACC)*, 2679–2684.

Cano, J., & Le Ny, J., (2021), Improving ranging-based location estimation with rigidity-constrained crlb-based motion planning, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5758–5764.

Cano, J., & Le Ny, J., (2023), Ranging-based localizability optimization for mobile robotic networks, *IEEE Transactions on Robotics*.

Capelli, B., & Sabattini, L., (2020), Connectivity maintenance: global and optimized approach through control barrier functions, *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 5590–5596.

Chen, F., Cao, Y., & Ren, W., (2012), Distributed average tracking of multiple time-varying reference signals with bounded derivatives, *IEEE Transactions on Automatic Control*, *57*(12), 3169–3174.

Chen, Y., Santillo, M., Jankovic, M., & Ames, A. D., (2020), Online decentralized decision making with inequality constraints: an admm approach, *IEEE Control Systems Letters*, *5*(6), 2156–2161.

Chezhegov, S., Novitskii, A., Rogozin, A., Parsegov, S., Dvurechensky, P., & Gasnikov, A., (2022), A general framework for distributed partitioned optimization, *arXiv preprint arXiv:2203.00681*.

Chung, T. H., Burdick, J. W., & Murray, R. M., (2006), A decentralized motion coordination strategy for dynamic target tracking, *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2416–2422.

Coleman, D., Bopardikar, S., & Tan, X., (2024), Using control barrier functions to incorporate observability: application to range-based target tracking, *Journal of Dynamic Systems, Measurement, and Control*, 1–12.

Cortes, J., & Egerstedt, M., (2017), Coordinated control of multi-robot systems: a survey, *SICE Journal of Control, Measurement, and System Integration*, *10*(6), 495–503.

Cossette, C. C., Shalaby, M. A., Saussie, D., Ny, J. L., & Forbes, J. R., (2022), Optimal multi-robot formations for relative pose estimation using range measurements, *arXiv preprint arXiv:2205.14263*.

Cristofaro, A., & Martinelli, A., (2010), Optimal trajectories for multi robot localization, *49th IEEE Conference on Decision and Control (CDC)*, 6358–6364.

De Carli, N., Restrepo, E., & Giordano, P. R., (2024), Adaptive observer from body-frame relative position measurements for cooperative localization, *IEEE Control Systems Letters*.

De Carli, N., Salaris, P., & Robuffo Giordano, P., (2021), Online decentralized perception-aware path planning for multi-robot systems, *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 128–136.

De Carli, N., Salaris, P., & Robuffo Giordano, P., (2023), Multi-robot active sensing for bearing formations, *MSR 2023-IEEE International Symposium on Multi-Robot & Multi-Agent Systems*, 1–7.

De Carli, N., Salaris, P., & Robuffo Giordano, P., (2024), Distributed control barrier functions for global connectivity maintenance, *2024 IEEE International Conference on Robotics and Automation (ICRA)*.

De Pasquale, G., Smith, K. D., Bullo, F., & Valcher, M. E., (2023), Dual seminorms, ergodic coefficients and semicontraction theory, *IEEE Transactions on Automatic Control*.

de Marina Peinado, H. J. G., (2016), Distributed formation control for autonomous robots, *University of Groningen*.

DeVries, L., Majumdar, S. J., & Paley, D. A., (2013), Observability-based optimization of coordinated sampling trajectories for recursive estimation of a strong, spatially varying flowfield, *Journal of intelligent & robotic systems*, *70*(1-4), 527–544.

Di Lorenzo, P., & Scutari, G., (2016), Next: in-network nonconvex optimization, *IEEE Transactions on Signal and Information Processing over Networks*, *2*(2), 120–136.

Dimarogonas, D. V., & Johansson, K. H., (2008), Decentralized connectivity maintenance in mobile networks with bounded inputs, *2008 IEEE International Conference on Robotics and Automation*, 1507–1512.

Dörfler, F., Simpson-Porco, J. W., & Bullo, F., (2018), Electrical networks and algebraic graph theory: models, properties, and applications, *Proceedings of the IEEE*, *106*(5), 977–1005.

Erskine, J., Briot, S., Fantoni, I., & Chriette, A., (2023), Singularity analysis of rigid directed bearing graphs for quadrotor formations, *IEEE Transactions on Robotics*.

Feller, C., & Ebenbauer, C., (2015), Weight recentered barrier functions and smooth polytopic terminal set formulations for linear model predictive control, *2015 American Control Conference (ACC)*, 1647–1652.

Fernandez-Ayala, V. N., Tan, X., & Dimarogonas, D. V., (2023), Distributed barrier function-enabled human-in-the-loop control for multi-robot systems, *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 7706–7712.

Fliess, M., Levine, J., Martin, P., & Rouchon, P., (1995), Flatness and defect of non-linear systems: introductory theory and examples, *International journal of control*, *61*(6), 1327–1361.

Franchi, A., & Robuffo Giordano, P., (2018), Online leader selection for improved collective tracking and formation maintenance, *IEEE Trans. on Control of Network Systems*, *5*(1), 3–13.

Franchi, A., Masone, C., Grabe, V., Ryll, M., Bülthoff, H. H., & Robuffo Giordano, P., (2012), Modeling and control of uav bearing formations with bilateral high-level steering, *The International Journal of Robotics Research*, *31*(12), 1504–1525.

Freeman, R. A., Yang, P., & Lynch, K. M., (2006), Stability and convergence properties of dynamic average consensus estimators, *Proceedings of the 45th IEEE Conference on Decision and Control*, 338–343.

Freundlich, C., Lee, S., & Zavlanos, M. M., (2017), Distributed active state estimation with user-specified accuracy, *IEEE Transactions on Automatic Control*, *63*(2), 418–433.

Galloway, K., Sreenath, K., Ames, A. D., & Grizzle, J. W., (2015), Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs, *IEEE Access*, *3*, 323–332.

Garg, K., Usevitch, J., Breeden, J., Black, M., Agrawal, D., Parwana, H., & Panagou, D., (2023), Advances in the theory of control barrier functions: addressing practical challenges in safe control synthesis for autonomous and robotic systems, *arXiv preprint arXiv:2312.16719*.

Garin, F., & Schenato, L., (2010), A survey on distributed estimation and control applications using linear consensus algorithms. *In Networked control systems* (pp. 75–107), Springer.

Gibson, J. J., (1962), Observations on active touch., *Psychological review*, *69*(6), 477.

Halsted, T., Shorinwa, O., Yu, J., & Schwager, M., (2021), A survey of distributed optimization methods for multi-robot systems, *arXiv preprint arXiv:2103.12840*.

Hamel, T., & Samson, C., (2016), Riccati observers for position and velocity bias estimation from direction measurements, *2016 IEEE 55th conference on decision and control (CDC)*, 2047–2053.

Hausman, K., Preiss, J., Sukhatme, G. S., & Weiss, S., (2017), Observability-aware trajectory optimization for self-calibration with application to uavs, *IEEE Robotics and Automation Letters*, *2*(3), 1770–1777.

Hinson, B. T., Binder, M. K., & Morgansen, K. A., (2013), Path planning to optimize observability in a planar uniform flow field, *2013 American Control Conference*, 1392–1399.

Hinson, B. T., & Morgansen, K. A., (2013), Observability optimization for the nonholonomic integrator, *2013 American Control Conference*, 4257–4262.

Hinson, B. T., (2014), *Observability-based guidance and sensor placement* (Doctoral dissertation).

Horn, R. A., & Johnson, C. R., (2012), *Matrix analysis*, Cambridge university press.

Houska, B., Telen, D., Logist, F., Diehl, M., & Van Impe, J. F., (2015), An economic objective for the optimal experiment design of nonlinear dynamic processes, *Automatica*, *51*, 98–103.

Jacquet, M., Kivits, M., Das, H., & Franchi, A., (2022), Motor-level n-mpc for cooperative active perception with multiple heterogeneous uavs, *IEEE Robotics and Automation Letters*, *7*(2), 2063–2070.

Jankovic, M., (2018), Robust control barrier functions for constrained stabilization of nonlinear systems, *Automatica*, *96*, 359–367.

Ji, M., & Egerstedt, M., (2007), Distributed coordination control of multiagent systems while preserving connectedness, *IEEE Transactions on Robotics*, *23*(4), 693–703.

Kamal, A. T., Farrell, J. A., & Roy-Chowdhury, A. K., (2013), Information weighted consensus filters and their application in distributed camera networks, *IEEE Transactions on Automatic Control*, *58*(12), 3112–3125.

Kay, S. M., (1993), *Fundamentals of statistical signal processing: estimation theory*, Prentice-Hall, Inc.

Kerrigan, E. C., & Maciejowski, J. M., (2000), Soft constraints and exact penalty functions in model predictive control.

Khalil, H., (2014), *Nonlinear control*, Pearson.

Khodayi-mehr, R., Aquino, W., & Zavlanos, M. M., (2019), Model-based active source identification in complex environments, *IEEE Transactions on Robotics*, *35*(3), 633–652.

Kia, S. S., Van Scoy, B., Cortes, J., Freeman, R. A., Lynch, K. M., & Martinez, S., (2019), Tutorial on dynamic average consensus: the problem, its applications, and the algorithms, *IEEE Control Systems Magazine*, *39*(3), 40–72.

Kolathaya, S., & Ames, A. D., (2018), Input-to-state safety with control barrier functions, *IEEE control systems letters*, *3*(1), 108–113.

Krener, A. J., & Ide, K., (2009), Measures of unobservability, *Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, 6401–6406.

Krick, L., Broucke, M. E., & Francis, B. A., (2009), Stabilisation of infinitesimally rigid formations of multi-robot networks, *International Journal of control*, *82*(3), 423–439.

Krstic, M., (2023), Inverse optimal safety filters, *IEEE Transactions on Automatic Control*.

Le Ny, J., & Chauvière, S., (2018), Localizability-constrained deployment of mobile robotic networks with noisy range measurements, *2018 Annual American Control Conference (ACC)*, 2788–2793.

Lee, B.-H., & Ahn, H.-S., (2016a), Distributed estimation for the unknown orientation of the local reference frames in n-dimensional space, *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1–6.

Lee, B.-H., & Ahn, H.-S., (2016b), Distributed formation control via global orientation estimation, *Automatica*, *73*, 125–129.

Lee, T., Kwon, J., & Park, F. C., (2018), A natural adaptive control law for robot manipulators, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–9.

Lee, T., Leok, M., & McClamroch, N. H., (2010), Geometric tracking control of a quadrotor uav on se (3), *49th IEEE conference on decision and control (CDC)*, 5420–5425.

Leonardos, S., Daniilidis, K., & Tron, R., (2019), Distributed 3-d bearing-only orientation localization, *2019 IEEE 58th Conference on Decision and Control (CDC)*, 1834–1841.

Li, J., Ning, Z., He, S., Lee, C.-H., & Zhao, S., (2022), Three-dimensional bearing-only target following via observability-enhanced helical guidance, *IEEE Transactions on Robotics*, *39*(2), 1509–1526.

Li, S., De Wagter, C., & de Croon, G. C., (2022), Nonlinear model predictive control for improving range-based relative localization by maximizing observability, *International Journal of Micro Air Vehicles*, *14*.

Li, X., Luo, X., & Zhao, S., (2019), Globally convergent distributed network localization using locally measured bearings, *IEEE Transactions on Control of Network Systems*, *7*(1), 245–253.

Lindemann, L., & Dimarogonas, D. V., (2020), Barrier function based collaborative control of multiple robots under signal temporal logic tasks, *IEEE Transactions on Control of Network Systems*, *7*(4), 1916–1928.

Lorussi, F., Marigo, A., & Bicchi, A., (2001), Optimal exploratory paths for a mobile rover, *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, *2*, 2078–2083.

Luft, L., Schubert, T., Roumeliotis, S. I., & Burgard, W., (2016), Recursive decentralized collaborative localization for sparsely communicating robots., *Robotics: Science and Systems*.

Luft, L., Schubert, T., Roumeliotis, S. I., & Burgard, W., (2018), Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication, *The International Journal of Robotics Research*, *37*(10), 1152–1167.

Lyu, Y., Hu, J., Chen, B. M., Zhao, C., & Pan, Q., (2019), Multivehicle flocking with collision avoidance via distributed model predictive control, *IEEE transactions on cybernetics*, *51*(5), 2651–2662.

Magnus, J. R., (1985), On differentiating eigenvalues and eigenvectors, *Econometric Theory*, *1*, 179–191.

Mahony, R., Hamel, T., & Pflimlin, J.-M., (2008), Nonlinear complementary filters on the special orthogonal group, *IEEE Transactions on automatic control*, *53*(5), 1203–1218.

Malli, I., Bechlioulis, C. P., & Kyriakopoulos, K. J., (2021), Robust distributed estimation of the algebraic connectivity for networked multi-robot systems, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 9155–9160.

Mao, G., Fidan, B., & Anderson, B. D., (2007), Wireless sensor network localization techniques, *Computer networks*, *51*(10), 2529–2553.

Marino, R., (1995), Nonlinear control design: geometric, *Adaptive and robust*.

Martinez, S., & Bullo, F., (2006), Optimal sensor placement and motion coordination for target tracking, *Automatica*, *42*(4), 661–668.

Mellinger, D., & Kumar, V., (2011), Minimum snap trajectory generation and control for quadrotors, *2011 IEEE international conference on robotics and automation*, 2520–2525.

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

Meyer, C. D., & Stewart, I., (2023), *Matrix analysis and applied linear algebra*, SIAM.

Molnar, T. G., & Ames, A. D., (2023), Composing control barrier functions for complex safety specifications, *IEEE Control Systems Letters*.

Montijano, E., Cristofalo, E., Zhou, D., Schwager, M., & Saguees, C., (2016), Vision-based distributed formation control without an external positioning system, *IEEE Transactions on Robotics*, *32*(2), 339–351.

Montijano, E., Montijano, J. I., Sagües, C., & Martınez, S., (2014), Robust discrete time dynamic average consensus, *Automatica*, *50*(12), 3131–3138.

Montijano, E., Zhou, D., Schwager, M., & Sagues, C., (2014), Distributed formation control without a global reference frame, *2014 American Control Conference*, 3862–3867.

Morbidi, F., & Mariottini, G. L., (2012), Active target tracking and cooperative localization for teams of aerial vehicles, *IEEE transactions on control systems technology*, *21*(5), 1694–1707.

Moreau, L., (2004), Stability of continuous-time distributed consensus algorithms, *2004 43rd IEEE conference on decision and control (CDC)(IEEE Cat. No. 04CH37601)*, *4*, 3998–4003.

Murphy, R. R., Dreger, K. L., Newsome, S., Rodocker, J., Steimle, E., Kimura, T., Makabe, K., Matsuno, F., Tadokoro, S., & Kon, K., (2011), Use of remotely operated marine vehicles at minamisanriku and rikuzentakata japan for disaster recovery, *2011 IEEE International symposium on safety, security, and rescue robotics*, 19–25.

Napolitano, O., Cognetti, M., Pallottino, L., Salaris, P., Kanoulas, D., & Modugno, V., (2023), Active sensing for data quality improvement in model learning.

Napolitano, O., Fontanelli, D., Pallottino, L., & Salaris, P., (2021), Gramian-based optimal active sensing control under intermittent measurements, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 9680–9686.

Napolitano, O., Fontanelli, D., Pallottino, L., & Salaris, P., (2022), Information-aware lyapunov-based mpc in a feedback-feedforward control strategy for autonomous robots, *IEEE Robotics and Automation Letters*, *7*(2), 4765–4772.

Nedić, A., Olshevsky, A., & Rabbat, M. G., (2018), Network topology and communication-computation tradeoffs in decentralized optimization, *Proceedings of the IEEE*, *106*(5), 953–976.

Nguyen, Q., & Sreenath, K., (2016), Exponential control barrier functions for enforcing high relative-degree safety-critical constraints, *2016 American Control Conference (ACC)*, 322–328.

Nguyen, Q., & Sreenath, K., (2021), Robust safety-critical control for dynamic robotics, *IEEE Transactions on Automatic Control*, *67*(3), 1073–1088.

Nocedal, J., & Wright, S. J., (1999), *Numerical optimization*, Springer.

Notomista, G., Pacchierotti, C., & Robuffo Giordano, P., (2022), Multi-robot persistent environmental monitoring based on constraint-driven execution of learned robot tasks, *2022 International Conference on Robotics and Automation (ICRA)*, 6853–6859.

Oh, K.-K., & Ahn, H.-S., (2013), Formation control and network localization via orientation alignment, *IEEE Transactions on Automatic Control*, *59*(2), 540–545.

Oh, K.-K., & Ahn, H.-S., (2014), Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space, *International Journal of Robust and Nonlinear Control*, *24*(12), 1809–1820.

Olfati-Saber, R., (2007), Distributed tracking for mobile sensor networks with information-driven mobility, *2007 American Control Conference*, 4606–4612.

Olfati-Saber, R., (2009), Kalman-consensus filter: optimality, stability, and performance, *Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, 7036–7042.

Ong, P., Capelli, B., Sabattini, L., & Cortes, J., (2023), Nonsmooth control barrier function design of continuous constraints for network connectivity maintenance, *Automatica*, *156*, 111209.

Pan, L., Shao, H., Mesbahi, M., Xi, Y., & Li, D., (2020), Consensus on matrix-weighted time-varying networks, *arXiv preprint arXiv:2001.11179*.

Panagou, D., Stipanovic, D. M., & Voulgaris, P. G., (2013), Multi-objective control for multi-agent systems using lyapunov-like barrier functions, *52nd IEEE Conference on Decision and Control*, 1478–1483.

Pengov, M., Richard, E., & Vivalda, J.-C., (2001), On the boundedness of the solutions of the continuous riccati equation., *Journal of Inequalities and Applications*, *6*(6), 641–649.

Preiss, J. A., Hausman, K., Sukhatme, G. S., & Weiss, S., (2018), Simultaneous self-calibration and navigation using trajectory optimization, *The International Journal of Robotics Research*, *37*(13-14), 1573–1594.

Pukelsheim, F., (2006), *Optimal design of experiments*, SIAM.

Qian, J., Nadri, M., & Dufour, P., (2017), Optimal input design for parameter estimation of nonlinear systems: case study of an unstable delta wing, *International Journal of Control*, *90*(4), 873–887.

Rawlings, J. B., Mayne, D. Q., & Diehl, M., (2017), *Model predictive control: theory, computation, and design* (Vol. 2), Nob Hill Publishing Madison, WI.

Restrepo, E., (2021), *Coordination control of autonomous robotic multi-agent systems under constraints* (Doctoral dissertation), Universite Paris-Saclay.

Restrepo, E., Loria, A., Sarras, I., & Marzat, J., (2021), Edge-based strict lyapunov functions for consensus with connectivity preservation over directed graphs, *Automatica*, *132*, 109812.

Restrepo, E., Loria, A., Sarras, I., & Marzat, J., (2022), Robust consensus of high-order systems under output constraints: application to rendezvous of underactuated uavs, *IEEE Transactions on Automatic Control*, *68*(1), 329–342.

Riz, F., Palopoli, L., & Fontanelli, D., (2023), Perception-aware trajectory planning for a pair of unicycle-like robots with absolute and relative ranging measurements, *IEEE Control Systems Letters*.

Robuffo Giordano, P., Franchi, A., Secchi, C., & Bülthoff, H. H., (2013), A passivity-based decentralized strategy for generalized connectivity maintenance, *The International Journal of Robotics Research*, *32*(3), 299–323.

Sabattini, L., Secchi, C., Chopra, N., & Gasparri, A., (2013), Distributed control of multirobot systems with global connectivity maintenance, *IEEE Transactions on Robotics*, *29*(5), 1326–1332.

Salaris, P., Cognetti, M., Spica, R., & Robuffo Giordano, P., (2019), Online optimal perception-aware trajectory generation, *IEEE Transactions on Robotics*, *35*(6), 1307–1322.

Salaris, P., Spica, R., Robuffo Giordano, P., & Rives, P., (2017), Online optimal active sensing control, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 672–678.

Sanalitro, D., Tognon, M., Cano, A. J., Cortes, J., & Franchi, A., (2022), Indirect force control of a cable-suspended aerial multi-robot manipulator, *IEEE Robotics and Automation Letters*, *7*(3), 6726–6733.

Schiano, F., Franchi, A., Zelazo, D., & Robuffo Giordano, P., (2016), A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5099–5106.

Schiano, F., & Giordano, P. R., (2017), Bearing rigidity maintenance for formations of quadrotor uavs, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1467–1474.

Schiano, F., & Tron, R., (2018), The dynamic bearing observability matrix nonlinear observability and estimation for multi-agent systems, *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3669–3676.

Schranz, M., Umlauft, M., Sende, M., & Elmenreich, W., (2020), Swarm robotic behaviors and current applications, *Frontiers in Robotics and AI*, 36.

Shorinwa, O., & Schwager, M., (2023), Distributed model predictive control via separable optimization in multi-agent networks, *IEEE Transactions on Automatic Control*.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G., (2009), *Force control*, Springer.

Singletary, A., Ahmadi, M., & Ames, A. D., (2022), Safe control for nonlinear systems with stochastic uncertainty via risk control barrier functions, *IEEE Control Systems Letters*, *7*, 349–354.

Slotine, J.-J. E., Li, W., et al., (1991), *Applied nonlinear control* (Vol. 199), Prentice hall Englewood Cliffs, NJ.

Sontag, E. D., (1989), A 'universal'construction of artstein's theorem on nonlinear stabilization, *Systems & control letters*, *13*(2), 117–123.

Sontag, E. D., Biswas, D., & Cowan, N. J., (2022), An observability result related to active sensing, *arXiv preprint arXiv:2210.03848*.

Spica, R., & Giordano, P. R., (2013), A framework for active estimation: application to structure from motion, *52nd IEEE conference on decision and control*, 7647–7653.

Spica, R., Giordano, P. R., & Chaumette, F., (2014), Active structure from motion: application to point, sphere, and cylinder, *IEEE Transactions on Robotics*, *30*(6), 1499–1513.

Spica, R., & Robuffo Giordano, P., (2016), Active decentralized scale estimation for bearing-based localization, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5084–5091.

Spica, R., Robuffo Giordano, P., & Chaumette, F., (2017), Coupling active depth estimation and visual servoing via a large projection operator, *The International Journal of Robotics Research*, *36*(11), 1177–1194.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., & Boyd, S., (2020), Osqp: an operator splitting solver for quadratic programs, *Mathematical Programming Computation*, *12*(4), 637–672.

Tan, X., Cortez, W. S., & Dimarogonas, D. V., (2021), High-order barrier functions: robustness, safety, and performance-critical control, *IEEE Transactions on Automatic Control*, *67*(6), 3021–3028.

Tan, X., & Dimarogonas, D. V., (2021), Distributed implementation of control barrier functions for multi-agent systems, *IEEE Control Systems Letters*, *6*, 1879–1884.

Tan, X., & Dimarogonas, D. V., (2022), Distributed control barrier function-based control scheme for multi-agent systems under a collective constraint, *the 25th International Symposium on Mathematical Theory of Networks and Systems Bayreuth, Germany, 12-16 September 2022*, 571–574.

Tang, W., & Daoutidis, P., (2019), Distributed nonlinear model predictive control through accelerated parallel admm, *2019 American Control Conference (ACC)*, 1406–1411.

Tang, Z., Cunha, R., Hamel, T., & Silvestre, C., (2022), Relaxed bearing rigidity and bearing formation control under persistence of excitation, *Automatica*, *141*, 110289.

Tang, Z., & Loria, A., (2022), Localization and tracking control of autonomous vehicles in time-varying bearing formation, *IEEE Control Systems Letters*.

Tee, K. P., Ge, S. S., & Tay, E. H., (2009), Barrier lyapunov functions for the control of output-constrained nonlinear systems, *Automatica*, *45*(4), 918–927.

Telen, D., Houska, B., Logist, F., Van Derlinden, E., Diehl, M., & Van Impe, J., (2013), Optimal experiment design under process noise using riccati differential equations, *Journal of Process Control*, *23*(4), 613–629.

Telen, D., Logist, F., Quirynen, R., Houska, B., Diehl, M., & Van Impe, J., (2014), Optimal experiment design for nonlinear dynamic (bio) chemical systems using sequential semidefinite programming, *AIChE Journal*, *60*(5), 1728–1739.

Trakas, P. S., Bechlioulis, C. P., & Rovithakis, G. A., (2022), Decentralized global connectivity maintenance for multi-agent systems using prescribed performance average consensus protocols, *2022 European Control Conference (ECC)*, 524–529.

Trinh, M. H., Van Nguyen, C., Lim, Y.-H., & Ahn, H.-S., (2018), Matrix-weighted consensus and its applications, *Automatica*, *89*, 415–419.

Trinh, M. H., Van Tran, Q., & Ahn, H.-S., (2019), Minimal and redundant bearing rigidity: conditions and applications, *IEEE Transactions on Automatic Control*, *65*(10), 4186–4200.

Ucinski, D., (2004), *Optimal measurement methods for distributed parameter system identification*, CRC press.

Van Scoy, B., Freeman, R. A., & Lynch, K. M., (2015), Design of robust dynamic average consensus estimators, *2015 54th IEEE Conference on Decision and Control (CDC)*, 6269–6275.

Van Tran, Q., & Ahn, H.-S., (2020), Distributed formation control of mobile agents via global orientation estimation, *IEEE Transactions on Control of Network Systems*, *7*(4), 1654–1664.

Van Tran, Q., Ahn, H.-S., & Anderson, B. D., (2018), Distributed orientation localization of multi-agent systems in 3-dimensional space with direction-only measurements, *2018 IEEE Conference on Decision and Control (CDC)*, 2883–2889.

Wang, L., Ames, A. D., & Egerstedt, M., (2017), Safety barrier certificates for collisions-free multirobot systems, *IEEE Transactions on Robotics*, *33*(3), 661–674.

Wensing, P. M., & Slotine, J.-J., (2020), Beyond convexity—contraction and global convergence of gradient descent, *Plos one*, *15*(8), e0236661.

Wieland, P., & Allgöwer, F., (2007), Constructive safety using control barrier functions, *IFAC Proceedings Volumes*, *40*(12), 462–467.

Xiao, W., & Belta, C., (2019), Control barrier functions for systems with high relative degree, *2019 IEEE 58th conference on decision and control (CDC)*, 474–479.

Xiao, W., & Belta, C., (2021), High order control barrier functions, *IEEE Transactions on Automatic Control*.

Xiao, W., Cassandras, C. G., & Belta, C., (2023), *Safe autonomy with control barrier functions: theory and applications*, Springer Nature.

Xu, X., Tabuada, P., Grizzle, J. W., & Ames, A. D., (2015), Robustness of control barrier functions for safety critical control, *IFAC-PapersOnLine*, *48*(27), 54–61.

Yang, P., Freeman, R. A., Gordon, G. J., Lynch, K. M., Srinivasa, S. S., & Sukthankar, R., (2010), Decentralized estimation and control of graph connectivity for mobile sensor networks, *Automatica*, *46*(2), 390–396.

Zelazo, D., Franchi, A., Bülthoff, H. H., & Robuffo Giordano, P., (2015), Decentralized rigidity maintenance control with range measurements for multi-robot systems, *The International Journal of Robotics Research*, *34*(1), 105–128.

Zhang, T., Qin, V., Tang, Y., & Li, N., (2022), Distributed information-based source seeking, *arXiv preprint arXiv:2209.09421*.

Zhang, W., Teague, B., & Meyer, F., (2022), Active planning for cooperative localization: a fisher information approach, *2022 56th Asilomar Conference on Signals, Systems, and Computers*, 795–800.

Zhao, S., & Zelazo, D., (2015), Bearing rigidity and almost global bearing-only formation stabilization, *IEEE Transactions on Automatic Control*, *61*(5), 1255–1268.

Zhao, S., & Zelazo, D., (2019), Bearing rigidity theory and its applications for control and estimation of network systems: life beyond distance rigidity, *IEEE Control Systems Magazine*, *39*(2), 66–83.

Zhou, K., & Roumeliotis, S. I., (2011), Multirobot active target tracking with combinations of relative observations, *IEEE Transactions on Robotics*, *27*(4), 678–695.

Zhu, M., & Martınez, S., (2010), Discrete-time dynamic average consensus, *Automatica*, *46*(2), 322–329.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Perception active et localisation pour les systèmes multirobots

Mot clés : Systèmes multi-robots, localisation coopérative, détection active, perception active, fonctions de barrière de contrôle

Résumé : Dans cette thèse, nous nous attaquons aux défis de la localisation des systèmes multi-robots, en nous concentrant sur la localisation coopérative dans des formations non infiniment rigides avec des contraintes de détection. Nos contributions introduisent un cadre dans lequel les objectifs éventuellement conflictuels du maintien de la connectivité, de l'exécution des tâches et de l'acquisition d'informations sont "médiés" à l'aide d'un programme quadratique et des fonctions de barrière de contrôle et du formalisme de la fonction de Lyapunov de contrôle.
Une autre contribution de cette thèse concerne la localisation active distribuée de cibles mobiles multiples par un groupe de robots volants utilisant des mesures basées sur des caméras, tout en accommodant d'autres tâches si la redondance du système le permet. Dans ce cas également, la formulation du problème utilise un programme quadratique et des fonctions de barrière de contrôle.
En nous appuyant sur la fonction de barrière de contrôle et le cadre du programme quadratique, nous identifions et abordons les limites de l'état actuel de la technique, en particulier en ce qui concerne les fonctions de barrière de contrôle distribuées. Nos modifications aboutissent à un contrôleur qui converge vers la solution optimale centralisée.
Enfin, nous présentons une méthodologie d'observation comme une nouvelle contribution, facilitant la localisation coopérative d'un système multi-robots dans un cadre commun en utilisant des mesures relatives au cadre du corps.

Title: Active Perception and Localization for Multi-Robot Systems

Keywords: Multi-robot systems, cooperative localization, active sensing, active perception, control barrier functions

Abstract: In this thesis, we tackle challenges in the localization of multi-robot systems, focusing on cooperative localization in non-infinitesimally rigid formations with sensing constraints. Our contributions introduce a framework in which the possibly conflicting goals of connectivity maintenance, task execution and the information acquisition are "mediated" using a quadratic program and the control barrier functions and control Lyapunov function formalism.
Another contribution of this thesis addresses distributed active localization of multiple moving targets by a group of flying robots using camera-based measurements, while accommodating other tasks if system redundancy permits. Also in this case, the problem formulation utilizes a quadratic program and control barrier functions.
Building on the control barrier function and quadratic program framework, we identify and address limitations in the existing state of the art, particularly in distributed control barrier functions. Our modifications result in a controller that converges to the centralized optimal solution.
Lastly, we present an observer methodology as a novel contribution, allowing cooperative localization of a multi-robot system in a common frame using body-frame relative measurements.